**Default of credit card clients**

**1. Introduction**

The focus of this research is about developing a model to predict the probability of default of credit customers. By developing the model, we can generate a model to predict the behavior of customers based on some other dependent factors. To do this, a logistic regression model is used. 23 different dependent variables will be used to define the response variable. The historical dataset is used to train and validate the model. For validation stage, the confusion matrix is used to calculate type I and II errors. The historical data contains the customer default payments in Taiwan and we use the binary variables as well as normal variables in the model.

**2. Experimental Results**

In this section, the dataset, Method and the results will be presented for the problem.

**2.1. Data Set Information**

This research aimed at the case of customers default payments in Taiwan and compares the predictive accuracy of probability. From the perspective of risk management, the result of predictive accuracy of the estimated probability of default will be more valuable than the binary result of classification - credible or not credible clients. Because the real probability of default is unknown, this study presented the novel Sorting Smoothing Method to estimate the real probability of default. With the real probability of default as the response variable (Y), and the predictive probability of default as the independent variable (X), the simple linear regression result (Y = A + BX) shows that the forecasting model produced by artificial neural network has the highest coefficient of determination; its regression intercept (A) is close to zero, and regression coefficient (B) to one.

**2.2. Attribute Information**

This research employed a binary variable, default payment (Yes = 1, No = 0), as the response variable. This study reviewed the literature and used the following 23 variables as explanatory variables:   
X1: Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit.   
X2: Gender (1 = male; 2 = female).   
X3: Education (1 = graduate school; 2 = university; 3 = high school; 4 = others).   
X4: Marital status (1 = married; 2 = single; 3 = others).   
X5: Age (year).   
X6 - X11: History of past payment. We tracked the past monthly payment records (from April to September, 2005) as follows: X6 = the repayment status in September 2005;

X7 = the repayment status in August, 2005;

X11 = the repayment status in April, 2005. The measurement scale for the repayment status is: -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.   
X12-X17: Amount of bill statement (NT dollar).

X12 = amount of bill statement in September 2005;

X13 = amount of bill statement in August 2005;

X17 = amount of bill statement in April, 2005.   
X18-X23: Amount of previous payment (NT dollar). X18 = amount paid in September 2005;

X19 = amount paid in August 2005;

X23 = amount paid in April 2005.

**3. Method**

The purpose of this project, is to estimate a Logistic Regression Model to the credit card customers, so we can calculate the Probability of default based on the inputs. First, let’s call the data set and run it in R.

setwd("/Users/Farzad/Desktop/Capstone Project")

Default <- read.csv("Default.csv")

Now, we take a look at the independent and dependent variables. but before taking a look at the variables, we should present the coded variables as a factor.

Default$X2 <- as.factor(Default$X2)

Default$X3 <- as.factor(Default$X3)

Default$X4 <- as.factor(Default$X4)

Default$X6 <- as.factor(Default$X6)

Default$X7 <- as.factor(Default$X7)

Default$X8 <- as.factor(Default$X8)

Default$X9 <- as.factor(Default$X9)

Default$X10 <- as.factor(Default$X10)

Default$X11 <- as.factor(Default$X11)

Default$Y <- as.factor(Default$Y)

dim(Default)

## [1] 30000 24

str(Default)

## 'data.frame': 30000 obs. of 24 variables:

## $ X1 : int 20000 120000 90000 50000 50000 50000 500000 100000 140000 20000 ...

## $ X2 : Factor w/ 2 levels "1","2": 2 2 2 2 1 1 1 2 2 1 ...

## $ X3 : Factor w/ 7 levels "0","1","2","3",..: 3 3 3 3 3 2 2 3 4 4 ...

## $ X4 : Factor w/ 4 levels "0","1","2","3": 2 3 3 2 2 3 3 3 2 3 ... ## $ X5 : int 24 26 34 37 57 37 29 23 28 35 ...

## $ X6 : Factor w/ 11 levels "-2","-1","0",..: 5 2 3 3 2 3 3 3 3 1 ...

## $ X7 : Factor w/ 11 levels "-2","-1","0",..: 5 5 3 3 3 3 3 2 3 1 ...

## $ X8 : Factor w/ 11 levels "-2","-1","0",..: 2 3 3 3 2 3 3 2 5 1 ...

## $ X9 : Factor w/ 11 levels "-2","-1","0",..: 2 3 3 3 3 3 3 3 3 1 ...

## $ X10: Factor w/ 10 levels "-2","-1","0",..: 1 3 3 3 3 3 3 3 3 2 ...

## $ X11: Factor w/ 10 levels "-2","-1","0",..: 1 4 3 3 3 3 3 2 3 2 ...

## $ X12: int 3913 2682 29239 46990 8617 64400 367965 11876 11285 0 ...

## $ X13: int 3102 1725 14027 48233 5670 57069 412023 380 14096 0 ... ## $ X14: int 689 2682 13559 49291 35835 57608 445007 601 12108 0 ... ## $ X15: int 0 3272 14331 28314 20940 19394 542653 221 12211 0 ... ## $ X16: int 0 3455 14948 28959 19146 19619 483003 -159 11793 13007 ...

## $ X17: int 0 3261 15549 29547 19131 20024 473944 567 3719 13912 ...

## $ X18: int 0 0 1518 2000 2000 2500 55000 380 3329 0 ...

## $ X19: int 689 1000 1500 2019 36681 1815 40000 601 0 0 ...

## $ X20: int 0 1000 1000 1200 10000 657 38000 0 432 0 ...

## $ X21: int 0 1000 1000 1100 9000 1000 20239 581 1000 13007 ...

## $ X22: int 0 0 1000 1069 689 1000 13750 1687 1000 1122 ... ## $ X23: int 0 2000 5000 1000 679 800 13770 1542 1000 0 ...

## $ Y : Factor w/ 2 levels "0","1": 2 2 1 1 1 1 1 1 1 1 ...

summary(Default)

## X1 X2 X3 X4 X5

## Min. : 10000 1:11888 0: 14 0: 54 Min. :21.00

## 1st Qu.: 50000 2:18112 1:10585 1:13659 1st Qu.:28.00

## Median : 140000 2:14030 2:15964 Median :34.00

## Mean : 167484 3: 4917 3: 323 Mean :35.49

## 3rd Qu.: 240000 4: 123 3rd Qu.:41.00

## Max. :1000000 5: 280 Max. :79.00

## 6: 51

## X6 X7 X8 X9

## 0 :14737 0 :15730 0 :15764 0 :16455

## -1 : 5686 -1 : 6050 -1 : 5938 -1 : 5687

## 1 : 3688 2 : 3927 -2 : 4085 -2 : 4348

## -2 : 2759 -2 : 3782 2 : 3819 2 : 3159

## 2 : 2667 3 : 326 3 : 240 3 : 180

## 3 : 322 4 : 99 4 : 76 4 : 69

## (Other): 141 (Other): 86 (Other): 78 (Other): 102

## X10 X11 X12 X13 ## 0 :16947 0 :16286 Min. :-165580 Min. :-69777 ## -1 : 5539 -1 : 5740 1st Qu.: 3559 1st Qu.: 2985 ## -2 : 4546 -2 : 4895 Median : 22382 Median : 21200 ## 2 : 2626 2 : 2766 Mean : 51223 Mean : 49179 ## 3 : 178 3 : 184 3rd Qu.: 67091 3rd Qu.: 64006 ## 4 : 84 4 : 49 Max. : 964511 Max. :983931 ## (Other): 80 (Other): 80 ## X14 X15 X16 X17 ## Min. :-157264 Min. :-170000 Min. :-81334 Min. :-339603

## 1st Qu.: 2666 1st Qu.: 2327 1st Qu.: 1763 1st Qu.: 1256

## Median : 20088 Median : 19052 Median : 18104 Median : 17071

## Mean : 47013 Mean : 43263 Mean : 40311 Mean : 38872

## 3rd Qu.: 60165 3rd Qu.: 54506 3rd Qu.: 50190 3rd Qu.: 49198

## Max. :1664089 Max. : 891586 Max. :927171 Max. : 961664

## ## X18 X19 X20 X21 ## Min. : 0 Min. : 0 Min. : 0 Min. : 0 ## 1st Qu.: 1000 1st Qu.: 833 1st Qu.: 390 1st Qu.: 296 ## Median : 2100 Median : 2009 Median : 1800 Median : 1500 ## Mean : 5664 Mean : 5921 Mean : 5226 Mean : 4826 ## 3rd Qu.: 5006 3rd Qu.: 5000 3rd Qu.: 4505 3rd Qu.: 4013 ## Max. :873552 Max. :1684259 Max. :896040 Max. :621000 ## ## X22 X23 Y

## Min. : 0.0 Min. : 0.0 0:23364

## 1st Qu.: 252.5 1st Qu.: 117.8 1: 6636

## Median : 1500.0 Median : 1500.0

## Mean : 4799.4 Mean : 5215.5

## 3rd Qu.: 4031.5 3rd Qu.: 4000.0

## Max. :426529.0 Max. :528666.0

##

Now, we divide the data set to training and testing data sets. 70 % of the data set is training data and the rest is the testing data.

set.seed(999)

test <- sample(nrow(Default),0.3\*nrow(Default))

data.train <- Default[-test,]

data.test <- Default[test,]

nrow(data.train)

## [1] 21000

nrow(data.test)

## [1] 9000

In this stage, we define the regression model for training data set. we can remove X10 or exclude (not remove) it from training data. the result will be the same.

# We can remove X10 or exclude (not remove) it from training data. The result will be the same.

glm.1 <- glm(Y~., data= data.train[,!colnames(data.train) %in% c("X10")],family=binomial)

glm.probs <- predict(glm.1,data.test, type="response")

Now, Convert probabilities into predicted class and define the confusion matrix.

glm.pred <- rep(0, 9000)

glm.pred[glm.probs > 0.5] = 1 # Confusion Matrix table(data.test$Y,glm.pred)

## glm.pred

## 0 1

## 0 6656 336

## 1 1294 714

Now, based on the output, we define type I and II errors. the following formulas and calculations will be presented.

1. 1

0 6656 336

1 1294 714

**4.** **Limitations and areas of improvements**

The trend of historical data can be estimated by different methods and most advanced methods can predict better future values. Logistic regression models consider a linear line for estimations, which is not necessarily true. So, using other machine learning techniques can be more applicable. In real world scenarios, the trend of response variables is not a linear one. So using non-linear regression analysis is more applicable., Although, some heuristic methods can lead us to a better result. ANN can be a better algorithm for solving the problem.

**5. Conclusion and Future Studies**

In this paper, the credit card client of a bank in Taiwan was studied. The goal was to predict the probability of default as a dependent variable by considering some independent variables. A Logistic Linear Regression Model was presented to solve the problem. Also, the confusion matrix was built to calculate the type I and II errors for the results. Methods and the code were presented in the project. Although, the method works well for the data set, some other machine learning approaches can perform better. Using some famous methods like Artificial Neural Network (ANN) can help us to get the better results. As future studies, other Non-Linear Regression models and heuristic methods can be applied to compare the results and conclude the best type of methods in these problems.