GROUP 12

*Project Halfway Report*

Default of Credit Card Clients Dataset

***Members:***  
Mary Ridhima Golamari  
Bilal Khan  
Manikandan Lalitprasad  
Tapas Kumar Mahanta  
Tirth Patel

***Data Description***

The dataset mentioned above is exciting for us since most of the group members have a financial background. This dataset contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005.

***Data Mining Method***

The objective of the Credit card clients default dataset is to build a model which can accurately detect whether a customer is a potential defaulter. The output ( i.e. our dependent variable) will be a 0 ( False or Not a defaulter) or a 1( True or Yes, a defaulter). This is a categorical output and hence we need to apply Machine Learning Classification algorithms to our dataset. This is because we have a target variable, set of attributes and we have a clearly defined objective.

***Data Mining Model***

Using R we will split the dataset into training, and testing data. We will then go ahead and conduct univariate, bivariate and multi-variate tests using the variables, conduct feature selection, and decide our final set of useful variables. We will build 3 models for the dataset which are Random Forest Classification (RFC), Logistic Regression (LG) and Neural Networks. We will then check the confusion matrix, accuracy and other factors to decide the best fitted model. If we notice overfitting, we can go ahead and regularize the model.

***Data Mining Objective***

The objective is to build the most accurate model to detect potential defaulters. In the process, we are also excited to learn more about R as a tool while using it for various purposes like model building, visualization, analysis and finding insights.

Method 1

Decision tree

We decided to start with decision tree as our classifier.

Steps

1.Remove ID column from the dataset

2.Split the data into train and validation splits (70/30 ratio)

3. Use the train data to build decision tree using rpart package.

4. Surprisingly the resulting tree has only a single root node.

Accuracy

Train: 82.19%

Validation set: 81.43%

5. The issue could be strict threshold for splitting used by default (cp =0.01) decision tree function in r.

6. We tried to relax the threshold by changing the complexity factor. We start from the basic i.e. cp = 0.001. This gives us more rules(nodes).

Accuracy

Train: 82.84%

Validation set: 81.53%

7. To get the optimum value of cp, we analyzed the xerror and real error for different values of cp from cptable. The lowest value of cp (0.001) has the least training error(rel error), but it might lead to overfitting. So we tried other cp values > 0.001 with with lowest xerror.

CP nsplit rel error xerror xstd

1 0.183366077 0 1.0000000 1.0000000 0.01306422

2 0.002837808 1 0.8166339 0.8166339 0.01210408

3 0.001673579 3 0.8109583 0.8209998 0.01212936

4 0.001309758 9 0.7972059 0.8183803 0.01211421

5 0.001200611 12 0.7932766 0.8181620 0.01211294

6 0.001100000 14 0.7908754 0.8183803 0.01211421

For cp-value 0.0022

Train: 82.31%

Validation set: 81.52%

height of tree – 4

Overall

BILL\_AMT1 21.468538

BILL\_AMT2 3.812081

BILL\_AMT5 5.037836

PAY\_0 1294.830795

PAY\_2 991.442393

PAY\_3 812.617154

PAY\_4 681.792226

PAY\_5 611.356296

PAY\_6 16.573848

PAY\_AMT1 3.901029

PAY\_AMT4 4.674725

LIMIT\_BAL 0.000000

SEX 0.000000

EDUCATION 0.000000

MARRIAGE 0.000000

AGE 0.000000

BILL\_AMT3 0.000000

BILL\_AMT4 0.000000

BILL\_AMT6 0.000000

PAY\_AMT2 0.000000

PAY\_AMT3 0.000000

PAY\_AMT5 0.000000

PAY\_AMT6 0.000000

For cp-value 0.0015

Train Set: 82.61%

Validation set: 81.46%

Height of tree – 10

For cp-value 0.0011

Train Set: 82.75%

Validation set: 81.5%

Height of tree – 15

Feature importance

Overall

BILL\_AMT1 21.468538

BILL\_AMT2 3.812081

BILL\_AMT3 9.953231

BILL\_AMT4 8.186492

BILL\_AMT5 7.827385

BILL\_AMT6 3.469726

LIMIT\_BAL 17.408146

PAY\_0 1520.378206

PAY\_2 1286.106510

PAY\_3 1077.914010

PAY\_4 890.130603

PAY\_5 625.925365

PAY\_6 16.573848

PAY\_AMT1 188.857563

PAY\_AMT2 19.505498

PAY\_AMT3 2.261756

PAY\_AMT4 12.556053

PAY\_AMT5 6.583416

PAY\_AMT6 6.998854

SEX 3.336695

EDUCATION 0.000000

MARRIAGE 0.000000

AGE 0.000000

As we can see PAY\_0, PAY\_1, PAY\_2 are heavily weighted, but about 50% of the values in each of the variables are ‘0’. We can’t simply remove these rows as this will reduce the training set and the variable is too important to be omitted. This is the issue we’ll try to address in the later half by using some Oversampling or Under-sampling methods.