**Credit Card Default Prediction**

**Technical Document**

**Mohammad Jibran**

**Siddhi Thakur**

| Data Science Trainees |

**ALMABETTER**

**Abstract**

1. **Problem Statement:**

This project is aimed at predicting the case of customers default payments in Taiwan. 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. We can use the [K-S chart](https://www.listendata.com/2019/07/KS-Statistics-Python.html) to evaluate which customers will default on their credit card payments.

1. **Introduction:**

For the banking and credit card industries, credit risk has typically been the biggest risk and the one that requires the most capital. Statistics and business reports from the sector can support this. For instance, "The Reserve Bank of India bases its assessment of credit card delinquencies on the proportion of accounts that are at least 90 days overdue. The key to increasing revenue and decreasing loss for the banking and credit card industries is therefore analysing, recognising, and controlling default risk.

Despite the fact that the banking industry has adopted machine learning and big data, the existing applications are primarily focused on predicting credit scores. The drawback of extensively relying on credit scores is that banks may lose out on valuable clients who come from generally underbanked nations and have no credit history or recent immigrants who have the ability to pay but no credit history.

The purpose of this project is to conduct quantitative analysis on credit card default risk by using interpretable machine learning models with accessible customer data, instead of credit score or credit history, with the goal of assisting and speeding up the human decision-making process.

1. **Understanding the data**

This dataset contains information on default payments, demographic factors, credit limit, history of payments, and bill statements of credit card clients in Taiwan from April 2005 to September 2005. It includes 30,000 rows and 25 columns, and there is no credit score or credit history information.

**Data Description** -

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:

There are 25 variables in this dataset:

• **ID**: 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\_1:** 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)

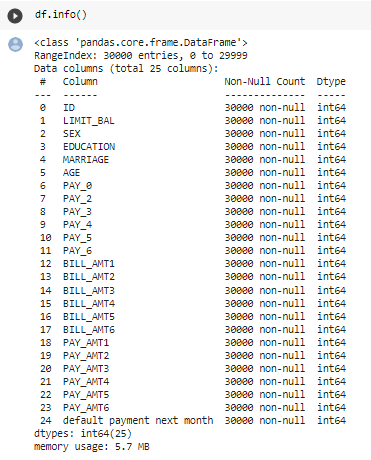
**• PAY\_AMT4:** Amount of previous payment in June, 2005 (NT dollar)

**• PAY\_AMT5:** Amount of previous payment in May, 2005 (NT dollar)

**• PAY\_AMT6:** Amount of previous payment in April, 2005 (NT dollar)

**• DEF\_PAY\_NMO:** Default payment (1=yes, 0=no)

Overall, the dataset is very clean, but there are several undocumented column values. As a result, most of the data wrangling effort was spent on searching information and interpreting the columns.



1. **Data Wrangling**

Data wrangling is a term often used to describe during data analysis. It helps to transform data from one format into another. The aim is to make data more accessible. These include things like data collection, exploratory analysis, data cleansing, creating data structures, and storage.

* **Importing important libraries**

Our major goal in this step was to import all of the necessary libraries to aid us in exploring the issue statement and doing EDA to draw conclusions based on the data collection.

**Libraries we used:**

1. **NumPy**

NumPy is the python library used for programming languages, adding support for large, multidimensional arrays and matrices with large collections.

1. **Pandas**

It is a software library written for python programming, flexible, and expressive data structures designed to make working with relational or labelled data both easy and intuitive. Pandas allow us to access many of Matplotlibs and NumPy’s methods with fewer codes.

1. **Matplotlib**

It is a pyplot collection of functions that make matplotlib work like MATLAB. e.g., creates a figure, lines a plotting area.

1. **Seaborn**

It is an open-source python library built on top of matplotlib. It is used for data visualization and EDA. Seaborn works easily with Data frames and pandas’ libraries. The graphs can also be customized.

1. **Scikit-learn** **(Sklearn)**

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modelling including classification, regression, and clustering and dimensionality reduction via a consistence interface in Python.

**Exploratory Data Analysis**

The purpose of exploratory data analysis is to identify the variables that impact payment default likelihood and the correlations between them. We use graphical and statistical data exploratory analysis tools to check every categorical variable. Each starts with a visualization and is followed by a statistical test to verify the findings.

**The main findings from exploratory analysis are as following:**

● Males have more delayed payment than females in this dataset. Keep in mind that this

finding only applies to this dataset, it does not imply this is true for other datasets.

● Customers with higher education have less default payments and higher credit limits.

● Customers aged between 30-50 have the lowest delayed payment rate, while younger

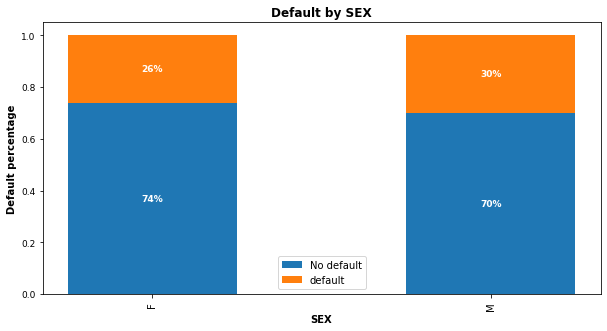
groups (20-30) and older groups (50-70) all have higher delayed payment rates.

However, the delayed rate drops slightly again in customers older than 70.

● There appears to be no correlation between default payment and marital status.

**Males have more delayed payments than females in this dataset.**

Since there are more females than males in the dataset, we use percentage of default within each sex group. Figure shows 30% males have default payment while only 26% females have default payment. The difference is not significant.

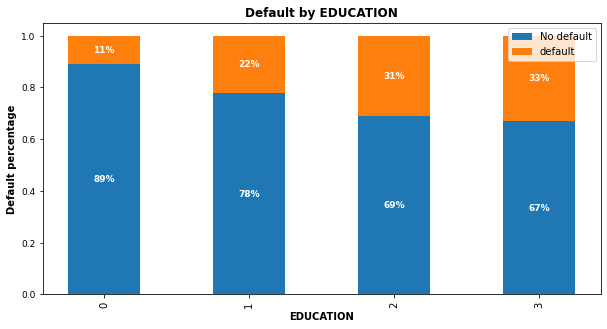


**Customers with higher education have less delayed payment.**

Figure indicates customers with lower education levels default more. Customers with high

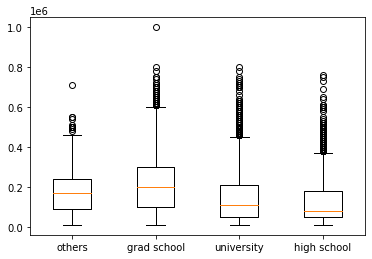
school and university educational level have higher default percentages than customers with

grad school education. Notice there is an education group “others” which appears to have the least default payment, but this group only has 468 (or 1.56%) customers, and we don’t know what consists of this group.



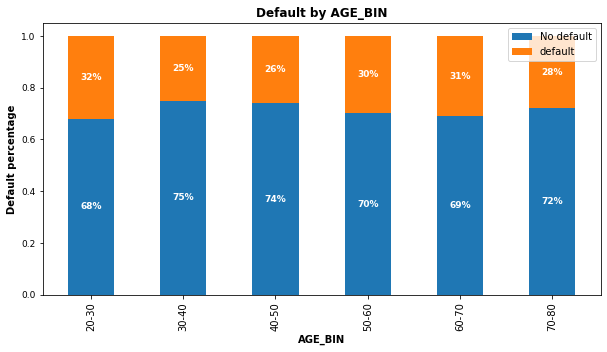
**Customers with a high education level get higher credit limits**

From the boxplot in figure, it is obvious that customers with grad school education have the highest 25% percentile, highest median, highest 75th percentile and highest maximum numbers, which suggests customers with higher education levels do get higher credit limits.



**Middle-aged customers have the lowest default rate**

The bar chart in figure 4 shows the default probability increases for customers younger than 30 and older than 70. Customers aged between 30 and 50 have the lowest delayed payment rate, while younger groups (20-30) and older groups (50-70) all have higher delayed payment rates. This aligns with social reality that customers aged 30-50 typically have the strongest earning power. We also notice the delayed rate drops slightly again in customers older than 70. This is understandable because elder customers' consumption tends to decrease.



**Modeling**

**Modeling Preparation**

Since there are labelled data and the expected outcome is the probability of customer default, we define this as supervised machine learning and it is a binary classification problem. For better model performance, we first take a few pre-processing steps to prepare for modeling.

● **Feature Selection:** There are 25 columns in this dataset and the target variable is the

column ‘DEF\_PAY\_NMO’ (means “default next month”). We drop the column ‘ID’ and

‘DEF\_PAY\_NMO’, save the rest 23 as predictor features. Those predictor variables

include categorical variables such as sex, age, education level and marital status, along

with numerical variables, such as payment status, credit limit, bill amount, etc. With this

dataset, we don’t need to do PCA or dimensionality reduction.

● **Check Class Imbalance:** It is common sense that most customers do not default. This

dataset is likely to be dominated by 0s (non-default) with rare 1s (default). Imbalanced

dataset will mislead machine learning algorithms and affect their performances.

‘DEF\_PAY\_NMO’ variable shows 22% of customers have default and 78% have no

default. The class ratio is roughly 1:4. We consider this dataset is imbalanced and will

use SMOTE oversampling technique after train-test data split to balance the data.

● **Transform Categorical Column**: In the dataset, ‘AGE’ column has continuous values

which is the individual customer’s age. In the business context, we are more concerned

about the age groups than the specific age, so we bin the ‘AGE’ column to 6 bins -

21~29,30~39,40~49,50~59,60~69, and 70~79. Finally, we convert this column into

numerical data type because sklearn does not accept categorical data type.

● **Data Rescaling:** The feature variables’ value varies vastly. For example, the credit limit

value is up to 100,000 NTD and the payment status only ranges from 0 to 8. In order to

make all variables have similar ranges, so the Logistic Regression model can perform

well in regularization, we rescale the training data. In this process, we make sure to only

fit training data (X\_train) and then transform training data and test data (X\_train, X\_test),

instead of fit and transform the entire X (consists of X\_train and X\_test).

● **Split Training and Test Data**: For each model, we use the same ratio for training and test data split (70% for training, 30% for test) to ensure consistency. After splitting the data,

we set the test data aside and leave it for the very end, which is the final testing after

hyperparameter tuning.

**Predictive Modeling**

This analysis uses 3 classification models - Logistic Regression, Random Forest and XGBoost. Since Random Forest and XGBoost are tree based on algorithms, rescaling is only performed on Logistic Regression, not on these 2 models.

For each model, we first try the model’s default parameters, train each model without SMOTE and with SMOTE samplings. Then tune each model’s hyperparameters to find the optimal performance.

As mentioned earlier, this dataset has imbalanced classes, therefore we use precision and recall, instead of accuracy as the performance metrics.

● **SMOTE Oversampling:** In the initial model fitting, we start by using all models’ default parameters. To compensate for the rare classes in the imbalance dataset, we use

SMOTE (Synthetic Minority Over-Sampling Technique) method to over sample the

minority class and ensure the sampling is not biased. What this technique does under

the hood is simply duplicating examples from the minority class in the training dataset

prior to fitting a mode. After SMOTE sampling, the dataset has equal size of 0s and 1s.

In order to verify if SMOTE improves models’ performance, all 3 models are trained with

SMOTE and without SMOTE. Below table shows the ROC\_ AUC scores on training data improved significantly with all models after over sampling with SMOTE. This proves

SMOTE is an effective method in sampling imbalanced dataset.

|  |  |  |
| --- | --- | --- |
| **Models** | **AUC Without SMOTE** | **AUC With SMOTE** |
| **Logistic Regression** | **0.725** | **0.797** |
| **Random Forest** | **0.767** | **0.920** |
| **XGBoost** | **0.781** | **0.860** |

● **Hyperparameters Tuning**: We utilize Scikit-Learn library’s built-in functions such as

cross-validation, randomized search and grid search to make this process easier. In

Logistic Regression, the only hyperparameter C penalizes a large number of features,

reduces model complexity and prevents overfitting. We use randomized search to find

the best C because C has a large search space and randomized search saves computing

time. With Random Forest, there are many hyperparameters available for tuning, but we

use most of the default settings in sklearn and only focus on a few. After creating a

parameter grid, we use grid search to find the best parameters combinations. The third

model XGBoost is known for its good performance on low-medium sized structured

tabular data, but the downside is there are quite some hyperparameters to tune. We

initially try grid search but this turns out to be not feasible because it requires substantial

computational resources, then we switch to randomized search and find a suitable

hyperparameters combination.

● **Performance Metrics**: Since this is a classification problem with imbalanced classes, accuracy is not the best metric because the data is dominated by non-default class, thus

precision and recall are a better choice. In the credit card default risk business context,

detecting as many defaults as possible is our ultimate goal because misclassifying a

default as non-default is costly; therefore, a high recall score is the best metric. However,

there is a known trade-off between precision and recall. We can raise recall to arbitrarily

high, but the precision will decrease. We use below metrics to measure model performances.

a. Confusion matrix

b. ROC\_AUC curve

c. Precision\_recall curve

● **Feature Importance**: By plotting the feature importance on tuned Random Forest model, it is clear that ‘PAY\_1’,’PAY\_2’ (the most recent 2 months’ payment status), along with credit limit (LIMIT\_BAL) are the most important predictors. Since we don’t have customer income data, generally speaking, higher credit limits are associated with lower default risk.

**Model Comparisons**

**Compare to Sklearn Dummy Classifier:**  In a real business context, we should have a

benchmark to measure a predictive model’s performance. For example, we could

compare model performance to existing ways of making the same classification. Since

we don't have any benchmark in this project, we compare the models to Scikit-Learn’s

dummy classifier. As shown in table 1, all 3 models have better classification performance than the dummy model, which suggests our modeling has some merit.

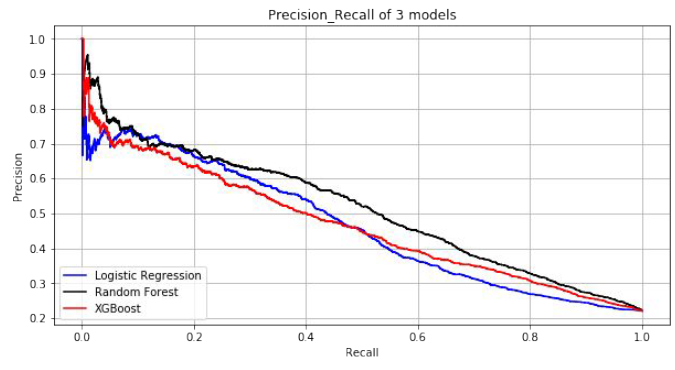
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Models** | **Precision** | **Recall** | **F-1 Score** | **Conclusion** |
| **Dummy Model** | **0.219** | **0.439** | **0.303** | **Benchmark** |
| **Logistic Regression** | **0.379** | **0.561** | **0.453** | **Best recall** |
| **Random Forest** | **0.523** | **0.505** | **0.514** | **Best F1** |
| **XGBoost** | **0.441** | **0.497** | **0.467** |  |

● **Compare within the 3 models:** Logistic Regression has the highest recall but also the

lowest precision. Random Forest outperforms Logistic Regression and XGBoost if

measured on their F1 scores, which is the balance between precision and recall.

XGBoost has a decent performance but it takes the most time to tune the model.



**Conclusions**

Based on the exploratory data analysis, we discover that human characteristics are not the

most important predictors of default, the payment status of the most 2 months and credit limit

are. From the modeling, we are able to classify default risk with accessible customer data and

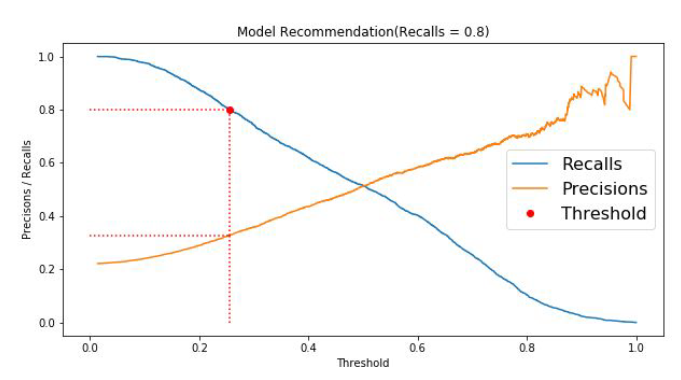
find a decent model.

● **Suggestions:** With every classification model, there is a general trade-off between precision and recall. A model’s recall can be adjusted to arbitrarily high at the cost of lower precision.

In these 3 models, Logistic Regression model has the highest recall but the lowest precision, if the firm expects high recall, then this model is the best candidate. If the balance of recall and precision is the most important metric, then Random Forest is the ideal model. Since Random Forest has slightly lower recall but much higher precision than Logistic Regression, we recommend the Random Forest model.

Below is our suggested recall plot. Note the threshold can be adjusted to reach higher

Recall.



● **Limitations:** The dataset only includes 30,000 records of non-American consumers,

there might be differences between US consumers and non-US consumers. Even with

the best model Random Forest, we can only detect 51.5% of default customers, and

among those that are being flagged as default, only 51.2% of them indeed have default.

Therefore, this model can only be served as an aid in decision making instead of

replacing human decision. Lastly, we suggest the model output probabilities rather than

predictions, so that we can achieve higher accuracy and allow more control for human

managers to quantify default risk.