Logistic Regression: R Exercise

Pilsung Kang
School of Industrial Management Engineering
Korea University

AGENDA

01	Logistic Regression: Formulation
02	Logistic Regression: Learning
03	Logistic Regression: Interpretation
04	Classification Performance Evaluation
05	R Exercise

R Exercise 1: Binary Classification

- Data Set: Personal Loan Prediction

Data Description:

ID	Customer ID
Age	Customer's Age in completed years
Experience	#years of professional experience
Income	Annual income of the customer ($000)
ZIPCode	Home Address ZIP code.
Family	Family size (dependents) of the customer
CCAvg	Avg. Spending on Credit Cards per month ($000)
Education	Education Level. 1: Undergrad; 2: Graduate; 3: Advanced/Professional
Mortgage	Value of house mortgage if any. ($000)
Personal Loan	Did this customer accept the personal loan offered in the last campaign?
Securities Account	Does the customer have a Securities account with the bank?
CD Account	Does the customer have a Certificate of Deposit (CD) account with the bank?
Online	Does the customer use internet banking facilities?
CreditCard	Does the customer use a credit card issued by UniversalBank?

R Exercise 1: Binary Classification

- Create a performance evaluation function
  - ✓ True positive rate, Precision, True negative rate, Accuracy, Balance correction rate, and F1-measure

```
# Performance Evaluation Function -----------------------------------------
perf_eval2 <- function(cm){
    # True positive rate: TPR (Recall)
    TPR <- cm[2,2]/sum(cm[2,])
    # Precision
    PRE <- cm[2,2]/sum(cm[,2])
    # True negative rate: TNR
    TNR <- cm[1,1]/sum(cm[1,])
    # Simple Accuracy
    ACC <- (cm[1,1]+cm[2,2])/sum(cm)
    # Balanced Correction Rate
    BCR <- sqrt(TPR*TNR)
    # F1-Measure
    F1 <- 2*TPR*PRE/(TPR+PRE)
    return(c(TPR, PRE, TNR, ACC, BCR, F1))
}
```

R Exercise 1: Binary Classification

- Initialize the performance matrix & Load the dataset

```r
# Initialize the performance matrix
perf_mat <- matrix(0, 1, 6)
colnames(perf_mat) <- c("TPR (Recall)", "Precision", "TNR", "ACC", "BCR", "F1")
rownames(perf_mat) <- "Logstic Regression"

# Load dataset
ploan <- read.csv("Personal Loan.csv")
input_idx <- c(2,3,4,6,7,8,9,11,12,13,14)
target_idx <- 10
ploan_input <- ploan[,input_idx]
ploan_target <- as.factor(ploan[,target_idx])
ploan_data <- data.frame(ploan_input, ploan_target)
```

✓ Column 1 & 5: id and zipcode (irrelevant variables)

✓ Column 10: target variable

✓ Convert the target variable type: numeric → factor

R Exercise 1: Binary Classification

- Normalize and split the dataset

```
# Conduct the normalization
ploan_input <- ploan[,input_idx]
ploan_input <- scale(ploan_input, center = TRUE, scale = TRUE)
ploan_target <- ploan[,target_idx]
ploan_data <- data.frame(ploan_input, ploan_target)

# Split the data into the training/validation sets
set.seed(12345)
trn_idx <- sample(1:nrow(ploan_data), round(0.7*nrow(ploan_data)))
ploan_trn <- ploan_data[trn_idx,] ploan_tst <- ploan_data[-trn_idx,]
```

  - ✓ Conduct normalization for stable learning
  - ✓ Divide the entire dataset into the training set (70%) and test set (30%)

R Exercise 1: Binary Classification

- Training the logistic regression model

```
# Train the Logistic Regression Model with all variables
full_lr <- glm(ploan_target ~ ., family=binomial, ploan_trn)
summary(full_lr)
```

✓ glm( ): generalized linear model

- Arg 1: Formula
- Arg 2: type of model (family = binomial → logistic regression)
- Arg 3: training dataset

R Exercise 1: Binary Classification

- Training the logistic regression model

```
> summary(full_lr)

Call:
glm(formula = ploan_target ~ ., family = binomial, data = ploan_trn)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2973  -0.2366  -0.1081  -0.0482   3.6007

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)        -4.21016    0.22999 -18.306  < 2e-16 ***
Age                -0.05479    1.06837  -0.051  0.95910
Experience          0.23514    1.06214   0.221  0.82480
Income              2.07961    0.17125  12.144  < 2e-16 ***
Family              0.80944    0.13411   6.036 1.58e-09 ***
CCAvg               0.30738    0.10800   2.846  0.00442 **
Education           1.13270    0.14325   7.907 2.63e-15 ***
Mortgage            0.07188    0.08685   0.828  0.40790
Securities.Account -0.44039    0.15266  -2.885  0.00392 **
CD.Account          0.94355    0.12160   7.760 8.52e-15 ***
Online             -0.13209    0.12191  -1.083  0.27859
CreditCard         -0.61753    0.15835  -3.900 9.63e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

R Exercise 1: Binary Classification

- Test the model and evaluate the classification performance

```
lr_response <- predict(full_lr, type = "response", newdata = ploan_tst)
lr_target <- ploan_tst$ploan_target
lr_predicted <- rep(0, length(lr_target))
lr_predicted[which(lr_response >= 0.5)] <- 1
cm_full <- table(lr_target, lr_predicted)
cm_full
```

✓ predict function

- type = "response": return the probability belonging to the positive (1) class
- Set the cut-off value to 0.5
- Compute the confusion matrix

```
> cm_full
         lr_predicted
lr_target   0   1
        0 667   4
        1  26  53
```

R Exercise 1: Binary Classification

- Test the model and evaluate the classification performance

```
perf_mat[1,] <- perf_eval2(cm_full)
perf_mat
```

```
> perf_mat
                    TPR (Recall) Precision       TNR  ACC       BCR        F1
Logstic Regression     0.6708861 0.9298246 0.9940387 0.96 0.8166313 0.7794118
```

✓ The 67% of actual loan users are correctly identified by the logistic regression model

✓ The 93% of customers being identified by the model are actual loan users

✓ The 99.4% of actual non-users are correctly identified by the model

✓ The 96% of customers are correctly identified

R Exercise 2: Multi-class Classification

- Dataset: Wine

Wine Data Set

Download: Data Folder, Data Set Description

Abstract: Using chemical analysis determine the origin of wines

Data Set Characteristics:	Multivariate	Number of Instances:	178	Area:	Physical
Attribute Characteristics:	Integer, Real	Number of Attributes:	13	Date Donated	1991-07-01
Associated Tasks:	Classification	Missing Values?	No	Number of Web Hits:	1087880

The attributes are (dontated by Riccardo Leardi,
1) Alcohol
2) Malic acid
3) Ash
4) Alcalinity of ash
5) Magnesium
6) Total phenols
7) Flavanoids
8) Nonflavanoid phenols
9) Proanthocyanins
10)Color intensity
11)Hue
12)OD280/OD315 of diluted wines
13)Proline

R Exercise 2: Multi-class Classification

- Install package, initiate the performance evaluation function

```
# Multinomial logistic regression
install.packages("nnet")
library(nnet)

perf_eval3 <- function(cm){
     # Simple accuracy
     ACC <- sum(diag(cm))/sum(cm)
     # ACC for each class
     A1 <- cm[1,1]/sum(cm[1,])
     A2 <- cm[2,2]/sum(cm[2,])
     A3 <- cm[3,3]/sum(cm[3,])
     BCR <- (A1*A2*A3)^(1/3)
     return(c(ACC, BCR))
}
```

R Exercise 2: Multi-class Classification

- Load dataset, set the baseline class, divide the dataset

```
wine <- read.csv("wine.csv")
# Define the baseline class
wine$Class <- as.factor(wine$Class)
wine$Class <- relevel(wine$Class, ref = "3")

trn_idx <- sample(1:nrow(wine), round(0.7*nrow(wine)))
wine_trn <- wine[trn_idx,]
wine_tst <- wine[-trn_idx,]
```

✓ Original type of Class variable is "int" → convert its type to "factor"

R Exercise 2: Multi-class Classification

- Train the models

```
# Train multinomial logistic regression
ml_logit <- multinom(Class ~ ., data = wine_trn)

# Check the coefficients
summary(ml_logit)
t(summary(ml_logit)$coefficients)
```

✓ summary( ) function provide the coefficients and standard deviations for each model

```
> summary(ml_logit)
Call:
multinom(formula = Class ~ ., data = wine_trn)

Coefficients:
  (Intercept)  Alcohol.2 Malic.acid.       Ash. Alcalinity.of.ash. Magnesium. Total.phenols. Flavanoids. Nonflavanoid.phenols
1   -150.8796   3.719582   22.733572   63.77061          -9.551572  0.2423116     -110.51008    93.31195            -56.18379
2    198.2972 -28.033655   -1.223123 -125.48033           8.010696  1.7445375      -61.48548    69.49118            220.15011
  Proanthocyanins. Color.intensity.      Hue OD280.OD315.of.diluted.wines.   Proline.
1        -4.839092        -19.49663  38.83918                      10.19197 0.24685710
2         2.687883        -23.41452 154.80004                       2.49729 0.02028825

Std. Errors:
  (Intercept) Alcohol.2 Malic.acid.     Ash. Alcalinity.of.ash. Magnesium. Total.phenols. Flavanoids. Nonflavanoid.phenols
1    22.52277  296.3198    543.0313 40.64535           669.1547   50.97645       74.43353    156.7126             26.32834
2    11.12063  237.4662    154.1817 34.19834           286.9405  216.32121      105.05729    102.8827             38.64278
  Proanthocyanins. Color.intensity.      Hue OD280.OD315.of.diluted.wines. Proline.
1         91.26217        142.65356 13.14472                     109.24921 31.87774
2        147.80114         38.88335 18.04335                      87.27646 32.33280

Residual Deviance: 0.000008193118
AIC: 56.00001
```

R Exercise 2: Multi-class Classification

- Train the models

```
# Train multinomial logistic regression
ml_logit <- multinom(Class ~ ., data = wine_trn)

# Check the coefficients
summary(ml_logit)
t(summary(ml_logit)$coefficients)
```

✓ Coefficients of each model

```
> t(summary(ml_logit)$coefficients)
                                      1             2
(Intercept)                 -150.8796315  198.29724653
Alcohol.2                      3.7195821  -28.03365495
Malic.acid.                   22.7335724   -1.22312289
Ash.                          63.7706125 -125.48032553
Alcalinity.of.ash.            -9.5515724    8.01069633
Magnesium.                     0.2423116    1.74453745
Total.phenols.              -110.5100808  -61.48547503
Flavanoids.                   93.3119457   69.49118380
Nonflavanoid.phenols         -56.1837869  220.15010991
Proanthocyanins.              -4.8390924    2.68788272
Color.intensity.             -19.4966267  -23.41452149
Hue                           38.8391791  154.80004270
OD280.OD315.of.diluted.wines. 10.1919660    2.49729004
Proline.                       0.2468571    0.02028825
```

R Exercise 2: Multi-class Classification

- Interpret the results

```r
# Conduct 2-tailed z-test to compute the p-values
z_stats <- summary(ml_logit)$coefficients/summary(ml_logit)$standard.errors
t(z_stats)

p_value <- (1-pnorm(abs(z_stats), 0, 1))*2
options(scipen=10)
t(p_value)
```

✓ multinorm( ) does not provide the p-values, so we manually compute them

```
> t(p_value)
                                               1                2
(Intercept)                  0.00000000002098766 0.00000000000000
Alcohol.2                    0.98998474329538033 0.90602547076899
Malic.acid.                  0.96660695408866371 0.99367045293444
Ash.                         0.11665910227299237 0.00024331650010
Alcalinity.of.ash.           0.98861131348134723 0.97772785523380
Magnesium.                   0.99620734776521647 0.99356547425947
Total.phenols.               0.13762822597308633 0.55837518665469
Flavanoids.                  0.55155361898111677 0.49939557325969
Nonflavanoid.phenols         0.03284554062986977 0.00000001218935
Proanthocyanins.             0.95771272346227398 0.98549062698237
Color.intensity.             0.89129072835830669 0.54705870613885
Hue                          0.00312936175614698 0.00000000000000
OD280.OD315.of.diluted.wines. 0.92567239450692451 0.97717279806758
Proline.                     0.99382134642228603 0.99949934191516
```

고려대학교 KOREA UNIVERSITY

DSBA Data Science & Business Analytics

R Exercise 2: Multi-class Classification

- Interpret the results

```
cbind(t(summary(ml_logit)$coefficients), t(p_value))
```

✓ Print the coefficients and p-values for each model

```
> cbind(t(summary(ml_logit)$coefficients), t(p_value))
                                    1            2                    1               2
(Intercept)                -150.8796315  198.29724653 0.0000000002098766 0.0000000000000
Alcohol.2                     3.7195821  -28.03365495 0.9899847432953803 0.9060254707689
Malic.acid.                  22.7335724   -1.22312289 0.9666069540886371 0.9936704529344
Ash.                         63.7706125 -125.48032553 0.1166591022729923 0.0002433165001
Alcalinity.of.ash.           -9.5515724    8.01069633 0.9886113134813472 0.9777278552338
Magnesium.                    0.2423116    1.74453745 0.9962073477652164 0.9935654742594
Total.phenols.             -110.5100808  -61.48547503 0.1376282259730863 0.5583751866546
Flavanoids.                  93.3119457   69.49118380 0.5515536189811167 0.4993955732596
Nonflavanoid.phenols        -56.1837869  220.15010991 0.0328455406298697 0.0000000121893
Proanthocyanins.             -4.8390924    2.68788272 0.9577127234622739 0.9854906269823
Color.intensity.            -19.4966267  -23.41452149 0.8912907283583066 0.5470587061388
Hue                          38.8391791  154.80004270 0.0031293617561469 0.0000000000000
OD280.OD315.of.diluted.wines. 10.1919660   2.49729004 0.9256723945069245 0.9771727980675
Proline.                      0.2468571    0.02028825 0.9938213464222860 0.9994993419151
```

Coefficients (1 vs. 3)	Coefficients (2 vs. 3)	p-values (1 vs. 3)	p-values (2 vs. 3)

고려대학교 KOREA UNIVERSITY

DSBA Data Science & Business Analytics

R Exercise 2: Multi-class Classification

- Check the classification accuracy

```
# Predict the class probability
ml_logit_haty <- predict(ml_logit, type="probs", newdata = wine_tst)
ml_logit_haty[1:10,]
```

✓ If we use type = "probs" option, the likelihood for each class is returned

```
> ml_logit_haty[1:10,]
              3 1            2
1    2.571434e-70 1  4.668832e-67
3    3.187174e-81 1  1.659844e-80
4    1.004466e-57 1 1.776978e-116
8    1.080070e-87 1  1.347261e-90
11  2.737317e-110 1  1.326322e-92
13   9.475704e-87 1  2.426455e-98
16   4.975362e-74 1  1.154407e-88
17   1.965882e-77 1  2.817637e-77
18   2.881156e-53 1  9.666261e-37
19  2.890111e-114 1 2.465335e-122
```

R Exercise 2: Multi-class Classification

- Check the classification accuracy

```
# Predict the class label
ml_logit_prey <- predict(ml_logit, newdata = wine_tst)
cfmatrix <- table(wine_tst$Class, ml_logit_prey)
cfmatrix perf_mat_wine[,2] <- perf_eval3(cfmatrix)
perf_mat_wine
```

✓ Without type = "prob" option, the class label with the highest likelihood is returned

```
> cfmatrix
   ml_logit_prey
     3  1  2
  3 12  0  0
  1  0 16  1
  2  3  0 21
> perf_eval3(cfmatrix)
[1] 0.9245283 0.9373311
```


ANY
questions?