PREDICTING CREDIT DEFAULT PROBABILITY

Kyle Nunn and Omoyeni Ogundipe

Final Project

Regression Analysis

MATH 5120

Introduction

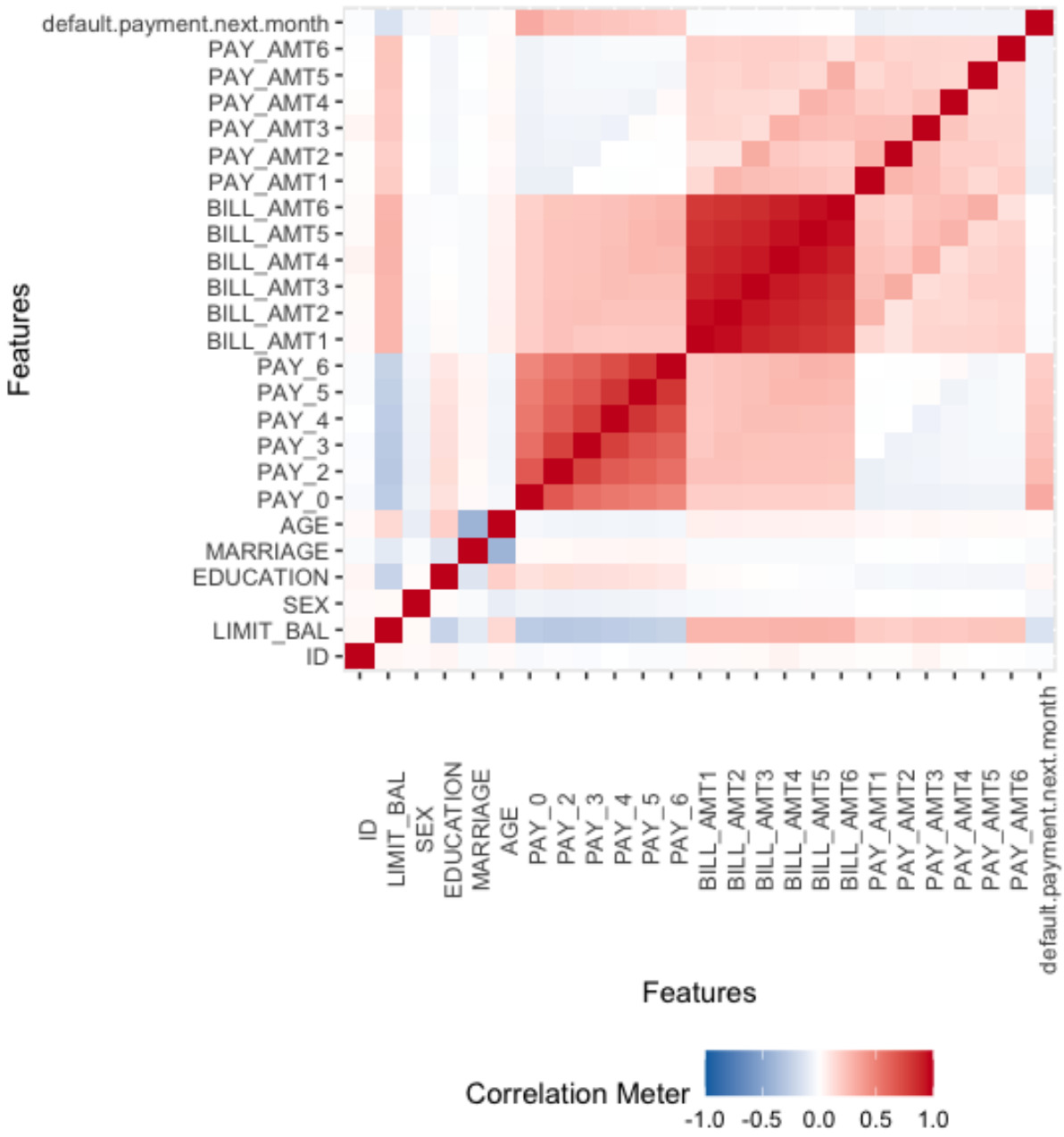
In this project, we will look at how well we can predict the probability of credit card default on loan using logistic regression. The data set we will use is the 'Default Payments of Credit Card Clients in Taiwan. The original dataset includes 30,000 rows and 30 columns. These columns include ID of each client, LIMIT_BAL: Amount of given credit in NT dollars (includes individual and family/supplementary credit, SEX: Gender (1=male, 2=female), EDUCATION: (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown), MARRIAGE: Marital status (1=married, 2=single, 3=others), AGE: Age in years, PAY_0: Repayment status in September, 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, … 8=payment delay for eight months, 9=payment delay for nine months and above), PAY_2: Repayment status in August, 2005 (scale same as above), PAY_3: Repayment status in July, 2005 (scale same as above), PAY_4: Repayment status in June, 2005 (scale same as above), PAY_5: Repayment status in May, 2005 (scale same as above), PAY_6: Repayment status in April, 2005 (scale same as above), BILL_AMT1: Amount of bill statement in September, 2005 (NT dollar), BILL_AMT2: Amount of bill statement in August, 2005 (NT dollar), BILL_AMT3: Amount of bill statement in July, 2005 (NT dollar), BILL_AMT4: Amount of bill statement in June, 2005 (NT dollar), BILL_AMT5: Amount of bill statement in May, 2005 (NT dollar), BILL_AMT6: Amount of bill statement in April, 2005 (NT dollar), PAY_AMT1: Amount of previous payment in September, 2005 (NT dollar), PAY_AMT2: Amount of previous payment in August, 2005 (NT dollar), PAY_AMT3: Amount of previous payment in July, 2005 (NT dollar), PAY_AMT4: Amount of previous payment in June, 2005 (NT dollar), PAY_AMT5: Amount of previous payment in May, 2005 (NT dollar), PAY_AMT6: Amount of previous payment in April, 2005 (NT dollar), 'default.payment.next.month': Default payment (1=yes, 0=no). Credit card and loan defaults are of high interest to banks. It is important for them to know which predictor variables increase the probability of a client defaulting so that they can maximize their returns. Obviously, this data sets only carries a few important predictor variables in relation to the amount of information actual banks have. Nevertheless, we will construct a logistic regression model to see how we this model predicts the probability of default.

Important questions for this project:

1. Which variables are most correlated? Furthermore, which variables increase the probability of a client defaulting in next month's payment?
2. How reliable is our model of predicting probability of loan default?
3. What can we do to improve the logistic regression model?

Analysis

Let's take a look at question at question 1. To find out which variables are the most important for the model, we will take a look at a heat map to visibly see which variables are highly correlated to next month's default and which ones don't matter at all.



As we can see, the repayment status of a client (PAY_0 : PAY_6) has the highest correlation of all the numerical variables, While it is probably insignificant to the model (we technically don't know for sure yet), it is pretty interesting to see that EDUCATION has the highest positive correlation to default. Below, we will run a model without BILL_AMT1 through BILL_AMT6 so that we can make a more efficient model. In addition, we will split the data into a training and testing set of 70% and 30% of the data, respectively.

```
Call:
glm(formula = default.payment.next.month ~ ., family = binomial(link = "logit"),
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0888  -0.6979  -0.5531  -0.3003   3.0202

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.45566    0.01956 -74.429  < 2e-16 ***
LIMIT_BAL   -0.07112    0.02375  -2.994 0.002752 **
SEX         -0.05453    0.01773  -3.075 0.002104 **
EDUCATION   -0.07045    0.01918  -3.673 0.000240 ***
MARRIAGE    -0.12594    0.01833  -6.872 6.32e-12 ***
PAY_0        0.63878    0.02358  27.094  < 2e-16 ***
PAY_2        0.11363    0.02867   3.964 7.38e-05 ***
PAY_3        0.06652    0.03223   2.064 0.039027 *
PAY_4        0.02316    0.03473   0.667 0.504791
PAY_5        0.06202    0.03611   1.718 0.085864 .
PAY_6        0.01138    0.03049   0.373 0.708959
BILL_AMT1   -0.13141    0.02345  -5.603 2.11e-08 ***
PAY_AMT1    -0.17603    0.03943  -4.464 8.04e-06 ***
PAY_AMT2    -0.16037    0.04724  -3.395 0.000686 ***
PAY_AMT3    -0.03547    0.02909  -1.219 0.222781
PAY_AMT4    -0.05785    0.02841  -2.036 0.041703 *
PAY_AMT5    -0.07364    0.02933  -2.511 0.012036 *
PAY_AMT6    -0.02951    0.02562  -1.152 0.249314
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 22198  on 20999  degrees of freedom
Residual deviance: 19604  on 20982  degrees of freedom
AIC: 19640

Number of Fisher Scoring iterations: 5
```

With the standardized training data, we can see which variables are the most significant in predicting the probability of default for a particular client. Next, we will make a confusion matrix to determine how accurate our model is in predicting loan default.

```
log.prediction.rd    0    1
               0 6774 1561
               1  188  477
```

Conclusions

Obviously, we can see that the model predicts a default or no default more times than a Type 1 or Type 2 error is committed. This is a good thing, but we still need to find the exact accuracy of the model. A bank's goal is to make money, so it is extremely important that we know how accurate the model is so that we can make smart decisions as to who gets a loan and who does not. For this model, the accuracy was 0.8056667. So, in answering question 2, we can predict the probability of default with around 81% accuracy.

While 81% is obviously better than half, a bank would not be satisfied with this model. The banking and finance industry is obviously very competitive, so the reliability of the model needs to be as accurate as possible. In terms of the above created model, we had limited variables that could have lessened the value of the model. Therefore, it is important to consider as many variables as possible to find the highest quality variables that determine the probability of the model. According to Walls Fargo, banks and lenders are guided by the five C's: credit history, capacity, collateral, capital and conditions. Many of the variables of the stated "five C's" were left out of our model. Some of these variables include credit history, income level, employment history, environmental conditions, and economic conditions. Each of these variables can be specific with numerical or categorical variables. All these variables are important in finding the probability of defaulting on a loan with the most reliable model as possible.

Accurate and extensive data is not easily obtained, which is most likely why there were so many limited-variable data sets on the internet. In future models, we will find a way to get a data set with a higher quantity of high-quality variables so that we can more accurately predict the probability of a client defaulting on next month's loan or for the whole loan in general.

References

https://www.wellsfargo.com/financial-education/credit-management/five-c/

https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset

https://www.r-bloggers.com/2019/11/logistic-regression-in-r-a-classification-technique-to-predict-credit-card-default/

https://rpubs.com/SameerMathur/LR_CreditCardDefault_Taiwan

Appendix

```
# Kyle Nunn and Omoyeni M. Ogundipe
# Regression Analysis : Project
# Logistic Regression: Credit Card/ Loan Defaults

library(data.table)
setwd("~/Desktop/Regression Analysis")
default.data <- read.csv("Credit Card Default Data.csv")
attach(default.data)
dim(default.data)
default.data

# Code stolen from https://rpubs.com/SameerMathur/LR_CreditCardDefault_Taiwan
# First, we are going to run logistic regression based on the first couple of variables for simplicity sake, including:
# Afterwards, we will run a logistic regression model with all of the variables in the data set.
# After running the first and second models, we will run a logistic regression model on a new set of data and compare the two data sets
# After comparing the two data sets, we will make our conclusion as to which variables have the best correlation to predicting default loans
# After drawing conclusions, we will make a hypothesis as to which variables that were not included in each data set might be beneficial for predicting probability of default.

head(default.data)
str(default.data)

# Now we want to convert some of the variables that are integers into factors including : ID, SEX, EDUCATION, MARRIAGE, and DEFAULT
default.data$ID <- as.factor(default.data$ID)
default.data$SEX <- as.factor(default.data$SEX)
default.data$EDUCATION <- as.factor(default.data$EDUCATION)
default.data$MARRIAGE <- as.factor(default.data$MARRIAGE)
default.data$default.payment.next.month <- as.factor(default.data$default.payment.next.month)
str(default.data)

# Changing levels of Default variable
levels(default.data$default.payment.next.month) <- c("No","Yes")

# Varifying conversion
str(default.data)

# Now we are going to split the data into training and testing data
library(caret)
set.seed(2341)

trainIndex <- createDataPartition(default.data$default.payment.next.month, p = .8, list = FALSE)
```

```
traindata <- default.data[trainIndex,]
testdata <- default.data[-trainIndex,]
dim(traindata)
dim(testdata)

# Now we will train the logistic regression model
# First, we will set the control parameters using the bootstrap
objControl <- trainControl(method = "boot",
                           number = 2,
                           returnResamp = 'none',
                           summaryFunction = twoClassSummary,
                           classProbs = TRUE,
                           savePredictions = TRUE)

# Now we will run the training model
set.seed(766)
# model building using caret package
LRModel <- train(default.payment.next.month ~ LIMIT_BAL
                 + SEX
                 + EDUCATION
                 + MARRIAGE
                 + AGE,
                 data = traindata,
                 method = 'glm',
                 trControl = objControl,
                 metric = "ROC")

# summary of the model
summary(LRModel)

# According the summary of the output of the model, it appears as if there are some statistically
significant variables that will sufficiently predict the probability of defaulting next month.
# While it may appear that way, we know that while they are technically statistically significant,
the estimates are so low that they are of no real value.

# predicting the model on test data set
PredLR <- predict(LRModel, testdata,type = "prob")

# Now we will predict the predicted probabilities
# plot of probabilities
plot(PredLR$Yes,
    main = "Scatterplot of Probabilities of default (test data)",
    xlab = "Customer ID",
    ylab = "Predicted Probability of default")
```

```
# Obviously, using the above predictor variables are not good predictors of whether someone
will default on their loan.
# Now we will locate the range of predicted probabilities
range <- range(PredLR$Yes)
format(range, scientific = FALSE)

# Now we will make confusion matrix cut-off probability at .20

pred.LR <- ifelse(PredLR$Yes > 0.20, "Yes", "No")
Predicted <- ordered(pred.LR, levels = c("Yes", "No"))

# actual and predicted data columns
Predicted <- as.factor(Predicted)
Actual <- as.factor(testdata$default.payment.next.month)
# making confusion matrix
cm <-confusionMatrix(data =Predicted,reference = Actual,
                     positive = "Yes")
cm

# Now we will calculate accuracy, sensitivity, and specificity.

# function to print confusion matrices for diffrent cut-off levels of probability
CmFn <- function(cutoff) {

  # predicting the test set results
  Pred.LR <- predict(LRModel, testdata,type = "prob")
  C1 <- ifelse(Pred.LR$Yes > cutoff, "Yes", "No")
  C2 <- testdata$default.payment.next.month
  predY   <- as.factor(C1)
  actualY <- as.factor(C2)

  # ordering the levels of predicted variable
  predY <- ordered(predY, levels = c("Yes", "No"))

  # use the confusionMatrix from the caret package
  cm1 <-confusionMatrix(data = predY,reference = actualY, positive = "Yes")
  # extracting accuracy
  Accuracy <- cm1$overall[1]
  # extracting sensitivity
  Sensitivity <- cm1$byClass[1]
  # extracting specificity
  Specificity <- cm1$byClass[2]
  # extracting value of kappa
  Kappa <- cm1$overall[2]

  # combined table
```

```
  tab <- cbind(Accuracy,Sensitivity,Specificity,Kappa)
  return(tab)}
# sequence of cut-off probability
cutoff1 <- seq( .01, .4, by = .03 )

# loop using "lapply"
tab2    <- lapply(cutoff1, CmFn)
# extra coding for saving table as desired format
tab3 <- rbind(tab2[[1]],tab2[[2]],tab2[[3]],tab2[[4]],tab2[[5]],tab2[[6]],tab2[[7]],
              tab2[[8]],tab2[[9]],tab2[[10]],tab2[[11]],tab2[[12]],tab2[[13]],tab2[[14]])
# printing the table
tab4 <- as.data.frame(tab3)
tab5 <- cbind(cutoff1,tab4$Accuracy,tab4$Sensitivity,tab4$Specificity,tab4$Kappa)
tab6 <- as.data.frame(tab5)

pm <- setnames(tab6, "cutoff1", "cutoff")
pm <- setnames(pm, "V2", "Accuracy")
pm <- setnames(pm, "V3", "Senstivity")
pm <- setnames(pm, "V4", "Specificity")
pm <- setnames(pm, "V5", "kappa")
pm

# Now we will plot ROC Curve
library(ROCR)

# Plotting the curves
PredLR <- predict(LRModel, testdata,type = "prob")
lgPredObj <- prediction(PredLR[2],testdata$default.payment.next.month)
lgPerfObj <- performance(lgPredObj, "tpr","fpr")
# plotting ROC curve
plot(lgPerfObj,main = "ROC Curve",col = 2,lwd = 2)
abline(a = 0,b = 1,lwd = 2,lty = 3,col = "black")

# Finding area under the curve
aucLR <- performance(lgPredObj, measure = "auc")
aucLR <- aucLR@y.values[[1]]
aucLR

# Now we move on to the full logistic regression model for the full data set

# Following R code borrowed from https://www.r-bloggers.com/2019/11/logistic-regression-in-
r-a-classification-technique-to-predict-credit-card-default/

# First step, we will import the neccessary libraries (including ones from above)

library(knitr)
```

```
library(tidyverse)
library(ggplot2)
library(lattice)
library(reshape2)

# Now we will import the data set again

DefaultData <- read.csv("Credit Card Default Data.csv")
head(DefaultData1)

# We will now rename "default payment next month" to simply "default_payment to avoid any discrepancies and to make things simpler.

colnames(DefaultData)[colnames(DefaultData)=="default payment next month"] <-
"default_payment"
head(DefaultData)

# As we have learned throughout the course, we will now conduct exploratory data analaysis
# In doing so, we will be able to visualize the data, find relations between different variables, and even deal with missing values and outliers.

dim(DefaultData)
str(DefaultData)

DefaultData[, 1:25] <- sapply(DefaultData[, 1:25], as.character)
str(DefaultData)

DefaultData[, 1:25] <- sapply(DefaultData[, 1:25], as.numeric)
str(DefaultData)

summary(DefaultData)

# Finding how much of each categorical variable there is in the dataset:
# Also, we are going to 'attach' the data so we can use headings in code
attach(DefaultData)
count(DefaultData, vars = EDUCATION)
count(DefaultData, vars = MARRIAGE)

# Now, for simplicity sake, we are going to converge 'like' variables or ones we don't know that much about..
DefaultData$EDUCATION[DefaultData$EDUCATION == 0] <- 4
DefaultData$EDUCATION[DefaultData$EDUCATION == 5] <- 4
DefaultData$EDUCATION[DefaultData$EDUCATION == 6] <- 4
DefaultData$MARRIAGE[DefaultData$MARRIAGE == 0] <- 3
count(DefaultData, vars = MARRIAGE)
count(DefaultData, vars = EDUCATION)
```

```
# Now we can move on the multi-variate analysis of the variables in the data set, introducing a heat map.
install.packages("DataExplorer")
library(DataExplorer)


plot_correlation(na.omit(DefaultData), maxcat = 5L)

# We can see right off the bat that the factors such as Sex, Education, Marriage, and Age don't have a strong correlation with next month default probability.
# Looking at the heat map, we can kind of tell which predictor variables have an extremely low (we don't technically know they are signficantly low)
# Knowing which predictors have low correlation, we can delete them from the data frame to make our model clearer

DefaultData_New <- select(DefaultData, -one_of('ID','AGE', 'BILL_AMT2',
                          'BILL_AMT3','BILL_AMT4','BILL_AMT5','BILL_AMT6'))
head(DefaultData_New)

# Now we need to standardize the data
# As we have learned, when we standardize, the mean of the data is now 0 and the standard deviation is 1

DefaultData_New[, 1:17] <- scale(DefaultData_New[, 1:17])
head(DefaultData_New)

# Now that we have standardized the data, we can split the data into a training and testing set
# We we use 70% of the data for training and 30% of the data for testing

data2 <- sort(sample(nrow(DefaultData_New), nrow(DefaultData_New)*.7))
train2 <- DefaultData_New[data2,]
test2 <- DefaultData_New[-data2,]
dim(train2)
dim(test2)

# Let's build our model now!! Logistic Regression is cool!

log.model <- glm(default.payment.next.month ~., data = train, family = binomial(link = "logit"))
summary(log.model)

test2[1:10,]

log.predictions <- predict(log.model, test2, type="response")
## Look at probability output
head(log.predictions, 10)
```

```
log.prediction.rd <- ifelse(log.predictions > 0.5, 1, 0)
head(log.prediction.rd, 10)

# Now we can evaluate the model using a confusion matrix to see what percentage of the time
our model will correctly predict a default or no default.

table(log.prediction.rd, test2[,18])
log.prediction.rd
accuracy <- table(log.prediction.rd, test2[,18])
sum(diag(accuracy))/sum(accuracy)
```

References

https://www.r-bloggers.com/2019/11/logistic-regression-in-r-a-classification-technique-to-predict-credit-card-default/

https://rpubs.com/SameerMathur/LR_CreditCardDefault_Taiwan

https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset