# Assignment 3:

## Bank Data Analysis using Classification Algorithm

Author: Yi Yang

ALY 6015-Intermediate Analytics Spring 2020

Instructor: Dr. Justin Rodgers

Due by 06/28/2020

# Introduction

This week's assignment is mainly about using classification algorithm to solve some questions and eventually help a bank management and sales team understand how to maximize clients signing up for a term deposit.

Specifically, we will use a csv file named banking_data from a marketing campaign implemented by a major banking institution. It contains total 41,888 observations of clients' data and each observation has 16 attributes including age, martial, education, occupation and etc. The outcome, y, is whether or not a bank salesperson was able to get a client to sign up for a term deposit (and is labeled 0 for no, and 1 for yes).

# Questions && Results

**Part 1**. We import the "banking_data.csv" dataset into R Studio and visualize the first 6 rows of data.

```
> # Load the banking data
> bank_data <- read.csv("C:/Users/sheny/Desktop/ALY6015/banking_data.csv")
> View(bank_data)
> bank_data<- data.frame(bank_data)
> head(bank_data)
  X age marital education occupation default housing   contact quarter day duration campaign pdays previous    poutcome  y
1 1  56 married         0          0      no      no telephone       0   1      261        1   999        0 nonexistent no
2 2  57 married         1          0 unknown      no telephone       0   1      149        1   999        0 nonexistent no
3 3  37 married         1          0      no     yes telephone       0   1      226        1   999        0 nonexistent no
4 4  40 married         0          1      no      no telephone       0   1      151        1   999        0 nonexistent no
5 5  56 married         1          0      no      no telephone       0   1      307        1   999        0 nonexistent no
6 6  45 married         0          0 unknown      no telephone       0   1      198        1   999        0 nonexistent no
```

**Part 2**. 1). To start, we preprocess to set the outcome variable equal to 0 for no means this client does not sign up for a term deposit, and 1 for yes means this client signs up for a term deposit. Then calculate the probability that any contacted client signs up for a term deposit.

```
> # Question 2: Pre-processing
> # The outcome y: 0 for no, 1 for yes
> bank_data$y <- ifelse(bank_data$y == "no", 0, 1)
>
> # Calculate the probability of y = 1(sign up for a term deposit)
> prop.table(bank_data$y)[bank_data$y == "1"] #p=0.0215% when y = 1
  [1] 0.0002155172 0.0002155172 0.0002155172 0.0002155172 0.0002155172 0.0002155172 0.0002155172 0.0002155172 0.0002155172 0.0002155172
 [11] 0.0002155172 0.0002155172 0.0002155172 0.0002155172 0.0002155172 0.0002155172 0.0002155172 0.0002155172 0.0002155172 0.0002155172
```

2). Fit an intercepts-only logistic regression model to the banking data (recall that an intercept only model can be fit in R as follows: y~1).

```
> # Create an Intercept Model
> intercept_model <-glm(y ~ 1,family = binomial(link = "logit"),data = bank_data)
>
> # Obtain the estimate for the intercept
> intercept_model$coefficients[1]
(Intercept)
  -2.063912
> summary(intercept_model)

Call:
glm(formula = y ~ 1, family = binomial(link = "logit"), data = bank_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.4889  -0.4889  -0.4889  -0.4889   2.0897

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.06391    0.01558  -132.4   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 28999  on 41187  degrees of freedom
Residual deviance: 28999  on 41187  degrees of freedom
AIC: 29001

Number of Fisher Scoring iterations: 4
```

3). Use your estimate for the intercept calculate the probability that y=1 using the formula $\frac{e^{\beta_0}}{1+e^{\beta_0}}$.

```
> # Use the function to calculate the probability
> (exp(intercept_model$coefficients[1]))/(1+exp(intercept_model$coefficients[1]))
(Intercept)
  0.1126542
> exp(-2.063912)/(1+exp(-2.063912))#p = 0.1126542
[1] 0.1126542
```

**Output:** we obtained the estimate -2.063912 above and use the function to get the probability is 0.1126542.

4). Confirm that this is correct by constructing a table for the outcome variable, y with no in one column and yes in the other column). Use these values to calculate the probability by hand. Do they match? [Note, a good library for constructing tables is library(gmodels) with the function: CrossTable(y)].

```
> # Use gmodels package to create a Cross Table
> # To verify with estimate, 0.1126542
> library(gmodels)
> CrossTable(bank_data$y)


   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  41188


          |         0 |         1 |
          |-----------|-----------|
          |     36548 |      4640 |
          |     0.887 |     0.113 |
          |-----------|-----------|




> CrossTable(bank_data$y, digits = 7)


   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  41188


          |         0 |         1 |
          |-----------|-----------|
          |     36548 |      4640 |
          | 0.8873458 | 0.1126542 |
          |-----------|-----------|
```

**Output:** According to the CrossTable above, the probability value for y=1 is 0.1126542 which matches the estimate.

**Part 3.** 1). Next, the bank marketing team would like to know whether their campaign was more successful among lower vs. higher educated clients.

Construct a logistic regression model to answer this question. [Remember to use factor(education) in your model so that R treats this as a categorical variable].

```
> ## Part 3:Logistic Regression for education
> # check the NAs under education
> education1 <- bank_data[-which(is.na(bank_data$education)),]
> str(education1$education)
 Factor w/ 3 levels "0","1","2": 1 2 2 1 2 1 3 3 2 2 ...
>
> # logistic regression model
> # covert education values from int to factor
> education1$education <- as.factor(education1$education)
> education_model <-glm(y ~ education,family = binomial(link = "logit"),data = education1)
> summary(education_model)

Call:
glm(formula = y ~ education, family = binomial(link = "logit"),
    data = education1)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.5280  -0.5280  -0.4789  -0.4272   2.2088

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.34801    0.03166 -74.164  < 2e-16 ***
education1   0.24036    0.04572   5.257 1.46e-07 ***
education2   0.44785    0.03886  11.526  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 27548  on 39456  degrees of freedom
Residual deviance: 27409  on 39454  degrees of freedom
AIC: 27415

Number of Fisher Scoring iterations: 5
```

**Output and interpretation:** each one-unit value in education-1 from education-0 will increase by 0.24036 and going from education-2 to education-1 will increase it by 0.44785.

Also, the difference between null deviance and residual deviance is 139 among 39456 observations (observations with NAs are removed), which tells us the model is a good fit.

2). What is the Odds Ratio for the highest education group (education=2) compared to the lowest education group (education=0)?

```
> # log odds
> summary(education_model)$coeff
              Estimate Std. Error    z value     Pr(>|z|)
(Intercept) -2.3480149 0.03165977 -74.164002 0.000000e+00
education1   0.2403621 0.04571807   5.257486 1.460380e-07
education2   0.4478532 0.03885699  11.525679 9.793722e-31
>
> # Odds ratios
> exp(coef(education_model))
(Intercept)  education1  education2
 0.09555866  1.27170956  1.56494898
>
> # odds ratio for education=2 compared to education=0
> exp(education_model$coefficients[3])#1.564949
education2
  1.564949
```

**Output:** The odds ratio for education=2 compared to education=0 is 1.564949
3). How would you interpret this in plain words to the marketing team?
Is this a significant association?
**Interpretation**: The odds ratio is 1.564949 means we expect to see about 56% increase in the odds of highest education group compared to lowest education group.

4). What is the probability that the lowest education group (education=0) signed up for a term deposit (y=1) in response to this campaign?

```
> #probability when education=0
> (exp(education_model$coefficients[1]))/(1+exp(education_model$coefficients[1]))
(Intercept)
 0.08722369
```

5). What is the probability that the highest education group (education=2) signed up for a term deposit (y=1) in response to this campaign?

```
> #probability when education=2
> pred_ex<-predict(education_model,newdata =education1[education1$education==2,],type="response")
> head(pred_ex,1)
        7
0.1300902
```

**Part 4.** Lastly, the IT team would like to build a program that prompts sales personnel to up their game when speaking to a client with a high probability of signing up. But first, they need you to build a predictive model.

1). First, split the data into a training dataset and a test dataset, with 80% of observations randomly going to the training data and 20% randomly going to the test data.

```
> ## Part 4:
> # remove missing values in bank data
> bank_omit <- na.omit(bank_data)
> View(bank_omit)
>
> # split the bank data into train and test data
> set.seed(12345)
> row.number <- sample(x=1:nrow(bank_omit), size=0.8*nrow(bank_omit))
> train = bank_omit[row.number,]
> test = bank_omit[-row.number,]
> head(train)
          X age  marital education occupation default housing   contact quarter day duration campaign pdays
11553 11553  36  married         0          0      no     yes telephone       1   5      560        3   999
34884 34884  32   single         2          1      no     yes  cellular       0   5      347        1   999
10584 10584  37   single         0          0 unknown unknown telephone       1   2       82        2   999
86       86  31 divorced         1          1      no      no telephone       0   1      246        1   999
31765 31765  32  married         1          1      no      no  cellular       0   4      142        1   999
27663 27663  48  married         2          1 unknown      no  cellular       2   5       71        9   999
      previous    poutcome   y
11553        0 nonexistent yes
34884        0 nonexistent  no
10584        0 nonexistent  no
86           0 nonexistent  no
31765        1     failure  no
27663        0 nonexistent  no
> head(test)
    X age marital education occupation default housing   contact quarter day duration campaign pdays previous
7   7  59 married         2          1      no      no telephone       0   1      139        1   999        0
10 10  25  single         1          0      no     yes telephone       0   1       50        1   999        0
12 12  25  single         1          0      no     yes telephone       0   1      222        1   999        0
18 18  46 married         0          0 unknown     yes telephone       0   1      440        1   999        0
21 21  30 married         1          2      no      no telephone       0   1       38        1   999        0
34 34  54 married         0          1 unknown     yes telephone       0   1      230        1   999        0
      poutcome  y
7  nonexistent no
10 nonexistent no
12 nonexistent no
18 nonexistent no
21 nonexistent no
34 nonexistent no
```

2). Then, using any or all of the data at your disposal please fit a logistic regression model with y as the outcome and the training data for the dataset.

```
> # logistic regression model using train data
> train_model <-glm(y ~ factor(education),family = binomial(link = "logit"),data = train)
> summary(train_model)

Call:
glm(formula = y ~ factor(education), family = binomial(link = "logit"),
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.5266  -0.5266  -0.4768  -0.4250   2.2132

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)        -2.35893    0.03580  -65.89  < 2e-16 ***
factor(education)1  0.24190    0.05169    4.68 2.87e-06 ***
factor(education)2  0.45350    0.04376   10.36  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 21850  on 31405  degrees of freedom
Residual deviance: 21738  on 31403  degrees of freedom
AIC: 21744

Number of Fisher Scoring iterations: 5
```

3). Next, use the predict function to get predicted values of the outcome from the test dataset (simulating future data). What percent of cases did you get correct (i.e., what was the prediction accuracy of your model)? Use a cut-off of 0.5 for translating your predicted probabilities into values of "yes" and "no".

```
> # predict function of test data
> predicted <- predict(train_model, data=test, type="response")
> predicted
     11553      34884      10584         86      31765      27663      22555      26805      37283      33245
0.08635858 0.12949486 0.08635858 0.10745234 0.10745234 0.12949486 0.10745234 0.12949486 0.08635858 0.08635858
       443      24919       9315      29506      22168       3045      14730      34218       5593      16377
0.08635858 0.10745234 0.12949486 0.08635858 0.10745234 0.12949486 0.12949486 0.10745234 0.10745234 0.08635858
     17959      15144      10386      34633      23172      10432      35241       1374      21084      34423
0.08635858 0.08635858 0.08635858 0.10745234 0.12949486 0.08635858 0.10745234 0.08635858 0.12949486 0.08635858
      4232      35198      27344      14875      26374      26122      40411      22140      35272      12957
> CrossTable(predicted)


   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  31406


                 | 0.0863585802406512 |  0.107452339688024 |  0.129494863013691 |
                 |--------------------|--------------------|--------------------|
                 |               9889 |               7501 |              14016 |
                 |              0.315 |              0.239 |              0.446 |
                 |--------------------|--------------------|--------------------|


> library(randomForest)
> fitRF <- randomForest(y~., train)
> fitRF

Call:
 randomForest(formula = y ~ ., data = train)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 8.61%
Confusion matrix:
       no  yes class.error
no  27106  825  0.02953707
yes  1879 1596  0.54071942
```

# Conclusion

Based on all these classification methods we used to solve the questions above, we can conclude that these can be served as great tools to better help business make suitable business decisions if being used properly in the real life.

# Appendix-R code

```
# ALY 6015 Assignment 3

# Yi Yang and 06/26/2020


## Part 1: Load the banking data

bank_data <- read.csv("C:/Users/sheny/Desktop/ALY6015/banking_data.csv")

View(bank_data)

bank_data<- data.frame(bank_data)

head(bank_data)


## Part 2: Pre-processing

# The outcome y: 0 for no, 1 for yes

bank_data$y <- ifelse(bank_data$y == "no", 0, 1)


# Calculate the probability of y = 1(sign up for a term deposit)

prop.table(bank_data$y)[bank_data$y == "1"] #p=0.0215% when y = 1


# Create an Intercept Model

intercept_model <-glm(y ~ 1,family = binomial(link = "logit"),data = bank_data)


# Obtain the estimate for the intercept

intercept_model$coefficients[1]#-2.063912

summary(intercept_model)


# Use the function to calculate the probability

(exp(intercept_model$coefficients[1]))/(1+exp(intercept_model$coefficients[1]))
```

```
exp(-2.063912)/(1+exp(-2.063912))#p = 0.1126542

# Use gmodels package to create a Cross Table
# To verify with the probability, 0.1126542
library(gmodels)
CrossTable(bank_data$y)
CrossTable(bank_data$y, digits = 7)

## Part 3: Logistic Regression for education
# check the NAs under education
education1 <- bank_data[-which(is.na(bank_data$education)),]
str(education1$education)

# logistic regression model
# covert education values from int to factor
education1$education <- as.factor(education1$education)
education_model <-glm(y ~ education,family = binomial(link = "logit"),data =
education1)
summary(education_model)

# log odds
summary(education_model)$coeff

# Odds ratios
exp(coef(education_model))

# odds ratio for education=2 compared to education=0
exp(education_model$coefficients[3])#1.564949

#probability when education=0
(exp(education_model$coefficients[1]))/(1+exp(education_model$coefficients[1]))
```

```
#predict function to verify

pred_ex<-predict(education_model,newdata
=education1[education1$education==0,],type="response")

head(pred_ex,1)


#probability when education=2

pred_ex<-predict(education_model,newdata
=education1[education1$education==2,],type="response")

head(pred_ex,1)


## Part 4:

# remove missing values in bank data

bank_omit <- na.omit(bank_data)

View(bank_omit)

# convert all attributes except y to numeric

bank_new <- data.frame(as.numeric(as.factor(bank_omit$X)),
                  as.numeric(as.factor(bank_omit$age)),
                  as.numeric(as.factor(bank_omit$marital)),
                  as.numeric(as.factor(bank_omit$education)),
                  as.numeric(as.factor(bank_omit$occupation)),
                  as.numeric(as.factor(bank_omit$default)),
                  as.numeric(as.factor(bank_omit$housing)),
                  as.numeric(as.factor(bank_omit$contact)),
                  as.numeric(as.factor(bank_omit$quarter)),
                  as.numeric(as.factor(bank_omit$day)),
                  as.numeric(as.factor(bank_omit$duration)),
                  as.numeric(as.factor(bank_omit$campaign)),
                  as.numeric(as.factor(bank_omit$pdays)),
                  as.numeric(as.factor(bank_omit$previous)),
                  as.numeric(as.factor(bank_omit$poutcome)),
                  bank$y)
```

```
# Split the bank data into train and test data
set.seed(12345)
row.number <- sample(x=1:nrow(bank_omit), size=0.8*nrow(bank_omit))
train = bank_omit[row.number,]
test = bank_omit[-row.number,]
head(train)
head(test)

# random forest
library(randomForest)
fitRF <- randomForest(y~., train)
fitRF

# logistic regression model using train data
train_model <-glm(y ~ factor(education).,family = binomial(link = "logit"),data =
train)
summary(train_model)

# predict function of test data
predicted <- predict(train_model, data=test, type="response")
head(predicted,1)
CrossTable(predicted)

#predicted probabilities as either Y=1 or Y=0 based on some cutoff value 0.05
y_pred_num <- ifelse(predicted > 0.5, 1, 0)
library(InformationValue)
optCutOff <- optimalCutoff(test$y, predicted, )[1]
y_pred_num <- ifelse(predicted >= optCutOff, 1, 0)

table(y_pred_num,test$y)
```