**Project: Detecting credit defaulters**

Link: https://github.com/LeenaDamle/CreditDefaulter

### **Introduction:**

Credit cards are important to the banks or organizations issuing them to earn money via profits.

Defaulting on credit card monthly installments causes huge losses. By observing client characteristics and their payment history, classification models are built to detect and predict default behaviors of the customers

**Problem:**

The aim of this project is to analyze the credit card of clients in Taiwan to determine whether the client will default or not in the next month. This data is obtained from the UC Irvine dataset collection:

<https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients>

To identify potential defaulters, I have explored the following Classification Algorithms using Caret package:

1. C5.0
2. NeuralNetwork
3. NaiveBayes

Data pre-processing techniques like Standardization, and k-fold cross validation are also explored to try and improve model performance.

**Data set:**

Our credit dataset consists of 30,000 observations and 23 factors along with 1 factor for classification as potential defaulter for the next month or not.

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. 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.   
 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.

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;

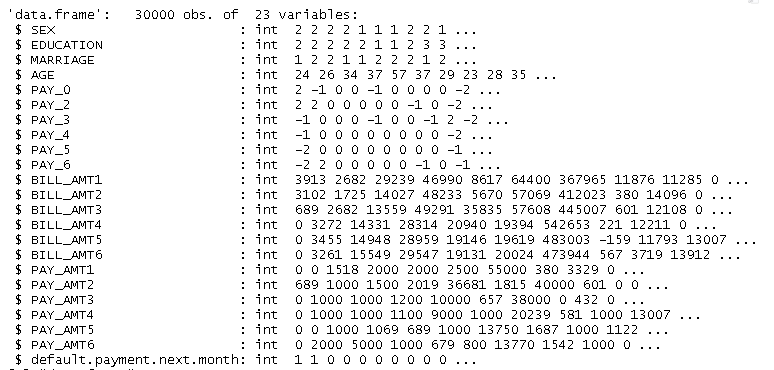
…

* X23 = amount paid in April, 2005.

**Exploring and Pre-processing the Data:**

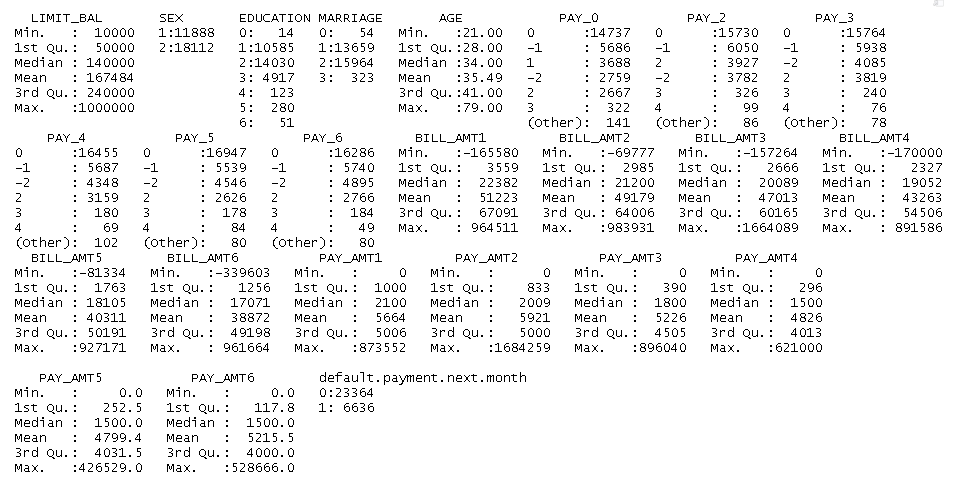
The credit dataset originally consisted of 30000 observations and 24 numeric variables.

* In the original data file, the first row contained an additional row with column names X1, X2.. which did not add business value. The data file was modified to exclude this row
* The column of ID variables is also removed to avoid overfitting of the model

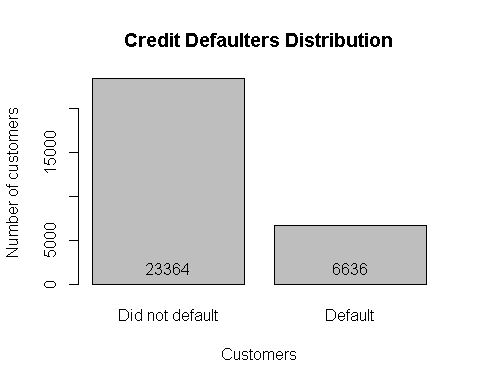


* The variables Sex, Education, Marriage, Past Monthly Repayment Status, Default Payment Next Month are in fact categorical variables. We convert them to factors

Here is a summary of the variables in our dataset:



* The data consists of 6636 customers (22.12%) who are defaulters and 23364 customers who are not defaulters



* We observe a large difference in the range of values of the numeric variables, for Example, Age ranges from 21 to 79 while PAY\_AMT2 ranges from 0 to 1684259. We standardize the data so that variables with larger values do not dominate those with smaller values.

**Training and Test Data:**

* Stratified Sampling is used to select Training and Test Data Sets.
* In stratified sampling, the data is divided into different strata or groups and samples are randomly selected proportionally from the different groups. The Class Variable default.payment.next.month is used to create 2 strata. 75% data is used as training sample and the rest is used for testing.
* createDataPartition() function from Caret package is used for this purpose.
* Distribution of class variable in Training and Test datasets is seen to verify whether it is representative of the population distribution.

**k-fold cross validation:**

In k-fold cross-validation, the original sample is randomly partitioned into k equal sized subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k − 1 subsamples are used as training data. The cross-validation process is then repeated k times, with each of the k subsamples used exactly once as the validation data. The k results can then be averaged to produce a single estimation.

[Source: Wikipedia: https://en.wikipedia.org/wiki/Cross-validation\_(statistics)#k-fold\_cross-validation]

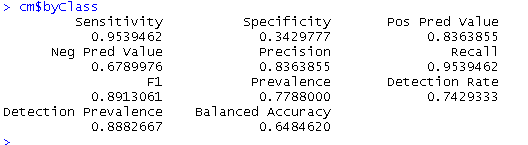
* 5- fold cross-validation is used due to processing limitations and the size of our dataset

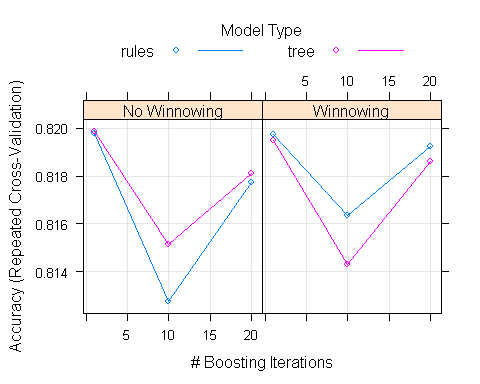
Here is a summary of the results provided by each method used:

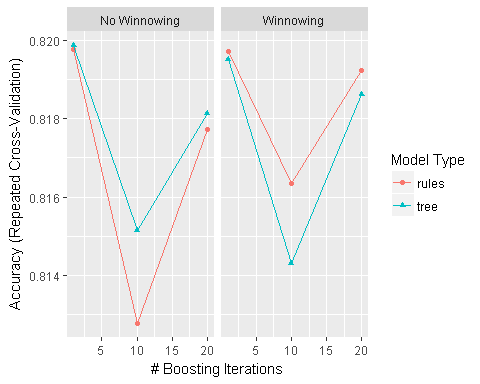
**Method1: C5.0 Decision Tree using raw data**

cm$overall

## Accuracy Kappa AccuracyLower AccuracyUpper AccuracyNull   
## 8.188000e-01 3.608522e-01 8.098926e-01 8.274589e-01 7.788000e-01   
## AccuracyPValue McnemarPValue   
## 7.521114e-18 1.302305e-109





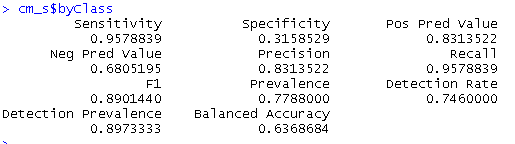


**Method 2: Decision Tree C50 using Standardized data:**

We observe that there is no improvement in accuracy:

cm\_s$overall

## Accuracy Kappa AccuracyLower AccuracyUpper AccuracyNull   
## 8.158667e-01 3.387130e-01 8.069059e-01 8.245812e-01 7.788000e-01   
## AccuracyPValue McnemarPValue   
## 1.564932e-15 3.411087e-126

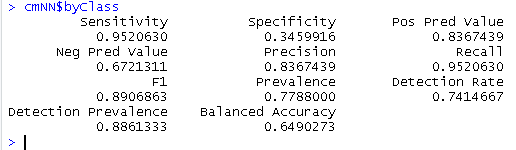


**Method 3: Neural Network classifier:**

A Neural Network is built to classify the data sample.

cmNN$overall

## Accuracy Kappa AccuracyLower AccuracyUpper AccuracyNull   
## 8.180000e-01 3.607127e-01 8.090780e-01 8.266741e-01 7.788000e-01   
## AccuracyPValue McnemarPValue   
## 3.364524e-17 5.370821e-105



**Method 4: Naïve Bayes classifier:**

This algorithm is based on Bayes Theorem:

Given training data X, posteriori probability of a hypothesis H,

P(H|X), follows the Bayes theorem

Informally, this can be written as

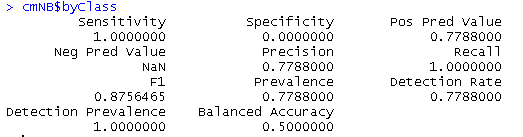
posteriori = likelihood x prior/evidence

Predicts X belongs to C2 iff the probability P(Ci|X) is the highest

among all the P(Ck|X) for all the k classes

cmNB$overall

## Accuracy Kappa AccuracyLower AccuracyUpper AccuracyNull   
## 0.7788000 0.0000000 0.7692325 0.7881503 0.7788000   
## AccuracyPValue McnemarPValue   
## 0.5065803 0.0000000



**Analysis of Results:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Sensitivity | Precision | Specificity | Recall |
| DecisionTree c5.0 | 0.8188 | 0.9539462 | 0.8363855 | 0.3429777 | 0.9539462 |
| NaiveBayes | 0.7788 | 1 | 0.7788 | 0 | 1.0000000 |
| NeuralNetwork | 0.8180 | 0.952063 | 0.8367439 | 0.345991 | 0.9520630 |
| DecisionTree c5.0 standardized data | 0. 8158667 | 0.9578839 | 0.8313522 | 0.3158529 | 0.9578839 |

The bank/ organization wishes to find those customers who might default on their next monthly payment.

Since the models constructed consider the positive class as ‘zero’- will not default) we are interested in the negative class- ‘one’ (Those who will default).

It is important in this case to minimize the number of False Negatives- Those who will default but are not identified as will default.

We can do this by increasing the number of True Negatives. (Customers correctly identified as interested in signing up for a term deposit.)

With respect to the models built,

Recall is what % of positive tuples (Not interested) did the classifier label as positive (Not interested)?

Precision is: What percent of what % of tuples that the classifier labelled as positive (Not interested) are actually positive (Not interested). It is important to identify this since we do not want to mis-classify someone as not interested, it would result in loss of business.

Hence, we are interested in a model with high Precision.

The Neural Network model gives the highest precision.

Recommendation:

The Taiwan based company/ bank can employ the Neural Network Model to predict which customer is going to default on the next monthly payment and take appropriate action like limiting their credit.

Future improvements:

I am now working on an ensemble model built using the above three algorithms (viz. Decision Tree C5.0, Naïve Bayes and Neural Network)