A comparison of Logistic Regression performances with adaboosting in credit scorecard model on imbalanced data

Dat Nguyen Ngoc

1. Introduction

The global financial status appears to be stable overall, but some countries are facing economic challenges. Many economies have recovered from the impacts of the CoVID-19 pandemic, and several countries have experienced significant growth in recent months. However, some countries are struggling with high levels of debt and limited access to credit. They are facing credit crunches due to a combination of factors, including political instability, natural disasters, and supply chain disruptions. Since then, the control and assessment of customers' credit scores have become more important.

In financial institutions or governments, a credit scorecard model is used to quantify an individual's or institution's profile into a credit score. Numerous methods have been employed to construct credit scorecards, with logistic regression models being the most prevalent. (Edwards, P. K., et al 2019) These models are desirable due to their robustness and transparency, but they have been surpassed in predictive accuracy by more recent techniques, such as adaBoosting.

The main contribution of this report is to determine whether adaboosting is actually better than Logistics Regression when training data to estimate each customer's credit score.

2. Theoretical background

2.1. Logistic regression

Logistic regression is a modified form of the typical regression technique, utilized when the dependent variable is binary in nature. In other words, it assumes a binary response, such as the occurrence or non-occurrence of an event. Independent variables in logistic regression can be continuous, categorical, or both. Linear relationship assumptions between independent and dependent variables are not made in logistic regression, which distinguishes it from ordinary linear regression. Additionally, logistic regression does not assume that the dependent variable or error terms have a normal distribution. The form of the model is:

$$log\frac{p}{1-p} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k$$

where p is the probability and $Y = 1$ and $X_1, X_2, ..., X_k$ are the independent variables (predictors). $\beta_1, \beta_2, ..., \beta_k$ known as regression coefficients, commonly referred to as beta coefficients, are estimated from the data. The probabilities of the occurrence of a specific event are then calculated using these regression coefficients.

As a result, the explanatory factors are combined linearly to generate the predictor variable $log\frac{p}{1-p}$ in a logistic regression. a logistic function is then used to convert this predictor variable's values into probabilities. Due to its clarity and simplicity, this has been extensively employed in credit rating applications.

2.2. adaboost

A highly well-liked boosting technique called adaBoost (adaptive Boosting) seeks to combine several weak classifiers into one powerful classifier. Yoav Freund and Robert Schapire wrote the first AdaBoost paper.

AdaBoost works with Decision Stumps. Like trees in a Random Forest, Decision Stumps are not yet "completely matured." They have two leaves and one node. AdaBoost substitutes a forest of these stumps for trees.

Making judgments based only on stomps is not a wise strategy. a fully developed tree incorporates the results of all actions to forecast the desired value. In contrast, a stump can only consider one factor when making a decision.

2.3. Scorecard model in credit risk

2.3.1. Weight of Evidence - WoE

WoE (weight of evidence) is one of the feature engineering and feature selection techniques commonly applied in the scorecard model. This method will rank the variables into strong, medium, weak, no impact, etc. based on their ability and strength to predict bad debt. The ranking criterion will be the information value index IV (information value) calculated from the WoE method. The model also generates feature values for each variable. This value will measure the difference in distribution between good and bad. An attribute's WoE is defined as follows:

$$WOE_{attribute} = ln\frac{p_{attr(nonevent)}}{p_{attr(event)}} = ln\frac{\frac{N_{attr(nonevent)}}{N_{tot(nonevent)}}}{\frac{N_{attr(event)}}{N_{tot(event)}}}$$

The WoE method will have different processing techniques for continuous and categorical variables:

In the case of continuous variables, the WoE will label each observation according to the bin label it belongs to. The bins will be consecutive intervals determined from the continuous variable such that the

number of observations in each bin is equal. To determine the bins, we need to determine the number of bins. We can imagine that the ends of bin intervals are quantiles. I will not explain further because this is basic knowledge in statistics.

In the case of categorical variables, the WoE may consider each class as a bin or may group several groups with a small number of observations into one bin. In addition, the degree of difference between the good/bad distribution measured through the WoE index can also be used to identify groups with the same categorical properties. The closer their WoE values are, the more likely they are to be grouped together. In addition, the Null case can also be considered a separate group if its number is significant or grouped into other groups if it is a minority.

2.3.2. Information value - IV

The Information Value (IV) is computed by taking the sum of the Weight of Evidence (WoE) of an attribute's features, which is then weighted. The weight is calculated by taking the difference between the conditional probability of the attribute occurring for an event and not occurring for an event. The formula for computing the IV involves the number of bins of a variable, which is represented by the variable *m*.

$$IV = \sum_{i=1}^{m} (\frac{N_{attr(nonevent)}}{N_{tot(nonevent)}} - \frac{N_{attr(event)}}{N_{tot(event)}}) WOE_i$$

The Information Value (IV), being a real number, does not have any restrictions on its value. In general, an attribute with a higher Information Value is deemed to be more predictive than those with a lower Information Value.

2.3.3. Credit Scorecard

To convert the model into a scorecard, we require the logistic regression coefficients obtained from the model fitting process, along with the converted WoE (Weight of Evidence) values. The conversion of the model scores from the log-odds unit to a points system is also necessary.

For each independent variable X, its corresponding score is:

$$Score = (\beta.WOE + \frac{\alpha}{n}).Factor + \frac{Offset}{n}$$

Where:

β_i : Logistic regression coefficient for the variable X_i

α : Logistic regression intercept

WoE: Weight of Evidence value for variable X_i

n: Number of independent variables X_i in the model
Factor, offset: Known as scaling parameter

3. Experiments

3.1. Preprocessing

We can observe that only two variables have missing values. Beside that, both of these variables are numeric variables. The number of missing values and the percentage of the total number of values are shown in Fig 1.

	Missing Values	% of Total Values
loan_int_rate	1910	9.6
person_emp_length	546	2.7

Fig 1. Features have missing value of the original data

In this case, we handle the missing data of numeric features by the means of each feature.

In the scorecard model, we also use Information value to determine how this variable affects the classification. The information value of each feature is shown in Fig 2.

Features	IV
loan_percent_income	0.88
loan_grade	0.875
loan_int_rate	0.65
person_income	0.48
person_home_ownership	0.38
cb_person_default_on_file	0.17
loan_intent	0.099
loan_amnt	0.085
person_emp_length	0.046
person_age	0.017
cb_person_cred_hist_length	0.005

Fig 2. Information value of each feature

We can completely remove variables if the IV is less than 0.02. However, to ensure fairness when comparing with adaboost models, I do not remove any variables in this part.

3.2. EDA

3.2.1. Loan status

The percentage of people in this data set who have the ability to repay their debts are relatively high - up to more than 78%. This demonstrates the extreme imbalance in the dataset's 'Loan Status' feature. Yet, since data imbalance is a common occurrence in practice, it is not a problem that needs to be addressed. "In real applications, class imbalance is by far the most common scenario. Indeed, many problems that are worth solving are inherently imbalanced. This happens because resources are limited. Since you want to invest your resources in the few cases that are likely to turn out positive, a model that can predict a rare event (such as a customer that churns, an earthquake, or a user that likes a movie) is extremely valuable." (Samuele Mazzanti, et al 2022). It can easily be observed in Fig 3 below.

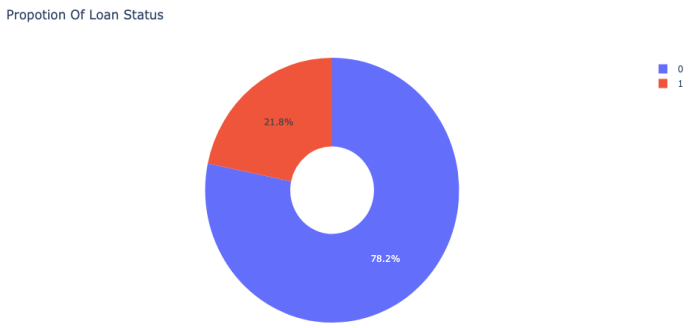

Fig 3. Percentage of defaulter and non-defaulter status in dataset

3.2.2. Person home ownership

50.4% renters of the total borrowers will make up the majority of borrowers, in which the default rate of this group is the highest with the rate of 3 out of 10 borrowers defaulting. Following that, those with mortgages (41.5%) and those who already own a house (7.8%) borrow very little, suggesting that their financial situations may be stable. The percentage of each group is shown in Fig 4 below.

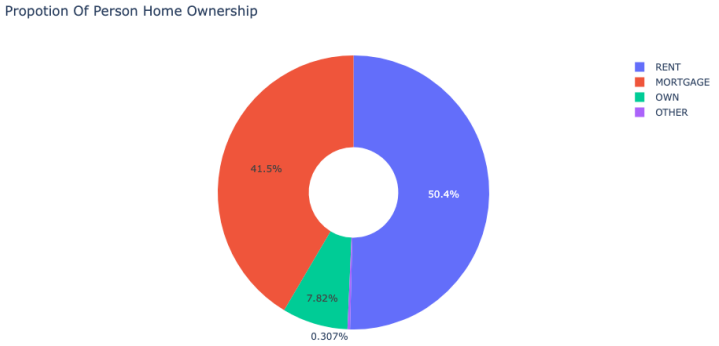

Fig 4. Percentage of home ownership category

3.2.3. Loan intent

Individuals will borrow money mostly for education, with 4000 people having loan intent for education, accounting for 19.6%. In addition to using their borrowed funds for education, people will also use them for medical purposes (18.8%), venture intent (17.5%) as well as personal objectives (16.8%). The percentage of each loan intent category is shown in Fig 5 below.

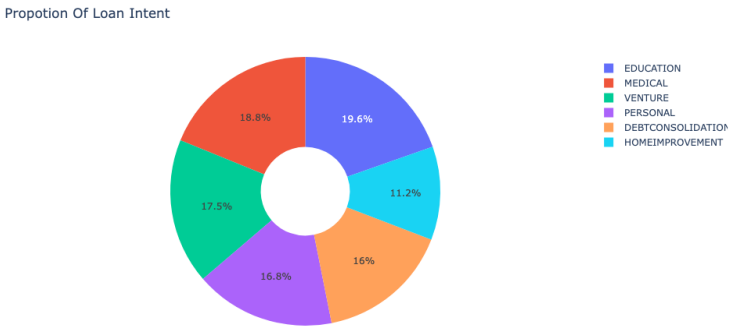

Fig 5. Percentage of each loan intent category

3.2.4. Loan grade

It should come as no surprise that nations with high debt ratings - a, B, and C had relatively low default rates. Yet, there are more people who do not pay their loans than those who do, meaning that the percentage of defaulters for applications scored below D may be more than 50%. The percentage of each loan grade is shown in Fig 6 below.

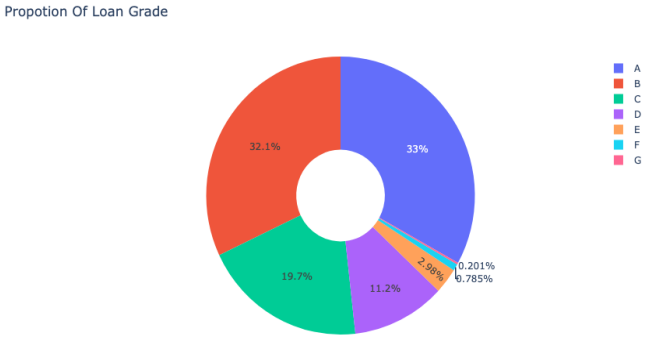

Fig 6. Percentage of each loan grade

3.2.5. Historical default

Those who have never defaulted before make up around 82.4% of the population, however their default rate on this loan is extremely high at roughly 23%. It is important to note that the default rate for borrowers with a history of defaulting on loans is still relatively high at 40%. The percent of people who default or not in history is shown in Fig 7 below.

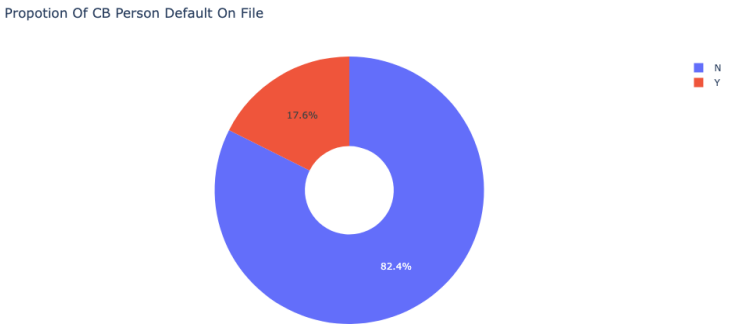

Fig 7. Percentage of people who default or not in history

4. Variables Used

We still hold 12 variables of the original dataset to build the adaboost and Logistic model. all of 12 variables were identified in Fig 8.

Feature	Type	Meaning
person_age	numeric	Age
person_income	numeric	Annual income
person_home_ownership	category	Home ownership
person_emp_length	numeric	Employment length (in years)
loan_intent	category	Loan intent
loan_grade	category	Loan grade
loan_amnt	numeric	Loan amount
loan_int_rate	numeric	Interest rate
loan_status	boolean	Loan status
loan_percent_income	numeric	Loan percent income
cb_person_default_on_file	category	Historical default
cb_person_cred_hist_length	numeric	Credit history length

Fig 8. Variable meaning

5. Result and discussions

5.1. Evaluate method

The models were tested for their effectiveness on several measures – true positive (TP), false positive (FP), F1 score and accuracy. These results are shown in Fig 9.

Results of classification				
	TP	TN	F1 score	accuracy
Model 1 (LR)	501	2970	0,66	0,87
Model 2 (AB)	517	2960	0,67	0,88

Fig 9. Result of two model

The AUC (area under curve) index measures the area under the ROC curve, indicating whether the classification ability of the GOOD/BAD contracts of the model is strong or weak. AUC $\in$ [0,1], the larger its value, the better the model. The AUC chart of 2 model are shown in Fig 10 and Fig 11.

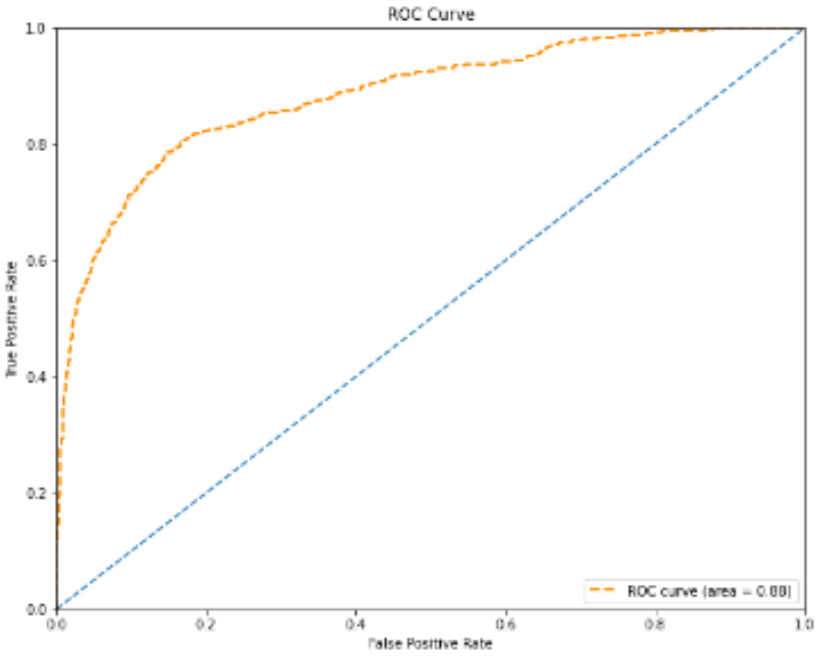

Fig 10. RoC Curve of Logistic Model

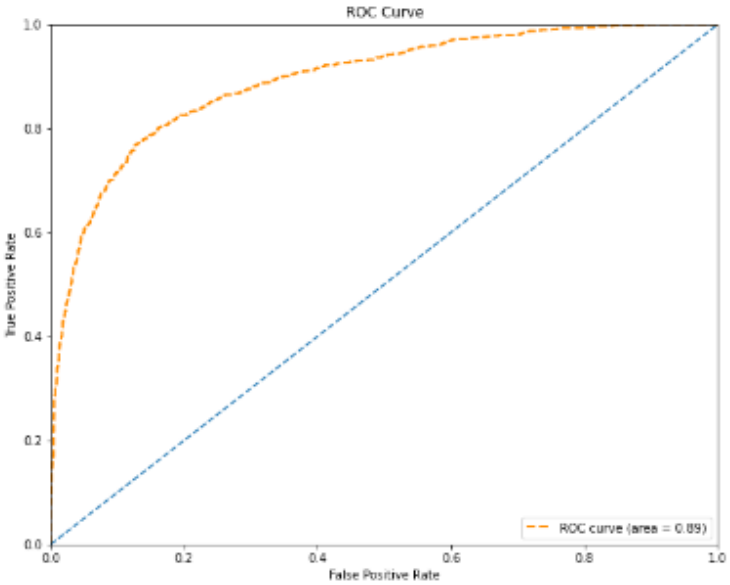

Fig 11. ROC Curve of adaboost Model

We can observe that the AUC of the two models are quite similar (~ 0.88), showing that two of these models' predictive ability is good and can be applied in practice. Besides that, relying on Fig 9, we see the accuracy of two models is also quite approximate (~ 0.87).

We can see not a big difference between these two models. So it's suitable to use Logistic Regression to build a scorecard model. There are many different algorithms applied to build credit scorecard models, such as: Boosting, Neural Networks, Random Forests, and SVM. These algorithms have better classification results, but their ability to explain the results is not good, so they are not often used to build credit scorecard models in practice.

Kolmogorov-Smirnov testing for LR is a test that measures the difference in the distribution between GooD and BaD according to threshold ratios. If the model is able to classify GooD and BaD well, then the cumulative probability distribution function (CDF) between GooD and BaD must have a large separation. on the contrary, if the model is very weak and its prediction result is only equal to a random selection. Then the cumulative probability distribution of GooD and BaD will be close to each other and asymptotically 45 degrees diagonal. The Kolmogorov-Smirnov curve is shown in Fig 12.

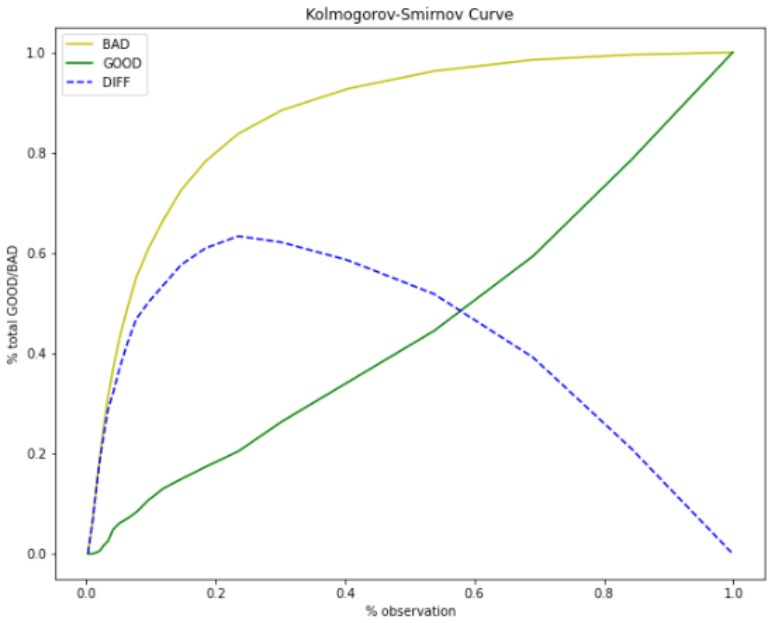

Fig 12. Kolmogorov-Smirnov curve of LR

The Kolmogorov-Smirnov test will test the hypothesis that Ho is two probability distributions Good and Bad have no difference. When the P-value < 0.05, the null hypothesis is rejected. The statistic value and P-value of Kolmogorov-Smirnov test are shown in Fig 13.

Statistic value	p-value
0,524	0,0055

Fig 13. Statistic value and P-value of Kolmogorov-Smirnov test

In this case, a P-value < 0.05 indicates that the cumulative distribution between BaD and GooD rates is different. Therefore, the model is significant in the classification of records.

5.2. Model's result

From the Logistic model, we have alpha and beta to calculate the score for each feature and also for each record in the data. alpha has a value of ~ -1.3489. The beta coefficient for each variable is shown in Fig 14.

Feature Name	Beta
person_age	0.241
person_income	0.783
person_emp_length	0.172
loan_amnt	0.25
loan_int_rate	0.11
cb_person_cred_hist_length	0.119
loan_percent_income	0.99
person_home_ownership	0.899
loan_intent	1.266
loan_grade	1.123
cb_person_default_on_file	-0.013

Fig 14. Beta coefficients estimated by LR

When we have all the values of the beta, alpha coefficients and the WoE, we put all of those values into the formula for calculating score to calculate the credit score of each feature. The credit score of each feature is shown in
Fig 15.

Features	Credit Score
person_age	49,80
person_income	73,30
person_home_ownership	92,3
person_emp_length	49,91
loan_intent	30,6
loan_grade	121,92
loan_amnt	50,16
loan_int_rate	57,60
loan_percent_income	112,43
cb_person_default_on_file	49,60
cb_person_cred_hist_length	49,33

Fig 15. Credit score of each feature

After we have the scores for each feature, the next thing we need to do is calculate the credit score for each profile in the dataset by adding the credit scores for each variable of that profile together. The distribution of credit score for each profile is shown in Fig 16 and Fig 17.

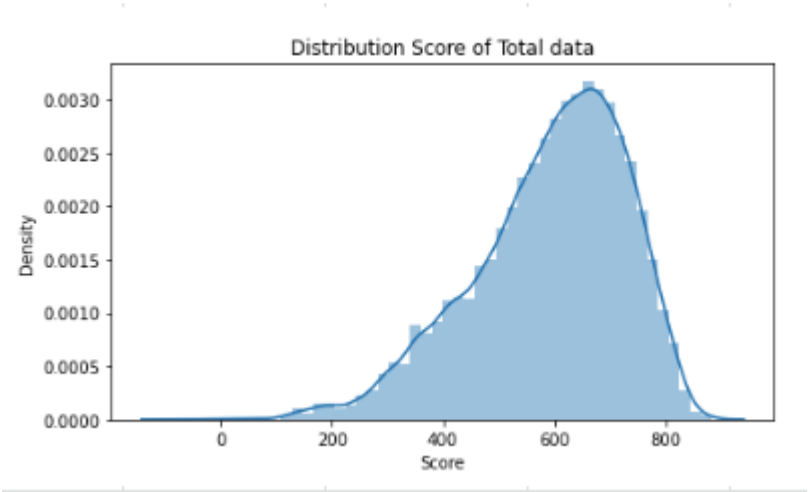

Fig 16. Distribution credit score of each profile

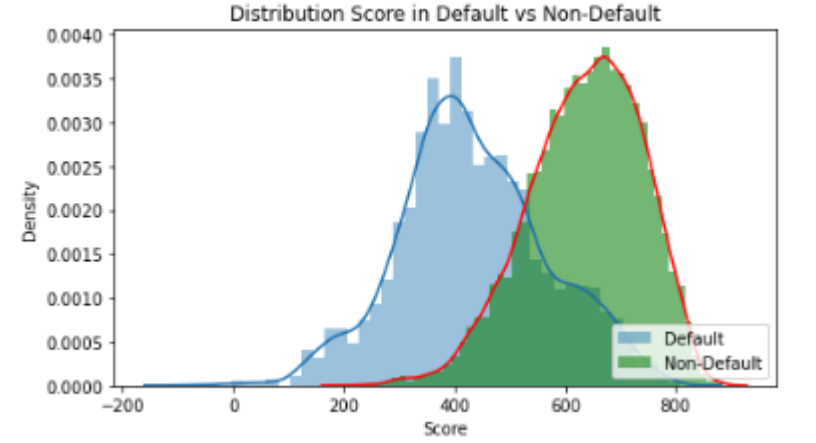

Fig 17. Distribution credit score of Default and Non-Default profile

We can see that the credit score distribution in the dataset is roughly normally distributed and tends to be right skewed. Most customers will have a credit score in the range of 500 to 700.

A good model will clearly separate the Default/Non-Default distributions. The ideal distribution is a smile shape**.** Good customers with high credit scores tend to be right-skewed. Default customers with low credit scores tend to be left-skewed.

Our current model has a different distribution of Default and Non-Default records. Credit score assesses the level of confidence for customers to commit to repaying the debt. A person with a high credit score has more confidence in themselves.

6. Conclusions

The objective of my report is to compare the performance of a logistic regression model and adaBoosting when training the dataset for constructing credit

scorecards. Our analysis shows that the predictive accuracy of the logistic regression model and adaBoosting don't have much difference, while logistic regression has a good ability to interpret the result. When applying the logistic regression model to this dataset, it's easy to observe the relatively ideal smile-shape distribution of defaults and non-defaults. Regrettably, the proposed model requires a considerably longer time to calculate parameters.

7. References

1. Dong, G., Lai, K. K., & Yen, J. (2010). Credit scorecard based on logistic regression with random coefficients. *Procedia Computer Science*, *1*(1), 2463-2468.
2. Liu, B., Lu, L., Zeng, Q., & Li, Y. (2021, May). Implementation of credit scoring card model based on logistic regression and lightgbm. In *2021 International Conference on Control Science and Electric Power Systems (CSEPS)* (pp. 175-178). IEEE.
3. Edwards, P. K., Duhon, D., & Shergill, S. (2019). Real adaBoost: boosting for credit scorecards and similarity to WoE logistic regression.
4. Rodeiro, C. L. V., & Nadas, R. (2010). Effects of modularisation. *Cambridge: Cambridge assessment*.
5. Maldonado, M., Haller, S., Czika, W., & Siddiqi, N. Creating Interval Target Scorecards with Credit Scoring for SaS® Enterprise Miner™.
6. Mazzanti S. (2022, august 24). *Your Dataset Is Imbalanced? Do Nothing!* Medium. https://towardsdatascience.com/your-dataset-is-imbalanced-do-nothing-abf6a0049813
7. Khanh, P.N.(n.d.) (2020, January 17). *Khanh's blog*. Khanh's Blog. https://phamdinhkhanh.github.io
8. arsenault, M. o. (2020, august 7). *KoLMoGoRoV–SMIRNoV TEST*. Medium. https://towardsdatascience.com/kolmogorov-smirnov-test-84c92fb4158d
9. Kinden Property, T. (2020, January 8). *Intro to Credit Scorecard*. Medium. https://towardsdatascience.com/intro-to-credit-scorecard-9afeaaa3725
10. *SaS Help Center*. (n.d.). SaS Help Center. https://documentation.sas.com/doc/en/vdmmlcdc/8.1/casstat/viyastat_binning_details02.htm
11. *a Guide To Understanding adaBoost | Paperspace Blog*. (2020, February 23). Paperspace Blog. https://blog.paperspace.com/adaboost-optimizer/

APPENDIX

All the figures in this report were calculated and visualized in Python. all the code was pushed on the Github repository's named score-model. Check these repositories for more detail.