Homework 5

Chapter 5, Exercise 7

In Sections 5.3.2 and 5.3.3, we saw that the `cv.glm()` function can be used in order to compute the LOOCV test error estimate. Alternatively, one could compute those quantities using just the `glm()` and `predict.glm()` functions, and a `for` loop. You will now take this approach in order to compute the LOOCV error for a simple logistic regression model on the Weekly data set. Recall that in the context of classification problems, the LOOCV error is given in (5.4).

```
library(ISLR)
# summary(Weekly)
set.seed(1)
attach(Weekly)
```

```
glm.fit = glm(Direction~Lag1+Lag2, data=Weekly, family=binomial)
summary(glm.fit)
```

(a) Fit a logistic regression model that predicts Direction using Lag1 and Lag2.

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2, family = binomial, data = Weekly)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.623  -1.261   1.001   1.083   1.506
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.22122    0.06147   3.599 0.000319 ***
## Lag1        -0.03872    0.02622  -1.477 0.139672
## Lag2         0.06025    0.02655   2.270 0.023232 *
## ---
## 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: 1488.2  on 1086  degrees of freedom
## AIC: 1494.2
##
## Number of Fisher Scoring iterations: 4
```

```
glm.fit = glm(Direction~Lag1+Lag2, data=Weekly[-1,], family=binomial)
summary(glm.fit)
```

(b) Fit a logistic regression model that predicts Direction using Lag1 and Lag2 using all but the first observation.

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2, family = binomial, data = Weekly[-1,
##     ])
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6258  -1.2617   0.9999   1.0819   1.5071
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.22324    0.06150   3.630 0.000283 ***
## Lag1        -0.03843    0.02622  -1.466 0.142683
## Lag2         0.06085    0.02656   2.291 0.021971 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1494.6  on 1087  degrees of freedom
## Residual deviance: 1486.5  on 1085  degrees of freedom
## AIC: 1492.5
##
## Number of Fisher Scoring iterations: 4
```

```
predict.glm(glm.fit, Weekly[1,], type="response") > 0.5
```

(c) Use the model from (b) to predict the direction of the first observation. You can do this by predicting that the first observation will go up if P (Direction = "Up"| Lag1, Lag2) > 0.5. Was this observation correctly classified?

```
##    1
## TRUE
```

Prediction was UP, true Direction was DOWN.

(d) Write a for loop from i = 1 to i = n, where n is the number of observations in the data set, that performs each of the following steps: i. Fit a logistic regression model using all but the ith observation to predict Direction using Lag1 and Lag2.

ii. Compute the posterior probability of the market moving up for the ith observation.

iii. Use the posterior probability for the ith observation in order to predict whether or not the market moves up.

iv. Determine whether or not an error was made in predicting the direction for the ith observation. If an error was made, then indicate this as a 1, and otherwise indicate it as a 0.

```
count = rep(0, dim(Weekly)[1])
for (i in 1:(dim(Weekly)[1])) {
   glm.fit = glm(Direction~Lag1+Lag2, data=Weekly[-i,], family=binomial)
   is_up = predict.glm(glm.fit, Weekly[i,], type="response") > 0.5
   is_true_up = Weekly[i,]$Direction == "Up"
   if (is_up != is_true_up)
     count[i] = 1
}
sum(count)
```

```
## [1] 490
```

There are 490 errors total counted with LOOCV.

```
mean(count)
```

(e) Take the average of the n numbers obtained in (d)iv in order to obtain the LOOCV estimate for the test error. Comment on the results.

```
## [1] 0.4499541
```

LOOCV estimates a test error rate of 45%.

```
N = dim(Weekly)[1]
count = rep(0,N)
for (i in 1:N) {
   glm.fit = glm(Direction~Lag1+Lag2, data=Weekly[-i,], family=binomial)
   is_up = predict.glm(glm.fit, Weekly[i,], type="response") > 0.5
   is_true_up = Weekly[i,]$Direction == "Up"
   if (is_up != is_true_up)
     count[i] = 1
}

mean(count)
```

(f) Find the 95% bootstrap confidence interval for the accuracy metric you derived in part (e).

```
## [1] 0.4499541
```

LOOCV estimates a test error rate of 45%.