ica9_shuangyu_zhao

shuangyu_zhao

2023-02-07

```
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.

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

2.

```
library(e1071)
mod <- naiveBayes(data = train, Purchase ~ PriceDiff + LoyalCH)
summary(mod)
```

```
##           Length Class  Mode
## apriori   2      table  numeric
## tables    2      -none- list
## levels    2      -none- character
## isnumeric 2      -none- logical
## call      4      -none- call
```

```
# confusion matrix for train
library(caret)
```

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

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

```
train_predict <- predict(mod, train)
confusionMatrix(data = as.factor(train_predict), reference = as.factor(train$Purchase))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  CH  MM
##         CH 432  76
##         MM  64 230
##
##                Accuracy : 0.8254
##                  95% CI : (0.7974, 0.8511)
##     No Information Rate : 0.6185
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6273
##
##  Mcnemar's Test P-Value : 0.3525
##
##             Sensitivity : 0.8710
##             Specificity : 0.7516
##          Pos Pred Value : 0.8504
##          Neg Pred Value : 0.7823
##              Prevalence : 0.6185
##          Detection Rate : 0.5387
##    Detection Prevalence : 0.6334
##       Balanced Accuracy : 0.8113
##
##        'Positive' Class : CH
##
```

```
# confusion matrix of test data
```

```
test_predict <- predict(mod, test)
confusionMatrix(data = as.factor(test_predict), reference = as.factor(test$Purchase))
```

```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction  CH  MM
##         CH 138  28
##         MM  19  83
##
##                Accuracy : 0.8246
##                  95% CI : (0.7737, 0.8682)
##     No Information Rate : 0.5858
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6343
##
##  Mcnemar's Test P-Value : 0.2432
##
##             Sensitivity : 0.8790
##             Specificity : 0.7477
##          Pos Pred Value : 0.8313
##          Neg Pred Value : 0.8137
##              Prevalence : 0.5858
##          Detection Rate : 0.5149
##    Detection Prevalence : 0.6194
##       Balanced Accuracy : 0.8134
##
##        'Positive' Class : CH
##
```

3.

```
mod2 <- naiveBayes(data = train, Purchase~STORE+PriceDiff + LoyalCH)
```

```
train_predict2 <- predict(mod2, train)
confusionMatrix(data = as.factor(train_predict2), reference = as.factor(train$Purchase))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  CH  MM
##         CH 437  74
##         MM  59 232
##
##                Accuracy : 0.8342
##                  95% CI : (0.8066, 0.8593)
##     No Information Rate : 0.6185
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6453
##
##  Mcnemar's Test P-Value : 0.2248
##
##             Sensitivity : 0.8810
##             Specificity : 0.7582
##          Pos Pred Value : 0.8552
```

```
##          Neg Pred Value : 0.7973
##              Prevalence : 0.6185
##          Detection Rate : 0.5449
##    Detection Prevalence : 0.6372
##       Balanced Accuracy : 0.8196
##
##        'Positive' Class : CH
##
```

```
test_predict2 <- predict(mod2, test)
confusionMatrix(data = as.factor(test_predict2), reference = as.factor(test$Purchase))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  CH  MM
##         CH 142  28
##         MM  15  83
##
##                Accuracy : 0.8396
##                  95% CI : (0.79, 0.8814)
##     No Information Rate : 0.5858
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.6636
##
##  Mcnemar's Test P-Value : 0.06725
##
##             Sensitivity : 0.9045
##             Specificity : 0.7477
##          Pos Pred Value : 0.8353
##          Neg Pred Value : 0.8469
##              Prevalence : 0.5858
##          Detection Rate : 0.5299
##    Detection Prevalence : 0.6343
##       Balanced Accuracy : 0.8261
##
##        'Positive' Class : CH
##
```