JIST Journal of Information Science and Technology
Volume 14, NO 2 | JUL – DEC 2024 | 36-42
ISSN: 2651-1053 (Online)
ISSN: 1906-9553 (Print)

An Application of Machine Learning Techniques for Loan Default Payment Prediction

Wilawan Inchamnam, Jesada Kajornrit and Waraporn Jirapanthong[*]

Collage of Creative Design and Entertainment Technology Dhurakij Pundit University

Received: June 26, 2024; Revised: December 20, 2024; Accepted: December 24, 2024; Published: December 28, 2024

ABSTRACT – In the banking business, predicting customer default payments has become a crucial operation to prevent and mitigate risks caused by non-performing loans. Presently, machine learning techniques are used alongside traditional methods for this task. This paper explores several ways to apply machine learning techniques in predicting default payments. The prediction development framework includes data encoding, data sampling, and model development. At each step, various techniques are tested and compared to find optimal solutions for business requirements. Our findings conclude that ensemble models are a good choice over a single model to increase the precision of the default payment class. The Over-sampling method is a suitable choice to increase recall of the default payment class, whereas the Under-sampling method is not recommended. Furthermore, if the size of the input vector is a concern, the Weight of Evidence encoding method can be used instead of One-hot encoding without a loss in performance.

KEY WORDS -- Loan Default Payment, One-Hot Encoding, Weight of Evidence Encoding, Over-Sampling Technique, Under- Sampling Technique, Decision Trees Classifier, Ensemble Methods

1. Introduction

In the banking business, assessing customer loans is a crucial procedure to minimize the impact of default payments and maintain risks at an acceptable level. Currently, the loan evaluation process relies not only on human expertise but also incorporates modern analytics like machine learning. The prediction model for loan default payments utilizes both current and historical customer information to assess their ability to repay on time [1]. An accurate prediction model improves the decision-making of human experts, instilling greater confidence. Consequently, the development of a precise loan default payments prediction system stands as a vital task for ensuring the profitability and sustainability of the bank.

In the current landscape, machine learning methodologies have become pervasive across diverse industries, and the banking sector is no exception. The utilization of machine learning prediction models enables banks to anticipate the likelihood of loan default payments in advance, thereby facilitating proactive risk mitigation strategies. It is undeniable that the efficacy of machine learning models depends upon the quality of the training data. Unfortunately, data related to loan default payments commonly exhibits imbalance [2], with the number of default payments smaller in comparison to non-default payments. Additionally, information about new customers, particularly those entering the workforce for the first time, is often deficient. Consequently, generating accurate predictions based on such limited information is rather challenging. This paper seeks to address these challenges.

The paper is structured as follows: Section 2 introduces relevant literatures. Section 3 presents the statistics of the loan default payments dataset. Section 4 proposes the methodology and related techniques. Section 5 presents the experimental results followed by discussions. Finally, section 6 is our conclusion.

2. Related Works

Several studies on predicting loan default payments favor the use of interpretable machine learning models such as Decision Trees and their ensemble techniques. These models offer favorable prediction results, relatively quick training times, require minimal data preprocessing, and provide a prediction

[*]*Corresponding Author: waraporn.jir@dpu.ac.th*

mechanism that is easily understandable to humans. The following literature reviews some of these studies.

Soni and Shankar [3] employed Random Forest classification to forecast bank loan defaults. They asserted that the ensemble technique surpasses single models like logistic regression, k-nearest neighbors, support vector machine, and decision tree classification. In a similar vein, Shaheen and ElFakharany [4] demonstrated that Random Forest and Gradient Boosting Tree outperform individual techniques in prediction accuracy when applied to predict loan default datasets.

Fan [5] conducted a comparison between LightGBM and Random Forest algorithms for predicting personal loan defaults. He asserted that LightGBM demonstrated superior predictive performance. Similarly, Lai [6] affirmed the effectiveness of AdaBoost, highlighting its superior performance compared to XGBoost, Random Forest, K- Nearest Neighbors, and Neural Network in predicting loan defaults using real-world datasets from a prestigious international bank. In another study, Barua et al. [7] investigated the use of the CatBoost algorithm for loan default prediction. CatBoost, known for its fast learning and ability to handle categorical data, was compared to Random Forest and Gradient Boosting Tree. The authors claimed that CatBoost achieved the highest accuracy among all other algorithms.

Al-qerem et al. [8] introduced various classification methods, such as Naïve Bayes, Decision Tree, and Random Forest, for predicting loan defaults. Additionally, they applied a range of preprocessing techniques to the dataset and utilized three different feature extraction algorithms to improve accuracy and performance. In a related study, Patel et al. [9] employed Logistic Regression, Gradient Boosting, CatBoost Classifier, and Random Forest for forecasting loan defaults. They contended that Gradient Boosting and CatBoost Classifier offer comparable accuracy, slightly surpassing Random Forest. However, Logistic Regression yielded unsatisfactory results.

Up to this point, one can see that the accuracy of predictions depends on the dataset's characteristics, especially its features and the information within those features. Handling this challenge becomes more complex when the dataset is imbalanced, leading classifiers to potentially misclassify rare samples from the minority class. It is not universally true that one machine learning algorithm outperforms others in all scenarios. Additionally, resolving this issue doesn't solely rely on machine learning algorithms; some additional techniques may provide assistance. Consequently, conducting a study is imperative to identify appropriate solutions for this task.

3. Loan Default Payments Dataset

The dataset in this paper is the loan default payments of individual customers. As the source of the data is confidential, the dataset is anonymized. The prediction features of the dataset are demographic information about the customers. The original data consists of sixteen features, as shown in Table 1. However, some redundant or impractical features, such as CUSTOMER_DOB, LOAN_DATE, or ZIP are not selected. The records containing missing values are also removed.

The dataset contains 51,018 records of customer information, which are labeled as default payments (class = 1) and non-default payments (class = 0). The dataset is divided into cross-validation data and final-validation data, with the number of cross-validation records being 38,263 and the number of final-validation records being 12,755. The imbalance ratio of cross-validation and final-validation data is 1:4.28 and 1:4.15, respectively. Table 2 presents some statistics of numeric features, and Table 3 displays some statistics of categorical features.

Table 1. Shows Loan Default Payments Dataset

No	Features	Type	Use	Value Range
1	AGE	Integer	Y	[18, 71]
2	COMPANY_TYPE	Category	Y	5 Unique Values
3	CUSTID	Identifier	Y	-
4	CUSTOMER_DOB	Category	N	-
5	EDUCATION	Category	Y	4 Unique Values
6	LOAN_AMOUNT	Integer	Y	[19415, 100513]
7	LOAN_DATE	Date	N	-
8	MARITAL_STATUS	Category	Y	4 Unique Values
9	NO_OF_DEPENDENT	Integer	Y	[0, 44]
10	SEX	Category	Y	2 Unique Values
11	STATE_NAME	Category	Y	21 Unique Values
12	TOTAL_MONTHLY_ INCOME	Integer	Y	[0, 1500000]
13	YEARS_OF_EXPERIENCE	Integer	Y	[0, 70]
14	YRS_IN_PRESENT_ JOB	Integer	Y	[0, 60]
15	ZIP	Category	N	-
16	LABEL	Category	Y	2 Unique Values

Table 2. The dataset statistics of numeric features

Statistics	Age	Loan Amount	Number of Dependent	Total Monthly Income	Year of Experience	Year in Present Job
mean	32.84	50895.28	0.95	16235.59	6.47	6.22
std	9.61	6474.09	1.28	11708.12	6.65	6.19
min	18.00	22830.00	0.00	0.00	0.00	0.00
max	71.00	96747.00	34.00	750000.00	70.00	60.00
skewness	0.75	-0.14	1.71	13.94	2.18	2.10
kurtosis	-0.24	0.78	12.65	511.93	6.03	4.88

Table 3. The dataset statistic of category features

Features	Category Code	Counts
COMPANY_TYPE	(0) Government, (1) Individual, (2) Private limited company, (3) Public limited company, (4) Others	(0) 8323, (1) 8467, (2) 22521, (3) 2360, (4) 9347
EDUCATION	(0) High school, (1) Graduate, (2) Postgraduate, (3) Others	(0) 18704, (1) 22286, (2) 1149 (3) 8879
MARITAL_STATUS	(0) Married, (1) Single, (2), Widowed, (3) Divorced	(0) 37346, (1) 13534, (2) 98, (3) 40
SEX	(0) Male, (1) Female	(1) 46040, (2) 4978
STATE	(0) 5701, (1) 5158, (2) 4972, (3) 4580, (4) 4543, (5) 3775, (6) 3323, (7) 3295, (8) 2831, (9) 2599, (10) 1904, (11) 1827, (12) 1810, (13) 1457, (14) 842, (15) 659, (16) 592, (17) 523, (18) 451, (19) 103, (20) 73	

Principal component analysis (PCA) is employed for the analysis of the dataset. Scatter plots between the first and second principal components are depicted in Figure 1. Observably, there is a mixing of minor class datapoints (light + sign) into major class datapoints (strong + sign), potentially introducing challenges in prediction. This pattern is typically observed in new customers for whom the bank has limited information.

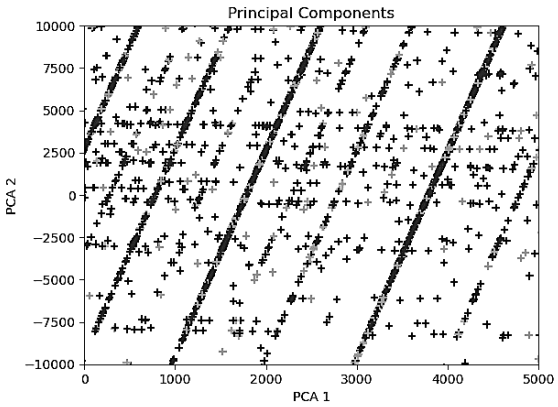

Figure 1 illustrates scatter plot between PCA1and PCA2 of the dataset.

4. Prediction Methodology

The complete prediction methodology is illustrated in Figure 2. The left segment of the figure represents the operational part, while the right side represents the modeling part. In the operational part, new vector inputs undergo preprocessing to encode category features into numeric features. Next, they are input into the prediction module to determine the class output. The input vector, comprising customer information, is utilized to predict whether the customer belongs to the default or non-default payment class.

In the modeling part, the training data are fed into the encoding process to convert category features into numeric features. Next, the encoded training data are sampled to train a machine learning model. It's important to note that the sampling process is optional, and the entirety of the training data may be utilized instead. The parameters obtained from the encoding process are employed in the preprocessing module, while the trained model is applied in the predicting module.

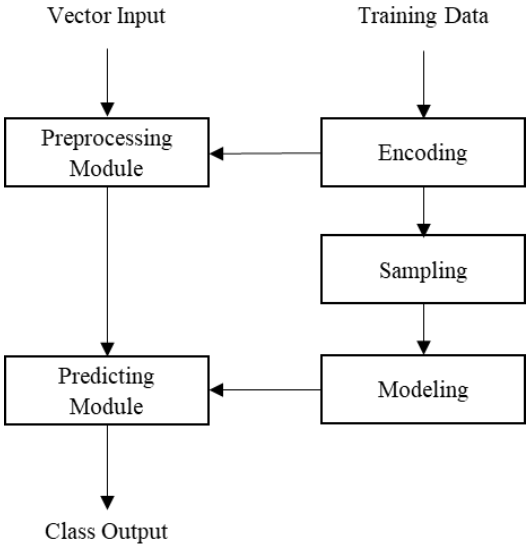

Figure 2 illustrates overall prediction methodology

In practical applications, numerous machine learning libraries tend to favor numeric features over categorical ones. Consequently, it becomes necessary to encode categorical features before training a model. In the Category Encoder website [10], Various category encoding methods have been proposed. This paper opts for two widely recognized encoding methods: namely, One-hot encoding and Weight of Evidence (WoE) encoding methods.

One-hot encoding provides a straightforward method to convert categorical data into numeric form using binary encoding. This process entails establishing binary columns for each category and designating the presence of a category with a "1" in the respective column. However, the number of binary features can expand significantly based on the cardinality of the original features. Consequently, this expansion may lead to an increase in the size of the input vector.

WoE is calculated based on the relationship between the categories of a categorical variable and the likelihood of the target event. The formula for calculating the WoE for a particular category is as follows:

$$WoE = \ln\left(\frac{\text{Distribution of Good Events}}{\text{Distribution of Bad Events}}\right)$$

In cases where the target variable is true (representing non-default payment), it is considered a good event, and vice versa is the bad event. WoE is an encoding method that does not augment the size of the input vector. However, when employed in a non-parametric model for interpretable reasons, it may pose challenges for human analysts in comprehending the insights.

To address the imbalanced class proportions often observed in datasets, a sampling process may be employed. Imbalance is a common occurrence in datasets related to default payment problems. This paper explores two imbalance sampling methods—Over-sampling and Under-sampling—from the Imbalanced-Learn community (https:// imbalanced - learn.org/stable/#).

Several ensemble techniques involving decision trees have gained popularity for addressing classification problems in the banking domain. This study evaluates various techniques, including Random Forests, Bagging, AdaBoost, and XGBoost, all built upon the Decision Trees classifier. The hyperparameters for all models are chosen through K-fold cross-validation, with k set to 5, to determine the most optimized configurations.

Table 4 provides a summary of the combinations of techniques. Given that the prediction objective centers on the default payment class, this paper will predominantly focus on the outcomes associated with that class.

Table 4. Combinations of techniques in the experiments

Experiment	Sampling	Encoding
A	No	One-hot
B	No	WoE
C	Over-sampling	One-hot
D	Over-sampling	WoE
E	Under-sampling	One-hot
F	Under-sampling	WoE

5. Experimental Results

Table 5 presents the results of Experiment A. In general, the ensemble techniques exhibit higher accuracy compared to the single model. Regarding Class 0, the precision of all models is relatively equal, but the recall is higher in the ensemble techniques. Conversely, for Class 1, the precision of ensemble techniques significantly improves compared to the single model, while the recall decreases notably.

Table 6 displays the results of Experiment B. The average accuracy slightly decreases from Experiment A. The average precision for Class 0 and Class 1 does not differ. The average recall for Class 0 decreases slightly, and vice versa for Class 1. Interestingly, the WoE method does not have a significant impact on the recall of prediction models. However, the Bagging technique benefits considerably from this encoding method, resulting in a 3% increase in the recall of Class 1. Similarly, Random Forests also derive advantages from this

method, with a 5% increase in the precision of Class 1.

Table 7 and Table 8 display the results of Experiments C and D, respectively. In these two experiments, the over-sampling method is applied. Overall, the accuracy of the models slightly decreases. Regardless of the encoding method used, the over-sampling method tends to decrease the recall of Class 0 and precision of Class 1, while increasing the recall of Class 1. In the case of One-hot encoding, AdaBoost seems to be significantly affected. Recall from Class 0 decreases by 9%, precision of Class 1 decreases by 7%, while the recall of Class 1 improves by 9%. With WoE encoding, XGBoost and Bagging exhibit a substantial effect. Recall of Class 0 decreases by 5%, precision of Class 1 decreases by about 9%, but the recall of Class 1 increases by about 8%.

Table 9 and Table 10 present the results of Experiments E and F, respectively, where the under-sampling method is applied. Under-sampling yields outcomes in the same direction as the over-sampling method, albeit with a larger magnitude. In One-hot encoding, the average recall of Class 0 decreases by 32%, the average precision of Class 1 decreases by about 16%, and the average recall of Class 1 increases by about 30%. In WoE encoding, the average recall of Class 0 decreases by more than 37%, the average precision of Class 1 decreases by about 21%, and the average recall of Class 1 increases by about 37%. It appears that under-sampling has a more pronounced effect on WoE compared to One-hot encoding schemes.

Table 5. Results of experiment A

Model	Accuracy	Class 0			Class 1		
		precision	recall	f1-score	precision	recall	f1-score
Decision Tree	0.743	0.845	0.834	0.840	0.347	0.366	0.356
Random Forests	0.813	0.838	0.953	0.891	0.541	0.233	0.326
AdaBoost	0.777	0.844	0.887	0.865	0.404	0.317	0.355
XGBoost	0.812	0.841	0.946	0.890	0.534	0.256	0.346
Bagging	0.808	0.835	0.949	0.888	0.510	0.219	0.306
Average	0.791	0.841	0.914	0.875	0.467	0.278	0.338

Table 6. Results of experiment B

Model	Accuracy	Class 0			Class 1		
		precision	recall	f1-score	precision	recall	f1-score
Decision Tree	0.743	0.843	0.837	0.840	0.341	0.350	0.345
Random Forests	0.820	0.839	0.962	0.896	0.593	0.230	0.332
AdaBoost	0.743	0.851	0.826	0.838	0.355	0.398	0.375
XGBoost	0.812	0.841	0.946	0.890	0.532	0.255	0.345
Bagging	0.808	0.839	0.943	0.888	0.513	0.247	0.333
Average	0.785	0.842	0.903	0.871	0.467	0.296	0.346

Table 7. Results of experiment C

Model	Accuracy	Class 0			Class 1		
		precision	recall	f1-score	precision	recall	f1-score
Decision Tree	0.737	0.848	0.822	0.835	0.343	0.388	0.364
Random Forests	0.801	0.843	0.925	0.882	0.477	0.285	0.357
AdaBoost	0.724	0.850	0.800	0.824	0.331	0.412	0.367
XGBoost	0.789	0.846	0.902	0.873	0.439	0.320	0.370
Bagging	0.799	0.844	0.920	0.881	0.472	0.295	0.363
Average	0.770	0.846	0.874	0.859	0.413	0.340	0.364

Table 8. Result of experiment D

Model	Accuracy	Class 0			Class 1		
		precision	recall	f1-score	precision	recall	f1-score
Decision Tree	0.745	0.848	0.833	0.840	0.353	0.377	0.365
Random Forests	0.801	0.843	0.924	0.882	0.477	0.287	0.358
AdaBoost	0.752	0.849	0.843	0.846	0.364	0.375	0.369
XGBoost	0.786	0.846	0.898	0.871	0.431	0.322	0.368
Bagging	0.788	0.845	0.902	0.872	0.434	0.313	0.364
Average	0.774	0.846	0.880	0.862	0.412	0.335	0.365

Table 9. Result of experiment E

Model	Accuracy	Class 0			Class 1		
		precision	recall	f1-score	precision	recall	f1-score
Decision Tree	0.555	0.855	0.539	0.661	0.245	0.620	0.351
Random Forests	0.598	0.870	0.589	0.703	0.271	0.634	0.379
AdaBoost	0.561	0.854	0.550	0.669	0.245	0.609	0.350
XGBoost	0.571	0.872	0.549	0.674	0.262	0.665	0.376
Bagging	0.578	0.873	0.558	0.681	0.265	0.662	0.378
Average	0.573	0.865	0.557	0.677	0.258	0.638	0.367

Table 10. Result of experiment F

Model	Accuracy	Class 0			Class 1		
		precision	recall	f1-score	precision	recall	f1-score
Decision Tree	0.564	0.858	0.550	0.671	0.250	0.623	0.357
Random Forests	0.575	0.874	0.553	0.677	0.264	0.668	0.379
AdaBoost	0.565	0.868	0.544	0.668	0.257	0.656	0.369
XGBoost	0.567	0.868	0.546	0.670	0.258	0.654	0.370
Bagging	0.567	0.866	0.548	0.671	0.256	0.647	0.367
Average	0.568	0.867	0.548	0.671	0.257	0.650	0.368

The experimental results yield the following recommendations for achieving default payment (class 1) prediction:

- Using One-hot encoding, a single decision tree model is a good choice for the use case when a higher recall measure is more important. In contrast, an ensemble method would be a suitable solution if precision measure is crucial. Random Forests and XGBoost methods are recommended for the latter scenario. It's worth noting that One-hot encoding with high cardinality category features can result in a large input vector. Model interpretability may become

challenging when extracting knowledge from decision trees in such cases.

- Using WoE encoding, prediction performance does not differ significantly from one-hot encoding, but the input vector to the model is smaller, thereby reducing some computational cost. Random Forest gains benefits from this encoding method with higher precision, and the Bagging technique also gains some benefit with higher recall. In practical terms, if the size of the input vector is a concern, the combination of WoE encoding with Random Forests or the Bagging technique is our suggestion.
- The over-sampling method outstandingly boosts the recall measures of all models, regardless of the encoding method used. However, a decrease in precision measure is a cost to pay. If the recall measure is the primary objective of modeling work, the over-sampling method is recommended. In the case of One-hot encoding, the Bagging method is a good solution. For WoE encoding, the Adaboost method may be our choice.
- Using the under-sampling method is not a good consideration for this problem. Even though the method outstandingly increases the recall measure of class 1, it also decreases the recall of class 0 and the precision of class 1 drastically. This condition occurs in both encoding methods, especially in the WoE method. If the recall measure is the primary concern, we recommend using the over-sampling method instead.
- In practice, data scientists do not limit themselves to using only one model to predict default payment; they may use a group of models working together to pursue better predictions. For example, employing two models simultaneously, where one excels in precision measure and another in recall measure. Furthermore, if more customer financial information is available, this information can be used to develop another prediction model in a modular manner. For instance, considering customers who are not new to credit or those whose transaction behavior is available via digital payments.
- In some cases, default payment prediction has been performed using a traditional rule-based method (traditional expert systems). This condition could lead to the problem of how to extract decision rules from trained machine learning. It may be trivial if a single decision tree model is used, but the prediction performance may be limited to the decision trees. Thus, extracting decision rules from ensemble methods is our future work.

6. Conclusion

The prediction of customer loan default payment holds significance within the banking sector for risk mitigation. The difficulty of this predictive task is particularly pronounced where customer information is constrained, especially among new entrants to the banking institution. This study introduces a utilization of machine learning techniques to forecast customer loan default events. The machine learning techniques encompasses Decision Trees, Random Forest, Bagging, AdaBoost, and XGBoost methods. These methods are employed in conjunction with two encoding methodologies, specifically One-hot encoding and Weight of Evidence encoding. Additionally, both Over-sampling and Under-sampling techniques are applied. Several combinations of these methodologies are evaluated to find out optimal solutions matching to modeling requirements. We found that, in general, all ensemble techniques demonstrate an enhancement in precision measures compared to individual models. Notably, One-hot encoding and Weight of Evidence encoding exhibit no difference in prediction performance but diverge in input vector size. The Over-sampling technique is observed to elevate recall measures but concurrently diminish certain precision measures. Finally, the deployment of machine learning techniques proposed herein is served as a pragmatic guideline for data scientists to design their methodologies with business requirement.

References

[1] A. K. I. Hassan and a. Abraham, “Modeling Consumer Loan Default Prediction Using Ensemble Neural Networks”, International Conference on Computing, Electrical and Electronic Engineering (ICCEEE), Khartoum, Sudan, pp. 719-724, 2013.

[2] T. Alam, K. Shaukat, I. A. Hameed, S. Luo, M. U. Sarwar, S. Shabbir, J. Li, and M. Khushi, “An Investigation of Credit Card Default Prediction in the Imbalanced Datasets”, IEEE Access, 8: 201173-201198, 2020.

[3] A. Soni and K. C. P. Shankar, “Bank Loan Default Prediction Using Ensemble Machine Learning Algorithm”, 2nd International Conference on Interdisciplinary Cyber Physical Systems (ICPS), Chennai, India, pp. 170-175, 2022.

[4] S. K. Shaheen and E. ElFakharany, “Predictive analytics for loan default in banking sector using machine learning techniques”, 28th International Conference on Computer Theory and Applications (ICCTA), Alexandria, Egypt, pp. 66-71, 2018.

[5] S. Fan, “Design and implementation of a personal loan default prediction platform based on

LightGBM model", 3rd International Conference on Power, Electronics and Computer Applications, Shenyang (ICPECA), China, pp. 1232-1236, 2023.

[6] L. Lai, "Loan Default Prediction with Machine Learning Techniques", International Conference on Computer Communication and Network Security (CCNS), Xi'an, China, pp. 5-9, 2020.

[7] S. Barua, D. Gavandi, P. Sangle, L. Shinde, and J. Ramteke, "Swindle: Predicting the Probability of Loan Defaults using CatBoost Algorithm", 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, pp. 1710-1715, 2021.

[8] A. Al-qerem, G. Al-Naymat, and M. Alhasan, "Loan Default Prediction Model Improvement through Comprehensive Preprocessing and Features Selection", Arab Conference on Information Technology (ACIT), Al Ain, United Arab Emirates, pp. 235-240, 2019.

[9] B. Patel, H. Patil, J. Hembram, and S. Jaswal, "Loan Default Forecasting using Data Mining", International Conference for Emerging Technology (INCET), Belgaum, India, pp. 1-4, 2020.

[10] Category Encoders, available at "https://contrib.scikit-learn/category_encoders/"