Code

HW-2 Classification: Piyaporn Puangprasert

Class: 33:136:487:01 LG SCALE DATA ANALY

Prof. Jin Wang

Piyaporn Puangprasert(pp712) team with Steven Panagakos (He may sent separate homework)

Due date: Mar 3, 2024

---

read the Weekly file

Hide

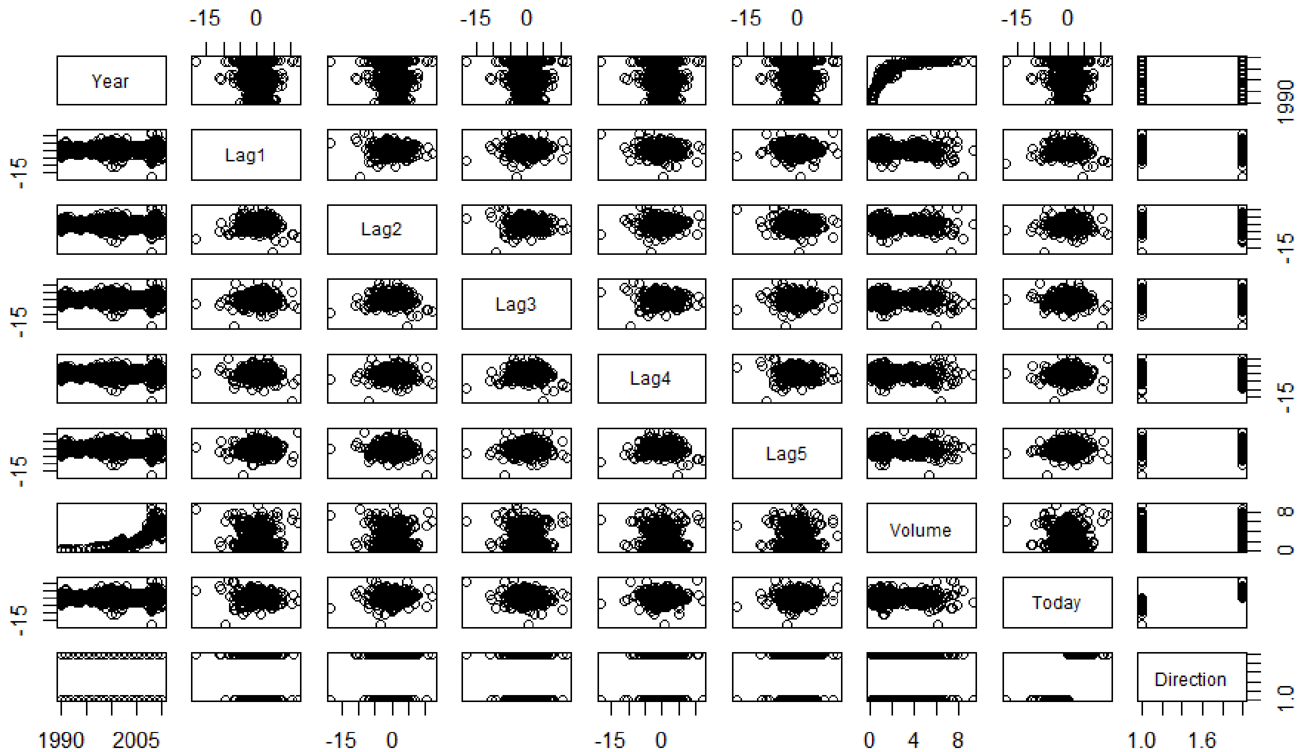
```
library(ISLR2)
# load "Weekly" data set
attach(Weekly)
```

```
The following objects are masked from Weekly (pos = 3):

    Direction, Lag1, Lag2, Lag3,
    Lag4, Lag5, Today, Volume, Year
```

Hide

```
pairs(Weekly)
```



Hide

```
names(Weekly)
```

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

Veiw full Weekly data set

Hide

```
View(Weekly)
```

model weeklyview: Direction with 5 lag

Hide

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

```
Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
    Volume, family = binomial, data = Weekly)

Deviance Residuals:
    Min       1Q   Median       3Q
-1.6949  -1.2565   0.9913   1.0849
    Max
 1.4579

Coefficients:
            Estimate Std. Error z value
(Intercept)  0.26686    0.08593   3.106
Lag1        -0.04127    0.02641  -1.563
Lag2         0.05844    0.02686   2.175
Lag3        -0.01606    0.02666  -0.602
Lag4        -0.02779    0.02646  -1.050
Lag5        -0.01447    0.02638  -0.549
Volume      -0.02274    0.03690  -0.616
            Pr(>|z|)
(Intercept)   0.0019 **
Lag1          0.1181
Lag2          0.0296 *
Lag3          0.5469
Lag4          0.2937
Lag5          0.5833
Volume        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
```

Answer I. Base on this result, only Lag 2 appears to be statistically significant because of the p-value < 0.05

II.Compute the Comfusion matrix and overall fraction of correct predictions

Hide

```
predicted_dir <- ifelse(predict(WeeklyView, type = "response") > 0.5, "Up", "Down")
conf_matrix <- table(predicted_dir, Weekly$Direction)
conf_matrix
```

```
predicted_dir Down  Up
         Down   54  48
         Up    430 557
```

Hide

```
overall_correct <- sum(diag(conf_matrix)) / sum(conf_matrix)
overall_correct
```

```
[1] 0.5610652
```

The confusion matric can showsthe type of mistakes make by the logistic regression model. It shows ho w correct classify Up or Down. /the identifies type of errors made by model that can be the wrong class. For example this True Positive (TP is up 557), True Negatives(TN)is “Down” 54 times when the market down. False Positive(FT) shows incorrect predicted at 48, and False Negative is Down 430

Calculate prediction =(57+557)/(54 + 48 + 430+557) = 0.56

Hide

```
(557+54)/(54+48+430+557)
```

```
[1] 0.5610652
```

the Up trends = 557/(48+557) = 0.92

Hide

```
557/(557+48)
```

```
[1] 0.9206612
```

the Down trend = 54/(430+54)

Hide

```
54/(54+430)
```

```
[1] 0.1115702
```

2. Divide the full data set and Training set

Hide

```
train_data = Weekly[1:900,]
test_data = Weekly [901:nrow(Weekly),]
```

2.2 Fit the logistic regression model using the training data set with Lag2 as only precidtor

Hide

```
train_model = glm(Direction~ Lag2, data = train_data, family = binomial)
```

2.3 Comput the confusion matrix and overall fraction of the test data

confusion matrix

Hide

```
# glm.all
glm.all = glm(Direction~., data = Weekly, family = "binomial")
```

```
Warning: glm.fit: algorithm did not convergeWarning: glm.fit: fitted probabilities numerically 0
or 1 occurred
```

Hide

```
# get predict probability
prob = predict(glm.all, newdata = Weekly, type = 'response')


pred = rep('Down', nrow(Weekly))
pred[prob>0.5]= 'Up'

# confusion matrix
table(pred,Weekly$Direction)
```

```
pred   Down  Up
  Down  484   0
  Up      0 605
```

from outsource

Hide

```
test_prob <- predict(train_model, newdata = test_data, type = "response")
test_predicted_direction <- ifelse(test_prob > 0.5, "Up", "Down")

test_conf_matrix <- table(test_predicted_direction, test_data$Direction)
test_overall_correct <- sum(diag(test_conf_matrix)) / sum(test_conf_matrix)

test_overall_correct
```

```
[1] 0.5396825
```

2.4

Hide

```
thresholds <- c(0.52, 0.53, 0.54)
best_correct <- 0
best_threshold <- 0

for (threshold in thresholds) {
  train_preds <- ifelse(predict(train_model, type = "response") > threshold, "Up", "Down")
  train_conf_matrix <- table(train_preds, train_data$Direction)
  train_correct <- sum(diag(train_conf_matrix)) / sum(train_conf_matrix)

  if (train_correct > best_correct) {
    best_correct <- train_correct
    best_threshold <- threshold
  }
}

best_threshold
```

```
[1] 0.53
```

Hide

```

```

That mean the 0.53 threshold gives the best result.

2.5

Hide

```
test_preds <- ifelse(predict(train_model, newdata = test_data, type = "response") > best_thresho
ld, "Up", "Down")
new_test_conf_matrix <- table(test_preds, test_data$Direction)
new_test_overall_correct <- sum(diag(new_test_conf_matrix)) / sum(new_test_conf_matrix)

new_test_overall_correct
```

```
[1] 0.5767196
```

3.1

Hide

```
library(MASS)
lda_model <- lda(Direction ~ Lag2, data = train_data)
```

3.2

Hide

```
lda_test_pred <- predict(lda_model, newdata = test_data)$class
lda_test_conf_matrix <- table(lda_test_pred, test_data$Direction)
lda_test_overall_correct <- sum(diag(lda_test_conf_matrix)) / sum(lda_test_conf_matrix)
lda_test_overall_correct
```

```
[1] 0.5449735
```

test_overall_correct is 0.5396835 and lda_test_overall_correct is 0.5449735 lad_test have more percentate correction prediction.