Project #3

Milica Cudina

2024-04-29

Problem #1 (5+10+5+10+10+10+10+10+5+25=100 points)

Solve **Problem 4.8.13** (pp. 192-193) from the textbook.

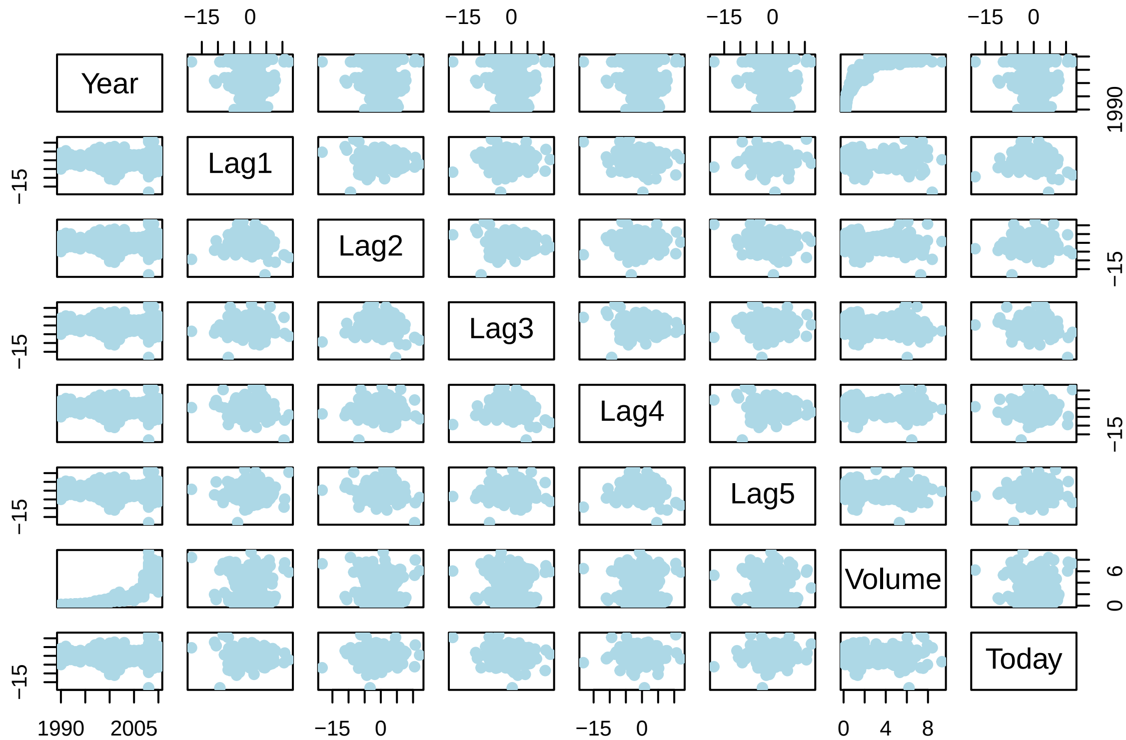
Hint: Here is a list of libraries you will need:

```
library(MASS)
library(ISLR2)
##
## Attaching package: 'ISLR2'
## The following object is masked from 'package:MASS':
##
##     Boston
library(e1071)
library(class)
```

Solution: First, here is some exploratory data analysis.

```
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
plot(Weekly[, -9], pch=19, col="lightblue")
```



As time goes by, there is more and more trading. So, there is a nice correlation between `Year` and `Volume`. Other than that, I cannot discern a pattern.

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

```
##
## 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 is the only significant one.

Now, it's time for the **confusion matrix**.

```
probs <- predict(mlr.fit, type = "response")
glm.pred=rep("Down", length(probs))
glm.pred[probs>0.5]<-"Up"
tab <- table(glm.pred, Weekly$Direction)
tab
##
## glm.pred Down  Up
##     Down   54  48
##     Up    430 557
sum(diag(tab))/sum(tab)
## [1] 0.5610652
mean(Weekly$Direction=="Up")
## [1] 0.5555556
```

The prediction is correct a bit over 56% of the time. However, the proportion of the realized "Up"s was just under 56%. So, constantly saying "Up" would work almost as well as our logistic regression.

Now, for training and testing.

```
attach(Weekly)
train <- (Year< 2009)
test=Weekly[!train,]
Direction.test=Direction[!train]
dim(test)
## [1] 104   9
fit.tr <- glm(Direction ~ Lag2, data = Weekly, subset=train, family = binomial)

probs.tr <- predict(fit.tr, test[,-9], type = "response")
probs.tr
##       986       987       988       989       990       991       992       993
## 0.5261291 0.6447364 0.4862159 0.4852001 0.5197667 0.5401255 0.6233482 0.4809930
```

```
##       994       995       996       997       998       999      1000      1001
## 0.4512204 0.4848808 0.4488192 0.6953567 0.5733026 0.6368201 0.5968501 0.5744959
##      1002      1003      1004      1005      1006      1007      1008      1009
## 0.5724070 0.5450566 0.5692901 0.6331171 0.4783830 0.5573436 0.6019835 0.5831351
##      1010      1011      1012      1013      1014      1015      1016      1017
## 0.5599792 0.5124689 0.5470007 0.5152843 0.5227821 0.6474861 0.6090744 0.5626685
##      1018      1019      1020      1021      1022      1023      1024      1025
## 0.5838410 0.5415393 0.5819483 0.5545612 0.5330760 0.5875345 0.5855762 0.5182874
##      1026      1027      1028      1029      1030      1031      1032      1033
## 0.5241299 0.6143180 0.5722506 0.5399379 0.4924147 0.5960111 0.5828808 0.5478786
##      1034      1035      1036      1037      1038      1039      1040      1041
## 0.5507839 0.5696462 0.5512007 0.5455175 0.5817080 0.5360825 0.5887869 0.5393750
##      1042      1043      1044      1045      1046      1047      1048      1049
## 0.4942153 0.5269836 0.5403419 0.5631688 0.5951015 0.5445667 0.5946395 0.5648403
##      1050      1051      1052      1053      1054      1055      1056      1057
## 0.5629973 0.5589196 0.5647832 0.5704009 0.5479362 0.5807465 0.5143121 0.4581345
##      1058      1059      1060      1061      1062      1063      1064      1065
## 0.5824712 0.4894383 0.5529103 0.5180988 0.5863795 0.5844761 0.4978605 0.4777452
##      1066      1067      1068      1069      1070      1071      1072      1073
## 0.6266569 0.5331483 0.6009391 0.5492598 0.5766247 0.4959289 0.5405584 0.5410922
##      1074      1075      1076      1077      1078      1079      1080      1081
## 0.6037500 0.5571859 0.5713260 0.5798975 0.5475908 0.5742261 0.5642262 0.5590485
##      1082      1083      1084      1085
## 0.5508557 0.6016494 0.5192446 0.5512582
##  [ reached getOption("max.print") -- omitted 4 entries ]
length(probs.tr)
## [1] 104
glm.pred=rep("Down", length(probs.tr))
glm.pred[probs.tr>0.5]<-"Up"
length(glm.pred)
## [1] 104
tab <- table(glm.pred, Direction.test)
tab
##         Direction.test
## glm.pred Down Up
##     Down    9  5
##     Up     34 56
sum(diag(tab))/sum(tab)
## [1] 0.625
```

Using LDA, we get

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

lda.tr
## 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.tr, test[,-9], type = "response")$class
#length(lda.pred)
tab <- table(lda.pred, Direction.test)
tab
##         Direction.test
## lda.pred Down Up
##     Down    9  5
##     Up     34 56
sum(diag(tab))/sum(tab)
## [1] 0.625
```

For QDA implementation, we just change the command above.

```
qda.tr <- qda(Direction ~ Lag2, data = Weekly, subset=train)

qda.tr
## 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
qda.pred <- predict(qda.tr, test[,-9], type = "response")$class
#length(lda.pred)
tab <- table(qda.pred, Direction.test)
tab
##         Direction.test
## qda.pred Down Up
##     Down    0  0
##     Up     43 61
sum(diag(tab))/sum(tab)
## [1] 0.5865385
```

For KNN, we need to remember that the syntax is slightly different in that the training and the testing are immediately input into the command.

```
knn.fit <- knn(Weekly[train, "Lag2", drop = FALSE],
               Weekly[!train, "Lag2", drop = FALSE],
               Weekly$Direction[train])

tab=table(knn.fit, Direction.test)
tab
##        Direction.test
## knn.fit Down Up
```

```
##    Down   21 29
##    Up     22 32
sum(diag(tab))/sum(tab)
## [1] 0.5096154
```

The logistic regression and the LDA - despite being the simplest - perform the best.

For the remainder of the project, solutions will vary.