ica8_shuangyu_zhao

shuangyu_zhao

2023-02-02

```r
library(ISLR2)
oj <- OJ
head(oj)
```

```
##   Purchase WeekofPurchase StoreID PriceCH PriceMM DiscCH DiscMM SpecialCH
## 1       CH            237       1    1.75    1.99   0.00    0.0         0
## 2       CH            239       1    1.75    1.99   0.00    0.3         0
## 3       CH            245       1    1.86    2.09   0.17    0.0         0
## 4       MM            227       1    1.69    1.69   0.00    0.0         0
## 5       CH            228       7    1.69    1.69   0.00    0.0         0
## 6       CH            230       7    1.69    1.99   0.00    0.0         0
##   SpecialMM  LoyalCH SalePriceMM SalePriceCH PriceDiff Store7 PctDiscMM
## 1         0 0.500000        1.99        1.75      0.24     No  0.000000
## 2         1 0.600000        1.69        1.75     -0.06     No  0.150754
## 3         0 0.680000        2.09        1.69      0.40     No  0.000000
## 4         0 0.400000        1.69        1.69      0.00     No  0.000000
## 5         0 0.956535        1.69        1.69      0.00    Yes  0.000000
## 6         1 0.965228        1.99        1.69      0.30    Yes  0.000000
##   PctDiscCH ListPriceDiff STORE
## 1  0.000000          0.24     1
## 2  0.000000          0.24     1
## 3  0.091398          0.23     1
## 4  0.000000          0.00     1
## 5  0.000000          0.00     0
## 6  0.000000          0.30     0
```

1.

```r
oj$target <- ifelse(oj$Purchase=="CH",1,0)
head(oj)
```

```
##   Purchase WeekofPurchase StoreID PriceCH PriceMM DiscCH DiscMM SpecialCH
## 1       CH            237       1    1.75    1.99   0.00    0.0         0
## 2       CH            239       1    1.75    1.99   0.00    0.3         0
## 3       CH            245       1    1.86    2.09   0.17    0.0         0
## 4       MM            227       1    1.69    1.69   0.00    0.0         0
## 5       CH            228       7    1.69    1.69   0.00    0.0         0
## 6       CH            230       7    1.69    1.99   0.00    0.0         0
##   SpecialMM  LoyalCH SalePriceMM SalePriceCH PriceDiff Store7 PctDiscMM
## 1         0 0.500000        1.99        1.75      0.24     No  0.000000
## 2         1 0.600000        1.69        1.75     -0.06     No  0.150754
## 3         0 0.680000        2.09        1.69      0.40     No  0.000000
## 4         0 0.400000        1.69        1.69      0.00     No  0.000000
## 5         0 0.956535        1.69        1.69      0.00    Yes  0.000000
## 6         1 0.965228        1.99        1.69      0.30    Yes  0.000000
##   PctDiscCH ListPriceDiff STORE target
## 1  0.000000          0.24     1      1
## 2  0.000000          0.24     1      1
## 3  0.091398          0.23     1      1
## 4  0.000000          0.00     1      0
## 5  0.000000          0.00     0      1
## 6  0.000000          0.30     0      1
```

```
# CH--1. MM--0
```

2.

```
split_pro <- 0.75
n <- length(oj$Purchase)*split_pro
row_samp <- sample(1:length(oj$Purchase), n, replace = FALSE)
train <- oj[row_samp,]
test <- oj[-row_samp,]
```

3.

```
mod <- glm(data = train, target ~ PriceDiff + LoyalCH, family = binomial)
summary(mod)
```

```
##
## Call:
## glm(formula = target ~ PriceDiff + LoyalCH, family = binomial,
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8774  -0.5240   0.2314   0.5612   2.7712
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.3255     0.2643 -12.582  < 2e-16 ***
## PriceDiff     2.8322     0.4003   7.076 1.49e-12 ***
## LoyalCH       6.6539     0.4628  14.377  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1067.33  on 801  degrees of freedom
## Residual deviance:  619.65  on 799  degrees of freedom
## AIC: 625.65
##
## Number of Fisher Scoring iterations: 5
```

a. they are all significant enough

b.

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
confusionMatrix(data = as.factor(as.integer(2*mod$fitted.values)), reference = as.factor(train$target))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 229  62
##          1  78 433
##
##                Accuracy : 0.8254
##                  95% CI : (0.7974, 0.8511)
##     No Information Rate : 0.6172
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6269
##
##  Mcnemar’s Test P-Value : 0.2049
##
##             Sensitivity : 0.7459
##             Specificity : 0.8747
##          Pos Pred Value : 0.7869
##          Neg Pred Value : 0.8474
##              Prevalence : 0.3828
##          Detection Rate : 0.2855
##    Detection Prevalence : 0.3628
##       Balanced Accuracy : 0.8103
##
##        ’Positive’ Class : 0
##
```

```
prediction <- predict(mod, test, type = "response")
```

```
confusionMatrix(data = as.factor(as.integer(2*prediction)), reference = as.factor(test$target))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  82  18
##          1  28 140
##
```

```
##                Accuracy : 0.8284
##                  95% CI : (0.7778, 0.8715)
##     No Information Rate : 0.5896
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6404
##
##  Mcnemar’s Test P-Value : 0.1845
##
##             Sensitivity : 0.7455
##             Specificity : 0.8861
##          Pos Pred Value : 0.8200
##          Neg Pred Value : 0.8333
##              Prevalence : 0.4104
##          Detection Rate : 0.3060
##    Detection Prevalence : 0.3731
##       Balanced Accuracy : 0.8158
##
##        ’Positive’ Class : 0
##
```

c.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## v purrr   0.3.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()
```

```
df <- data.frame('y' = mod$y, 'fit' = mod$fitted.values)
calib <- data.frame('count' = numeric(0), 'bin' = numeric(0), 'prob' = numeric(0))
for(i in 1:10){
  temp <- filter(df, fit > (i-1)/10 & fit < i/10)
  calib[nrow(calib) + 1,]$count <- nrow(temp)
  calib[nrow(calib),]$bin <- (i - .5)/10
  calib[nrow(calib),]$prob <- mean(temp$y)
}
calib
```

```
##    count  bin       prob
## NA    97 0.05 0.08247423
## 2     59 0.15 0.08474576
## 3     46 0.25 0.28260870
## 4     49 0.35 0.32653061
## 5     40 0.45 0.50000000
## 6     43 0.55 0.51162791
## 7      60 0.65 0.65000000
## 8      57 0.75 0.73684211
## 9      85 0.85 0.83529412
## 10    266 0.95 0.97368421
```

4.

```
coeff1 <- rep(0, 1000)
coeff2 <- rep(0, 1000)
n <- nrow(oj)
for(i in 1:1000){
  row_samp <- sample(1:n, replace = TRUE)
  oj_samp <- oj[row_samp,]
  temp_mod <- glm(data = oj_samp, target ~ PriceDiff  + LoyalCH, family = binomial)
  coeff1[i] <- temp_mod$coefficients[2]
  coeff2[i] <- temp_mod$coefficients[3]
}
quantile(coeff1, c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 2.194292 3.572976
```

```
quantile(coeff2, c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 5.688211 7.268618
```