**Fintech Hackathon Project**



**Predicting default on Credit Card Applications for Taiwanese Credit Card Customers**

Vinayak khamkar, Amrinder Singh Bedi,Shravani,Sagar Rawale,Nikhil Gupta, Raviraj Pandit,Prasad Patil

Team 3

Held on 21st and 22nd April, 2018

**Abstract**

The purpose of this project was to predict whether customer is going to default credit card payment of next month in Taiwan. By comparing different machine learning algorithms for classification; we aim to deter-mine the best method. Data comprised of 72% non-defaulters and 28% defaulters. Proportion indicates there is not much of an imbalance in data,. Objective of using various machine learning algorithms is to predict the best possible cost-effective outcome. It was concluded that original data with Stacking Ensemble algorithm is the best in terms of performance parameters.

**1** **Background**

Problem statement and the data set originated from the real events occurred in Taiwan. While we were doing our research we found the real story behind this project which goes like this – In the 1990s, the Taiwanese government formed new banks. These new banks lent large sums of money to real estate companies with the goal of expanding their businesses and increasing profits. However, after a couple of years of expansion, the real estate market became saturated and profits from the sector stopped growing.

The new banks turned to other new business – credit cards. In expanding this area of business, banks lavished money on commercials encouraging people to apply for credit cards to consume, apparently without consequences. These banks lowered the requirements for credit card approvals to get more customers. In time, young people became target customers. Although, many young people did not have enough income, banks still issued credits cards to them.

In Taiwan, in February 2006, debt from credit cards reached $268 billion USD. More than half a million people were not able to repay their loans. They became “credit card slaves”, a term coined in Taiwan to refer to people who could only pay the minimum balance on their credit card debt every month. This issue resulted in significant societal problems. Some debtors and their families committed suicide because of the debt, some became homeless due to repossession of their homes, and others could not afford to pay their children’s tuition. Some credit card slaves sold illegal drugs to repay the banks. Sometimes violent and threatening collection practices of certain banks added pressure to lenders, particularly those in lower income groups.

The Taiwanese government was forced to solve these problems to save the financial system and prevent further societal problems. Based on the Taiwanese Department of Health report, 2,172 people committed suicide in 1997, 4,406 in 2006, and 4,128 in 2008. The suicide rate in Taiwan was the second highest in the world at that time. The suicide rate increased 22.9% compared to the rate in 2005, and the main reason is unemployment and credit card debt.

Using our machine learning knowledge we as a team tried to help Taiwan’s banks and Government to resolve the dreadful societal problems they were facing. If there had been our algorithm in place,

Banks would have taken necessary measures and many people would have not committed suicides is what we claim.

**2** **Introduction**

The dataset includes demographic information like age, gender, marital status etc. and the credit history showing billing and pay-ment records to predict performance of an individual’s risk when it comes to a potential defaulter.

Many advanced machine learning methods can be used for classification of status of an individual whether he/she is going to default or not in next month. We used several different machine learning algorithms and concluded Stacking Ensemble is the best when it comes to predictive accuracy across multiple metrics. These algorithms can detect a client who might default on next payment with a high accuracy. But it might categorize a lot of good customers as defaulters when the default detection is so specific; it might list lot of potential good customers to fall in category of defaulters. Just to get a high predictability of defaulters, one cannot afford to lose such good customers

As it might very well prove detrimental for financial institutions issuing credit cards. A good prediction will potentially have a mix of risky and non-risky clients with a better accuracy in predicting a defaulter in a cost-effective manner.

The five machine learning algorithms used in this project are as follows:

1. Logistics Regression
2. Decision Trees
3. Random Forest.
4. Stacking Ensemble
5. XgBoost
6. **Data Attributes at a Glance**

This project is based on binary classification dataset provided on UCI Machine Learning Repository. The data contains 30,000 clients with 23 attributes with no missing information. Attributes X1 through X23 are independent variables; and class is the dependent variable with binary classes (0,1); 0 – Not defaulted, 1 – Defaulted on credit card pay-ment.

All of the 23 variables from the dataset are described below and have been utilized in this research

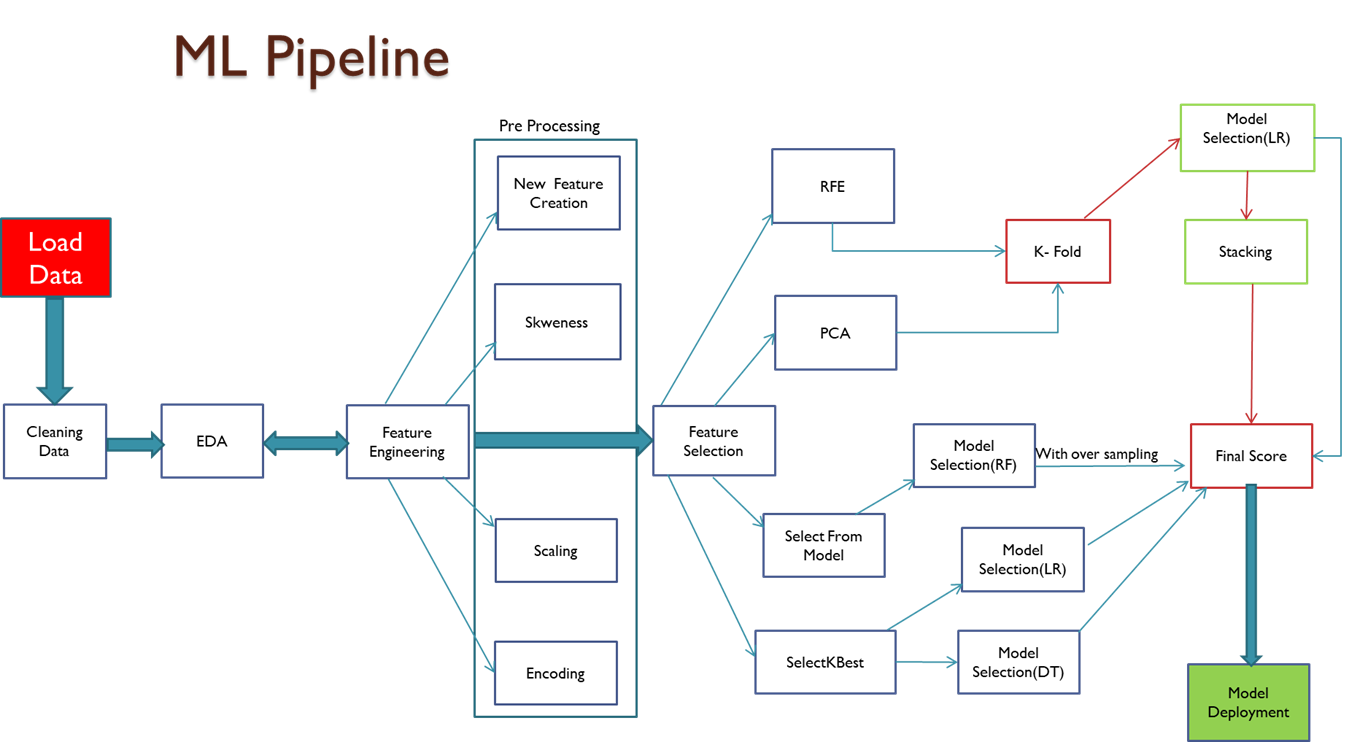
* X1: Amount of the given credit (NT dollar): it includes both the

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.

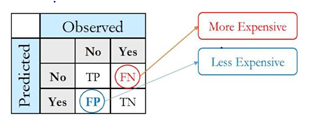


1. **Performance Parameter Selection**

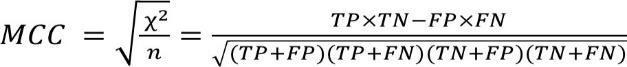
It is crucial to select the appropriate measure in order to gauge predictions. From the problem statement, we figured out that positive class here is credit card payment defaulters and negative class credit card bill payers. So confusion matrix explains:

* **Accuracy** = Overall how often model predicts correctly defaulter and non-defaulters
* **Recall** = Out of total actual defaulter how many model has picked correctly
* **Precision** = Out of total predicted defaulters how many actually were defaulters
* **False Positive** = Person who is predicted defaulter actually end up paying credit card bills
* **False Negative** = Person who is predicted not a defaulter will end up defaulting credit card bills

From bank’s perspective, business would not mind FP’s appearing predictions but it will be huge cost for business if there are numerous FN’s in predictions. Hence recall seems to be a good measure for performance check. Along with, recall one just cannot ignore precision which is an equally important measure. In order to find balanced measure we selected **f1\_score,** which is trade-off between recall and precision.



However, after digging deeper in sklearn documentation, we found there is another metric called **Matthew’s Correlation Coefficient** which is a trade –off between all.The Matthews correlation coefficient is used in machine learning as a measure of the quality of binary (two-class) classifications. It takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes. The MCC is in essence a correlation coefficient value between -1 and +1. A coefficient of +1 represents a perfect prediction, 0 an average random prediction and -1 an inverse prediction. The statistic is also known as the phi coefficient**.**

**We followed below Machine Learning Pipeline.**

Apart from above discussed measures we thought of introducing cost factor to our predictions by implying a higher cost to defaulters classified not correctly thereby penalizing false predictions of defaulters. With a good estimation on prediction of defaulters while maintaining a good number of consumers and reduce the overall cost.



**Thought Process:**

Assume an arbitrary number say 10, multiply it with number of false negatives that you get from the model predictions and add number of false positives to that number. The resultant output of this calculation is overall cost for the business. Data scientist’s sole purpose is to minimize this cost as much as possible.

The only flaw to this idea was which arbitrary number to select and justification for selected figure .After consulting with mentors, we realized that that is a business call and one should not assign equal weightage to all false negatives. But still we can use it in order to compare models on performances.

1. **Result Summary**

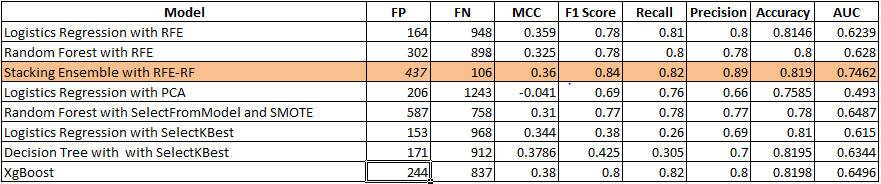
After feature engineering where we have done label encoding and scaling, we have experimented several feature selection techniques during hackathon with sole aim to achieve better result as shown in ML Pipeline. Comparing several algorithms, we have achieved better result on Stacking Ensemble algorithm.

**Final Model Selection and Evaluation –**

**I.** Initially, we applied recursive feature elimination technique for feature selection and tested it on three models (1.Logistics regression –baseline model, 2. Decision Trees – to generalize non-linearity if any in data, 3. Random Forest- bagging ensemble of Decision trees since multiple weak learners combined makes strong learner) for 10 important features.

**II.** Secondly, we checked our three models using KFold validation, and Logistics regression and Random forest found to be performed well comparing cross validation mean scores with just 0.01units of difference .

**III.** We already had important features from step **I**, we applied train test split based on those and ran Stacking Ensemble for both Logistics regression with its RFE features and Random Forest with its RFE features. Both models showed more or less same results across all performance parameters which in the end outperformed all other algorithms that we tried using other feature selection techniques. As shown in figure below:



**5** **Notable Approaches**

**1. Month-Wise Models Approach**

The approach is basically to fit a model on each month's data instead of clubbing all the months together in single model. The logicbehind this is to predict for each month separately and then combining the predictions for predicting for the final predictions on basis of the weights given to different months. The following steps are taken:

- Load data

- Split the data into 90:10

- Split the 90% data again into 80:20 (Train: Test)

- split the dataset into 6 parts (for every month) where all the categorical variables remain same for all the parts, the only change is in terms of Bill\_Amt, Pay\_Amt and Pay\_status variables, which are different for different months

- Consider Pay\_status variables as your dependent variable for every dataset. (Logic: Pay\_status is nothing but the payment status for that particular month. So if someone has made advance payment, or partial payment or paid a minimum amount he/she is not a default for that particular month. So I considered all such instances as 0. Same way if someone has delayed the payment for 1 or more months, that entry has defaulted for that particular month. So such instances can be considered as 1)

- Converting pay\_status values into 0 and 1 on the basis of above logic

- RFE and K-fold validation for each of the model.

- Remove the least important features from all the models

- Fit DT, Logistic Regression and RF. Test which one is best (DT was giving best result hence I continued with DT)

- Now I get 6 predictions for my 6 models

- Fit Logistic Regression between all these 6 predicted columns and my actual target variable which is Default\_status (The idea behind this is giving weights to each prediction column on basis of its importance)

- Now fit the 6 models on my remaining 10% data

And test the predicted columns against Default status from this 10% data

- It gave me F1 score as 0.72

**More to do:**

- I tried oversampling with SMOTE but it didn't work as it converted my pay\_status variable into continuous variable (which is my dependent variable in every month's dataset). So I dropped the idea. But I can try random sampling.

- I can also try fitting some other models between my predictions from six models and default status.

- Need to explore more ML techniques that can be tried for this approach.

**Ms.Shravani Ghantnekar**

**2. Deriving Features**

The concept that we wanted to explore was: if you would want to decide someone as defaulter, then we can take a look at how many times the customer has defaulted, how many times, he has made payments on time or minimum payment and check if customer’s overall payment dues are within the credit limit assigned to him or not, check his total/average payments made, total/average bill amount that customer is paying till date. We wanted to impute these summary features into the dataset and check the relations of these summary features with the defaulter status.

- Looking at Pay status columns, we have values ranging from -2 to 8. Based on these derive advance pay, minimum pay and delayed pay columns to represent the number of times the payments were made into these categories

- Total amount owed till date and whether the amount owed is greater than the credit limit.

- Total amount paid and average amount paid based on the 6 months bill amount paid

- Total bill amount and average bill amount being charged to the customer

- Based on these columns check the relationship, perform feature engineering & feature selection and build classification models.

- Yet to complete this approach. I tried initial derived feature imputation but next steps are pending.

**Mr.Vinayak Khamkar**

**3. Iterative Predictions**

Analyzing the data we see that there is a delayed relation between bill amount and pay amount and in the problem statement we were asked to find out if someone will default their payment for the 7th month or not( We have 6 months of bill and payment data).Now while working on the data set we came across some intriguing questions like if we get the predictions for 7th month can we create a model that will run iteratively and give us the next month’s predictions using the given data and predicted values? One way to begin for the same was that say if we get the predicted value as 0 for 7th month which means the person is not going to default for the next month. So we can infer that either he will pay fully for 7th month or minimum amount. Assuming that he pays fully we can infer payment\_amount7 columns value as bill\_amount6 columns value.

Now we cannot get exact amount for bill\_amount7 but we can infer the same using an assumption that we take the mean all of his past bill amounts. So in a way from six months of data and running an iteration of model which gives us Default/no default status for 7th payment we can infer the bill amount7 and payment amount7 and then run the cycle again for making predictions for the 8th month and so on. The same way if he defaults (output 1) we can infer the payment amount and bill amount and only the payment amount will change to 0 and we can use the same above process to infer bill amount for next month.

We wanted to create an iterative model for the same and we also did the same and developed up to inferring columns but the only bottleneck was that we were struck since we cannot test our results for our consecutive month’s predictions due to absence of values for comparison. The other thing that makes this approach feasible is because the other features like age, sex, marriage, education do not change drastically. If we were/are able to build such a model we can increase the age by 1 every 12 months and sex of a person will never change except in rarest of cases.

**Other Questions:**

Some other questions that came out while working with the data set were like presence of columns like job status, income , credit card consumption use( means whether a person is using amount for assets like flats, cars or education, medical etc.) would be very beneficial in building a more robust model. At the same time it was observed in the data set that a person can consume his limit balance amount every month and end up paying just a minimum balance or does not even pay. Sometimes amount greater than limit balance was also

consumed in a particular month. So instead of finding out just whether a person will default his payment or not we can also predict that if a person is consuming every month and paying very less whether he/she should be given same privileges for the following months or not( Although the same can be done using simple mathematics also)

From a business perspective the bank should stop giving him more money until he repays the money consumed till now. These were some additional queries that came to our mind while working with the data set.

**Mr.Amrinder Singh Bedi**

**Articles and References:**

**Background and Story:** [**https://sevenpillarsinstitute.org/case-studies/taiwans-credit-card-crisis/**](https://sevenpillarsinstitute.org/case-studies/taiwans-credit-card-crisis/)

**DataSet and Citation:** [**https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients**](https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients)

**Referred to paper** “The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications” given in the above URL

**Kaggle Site for some of the approaches**