Bank Customer Churn Prediction

Berk Yiğit ÖZBEK

Uploading the data

```
data1 <- read.csv("Bank Customer Churn Prediction.csv")
```

Downloading necessary packages.

```
install.packages("DALEX")
```

```
Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
(as 'lib' is unspecified)
```

```
install.packages("caret")
```

```
Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
(as 'lib' is unspecified)
```

```
install.packages("ROCR")
```

```
Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
(as 'lib' is unspecified)
```

```
library(DALEX)
```

```
Welcome to DALEX (version: 2.4.3).
Find examples and detailed introduction at: http://ema.drwhy.ai/
Additional features will be available after installation of: ggpubr.
Use 'install_dependencies()' to get all suggested dependencies
```

```r
library(caret)
```

```
Loading required package: ggplot2
```

```
Loading required package: lattice
```

```r
library(ROCR)
```

```r
str(data1)
```

```
'data.frame':	10000 obs. of  12 variables:
 $ customer_id     : int  15634602 15647311 15619304 15701354 15737888 15574012 15592531 1565
 $ credit_score    : int  619 608 502 699 850 645 822 376 501 684 ...
 $ country         : chr  "France" "Spain" "France" "France" ...
 $ gender          : chr  "Female" "Female" "Female" "Female" ...
 $ age             : int  42 41 42 39 43 44 50 29 44 27 ...
 $ tenure          : int  2 1 8 1 2 8 7 4 4 2 ...
 $ balance         : num  0 83808 159661 0 125511 ...
 $ products_number : int  1 1 3 2 1 2 2 4 2 1 ...
 $ credit_card     : int  1 0 1 0 1 1 1 1 0 1 ...
 $ active_member   : int  1 1 0 0 1 0 1 0 1 1 ...
 $ estimated_salary: num  101349 112543 113932 93827 79084 ...
 $ churn           : int  1 0 1 0 0 1 0 1 0 0 ...
```

Task Details

- Data set includes 12 variables, 10000 rows. And it also includes categorical and numerical variables. (mixed) "Country" and "Gender" are categorical variables here. Others are numerical ones. (CreditScore, Age, Tenure, Balance, NumberProducts, HasCard, ActiveMember, EstimatedSalary, and Churn)
- The problem is about the churning of the bank. And the target is predicting the churn of the spesific bank. # Features
- CustomerId: Customer's identification number.
- CreditScore: Customer's credit score
- Country: Countries of customers (Spain, France, or Germany)
- Gender: Gender of customers. (Male or Female)
- Age: Customer's age

- Tenure: Account years
- Balance: Customer's account balance
- NumOfProducts: Number of the bank products which used by the customers
- HasCrCard: If customers have a credit card? (No = 0, Yes = 1)
- IsActiveMember: If customers an active member of the bank? (0 = No, 1 = Yes)
- EstimatedSalary: Customer's estimated salaries
- Churn: If customers left the bank or not (0 = No, 1 = Yes)

```r
data2 <- data1[, -c(1,3,4)]
```

```r
data2 <- na.exclude(data2)
```

```r
set.seed(123)
index <- sample(1 : nrow(data2), round(nrow(data2) * 0.80))
train <- data2[index, ]
test  <- data2[-index, ]
```

```r
lr_model <- glm(churn ~ ., data = train, family = "binomial")
```

```r
lr_model
```

```

Call:  glm(formula = churn ~ ., family = "binomial", data = train)

Coefficients:
     (Intercept)      credit_score               age            tenure
      -3.809e+00        -6.166e-04         7.424e-02        -2.135e-02
         balance   products_number       credit_card     active_member
       4.918e-06        -2.703e-02        -1.727e-02        -1.096e+00
estimated_salary
       5.690e-07

Degrees of Freedom: 7999 Total (i.e. Null);  7991 Residual
Null Deviance:      8100
Residual Deviance: 7042     AIC: 7060
```

```r
summary(lr_model)
```

```
Call:
glm(formula = churn ~ ., family = "binomial", data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0960  -0.6773  -0.4747  -0.2886   2.8943

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)      -3.809e+00  2.677e-01 -14.227   <2e-16 ***
credit_score     -6.166e-04  3.081e-04  -2.001   0.0454 *
age               7.424e-02  2.859e-03  25.964   <2e-16 ***
tenure           -2.135e-02  1.034e-02  -2.066   0.0389 *
balance           4.918e-06  5.102e-07   9.639   <2e-16 ***
products_number  -2.703e-02  5.180e-02  -0.522   0.6018
credit_card      -1.727e-02  6.534e-02  -0.264   0.7916
active_member    -1.096e+00  6.382e-02 -17.176   <2e-16 ***
estimated_salary  5.690e-07  5.238e-07   1.086   0.2774
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 8099.8  on 7999  degrees of freedom
Residual deviance: 7042.2  on 7991  degrees of freedom
AIC: 7060.2

Number of Fisher Scoring iterations: 5
```

```r
predicted_probs <- predict(lr_model, test, type = "response")
head(predicted_probs)
```

```
         6         19         23         34         35         38
0.36883013 0.29016683 0.20359284 0.09218654 0.03293496 0.06328798
```

```r
predicted_classes <- ifelse(predicted_probs > 0.5, 1, 0)
head(predicted_classes)
```

```
 6 19 23 34 35 38
 0  0  0  0  0  0
```

```
TP <- sum(predicted_classes[which(test$churn == 1)] == 1)
FP <- sum(predicted_classes[which(test$churn == 1)] == 0)
TN <- sum(predicted_classes[which(test$churn == 0)] == 0)
FN <- sum(predicted_classes[which(test$churn == 0)] == 1)
recall      <- TP / (TP + FN)
specificity <- TN / (TN + FP)
precision   <- TP / (TP + FP)
accuracy    <- (TN + TP) / (TP + FP + TN + FN)
recall
```

```
[1] 0.5371901
```

```
specificity
```

```
[1] 0.8201171
```

```
precision
```

```
[1] 0.1612903
```

```
accuracy
```

```
[1] 0.803
```

```
table(train$churn) / dim(train)[1]
```

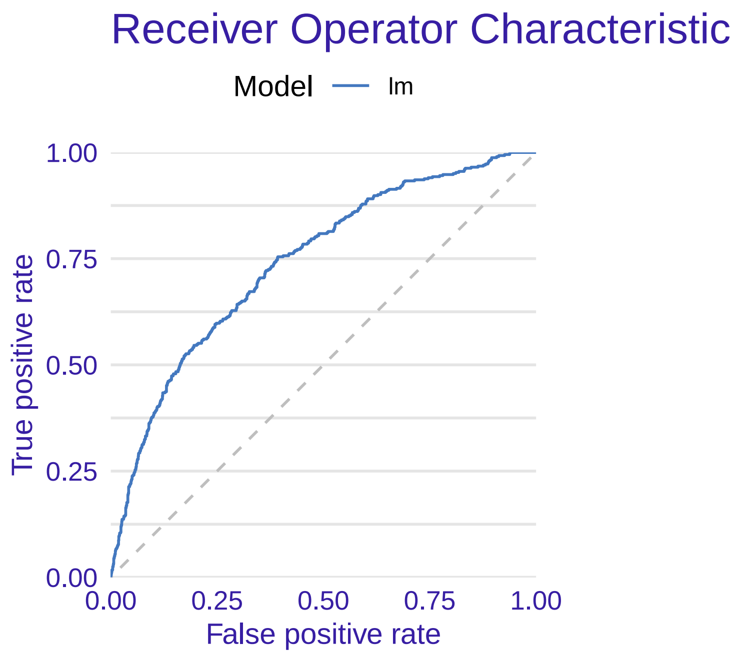
```

      0       1 
0.79575 0.20425 
```

```r
confusionMatrix(table(test$churn,
                      predicted_classes),
                positive = "1")
```

```
Confusion Matrix and Statistics

   predicted_classes
       0    1
  0 1541   56
  1  338   65

               Accuracy : 0.803
                 95% CI : (0.7849, 0.8202)
    No Information Rate : 0.9395
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1709

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.5372
            Specificity : 0.8201
         Pos Pred Value : 0.1613
         Neg Pred Value : 0.9649
             Prevalence : 0.0605
         Detection Rate : 0.0325
   Detection Prevalence : 0.2015
      Balanced Accuracy : 0.6787

       'Positive' Class : 1

```

```r
explain_lr <- explain(model   = lr_model,
                      data    = test[, -9],
                      y       = test$churn == 1,

                      type    = "classification",
                      verbose = FALSE)
```

```r
performance_lr <- model_performance(explain_lr)
plot(performance_lr, geom = "roc")
```




```r
performance_lr
```

```
Measures for:  classification
recall     : 0.1612903
precision  : 0.5371901
f1         : 0.2480916
accuracy   : 0.803
auc        : 0.7410654

Residuals:
         0%         10%         20%         30%         40%         50%
-0.84349822 -0.32864557 -0.23744919 -0.18006259 -0.13873122 -0.10330616
        60%         70%         80%         90%        100%
-0.07555967 -0.05360338  0.21206633  0.71043957  0.95907708
```