In-Class Exercise 7

Ryan Standridge

10/13/2021

Contents

Logistic Regression and Feature Selection in R

- set working directory and load packages

```
setwd("~/Documents/Data Science/DSA_8590")
# We need to load the following packages

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 confidentiality of the data.

Our 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

```
# Import the dataset and quick glance
credit <- read.csv('credit.csv')
head(credit)
```

```
##   A1    A2    A3 A4 A5 A6 A7   A8 A9 A10 A11 A12 A13 A14 A15 A16
## 1  b 30.83 0.000  u  g  w  v 1.25  t   t   1   f   g 202   0   +
## 2  a 58.67 4.460  u  g  q  h 3.04  t   t   6   f   g  43 560   +
## 3  a 24.50 0.500  u  g  q  h 1.50  t   f   0   f   g 280 824   +
## 4  b 27.83 1.540  u  g  w  v 3.75  t   t   5   t   g 100   3   +
## 5  b 20.17 5.625  u  g  w  v 1.71  t   f   0   f   s 120   0   +
## 6  b 32.08 4.000  u  g  m  v 2.50  t   f   0   t   g 360   0   +
```

```
# make the output variable a factor
credit$A16 <- factor(credit$A16)

# Split it to 80% training and 20% testing
set.seed(100)
train_rows <- createDataPartition(y = credit$A16,
                                  p = 0.80, list = FALSE)
credit_train <- credit[train_rows,]
credit_test <- credit[-train_rows,]
```

P2: Build logistic regression model with all features

```
logit_model <- glm(A16 ~ ., data = credit_train,
                   family = binomial(link = "logit"))
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(logit_model)
```

```
##
## Call:
## glm(formula = A16 ~ ., family = binomial(link = "logit"), data = credit_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3347  -0.3212  -0.1497   0.4250   3.4272
##
## Coefficients: (2 not defined because of singularities)
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.126e+01  1.455e+03   0.008 0.993824
## A1b          1.063e-01  3.642e-01   0.292 0.770368
## A2           1.432e-02  1.599e-02   0.896 0.370495
## A3          -1.781e-02  3.376e-02  -0.528 0.597742
## A4u         -1.578e+01  1.455e+03  -0.011 0.991350
## A4y         -1.634e+01  1.455e+03  -0.011 0.991040
## A5gg                NA         NA      NA       NA
## A5p                 NA         NA      NA       NA
## A6c          7.053e-01  6.171e-01   1.143 0.253048
## A6cc         1.476e+00  8.422e-01   1.753 0.079618 .
## A6d          1.367e+00  9.851e-01   1.388 0.165131
## A6e          2.434e+00  1.296e+00   1.879 0.060303 .
## A6ff        -3.038e+00  2.109e+00  -1.440 0.149732
```

```
## A6i          5.177e-01  8.133e-01   0.637 0.524390
## A6j         -3.711e+00  2.225e+00  -1.668 0.095323 .
## A6k          4.359e-02  7.696e-01   0.057 0.954833
## A6m          4.402e-01  7.625e-01   0.577 0.563711
## A6q          2.860e-01  6.499e-01   0.440 0.659835
## A6r         -1.071e+01  4.006e+01  -0.267 0.789171
## A6w          1.019e+00  6.442e-01   1.582 0.113687
## A6x          3.689e+00  1.060e+00   3.480 0.000501 ***
## A7dd        -1.427e+00  2.394e+00  -0.596 0.551234
## A7ff         2.668e+00  1.983e+00   1.345 0.178489
## A7h          9.176e-01  6.752e-01   1.359 0.174139
## A7j          4.330e+00  2.069e+00   2.093 0.036351 *
## A7n          1.247e+01  4.007e+01   0.311 0.755564
## A7o         -1.234e+01  1.455e+03  -0.008 0.993237
## A7v          7.910e-01  6.295e-01   1.257 0.208864
## A7z         -3.469e+00  1.880e+00  -1.845 0.064992 .
## A8           9.308e-02  5.835e-02   1.595 0.110655
## A9t          3.773e+00  3.961e-01   9.525  < 2e-16 ***
## A10t         6.680e-01  4.353e-01   1.535 0.124856
## A11          9.735e-02  6.584e-02   1.479 0.139241
## A12t        -1.187e-01  3.212e-01  -0.369 0.711801
## A13p        -1.289e+01  1.455e+03  -0.009 0.992935
## A13s        -2.797e-02  5.327e-01  -0.053 0.958123
## A14         -2.931e-03  1.043e-03  -2.811 0.004945 **
## A15          5.762e-04  2.217e-04   2.600 0.009330 **
## ---
## 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: 299.70  on 487  degrees of freedom
## AIC: 371.7
##
## Number of Fisher Scoring iterations: 14
```

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

```
# by default, predict() gives you the log odds
# pred <- predict(logit_model, credit_test)
# to get predicted probability:
pred_prob <- predict(logit_model, credit_test, type = "response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
# to convert to binary predictions
pred_binary <- ifelse(pred_prob > 0.5, "+", "-")
# confusion matrix
confusionMatrix(factor(pred_binary),
                credit_test$A16,
```

```
                mode = "prec_recall",
                positive = "+")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  -  +
##          - 58  2
##          + 13 57
##
##                Accuracy : 0.8846
##                  95% CI : (0.8168, 0.934)
##     No Information Rate : 0.5462
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7709
##
##  Mcnemar's Test P-Value : 0.009823
##
##               Precision : 0.8143
##                  Recall : 0.9661
##                      F1 : 0.8837
##              Prevalence : 0.4538
##          Detection Rate : 0.4385
##    Detection Prevalence : 0.5385
##       Balanced Accuracy : 0.8915
##
##        'Positive' Class : +
##
```

```
# AUC (using library(pROC))
#roc_logit <- roc(response = ifelse(credit_test$A16 == "+", 1, 0), predictor = pred_prob)
#plot(roc_logit)
#auc(roc_logit)
auc(ifelse(credit_test$A16 == "+",1,0), pred_prob)
```

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

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

```
## Area under the curve: 0.9477
```

As seen in the results above, the accuracy of this logit model is 0.8846 with a Precision of 0.8143, a Recall of 0.9661 and a F-measure of 0.8837. Also, the AUC is 0.9477.

P4: Select top 5 features by information gain, then build another model.

*Do you get better performance in terms of AUC?

```r
set.seed(100)
# using library(FSelectorRcpp)
# select top 5 features by information gain
IG <- information_gain(A16 ~ ., data = credit_train)
topK <- cut_attrs(IG, k = 5)

# build another model
credit_topK_train <- credit_train %>% select(topK, A16)
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(topK)` instead of `topK` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```r
credit_topK_test <- credit_test %>% select(topK, A16)
logit_model_topK <- glm(A16 ~ ., data = credit_topK_train,
                        family = binomial(link = "logit"))
summary(logit_model_topK)
```

```
##
## Call:
## glm(formula = A16 ~ ., family = binomial(link = "logit"), data = credit_topK_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4032  -0.3426  -0.2894   0.5351   2.5309
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.1617352  0.2933661 -10.777  < 2e-16 ***
## A9t          3.3618152  0.3126811  10.752  < 2e-16 ***
## A11          0.0830945  0.0597064   1.392  0.16401
## A10t         0.7159345  0.3658600   1.957  0.05036 .
## A8           0.1156761  0.0514086   2.250  0.02444 *
## A15          0.0004972  0.0001688   2.945  0.00323 **
## ---
## 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: 350.50  on 517  degrees of freedom
## AIC: 362.5
##
## Number of Fisher Scoring iterations: 6
```

```r
# to get predicted probability:
pred_prob_topK <- predict(logit_model_topK, credit_topK_test, type = "response")
# to convert to binary predictions
pred_binary_topK <- ifelse(pred_prob_topK > 0.5, "+", "-")
# confusion matrix
confusionMatrix(factor(pred_binary_topK),
```

```
                credit_topK_test$A16,
                mode = "prec_recall",
                positive = "+")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  -  +
##          - 54  0
##          + 17 59
##
##                Accuracy : 0.8692
##                  95% CI : (0.7989, 0.9219)
##     No Information Rate : 0.5462
##     P-Value [Acc > NIR] : 3.018e-15
##
##                   Kappa : 0.7425
##
##  Mcnemar's Test P-Value : 0.0001042
##
##               Precision : 0.7763
##                  Recall : 1.0000
##                      F1 : 0.8741
##              Prevalence : 0.4538
##          Detection Rate : 0.4538
##    Detection Prevalence : 0.5846
##       Balanced Accuracy : 0.8803
##
##        'Positive' Class : +
##
```

```
# AUC (using library(pROC))
auc(ifelse(credit_topK_test$A16 == "+", 1, 0), pred_prob_topK)
```

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

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

```
## Area under the curve: 0.9566
```

As seen in the results above, the accuracy of the logit model with the top 5 features by information gain is 0.8692 which is less than (original = 0.8846). Also this model has a Precision of 0.7763 (less than original = 0.8143), a Recall of 1.0000 (greater than original = 0.9661), and a F-measure of 0.8741 (less than original = 0.8837). Also, the AUC for this model is 0.9566 which is greater than the original, being 0.9477.

Thus, in terms of AUC, this model has a better performance than the original model with all features.

P5: Implement backward Elimination.

- Consider AUC increase of any positive amount to be an improvement when selecting features.
- Report the selected features.

```r
set.seed(100)
# build the model to get starting Best AUC
model_all = glm(A16 ~ .,
                data = credit_train,
                family = binomial(link = "logit"))
# to get predicted probability:
pred_prob_all = predict(model_all, credit_test, type = "response")
# best AUC (beginning AUC from the all features model)
best_auc = auc(ifelse(credit_test$A16 == "+",1,0), pred_prob_all)

# Implement backward Elimination. 1-15 because using A16 as output column
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
}
```

```
##  [1]  1  2  3  4  5  6  8  9 10 11 12 13 14 15
## Area under the curve: 0.9632
##  [1]  1  2  3  4  5  6  8  9 11 12 13 14 15
## Area under the curve: 0.9675
```

```r
# Report the selected features
print(selected_features)
```

```
##  [1]  1  2  3  4  5  6  8  9 11 12 13 14 15
```

```r
print(best_auc)
```

```
## Area under the curve: 0.9675
```

As seen in the results from the backwards elimination, the selected features are A1, A2, A3, A4, A5, A6, A8, A9, A11, A12, A13, A14, and A15.

Also, the AUC value from the logistic model with these selected features is 0.9675.