Logistic Regression For Binary Classification For Bank Personal-Loan Modelling

Matthew Curcio

Contents

1 Executive Summary

The management of Thera Bank wanted to explore ways of converting its customers to personal loan customers. A previous advertising campaign was run last year to elicit loans from customers. From the 5000 customers selected for the campaign a conversion rate of over 9% success was found. One Logistic Regression (Model #2) was found to have high accuracy (96.10%) and high specificity (98.67%). and sensitivity (71.88%).

If Model #2 were provided a list of 1000 banking customers, 81 would be possible candidates with a cost savings of 92.9% compared to the first campaign. From the 81 customers approximately 85% would be converted to Personal loans, (69 customers out of 81).

The optimum customer for Personal loans was found to be;

```
1. One with higher education,
2. Who owns CD's,
3. Who does Not have a Securities Act.,
4. A customer who does Not have a Credit Card.
5. A customer with an income >= $65 k.
```

2 Introduction

2.1 Understanding Logistic Regression

Logistic Regression is often used as a binary classifier where the dependent variable is a categorical outcome, for example; go/no-go, loan/reject-loan. Using Logistic Regression, we may also calculate the presence or absence of a product or quality when the *decision boundary* is not clear.

Logistic regression may also be familiar as the *exponential growth curve* (Eq. #1) given a limited set of resources. It may be used for other biological situations such as dose-response curves, enzyme kinetic curves, median lethal dose curve (LD-50), and survival.

$$f(x) \;=\; \frac{M}{1 + Ae^{-r(x-x_0)}} \tag{1}$$

where M is the curve's maximum value, r is the maximum growth rate, x_0 is the midpoint of the curve, and A is the number of doublings to reach M.[1]

In the specific case of *Logistic Regression for Binary Classification* (Eq. #2), M, A and r take on the value 1 and a probability between 0 and 1 is generated.

$$f(x) \;=\; \frac{1}{1 + e^{-(WX+b)}} \tag{2}$$

Because the logistic equation is exponential, it is easier to work with the formula in terms of its *log-odds*. Where odds are the probabilities of success over failure. By using log-odds, logistic regression may be more easily expressed as a set of linear equations in terms of x. Therefore Logistic Regression is considered a *generalized linear model* (GLM).

$$ln\left(\frac{p}{1-p}\right) = \;\sum_{i}^{k} \beta_i x_i \tag{3}$$

[1] https://en.wikipedia.org/wiki/Malthusian_growth_model

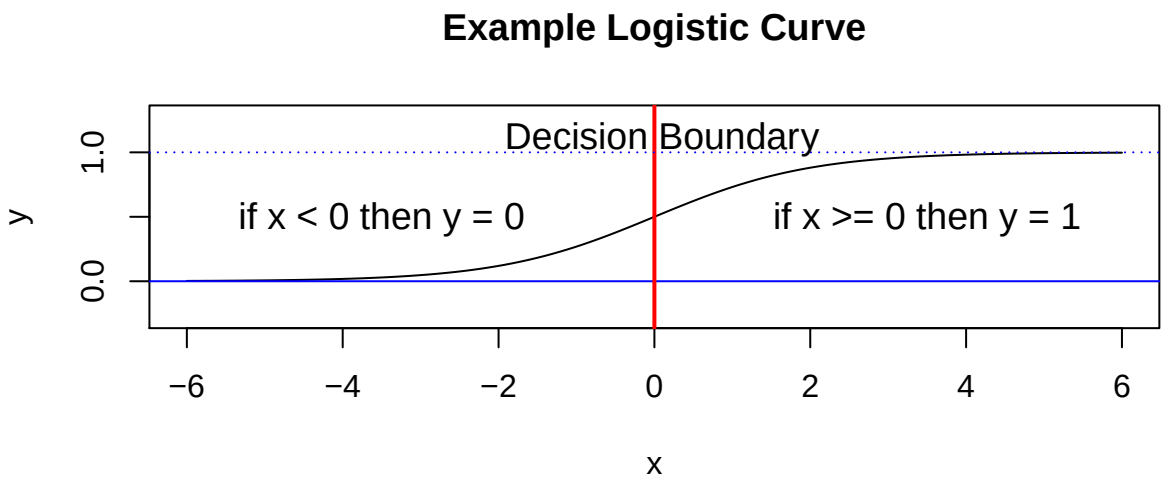

Figure 1: Example Logistic Curve

Eliminate the natural log by taking the exponent on both sides;

$$\frac{p}{1-p} = exp\left(\sum_{i}^{k} \beta_i x_i\right) \tag{4}$$

To evaluate the 2 models the Akaike Information Criterion [2] will be used. With AIC smaller values indicate better fitting models.

$$AIC = 2K - 2ln(\widehat{L}) \tag{5}$$

Where $ln(\widehat{L})$ is the log-likelihood estimate, K is the number of parameters.

2.2 Thera Bank Loan Campaign

The management of Thera Bank wants to explore various ways of converting its liability customers to personal loan customers yet to also retain them as depositors. A previous advertising campaign was run last year to elicit loans from customers. From the 5000 customers selected for the campaign a conversion rate of over 9% success was found. The purpose of this work is to understand if it is possible to convert a higher percentage of bank users to loan recipients.

A large portion of the feature variables relate to whether bank customers have accounts with Thera Bank. Will a study of these features provide information on which ones to focus on for future loan campaign advertising?

At the present time, the feature `Zip code` will not be used. It is possible to correlate home zip code and neighborhood qualities but this is for a future project.

`Bank_Personal_Loan_Modeling data file can be found at:`
`https://www.kaggle.com/krantiswalke/bank-personal-loan-modelling`

[2] https://en.wikipedia.org/wiki/Akaike_information_criterion

2.2.1 13 Feature Attributes Were Collected

Abbreviation	Attribute
ID	Customer ID
Age	Age
Experience	Yrs experience
ZIP Code	Zip code
Family	Family size
Income	Annual income, $k
Mortgage	Home mortgage, $k
CCAvg	Mean credit card spending, $k
Education	Education Level (1,2,3)
Securities Account	Customer has securities (0,1)
CD Account	Customer has CDs (0,1)
Online	Customer uses internet banking (0,1)
Credit card	Customer uses credit card (0,1)

2.2.2 Target - Personal Loan Conversion

Abbreviation	Target
Personal Loan	Customer accepted loan? (0,1)

3 Model # 1 - Using 12 features and 1 Target variable

3.1 R Libraries

```
# Load Libraries
Libraries <- c("doMC", "knitr", "readr", "tidyverse", "caret", "e1071")
for (p in Libraries) {
  library(p, character.only = TRUE)
}
```

Import Data

```
model_1 <- read_csv("Bank_Personal_Loan_Modelling.csv",
                    col_types = cols(ID = col_skip(),
                                     `ZIP Code` = col_skip(),
                                     `CD Account` = col_factor(levels = c("0", "1")),
                                     CreditCard = col_factor(levels = c("0", "1")),
                                     Education = col_factor(levels = c("1", "2", "3")),
                                     Family = col_integer(),
                                     Online = col_factor(levels = c("0", "1")),
                                     `Personal Loan` = col_factor(levels = c("0", "1")),
                                     `Securities Account` = col_factor(levels = c("0", "1"))))
# View(model_1)
dim(model_1)
```

```
## [1] 5000   12
```

3.2 Model 1 Generation

```
set.seed(1000)
index <- createDataPartition(model_1$`Personal Loan`, p = 0.8, list = FALSE)
training_set_1 <- model_1[index,]

# The `test_set_1` and `Class_test` data sets are not produced since the Logit
# run with 11 features was is not the final model.

# The first training run is to determine if all 11 features are necessary for
# our final logistic regression model.

set.seed(1000)
registerDoMC(cores = 3)       # Start multi-processor mode
start_time <- Sys.time()      # Start timer

# Create model, 10X fold CV repeated 5X
tcontrol <- trainControl(method = "repeatedcv",
                         number = 10,
                         repeats = 5)

model_result_1 <- train(`Personal Loan` ~ .,
                        data = training_set_1,
                        trControl = tcontrol,
                        method = "glm",         # glm = 'Generalized Linear Model'
                        family = "binomial")

end_time <- Sys.time()       # End timer
end_time - start_time        # Display time
```

```
## Time difference of 1.009425 secs
```

```
registerDoSEQ()              # Stop multi-processor mode
```

3.3 Summary - Model 1

```
summary(model_result_1)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3243  -0.1860  -0.0694  -0.0219   4.1907
##
## Coefficients:
```

```
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -1.126e+01  2.085e+00  -5.399 6.70e-08 ***
## Age                          -7.493e-02  7.828e-02  -0.957  0.33850
## Experience                    7.905e-02  7.763e-02   1.018  0.30859
## Income                        6.085e-02  3.349e-03  18.170  < 2e-16 ***
## Family                        6.486e-01  8.731e-02   7.429 1.10e-13 ***
## CCAvg                         1.615e-01  4.916e-02   3.286  0.00102 **
## Education2                    3.812e+00  2.991e-01  12.747  < 2e-16 ***
## Education3                    3.932e+00  2.964e-01  13.265  < 2e-16 ***
## Mortgage                      1.417e-04  6.616e-04   0.214  0.83041
## `\\`Securities Account\\`1` -9.014e-01  3.384e-01  -2.663  0.00774 **
## `\\`CD Account\\`1`          3.839e+00  3.866e-01   9.929  < 2e-16 ***
## Online1                      -7.283e-01  1.862e-01  -3.911 9.21e-05 ***
## CreditCard1                  -1.013e+00  2.406e-01  -4.213 2.52e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2529.63  on 3999  degrees of freedom
## Residual deviance:  928.51  on 3987  degrees of freedom
## AIC: 954.51
##
## Number of Fisher Scoring iterations: 8
```

3.4 Observations - Model 1

- Model contains 12 features and 1 dependent variable.
- Age, Experience and Mortgage have P-values greater than 5%.

Feature	P-value
Age	0.33850
Experience	0.30859
Mortgage	0.83041

- The Akaike information criterion (AIC)[3] for model #1 is 954.51.

3.5 Decision - Model 1

- Model should DROP: **Age, Experience, Mortgage, Zip code**

---

4 Model # 2 - Uses 9 features and 1 Target variable

The second model uses 9 features:

[3] https://en.wikipedia.org/wiki/Akaike_information_criterion

1. Income
2. Family
3. Average Credit Card Debt
4. CD Account
5. Education-2
6. Education-3
7. Online banking
8. Securities Account
9. Has Credit Card

Import Bank Loan Data

```
model_2  <- read_csv("Bank_Personal_Loan_Modelling.csv",
                     col_types = cols(Age = col_skip(),
                                      ID = col_skip(),
                                      Mortgage = col_skip(),
                                      Experience = col_skip(),
                                      `ZIP Code` = col_skip(),
                                      `CD Account` = col_factor(levels = c("0", "1")),
                                      CreditCard = col_factor(levels = c("0", "1")),
                                      Education = col_factor(levels = c("1", "2", "3")),
                                      Family = col_integer(),
                                      Online = col_factor(levels = c("0", "1")),
                                      `Personal Loan` = col_factor(levels = c("0", "1")),
                                      `Securities Account` = col_factor(levels = c("0", "1"))))
dim(model_2)
```

```
## [1] 5000    9
```

4.1 Model 2 Generation

```
# Partition data into training and testing sets
set.seed(1000)
index <- createDataPartition(model_2$`Personal Loan`, p = 0.8, list = FALSE)

training_set_2 <- model_2[ index, ]
test_set_2     <- model_2[-index, ]

Class_test_2 <- as.factor(test_set_2$`Personal Loan`)

set.seed(1000)
registerDoMC(cores = 3)           # Start multi-core
start_time <- Sys.time()          # Start timer

# Create model, 10X fold CV repeated 5X
tControl <- trainControl(method = "repeatedcv",
                         number = 10,
                         repeats = 5,
                         savePredictions = "final") # IMPORTANT: Saves predictions
```

```r
model_result_2 <- train(`Personal Loan` ~ .,
                        data = training_set_2,
                        trControl = tControl,
                        method = "glm",
                        family = "binomial")

end_time <- Sys.time()           # End timer
end_time - start_time            # Display time
```

```
## Time difference of 1.069178 secs
```

```r
registerDoSEQ()                  # Stop multi-core
```

4.2 Summary - Model 2

```r
summary(model_result_2)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3571  -0.1893  -0.0691  -0.0219   4.1676
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -13.042368   0.635359 -20.528  < 2e-16 ***
## Income                        0.061022   0.003311  18.430  < 2e-16 ***
## Family                        0.646500   0.087081   7.424 1.14e-13 ***
## CCAvg                         0.157995   0.048693   3.245  0.00118 **
## Education2                    3.795839   0.298495  12.717  < 2e-16 ***
## Education3                    3.881894   0.291956  13.296  < 2e-16 ***
## `\\`Securities Account\\`1`  -0.888602   0.337540  -2.633  0.00847 **
## `\\`CD Account\\`1`           3.847597   0.385513   9.980  < 2e-16 ***
## Online1                      -0.728666   0.186064  -3.916 9.00e-05 ***
## CreditCard1                  -1.003529   0.239914  -4.183 2.88e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2529.63  on 3999  degrees of freedom
## Residual deviance:  929.95  on 3990  degrees of freedom
## AIC: 949.95
##
## Number of Fisher Scoring iterations: 8
```

4.3 Confusion Matrix - Model 2

```
Predicted_test_vals <- predict(model_result_2, test_set_2[, -5])

confusionMatrix(Predicted_test_vals, Class_test_2, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 892  27
##          1  12  69
##
##                Accuracy : 0.961
##                  95% CI : (0.9471, 0.9721)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : 5.695e-12
##
##                   Kappa : 0.7584
##
##  Mcnemar's Test P-Value : 0.02497
##
##             Sensitivity : 0.7188
##             Specificity : 0.9867
##          Pos Pred Value : 0.8519
##          Neg Pred Value : 0.9706
##              Prevalence : 0.0960
##          Detection Rate : 0.0690
##    Detection Prevalence : 0.0810
##       Balanced Accuracy : 0.8527
##
##        'Positive' Class : 1
##
```

4.4 Observations - Model 2

- All Model 2 features have (**P-values <= 0.01**).
- Model 1 & 2: Akaike information criterion

Model #	Features	AIC
1	12	954.51
2	9	949.95

- Model #2 has a **lower AIC** and will be used.
- Model #2 Beta-Parameters ranked in order of importance:

Feature	Beta-Parameters
Education-3	3.881894
CD Account-1	3.847597
Education-2	3.795839
Family	0.646500
CCAvg	0.157995
Income	0.061022
Online-1	-0.728666
Securities Account-1	-0.888602
CreditCard-1	-1.003529

- The 3 most *Positive Beta-Parameters* are;
  1. Education-3,
  2. Ownership of Certificates of Deposit,
  3. Education-2.
- The 3 most *Negative Beta-Parameters* are;
  1. Ownership of Credit Card,
  2. Ownership of Securities,
  3. Use of Online banking.
- Model 2 Statistics

Model Statistics	Value
Accuracy	0.9610
Sensitivity	0.7188
Specificity	0.9867

- Although Accuracy and Specificity are very high (> 0.95), Sensitivity is 0.7188. This means that any list of customers produced from Model 2 will have a relatively high rate of False-Positives.

4.5 Decisions - Model 2

- Model 2 had 9 parameters and should be used.
- All Model #2 Beta-parameters had P-values < 0.01.
- Model 2 Accuracy (0.9610) and Specificity (0.9867) are very high .
- Sensitivity is 0.7188, which can lead to a higher rate of False-Positives. The higher rates of False-Positives may require larger campaigns.

5 Conclusion

- Model 2 had 9 parameters and should be used.
- All Model #2 Beta-parameters had P-values < 0.01.
- Model 2 Accuracy (0.9610) and Specificity (0.9867) are very high .
- Sensitivity is 0.7188, which can lead to a higher rate of False-Positives. The higher rates of False-Positives may require larger campaigns.

If Model #2 were provided a list of 1000 banking customers 81 would be possible candidates with a cost savings of 92.9% compared to the first campaign. From the 8.1% of new customers approximately 85% would be converted (69 customers out of 81).

with 6.9% overall converted to Personal loans.

- A good customer profile would be:
    1. One **with higher education**,
    2. who **owns CD's**,
    3. who does **Not have a Securities Act.**,
    4. A customer who does **Not have a Credit Card**.
    5. AND, of course, a person with an **income >= $55 k**.