Northwestern MSDS-498 Artificial

Model #101: Credit Card Default Model

Model Development Guide

Andrew Stevens

September 1, 2022

## Introduction

Credit risk modeling has advanced in the last several decades from local or collateral only based decisions to demographic and data-based industry. The features available for developing models has expanded while more data becomes available both at the individual and population levels. When these data are properly collected, cleaned and engineered to produce accurate predictions a bank can have an advantage and lend money with less risk producing hire profits.

## The Data

University of California Irvine hosts a Machine Learning Repository (Dua and Graff 2019) which includes the default of credit card clients in Taiwan prepared to compare the predictive abilities of selected data mining methods (Dua and Graff 2019). The response variable of the dataset is a binary indicator for whether a customer defaulted on their credit card debt. Delinquency is defined as missing a single payment due date, while default is not making a specific number of consecutive payments (Cagan 2020). Entering default involves collections actions and likely losses for the creditor, so a company would seek customers unlikely to default.

The predictor variables included in the dataset can be divided into two categories about the customer: demographic attributes and billing/payment history. The demographic attributes are comprised of SEX, EDUCATION, MARRIAGE, and AGE. The billing/payment history variables comprise six months of history including repayment status, billing amount, and payment amount.

Before the data can be engineered into features consumable by different modeling methods, each datatype and feature class must be reviewed for sufficient and consistent data quality. The dataset was first checked for empty values, and zero nullity was reported across all fields. The dataset was focused to only explanatory and target variables (ignoring the indices and those generated for splitting train/test/validate).

The data dictionary provides definitions and bounds for each variable and can be used to tell whether invalid values exist and must be cleaned. Appendix A provides the complete dictionary with each field explicitly defined.

|  |  |  |  |
| --- | --- | --- | --- |
| Table 1: Data Dictionary - Abridged | | | |
| Fields | | Variable | Valid  Values |
| X1 | LIMIT\_BAL | > 0 | Scalar positive |
| X2 | SEX | (1,2) | Binary |
| X3 | EDUCATION | (1:4) | Integer, categorical |
| X4 | MARRIAGE | (1:3) | Integer, categorical |
| X5 | AGE | Int, >0,<120 | Discrete, potentially ordinal integer |
| X6-X11 | PAY\_# | (-1,1:9) | Discrete ordinal integer |
| X12-X17 | BILL\_AMT# | (1:3) | Scalar pos/neg |
| X18-X23 | PAY\_AMT# | (1:3) | Scalar positive |
| Z | DEFAULT | (0,1) | Binary |
|  | |  |  |

In order to understand the data as received and determine what cleaning and engineering steps are necessary, a high-level quality report assists in reviewing each variable’s requirements & alignment to them. This was executed following the guidelines from Dempsey (2015).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Table 2: Data Quality Overview - Raw | | | | | | | |
|  |  | |  |  |  |  |  |
| Column Name | | Data Type | Present Values | Missing Values | Unique Values | Minimum Value | Maximum Value |
| ID | int64 | | 30000 | 0 | 30000 | 1 | 30000 |
| LIMIT\_BAL | int64 | | 30000 | 0 | 81 | 10000 | 1000000 |
| SEX | int64 | | 30000 | 0 | 2 | 1 | 2 |
| EDUCATION | int64 | | 30000 | 0 | 7 | 0 | 6 |
| MARRIAGE | int64 | | 30000 | 0 | 4 | 0 | 3 |
| AGE | int64 | | 30000 | 0 | 56 | 21 | 79 |
| PAY\_0 | int64 | | 30000 | 0 | 11 | -2 | 8 |
| PAY\_2 | int64 | | 30000 | 0 | 11 | -2 | 8 |
| PAY\_3 | int64 | | 30000 | 0 | 11 | -2 | 8 |
| PAY\_4 | int64 | | 30000 | 0 | 11 | -2 | 8 |
| PAY\_5 | int64 | | 30000 | 0 | 10 | -2 | 8 |
| PAY\_6 | int64 | | 30000 | 0 | 10 | -2 | 8 |
| BILL\_AMT1 | int64 | | 30000 | 0 | 22723 | -165580 | 964511 |
| BILL\_AMT2 | int64 | | 30000 | 0 | 22346 | -69777 | 983931 |
| BILL\_AMT3 | int64 | | 30000 | 0 | 22026 | -157264 | 1664089 |
| BILL\_AMT4 | int64 | | 30000 | 0 | 21548 | -170000 | 891586 |
| BILL\_AMT5 | int64 | | 30000 | 0 | 21010 | -81334 | 927171 |
| BILL\_AMT6 | int64 | | 30000 | 0 | 20604 | -339603 | 961664 |
| PAY\_AMT1 | int64 | | 30000 | 0 | 7943 | 0 | 873552 |
| PAY\_AMT2 | int64 | | 30000 | 0 | 7899 | 0 | 1684259 |
| PAY\_AMT3 | int64 | | 30000 | 0 | 7518 | 0 | 896040 |
| PAY\_AMT4 | int64 | | 30000 | 0 | 6937 | 0 | 621000 |
| PAY\_AMT5 | int64 | | 30000 | 0 | 6897 | 0 | 426529 |
| PAY\_AMT6 | int64 | | 30000 | 0 | 6939 | 0 | 528666 |
| DEFAULT | int64 | | 30000 | 0 | 2 | 0 | 1 |
|  | | |  |  |  |  |  |
|  | | |  |  |  |  |  |

The data must be divided for the training and testing models in prediction of the target variable. Table 3 provides the counts of observations within each of the groups

|  |  |  |  |
| --- | --- | --- | --- |
| **Table 3: Data Modeling Splits** | | | |
| Group | Count | | |
| Train | | 15180 | | |
| Test | | 7323 | | |
| Validate | | 7497 | | |
|  | | |  |

## Feature Engineering

In the practice of credit risk modeling, features are usually engineered by aggregating customer transactional data to determine behavioral patterns (Bahnsen et al 2016). We will also consider and test approaches for binning and combining demographic attributes of the customer, dependent on each specific model’s needs.

The AGE attribute is received as integers indicating years of age for each customer. Because age is a discrete variable with high cardinality, discretization can bring it closer to a knowledge-level representation (Peng et al 2009) and is essential for models such as trees/forests. Age has been initially separated by decade, and testing will be performed on more evenly distributed bins or perhaps other approaches.

|  |  |
| --- | --- |
| **Table 4: Resulting distribution of Age Binning** | |
|  | |
| Age\_Group | Freq |
|  | |
| 1-10 | 0 |
| 11-20 | 0 |
| 21-30 | 11,013 |
| 31-40 | 10,713 |
| 41-50 | 6,005 |
| 51-60 | 1,997 |
| 61-70 | 257 |
| 71-80 | 15 |
|  | |

Weight of evidence binning will also be tested, which divides the AGE attribute into four classes and the ‘separation’ of response results indicates that it will be a more effective means than based only on decade.

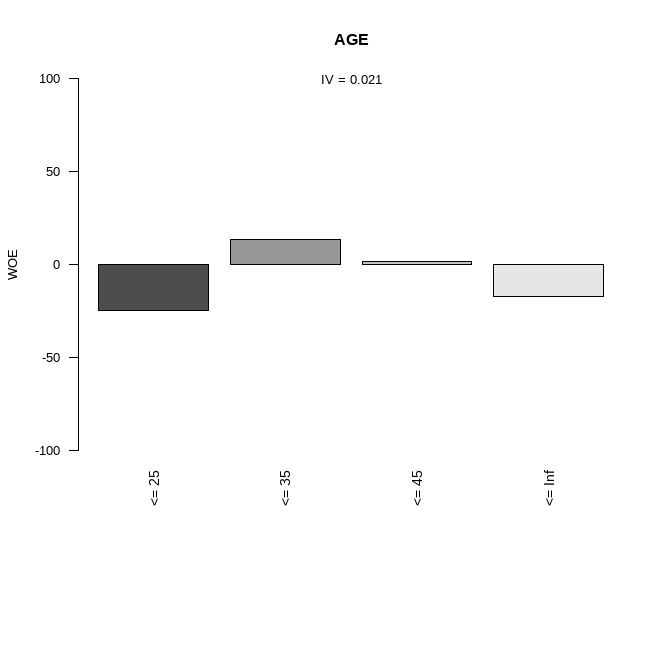


Figure 1: Weight of Evidence Binning Result - Age

In order to produce variables useful and potentially meaningful to statistical models, all transactional data will be replaced with aggregated and computed statistics.

* **Utilization** - balance divided by the consumer’s credit line each month. The Values resulting are between approximately negative two and positive eleven. After averaging across all months to produce util\_avg, these will be scaled between negative one and positive one, testing normalization first.
* **Payment\_ratio** - payment each month divided by the previous month’s balance. To be initially normalized between zero and one. Any month with zero balance will be set to one as this is a “perfect” payment; no remaining balance due. The average was calculated across all then the original time series values were dropped.
* **Age\_bins** – Initial binning by decade; Weigh of Evidence has also been used and the two will be compared for performance/correlation with target.
* **Other binning** - Binning was also tested on the following categorical fields: PAY, education. Effect was minimal and often detrimental to model quality so it was abandoned
* **Sex, Education and Marriage** – categorial variables that after testing for minor engineering efforts resolved to remain as received
* **bill\_max** – the maximum value across all month’s bills
* **payment\_max** – the maximum value across all months’ payment values
* **pay\_max** – each month’s pay field indicates how delinquent a customer is. This field indicates the longest (highest value) that a customer has been delinquent in available history

## Exploratory Data Analysis

After initial engineering of the data for feature generation, the distribution and key statistics of the explanatory variables should be reviewed for consideration of further engineering or elimination from usage.

A picture containing shoji, crossword puzzle, building, shrimp

Description automatically generated

Figure 2: Binned Distribution of Variables

A picture containing crossword puzzle, shoji, scoreboard, clipart

Description automatically generated

Figure 3: Binned Distribution of Variables – Log Transformed

A common practice for modeling data preparation is to log-transform the data in each variable to reduce skewness and achieve a more normal distribution (Feng et. al. 2014). Figure 3 shows how the transformation effects certain variables particularly in comparison with the original seen in Figure 2 – and is essential prior to using standardization or normalization across the set.

The target variable, DEFAULT, is found to be imbalanced, though not severely. This will require adjustment to accuracy measure and potentially modeling choices. Simple accuracy would not accurately call out a high false positive rate, while using F1 or confusion matrices as we plan to would help with representation.



Figure 4: Histogram of Target Variable

Rebalancing was tested by oversampling the positive default case (represented by “1”).

Icon

Description automatically generated

Figure 5: Re-sampled Target Variable Counts

The method used to calculate some of the engineered variables will require further adjustment. NAs result from 0 in the denominator (divide by 0 error) of division engineered variables. These fields must be filled in logically. Since a Pay Ratio is engineered by dividing the payment each month by the previous month’s balance, a balance of zero is a positive result and should be set to the maximum value for this field.



Figure 6: Engineered Variable EDA Excerpt

Chart, box and whisker chart

Description automatically generatedChart, histogram

Description automatically generated

**Figure 7: Payment and Utilization distribution over class by Age**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Table 5: Data Quality Overview - Engineered** | | | | | | | |
|  | | |  |  |  |  |  |
|  |  | |  |  |  |  |  |
| Column Name | | Data Type | Present Values | Missing Values | Unique Values | Minimum Value | Maximum Value |
| DEFAULT | int32 | | 30000 | 0 | 2 | 0 | 1 |
| age\_bins | category | | 30000 | 0 | 6 | 21-30 | 71-80 |
| bill\_avg | float64 | | 30000 | 0 | 27370 | -56043.166667 | 877313.833333 |
| payment\_avg | float64 | | 30000 | 0 | 19180 | 0.0 | 627344.333333 |
| pay\_ratio1 | float64 | | 30000 | 0 | 20209 | 0.0 | 4444.333333 |
| pay\_ratio2 | float64 | | 30000 | 0 | 20042 | 0.0 | 5001.0 |
| pay\_ratio3 | float64 | | 30000 | 0 | 19411 | 0.0 | 4444.333333 |
| pay\_ratio4 | float64 | | 30000 | 0 | 18580 | 0.0 | 129.705128 |
| pay\_ratio5 | float64 | | 30000 | 0 | 18025 | 0.0 | 690.655172 |
| ratio\_avg | float64 | | 30000 | 0 | 24820 | 0.0 | 2667.199955 |
| util1 | float64 | | 30000 | 0 | 25565 | -0.619892 | 6.4553 |
| util2 | float64 | | 30000 | 0 | 25088 | -1.39554 | 6.3805 |
| util3 | float64 | | 30000 | 0 | 24738 | -1.0251 | 10.688575 |
| util4 | float64 | | 30000 | 0 | 24452 | -1.3745 | 5.14685 |
| util5 | float64 | | 30000 | 0 | 24075 | -0.876743 | 4.9355 |
| util6 | float64 | | 30000 | 0 | 24075 | -0.876743 | 4.9355 |
| util\_avg | float64 | | 30000 | 0 | 28402 | -0.23259 | 5.537758 |
| balance\_growth\_6mo | float64 | | 30000 | 0 | 27137 | -4.7004 | 1.7911 |
| bill\_max | int32 | | 30000 | 0 | 23979 | -6029 | 1664089 |
| payment\_max | int32 | | 30000 | 0 | 11670 | 0 | 1684259 |
| pay\_max | float64 | | 30000 | 0 | 9 | 0.0 | 8.0 |
| DEFAULT | int32 | | 30000 | 0 | 2 | 0 | 1 |
|  | | |  |  |  |  |  |

After cleaning the engineered variables, a review of the high level data quality report helps to determine if any further actions are necessary.

“Correlation allows you to interpret the covariance further by identifying both the direction and the strength of any association” (Tilman, 2016), and a correlation matrix makes this visually accessible across the range of explanatory variables and the target.

Chart

Description automatically generated

Figure 8: Correlation Plot of Target Variable and Engineered Variables

The matrix shows that several of the variables have very low coefficients with the target variable, “DEFAULT” and should be considered for dimension reduction. Explanatory variables that have high correlation with each other should also be considered for reduction (Gelman, 2014) as the intent of building a model is locating the variance or finding the signal in the noise. Multiple correlated signals does not contribute to the model; it is redundant & models perform better with fewer inputs.

The columns with the highest correlations (either negative or positive) with the target default are: payment\_avg , util\_avg , payment\_max, and pay\_max. payment\_avg and payment\_max are too highly correlated with each other to provide variance, and payment\_max having the slightly lower correlation to the target means it should be eliminated. The boxplots display a lack of distinction of classes across two of the credit history-based variables, and some evidence of variance by age group though not drastic.

## Predictive Modeling: Methods and Results

In addition to experimentation with data preparation and feature engineering, testing different model types and hyperparameters can yield vastly different prediction performance results. The model must be chosen based on the input data types and the priorities in performance. Although the data as it was provided has some time-series features, the intent of the problem statement is not a time-series problem. The fluctuations over time have been used to engineer features and we are not looking to predict “when” a consumer may default but “if”.

### Random Forest

The first model tested is random forest. Random forest models tend to perform well with few features and either categorical variables or binning implemented to allow each node in the constituent trees to have finite branches.

Chart, treemap chart

Description automatically generated

Figure 9: Confusion Matrix: Random Forest

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Table 6: Classification Report: Random Forest** | | | | | | | |
|  | |  | |  | |  | |
|  | Precision | | Recall | | F1-Score | | Support |
|  | |  | |  | |  | |
| 0 | 0.83 | | 0.93 | | 0.88 | | 5766.00 |
| 1 | 0.52 | | 0.27 | | 0.36 | | 1557.00 |
| accuracy | 0.79 | | 0.79 | | 0.79 | | 0.79 |
| macro avg | 0.67 | | 0.60 | | 0.62 | | 7323.00 |
| weighted avg | 0.76 | | 0.79 | | 0.76 | | 7323.00 |
|  | |  | |  | |  | |

Random forest is a very flexible model that can be used across many domains and is a good general purpose tool. Right out of the box, performance is fairly good, though one area we will focus is the True Positive rate. Since the goal of this study is to predict when a customer will default, error here would likely be costly – with a borrower not paying back the money owed. A TPR of 27% is not sufficient for industry.

The variable performance plot is provided in Figure 7 and shows that by a factor of five pay\_max is the most influential input variable.

Graphical user interface, text

Description automatically generated

Figure 10: Feature Importance: Random Forest

### Gradient Boosting

The second model tested is XGBoost. XGBoost is similar to random forest in that it builds off decision trees. There are many difference though in how this is executed. While Random Forest uses hyperparameters to randomly generate many trees and develops an ensemble, XGboost uses the gradient to build one tree at a time and converge towards a solution by minimizing error (Chen, Tianqi, and Guestrin, 2016). XGBoost was able to produce higher accuracy and F1 score across the classes, making it a superior model to random forest in this case.

Chart, treemap chart

Description automatically generated

Figure 11: Confusion Matrix: XGBoost

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Table 7: Classification Report: XGBoost** | | | | | | | |
|  | |  | |  | |  | |
|  | Precision | | Recall | | F1-Score | | Support |
|  | |  | |  | |  | |
| 0 | 0.88 | | 0.98 | | 0.93 | | 5766.00 |
| 1 | 0.87 | | 0.52 | | 0.65 | | 1557.00 |
| accuracy | 0.88 | | 0.88 | | 0.88 | | 0.88 |
| macro avg | 0.88 | | 0.75 | | 0.79 | | 7323.00 |
| weighted avg | 0.88 | | 0.88 | | 0.87 | | 7323.00 |
|  | |  | |  | |  | |

The variable performance plot is provided in Figure 9 and shows that by a factor of two pay\_max is the most influential input variable. This agrees with random forest result, though the remaining variable order is different.

Graphical user interface, text, chat or text message

Description automatically generated

Figure 12: Feature Importance: XGBoost

### Logistic Regression with Variable Selection

To develop a logistic regression model, the feature importance from the previous two models, Random Forest and XGboost, were used and the top three were taken from each. Logistic regression is one of the simplest data science approaches to fitting a model – simply using an algebraic expression with the number of variables chosen as a hyperparameter.

Chart, treemap chart

Description automatically generated

Figure 13: Confusion Matrix: Logistic Regression

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Table 8: Classification Report: Logistic Regression | | | | | | | |
|  | |  | |  | |  | |
|  | Precision | | Recall | | F1-Score | | Support |
|  | |  | |  | |  | |
| 0 | 0.79 | | 0.99 | | 0.88 | | 5766.00 |
| 1 | 0.64 | | 0.03 | | 0.06 | | 1557.00 |
| accuracy | 0.79 | | 0.79 | | 0.79 | | 0.79 |
| macro avg | 0.72 | | 0.51 | | 0.47 | | 7323.00 |
| weighted avg | 0.76 | | 0.79 | | 0.71 | | 7323.00 |
|  | |  | |  | |  | |

The results shown below are for a model using a second-degree equation.

Text

Description automatically generated

Figure 14: Variable and Result summary: Logistic Regression

### Support Vector Machine

The support vector machine is implemented as a support vector classifier. SVMs can be thought similar to logistic regression, but instead of a single line to predict trends in the target variable, it uses geometry, potentially more complex, to split between the classes. Using a liner model to start makes sense since it the target class is binary.

A picture containing shape

Description automatically generated

Figure 15: SVM Separating Hyperline – needs work

Chart, treemap chart

Description automatically generated

Figure 16: Confusion Matrix: SVM

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Table 9: Classification Report: SVM** | | | | | | | |
|  | |  | |  | |  | |
|  | Precision | | Recall | | F1-Score | | Support |
|  | |  | |  | |  | |
| 0 | 0.81 | | 0.94 | | 0.87 | | 5766.00 |
| 1 | 0.42 | | 0.16 | | 0.23 | | 1557.00 |
| accuracy | 0.78 | | 0.78 | | 0.78 | | 0.78 |
| macro avg | 0.62 | | 0.55 | | 0.55 | | 7323.00 |
| weighted avg | 0.72 | | 0.78 | | 0.73 | | 7323.00 |
|  | |  | |  | |  | |

## Comparison of Results

Throughout the initial models true negative rate is the highest class. False negatives are the common issue that is concerning, as failure to predict most of the defualting customers would be extremely costly for the company. False positives are the next concern, but less so as a lost customer is not as expensive as letting through a defaulting one. XGBoost is the best initial model that is able to deal with the target class imbalance.

After further feature engineering, balancing of the target variables, feature reductions, and hyperparameter optimization all of the models were able to be improved. A Recurrent Neural Network was also tested, but not further pursued and would be in a future study. Macro-averaged F1 score will be one measure to be focused on as it well represents both classes, and TPR would be important given the problem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table 10: Model Results: Baseline** | | | | | |
|  | |  | |  |
|  | Model Type | | Macro Avg | TPR |
|  | |  | |  |
|  |  | |  |  |
| RF | | 0.62 | 0.27 |
| XG | | 0.79 | 0.52 |
| LR | | 0.79 | 0.03 |
| SVM | | 0.55 | 0.16 |
| RNN | | 0.66 | 0.59 |
|  | |  | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table 11: Model Results: Re-balanced** | | | | | |
|  | |  | |  |
|  | Model Type | | Macro Avg | TPR |
|  | |  | |  |
|  |  | |  |  |
| RF | | 0.65 | 0.42 |
| XG | | 0.65 | 0.59 |
| LR | | 0.46 | 0.89 |
| SVM | | 0.47 | 0.81 |
|  | |  | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table 12: Model Results: Log-Scaled, Normalized** | | | | | |
|  | |  | |  |
|  | Model Type | | Macro Avg | TPR |
|  | |  | |  |
|  |  | |  |  |
| RF | | 0.65 | 0.42 |
| XG | | 0.73 | 0.59 |
| LR | | 0.73 | 0.67 |
| SVM | | 0.49 | 0.78 |
|  | |  | |  |

## Conclusions

The objective of credit risk modeling is to predict whether a customer will default on the money borrowed from an institution. If a firm relies on modeling that is unsuccessful, it cannot make profit and would thus not survive. In modern day credit industry, accurate models are required to remain competitive, and have allowed certain companies to lead. The models tested in this study produce a variety of outputs which could be chosen based on priority. A cost function would need to be considered before choosing a model, weighing costly false negatives against false positives that lose good customers. While logistic regression was able to achieve very high true positive rates, XGBoost generally performed better and with further feature engineering may yield the greatest profits.

## Bibliography

1. Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
2. Yeh, I. C., and Lien, C. H. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications, 36(2), 2473-2480.
3. Cagan, Michele. Debt 101: From Interest Rates and Credit Scores to Student Loans and Debt Payoff Strategies, an Essential Primer on Managing Debt. First Adams Media hardcover edition, Adams Media, 2020.
4. Dempsey, Robert. Python Business Intelligence Cookbook: Leverage the Computational Power of Python with More than 60 Recipes That Arm You with the Required Skills to Make Informed Business Decisions. 2015. Open WorldCat, https://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=1131993.
5. Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables. R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
6. Correa Bahnsen, Alejandro, et al. “Feature Engineering Strategies for Credit Card Fraud Detection.” *Expert Systems with Applications*, vol. 51, June 2016, pp. 134–42. *DOI.org (Crossref)*, https://doi.org/10.1016/j.eswa.2015.12.030.
7. L. Peng, W. Qing and G. Yujia, "Study on Comparison of Discretization Methods," 2009 International Conference on Artificial Intelligence and Computational Intelligence, 2009, pp. 380-384, doi: 10.1109/AICI.2009.385.
8. Waskom, Michael L. “Seaborn: Statistical Data Visualization.” Journal of Open Source Software, vol. 6, no. 60, 2021, p. 3021, <https://doi.org/10.21105/joss.03021>.
9. Gelman, Andrew. Bayesian Data Analysis. Third edition, CRC Press, 2014.
10. Feng C, Wang H, Lu N, Chen T, He H, Lu Y, Tu XM. Log-transformation and its implications for data analysis. Shanghai Arch Psychiatry. 2014 Apr;26(2):105-9. doi: 10.3969/j.issn.1002-0829.2014.02.009. Erratum in: Gen Psychiatr. 2019 Sep 6;32(5):e100146corr1. PMID: 25092958; PMCID: PMC4120293.
11. Chen, Tianqi, and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System.” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 785–94. DOI.org (Crossref), https://doi.org/10.1145/2939672.2939785.

## Appendix A: Data Dictionary

|  |  |  |
| --- | --- | --- |
| X1 | LIMIT\_BAL | Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit. |
| X2 | SEX | Gender (1 = male; 2 = female). |
| X3 | EDUCATION | Education (1 = graduate school; 2 = university; 3 = high school; 4 = others). |
| X4 | MARRIAGE | Marital status (1 = married; 2 = single; 3 = others). |
| X5 | AGE | Age (year). |
| History of monthly 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. | | |
| X6 | PAY\_1 | repayment status in September, 2005 |
| X7 | PAY\_2 | repayment status in August, 2005 |
| X8 | PAY\_3 | repayment status in July, 2005. |
| X9 | PAY\_4 | repayment status in June, 2005. |
| X10 | PAY\_5 | repayment status in May, 2005. |
| X11 | PAY\_6 | repayment status in April, 2005. |
| Amount of bill statement (NT dollar) | | |
| X12 | BILL\_AMT1 | bill statement amount in September, 2005 |
| X13 | BILL\_AMT2 | bill statement amount in August, 2005 |
| X14 | BILL\_AMT3 | bill statement amount in July, 2005. |
| X15 | BILL\_AMT4 | bill statement amount in June, 2005. |
| X16 | BILL\_AMT5 | bill statement amount in May, 2005. |
| X17 | BILL\_AMT6 | bill statement amount in April, 2005. |
| Amount of previous payment (NT dollar) | | |
| X18 | PAY\_AMT1 | amount paid in September, 2005 |
| X19 | PAY\_AMT2 | amount paid in August, 2005 |
| X20 | PAY\_AMT3 | amount paid in July, 2005. |
| X21 | PAY\_AMT4 | amount paid in June, 2005. |
| X22 | PAY\_AMT5 | amount paid in May, 2005. |
| X23 | PAY\_AMT6 | amount paid in April, 2005. |