JieTang_Lab6.R

tjj

2020-08-04

```
#Name:Jie Tang
#Course:Machine learning using R 374815
#Quarter:Summer
#Instructor name : Michael Chang

#Quiz part

library(ISLR)
names(Smarket)
```

```
## [1] "Year"      "Lag1"      "Lag2"      "Lag3"      "Lag4"      "Lag5"
## [7] "Volume"    "Today"     "Direction"
```

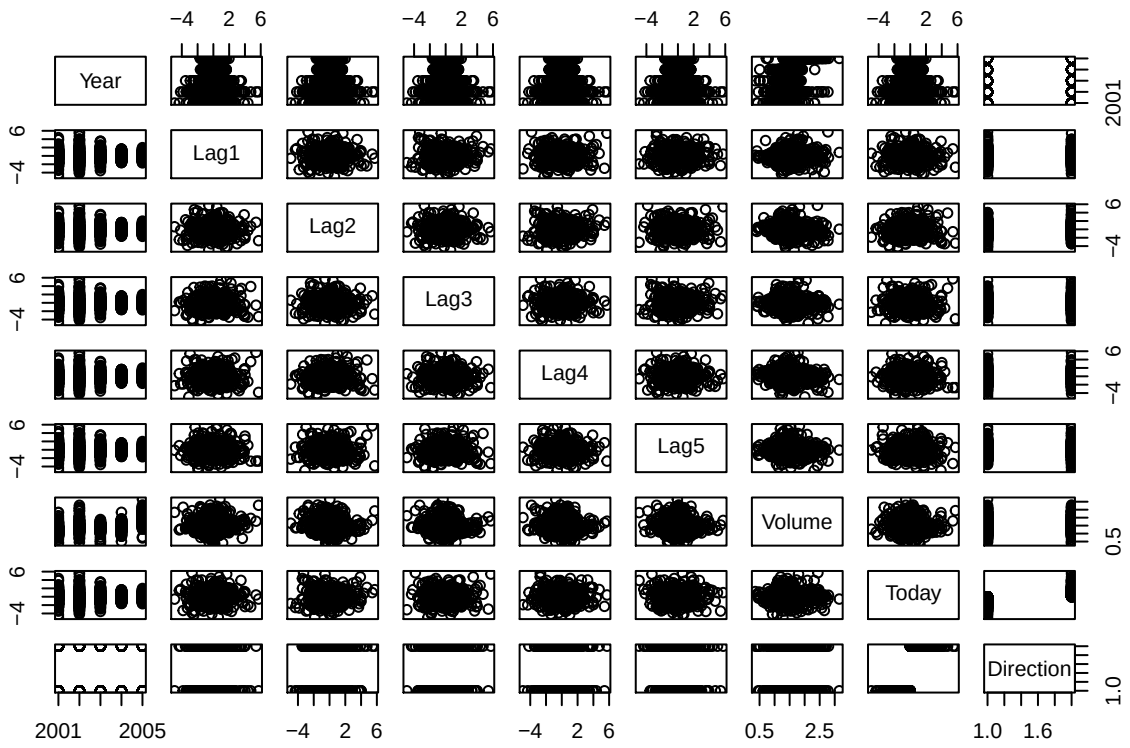
```
dim(Smarket)
```

```
## [1] 1250    9
```

```
summary(Smarket)
```

```
##       Year           Lag1                Lag2                Lag3
##  Min.   :2001   Min.   :-4.922000   Min.   :-4.922000   Min.   :-4.922000
##  1st Qu.:2002   1st Qu.:-0.639500   1st Qu.:-0.639500   1st Qu.:-0.640000
##  Median :2003   Median : 0.039000   Median : 0.039000   Median : 0.038500
##  Mean   :2003   Mean   : 0.003834   Mean   : 0.003919   Mean   : 0.001716
##  3rd Qu.:2004   3rd Qu.: 0.596750   3rd Qu.: 0.596750   3rd Qu.: 0.596750
##  Max.   :2005   Max.   : 5.733000   Max.   : 5.733000   Max.   : 5.733000
##       Lag4                Lag5              Volume           Today
##  Min.   :-4.922000   Min.   :-4.92200   Min.   :0.3561   Min.   :-4.922000
##  1st Qu.:-0.640000   1st Qu.:-0.64000   1st Qu.:1.2574   1st Qu.:-0.639500
##  Median : 0.038500   Median : 0.03850   Median :1.4229   Median : 0.038500
##  Mean   : 0.001636   Mean   : 0.00561   Mean   :1.4783   Mean   : 0.003138
##  3rd Qu.: 0.596750   3rd Qu.: 0.59700   3rd Qu.:1.6417   3rd Qu.: 0.596750
##  Max.   : 5.733000   Max.   : 5.73300   Max.   :3.1525   Max.   : 5.733000
##  Direction
##  Down:602
##  Up  :648
##
##
##
##
```

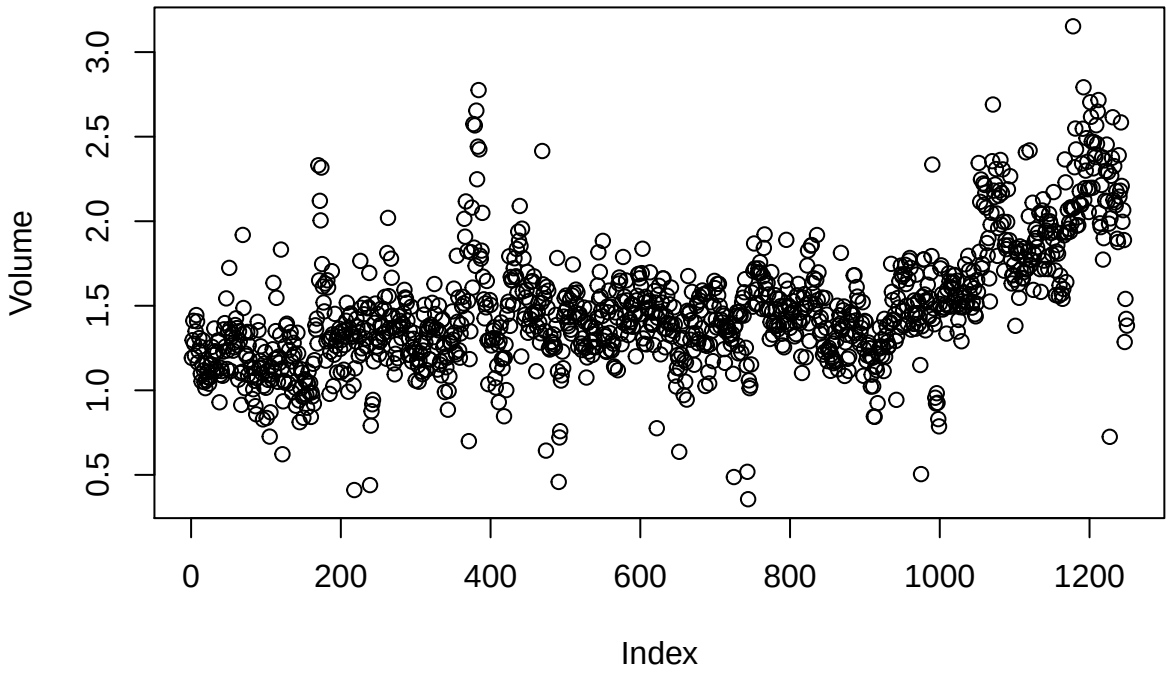
```
pairs(Smarket)
```




```
#cor(Smarket)
cor(Smarket[,-9])
```

```
##              Year         Lag1         Lag2         Lag3         Lag4
## Year   1.00000000  0.029699649  0.030596422  0.033194581  0.035688718
## Lag1   0.02969965  1.000000000 -0.026294328 -0.010803402 -0.002985911
## Lag2   0.03059642 -0.026294328  1.000000000 -0.025896670 -0.010853533
## Lag3   0.03319458 -0.010803402 -0.025896670  1.000000000 -0.024051036
## Lag4   0.03568872 -0.002985911 -0.010853533 -0.024051036  1.000000000
## Lag5   0.02978799 -0.005674606 -0.003557949 -0.018808338 -0.027083641
## Volume 0.53900647  0.040909908 -0.043383215 -0.041823686 -0.048414246
## Today  0.03009523 -0.026155045 -0.010250033 -0.002447647 -0.006899527
##                Lag5      Volume        Today
## Year    0.029787995  0.53900647  0.030095229
## Lag1   -0.005674606  0.04090991 -0.026155045
## Lag2   -0.003557949 -0.04338321 -0.010250033
## Lag3   -0.018808338 -0.04182369 -0.002447647
## Lag4   -0.027083641 -0.04841425 -0.006899527
## Lag5    1.000000000 -0.02200231 -0.034860083
## Volume -0.022002315  1.00000000  0.014591823
## Today  -0.034860083  0.01459182  1.000000000
```

```
attach(Smarket)
plot(Volume)
```




```
#q1 check by coef and summary functions
train=(Year<2005)
glm.fits=glm(Direction~Lag1+Lag2,data=Smarket,family=binomial,subset=train)
summary(glm.fits)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2, family = binomial, data = Smarket,
##     subset = train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.345  -1.188   1.074   1.164   1.326
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.03222    0.06338   0.508    0.611
## Lag1        -0.05562    0.05171  -1.076    0.282
## Lag2        -0.04449    0.05166  -0.861    0.389
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 1383.3  on 997  degrees of freedom
## Residual deviance: 1381.4  on 995  degrees of freedom
## AIC: 1387.4
##
## Number of Fisher Scoring iterations: 3
```

```
coef(glm.fits)
```

```
## (Intercept)        Lag1        Lag2
##  0.03221753 -0.05562122 -0.04448684
```

```
#Q2 use glm model get the predictions. use table() to create confusion matrix
#calculate the accuracy
train=(Year<2005)
Smarket.2005=Smarket[!train,]
dim(Smarket.2005)
```

```
## [1] 252   9
```

```
Direction.2005=Direction[!train]
glm.probs=predict(glm.fits,Smarket.2005,type="response")
glm.pred=rep("Down",252)
glm.pred[glm.probs>.5]="Up"
table(glm.pred,Direction.2005)
```

```
##         Direction.2005
## glm.pred Down  Up
##     Down   35  35
##     Up     76 106
```

```
mean(glm.pred==Direction.2005)
```

```
## [1] 0.5595238
```

```
(106+35)/(106+76+35+35)
```

```
## [1] 0.5595238
```

```
#Q3
#calculate correct negative predictions (TN) divided by the total number of negatives (N).
table(glm.pred,Direction.2005)
```

```
##         Direction.2005
## glm.pred Down  Up
##     Down   35  35
##     Up     76 106
```

```r
106/(106+35)
```

```
## [1] 0.751773
```

```r
1 - (76/(35+76))
```

```
## [1] 0.3153153
```

```r
#Q4
#use max and min to get the value of posterior under "UP" perdiction.
library(MASS)
lda.fit=lda(Direction~Lag1+Lag2,data=Smarket,subset=train)
lda.fit
```

```
## Call:
## lda(Direction ~ Lag1 + Lag2, data = Smarket, subset = train)
##
## Prior probabilities of groups:
##     Down       Up
## 0.491984 0.508016
##
## Group means:
##             Lag1        Lag2
## Down  0.04279022  0.03389409
## Up   -0.03954635 -0.03132544
##
## Coefficients of linear discriminants:
##             LD1
## Lag1 -0.6420190
## Lag2 -0.5135293
```

```r
plot(lda.fit)
lda.pred=predict(lda.fit, Smarket.2005)
names(lda.pred)
```

```
## [1] "class"     "posterior" "x"
```

```r
max(lda.pred$posterior[1:252,2])
```

```
## [1] 0.5422133
```

```r
min(lda.pred$posterior[1:252,2])
```

```
## [1] 0.479765
```

```r
#Q5  correct positive prediction rate sensitivity
qda.fit=qda(Direction~Lag1+Lag2,data=Smarket,subset=train)
qda.fit
```

```
## Call:
## qda(Direction ~ Lag1 + Lag2, data = Smarket, subset = train)
##
## Prior probabilities of groups:
##     Down       Up
## 0.491984 0.508016
##
## Group means:
##             Lag1        Lag2
## Down  0.04279022  0.03389409
## Up   -0.03954635 -0.03132544
```

```
qda.class=predict(qda.fit,Smarket.2005)$class
table(qda.class,Direction.2005)
```

```
##          Direction.2005
## qda.class Down  Up
##      Down   30  20
##      Up     81 121
```

```
121/(121+20)
```

```
## [1] 0.858156
```

```
mean(qda.class==Direction.2005)
```

```
## [1] 0.5992063
```

```
#Q6
#calculate type 1 error and type 2 error
library(class)
train.X=cbind(Lag1,Lag2)[train,]
test.X=cbind(Lag1,Lag2)[!train,]
train.Direction=Direction[train]
RNGkind(sample.kind = "Rounding")
```

```
## Warning in RNGkind(sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
set.seed(1)
RNGkind(sample.kind = "Rounding")
```

```
## Warning in RNGkind(sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
knn.pred=knn(train.X,test.X,train.Direction,k=4)
table(knn.pred,Direction.2005)
```

```
##         Direction.2005
## knn.pred Down Up
##     Down   45 58
##     Up     66 83
```

```r
#Type 1 error
66/(45+66)
```

```
## [1] 0.5945946
```

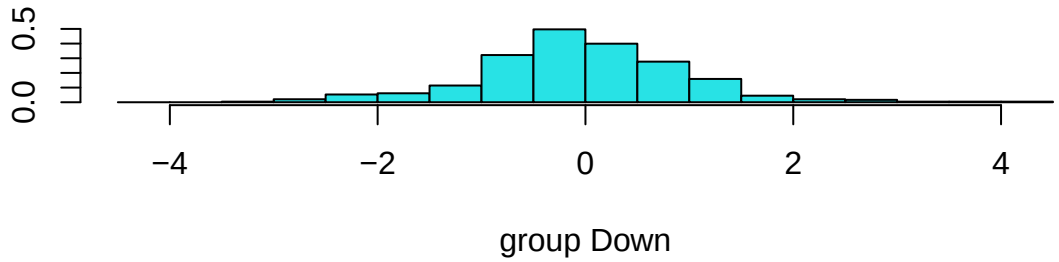
```r
#Type 2 error
58/(58+83)
```

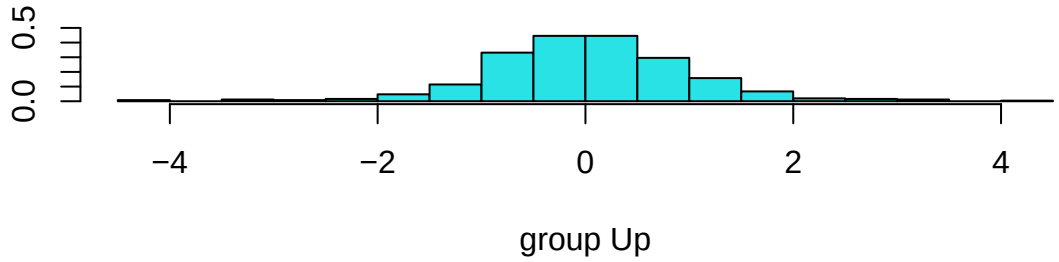
```
## [1] 0.4113475
```

```r
#Q7
test=1:1000
library(MASS)
lda.fit=lda(Direction~Lag1+Lag2,data=Smarket,subset=train)
lda.fit
```

```
## Call:
## lda(Direction ~ Lag1 + Lag2, data = Smarket, subset = train)
##
## Prior probabilities of groups:
##     Down       Up
## 0.491984 0.508016
##
## Group means:
##             Lag1        Lag2
## Down  0.04279022  0.03389409
## Up   -0.03954635 -0.03132544
##
## Coefficients of linear discriminants:
##             LD1
## Lag1 -0.6420190
## Lag2 -0.5135293
```

```r
plot(lda.fit)
```






```
lda.pred=predict(lda.fit, Smarket.2005)
lda.fit=lda(Purchase~.,data=Caravan,subset=-test)
lda.probs=predict(lda.fit, Caravan[test,])$posterior[,2]
lda.class=lda.pred$class
table(lda.class,Direction.2005)
```

```
##          Direction.2005
## lda.class Down  Up
##      Down   35  35
##      Up     76 106
```

```
sum(lda.pred$posterior[,1]>=.2)
```

```
## [1] 252
```

```
sum(lda.pred$posterior[,1]<.2)
```

```
## [1] 0
```

```
lda.pred$posterior[1:20,1]
```

```
##       999      1000      1001      1002      1003      1004      1005      1006
## 0.4901792 0.4792185 0.4668185 0.4740011 0.4927877 0.4938562 0.4951016 0.4872861
```

```
##      1007      1008      1009      1010      1011      1012      1013      1014
## 0.4907013 0.4844026 0.4906963 0.5119988 0.4895152 0.4706761 0.4744593 0.4799583
##      1015      1016      1017      1018
## 0.4935775 0.5030894 0.4978806 0.4886331
```

```
lda.class[1:20]
```

```
##  [1] Up   Up   Up   Up   Up   Up   Up   Up   Up   Up   Up   Down Up   Up   Up
## [16] Up   Up   Down Up   Up
## Levels: Down Up
```

```
sum(lda.pred$posterior[,1]>=.3)
```

```
## [1] 252
```

```
sum(lda.pred$posterior[,1]<.3)
```

```
## [1] 0
```

```
# Chapter 4 Lab: Logistic Regression, LDA, QDA, and KNN

# The Stock Market Data
#
# library(ISLR)
# names(Smarket)
# dim(Smarket)
# summary(Smarket)
# pairs(Smarket)
# #cor(Smarket)
# cor(Smarket[,-9])
# attach(Smarket)
# plot(Volume)
#
# # Logistic Regression
#
# glm.fits=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,data=Smarket,family=binomial)
# summary(glm.fits)
# coef(glm.fits)
# summary(glm.fits)$coef
# summary(glm.fits)$coef[,4]
# glm.probs=predict(glm.fits,type="response")
# glm.probs[1:10]
# contrasts(Direction)
# glm.pred=rep("Down",1250)
# glm.pred[glm.probs>.5]="Up"
# table(glm.pred,Direction)
# (507+145)/1250
# mean(glm.pred==Direction)
# train=(Year<2005)
# Smarket.2005=Smarket[!train,]
# dim(Smarket.2005)
```

```
# Direction.2005=Direction[!train]
# glm.fits=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,data=Smarket,family=binomial,subset=train)
# glm.probs=predict(glm.fits,Smarket.2005,type="response")
# glm.pred=rep("Down",252)
# glm.pred[glm.probs>.5]="Up"
# table(glm.pred,Direction.2005)
# mean(glm.pred==Direction.2005)
# mean(glm.pred!=Direction.2005)
# glm.fits=glm(Direction~Lag1+Lag2,data=Smarket,family=binomial,subset=train)
# glm.probs=predict(glm.fits,Smarket.2005,type="response")
# glm.pred=rep("Down",252)
# glm.pred[glm.probs>.5]="Up"
# table(glm.pred,Direction.2005)
# mean(glm.pred==Direction.2005)
# 106/(106+76)
# predict(glm.fits,newdata=data.frame(Lag1=c(1.2,1.5),Lag2=c(1.1,-0.8)),type="response")
#
# # Linear Discriminant Analysis
#
# library(MASS)
# lda.fit=lda(Direction~Lag1+Lag2,data=Smarket,subset=train)
# lda.fit
# plot(lda.fit)
# lda.pred=predict(lda.fit, Smarket.2005)
# names(lda.pred)
# lda.class=lda.pred$class
# table(lda.class,Direction.2005)
# mean(lda.class==Direction.2005)
# sum(lda.pred$posterior[,1]>=.5)
# sum(lda.pred$posterior[,1]<.5)
# lda.pred$posterior[1:20,1]
# lda.class[1:20]
# sum(lda.pred$posterior[,1]>.9)
#
# # Quadratic Discriminant Analysis
#
# qda.fit=qda(Direction~Lag1+Lag2,data=Smarket,subset=train)
# qda.fit
# qda.class=predict(qda.fit,Smarket.2005)$class
# table(qda.class,Direction.2005)
# mean(qda.class==Direction.2005)
#
# # K-Nearest Neighbors
#
# library(class)
# train.X=cbind(Lag1,Lag2)[train,]
# test.X=cbind(Lag1,Lag2)[!train,]
# train.Direction=Direction[train]
# set.seed(1)
# knn.pred=knn(train.X,test.X,train.Direction,k=1)
# table(knn.pred,Direction.2005)
# (83+43)/252
# knn.pred=knn(train.X,test.X,train.Direction,k=3)
```

```
# table(knn.pred,Direction.2005)
# mean(knn.pred==Direction.2005)
#
# # An Application to Caravan Insurance Data
#
# dim(Caravan)
# attach(Caravan)
# summary(Purchase)
# 348/5822
# standardized.X=scale(Caravan[,-86])
# var(Caravan[,1])
# var(Caravan[,2])
# var(standardized.X[,1])
# var(standardized.X[,2])
# test=1:1000
# train.X=standardized.X[-test,]
# test.X=standardized.X[test,]
# train.Y=Purchase[-test]
# test.Y=Purchase[test]
# set.seed(1)
# knn.pred=knn(train.X,test.X,train.Y,k=1)
# mean(test.Y!=knn.pred)
# mean(test.Y!="No")
# table(knn.pred,test.Y)
# 9/(68+9)
# knn.pred=knn(train.X,test.X,train.Y,k=3)
# table(knn.pred,test.Y)
# 5/26
# knn.pred=knn(train.X,test.X,train.Y,k=5)
# table(knn.pred,test.Y)
# 4/15
# glm.fits=glm(Purchase~.,data=Caravan,family=binomial,subset=-test)
# glm.probs=predict(glm.fits,Caravan[test,],type="response")
# glm.pred=rep("No",1000)
# glm.pred[glm.probs>.5]="Yes"
# table(glm.pred,test.Y)
# glm.pred=rep("No",1000)
# glm.pred[glm.probs>.25]="Yes"
# table(glm.pred,test.Y)
# 11/(22+11)
```