Logistic Regression and Feature Selection

Blake Pappas

2023-12-17

Logistic Regression and Feature Selection in R

Load the following packages:

```r
library(caret)
library(dplyr)
library(FSelectorRcpp)
library(pROC)
```

In this exercise, we use the "credit.csv" file. This file concerns credit card applications. All attribute names and values have been changed to meaningless symbols to protect the confidentiality of the data.

The goal is to predict the "A16" (credit decision) based on A1-A15 as features.

P1: Import the dataset. Split it to 80% training and 20% testing.

```r
credit = read.csv("credit.csv")

train_rows = createDataPartition(y = credit$A16,
                                 p = 0.80, list = FALSE)

# Training Dataset
credit_train = credit[train_rows, ]

# Testing Dataset
credit_test = credit[-train_rows, ]
```

P2: Build logistic regression model with all features.

```r
# Build the Model
credit_train$A16 = ifelse(credit_train$A16 == "+", 1, 0)
logit_model = glm(A16 ~ ., data = credit_train,
                  family = binomial(link = "logit"))

# Examine the Model
summary(logit_model)
```

```
##
## Call:
## glm(formula = A16 ~ ., family = binomial(link = "logit"), data = credit_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6781  -0.2938  -0.1080   0.3701   3.3266
##
## Coefficients: (2 not defined because of singularities)
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)  9.924e+00  7.380e+02   0.013  0.98927
## A1b          4.021e-02  3.657e-01   0.110  0.91244
## A2           1.012e-02  1.655e-02   0.612  0.54062
## A3          -3.805e-02  3.335e-02  -1.141  0.25382
## A4u         -1.379e+01  7.380e+02  -0.019  0.98510
## A4y         -1.470e+01  7.380e+02  -0.020  0.98411
## A5gg                NA         NA      NA       NA
## A5p                 NA         NA      NA       NA
## A6c          4.333e-01  6.078e-01   0.713  0.47594
## A6cc         1.577e+00  9.173e-01   1.720  0.08551 .
## A6d          7.255e-01  1.053e+00   0.689  0.49101
## A6e          2.078e+00  1.586e+00   1.310  0.19012
## A6ff        -3.526e+00  2.296e+00  -1.535  0.12470
## A6i         -5.613e-01  8.058e-01  -0.697  0.48607
## A6j         -4.227e+00  2.374e+00  -1.780  0.07507 .
## A6k         -6.170e-01  7.259e-01  -0.850  0.39533
## A6m          5.568e-01  8.140e-01   0.684  0.49394
## A6q          3.206e-01  6.493e-01   0.494  0.62142
## A6r         -1.928e+00  4.640e+00  -0.416  0.67771
## A6w          1.115e+00  7.219e-01   1.545  0.12246
## A6x          3.120e+00  1.109e+00   2.813  0.00490 **
## A7dd        -6.838e-01  2.274e+00  -0.301  0.76365
## A7ff         2.674e+00  2.169e+00   1.233  0.21772
## A7h          5.167e-01  6.848e-01   0.754  0.45057
## A7j          4.474e+00  2.251e+00   1.987  0.04688 *
## A7n          3.365e+00  1.659e+00   2.029  0.04246 *
## A7o         -1.758e+01  3.182e+04  -0.001  0.99956
## A7v          3.613e-01  6.299e-01   0.574  0.56624
## A7z         -4.029e+00  2.190e+00  -1.839  0.06585 .
## A8           8.355e-02  6.461e-02   1.293  0.19595
## A9t          4.151e+00  4.336e-01   9.574  < 2e-16 ***
## A10t         3.493e-01  4.348e-01   0.803  0.42170
## A11          1.460e-01  6.737e-02   2.168  0.03018 *
## A12t        -1.529e-01  3.346e-01  -0.457  0.64759
## A13p        -1.849e+01  3.182e+04  -0.001  0.99954
## A13s         3.860e-02  6.463e-01   0.060  0.95238
## A14         -2.857e-03  1.105e-03  -2.585  0.00975 **
## A15          5.320e-04  2.105e-04   2.528  0.01148 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 720.43  on 522  degrees of freedom
## Residual deviance: 279.52  on 487  degrees of freedom
## AIC: 351.52
##
## Number of Fisher Scoring iterations: 14
```

```r
# Make Predictions

# Log Odds
pred = predict(logit_model, credit_test)

# Predicted Probability
pred_prob = predict(logit_model, credit_test,
                    type = "response")

# Binary Predictions
pred_binary = ifelse(pred_prob > 0.5, "yes", "no")
```

P3: Evaluate the performance of the logit model (confusion matrix, AUC, etc.).

```r
# Confusion Matrix
confusionMatrix(factor(pred_binary), factor(ifelse(credit_test$A16 == "+", "yes", "no")),
                mode = "prec_recall", positive = "yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction no yes
##        no  60   7
##        yes 11  52
##
##                Accuracy : 0.8615
##                  95% CI : (0.79, 0.9158)
##     No Information Rate : 0.5462
##     P-Value [Acc > NIR] : 1.598e-14
##
##                   Kappa : 0.7223
##
```

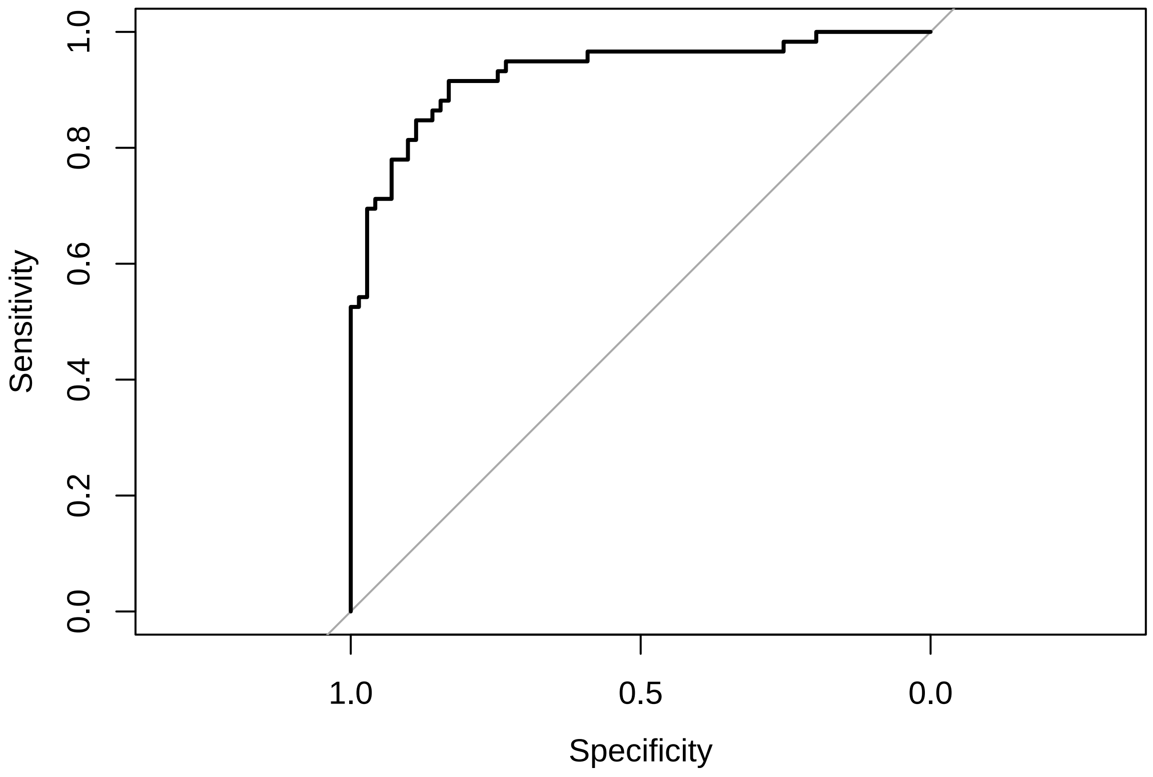
```
##  Mcnemar's Test P-Value : 0.4795
##
##               Precision : 0.8254
##                  Recall : 0.8814
##                      F1 : 0.8525
##              Prevalence : 0.4538
##          Detection Rate : 0.4000
##    Detection Prevalence : 0.4846
##       Balanced Accuracy : 0.8632
##
##        'Positive' Class : yes
##
```

```
# AUC for class "+"
library(pROC)
roc_logit = roc(response = ifelse(credit_test$A16 == "+", 1, 0),
                predictor = pred_prob)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
plot(roc_logit)
```



```r
auc(roc_logit)
```

```
## Area under the curve: 0.9301
```

P4: Select the top 5 features by information gain. Then, build another model. Do you get better performance in terms of AUC?

```r
# Information Gain
library(FSelectorRcpp)

IG = information_gain(A16 ~ ., data = credit_train)

# Select Top 5 Attributes
topK = cut_attrs(IG, k = 5)

credit_topK_train = credit_train %>% select(topK, A16)
credit_topK_test = credit_test %>% select(topK, A16)

# Build Logistic Regression Model
library(pROC)

logit_model2 = glm(A16 ~ ., data = credit_topK_train,
                   family = binomial(link = "logit"))

pred_prob = predict(logit_model2, credit_topK_test,
                    type = "response")

auc(ifelse(credit_topK_test$A16 == "+", 1, 0), pred_prob)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
## Area under the curve: 0.9211
```

Answer: Yes, the model gets better performance in terms of AUC.

P5: Implement Backward Elimination. Consider an AUC increase of any positive amount to be an improvement when selecting features. Report the selected features.

```r
model_all = glm(A16 ~ ., data = credit_train,
                family = binomial(link = "logit"))
```

```r
pred_prob = predict(model_all, credit_test,
                    type = "response")

best_auc = auc(ifelse(credit_test$A16 == "+", 1, 0), pred_prob)
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```

```r
selected_features = 1:15
while (TRUE) {
  feature_to_drop = -1
  for (i in selected_features) {
    train = credit_train %>% select(setdiff(selected_features, i), A16)
    test = credit_test %>% select(setdiff(selected_features, i), A16)

    logit_model = glm(A16 ~ ., data = train,
                      family = binomial(link = "logit"))

    pred_prob = predict(logit_model, test,
                        type = "response")

    auc = auc(ifelse(test$A16 == "+", 1, 0), pred_prob)

    if (auc > best_auc) {
      best_auc = auc
      feature_to_drop = i
    }
  }

  if (feature_to_drop != -1) {
    selected_features = setdiff(selected_features, feature_to_drop)
    print(selected_features)
    print(best_auc)
  }
  else break
}
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

##  [1]  1  2  4  5  6  7  8  9 10 11 12 13 14 15
## Area under the curve: 0.9329
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

##  [1]  1  2  4  5  6  7  8  9 10 12 13 14 15
## Area under the curve: 0.9348

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
##  [1]  1  2  4  5  6  7  8  9 10 12 14 15
## Area under the curve: 0.936
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

##  [1]  1  4  5  6  7  8  9 10 12 14 15
## Area under the curve: 0.9365

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```