ENTHUSIASTIC
INTERNATIONAL JOURNAL OF APPLIED STATISTICS AND DATA SCIENCE
Volume 4, Issue 2, October 2024, pp. 85-95

Loan Approval Classification Using Ensemble Learning on Imbalanced Data

Rahmi Anadra [a,1,*], Kusman Sadik [a,2], Agus M Soleh [a,3], Reka Agustia Astari [a,4]

[a] Department of Statistics, IPB University, Kampus IPB, Jl. Raya Dramaga, Babakan, Kec. Dramaga, Kabupaten Bogor, 16680, Indonesia
[1] rahmianadra@apps.ipb.ac.id*; [2] kusmans@apps.ipb.ac.id; [3] agusms@apps.ipb.ac.id; [4] rekaagustiaastari@apps.ipb.ac.id
* Corresponding author

ARTICLE INFO

Keywords
classification
ensemble learning
loan
random forest
XGBoost

ABSTRACT

Loan processing is an important aspect of the financial industry, where the right decisions must be made to determine loan approval or rejection. However, the issue of default by loan applicants has become a significant concern for financial institutions. Hence, ensemble learning needs to be used with random forest and Extreme Gradient Boosting (XGBoost) algorithms. Unbalanced data are handled using the Synthetic Minority Over-sampling Technique (SMOTE). This research aimed to improve accuracy and precision in credit risk assessment to reduce human workload. Both algorithms used a dataset of 4,296 with 13 variables relevant to making loan approval decisions. The research process involved data exploration, data preprocessing, data sharing, model training, model evaluation with accuracy, sensitivity, specificity, and F1-score, model selection with 10-fold cross-validation, and important variables. The results showed that XGBoost with imbalanced data handling had the highest accuracy rate of 98.52% and a good balance between sensitivity of 98.83%, specificity of 98.01, and F1-score of 98.81%. The most important variables in determining loan approval are credit score, loan term, loan amount, and annual income.

1. Introduction

Consumer spending is one of the systemic drivers of macroeconomics and risk. As a result, analysis of consumer lending is critical as people may ultimately seek loans to meet their needs [1]. Loan processing is an important part of the financial industry, where decisions must be made appropriately to determine whether a loan should be approved or denied.

Financial institutions can obtain a source of profit by providing loans. However, the problem of default by applicants has become a significant concern for financial institutions [2]. The current crisis scenario has caused the financial industry around the world to take necessary measures to avoid the risk of not being able to repay the money lent to borrowers. The rampant debt defaults have made many experts rethink whether the current standards and practices are feasible enough to safeguard companies from such occurrences [3].

p-ISSN 2798-253X
e-ISSN 2798-3153

Nowadays, many banks and other financial organizations issue loans after a thorough verification and validation procedure, but there's no guarantee that the person selected is the worthiest applicant. Hence, advanced machine learning techniques are now needed to predict whether a loan should be approved or not [3].

An extension of artificial intelligence, machine learning allows robots to gain new skills by defining models with human input and learning from data. Machine learning algorithms can produce predictions using conditions and logic [4]. Ensemble machine learning is a method that combines various heterogeneous and homogeneous machine learning base models to improve predictions by shrinking the error between observed and predicted data. Many categorize ensemble methods into bootstrap aggregating (bagging), boosting, and stacking categories [5]. This study used group learning algorithms such as random forest and Extreme Gradient Boosting (XGBoost).

Classification and regression trees are the foundation of random forest, an effective ensemble learning method (CART). Ho created the first random choice forest method in 1995, and Breiman improved it even further in 2001. Based on statistical learning theory, this method generates numerous samples from the original dataset using a bootstrap resampling strategy. A decision tree is constructed for every bootstrap sample, and the ultimate forecast is obtained by summing and averaging the forecasts from several decision trees [6]. In the meantime, previous study has developed a novel method called XGBoost to enhance gradient trees by predicting a result utilizing a variety of decision trees [7]. Studies have shown that the XGBoost model outperforms the random forest model, which implies that the XGBoost model is more accurate and better at classifying credit judgments than the random forest model, which performs worse with smaller and imbalanced data [8].

Machine learning algorithms do not work efficiently with imbalanced data. One way to balance the information in the data classes is to use sampling methods such as the Synthetic Minority Over-Sampling Technique (SMOTE). SMOTE uses Euclidean distance to generate random synthetic data from the minority class of its nearest neighbor, increasing the number of data samples. Because the new samples are created using the same features, they resemble the original data [9].

The research aimed to create a loan status prediction system using training and test data and machine learning methods on imbalanced data. Another goal was to minimize the human workload required to assess and analyze loan risk.

2. Methodology

2.1. Data

This research used secondary data sourced from the Kaggle website. These data were processed using R Studio software. The dataset was made up of loan approval data, which was a compilation of financial records and associated data used to assess a person’s or an organization's eligibility for loans from lending institutions. There were 4,269 records in the dataset, including 12 explanatory factors and 1 response variable. The variables used in this study are shown in Table 1.

Table 1. List of Variables Used

No	Variable	Description	Information
1.	Loan Id	Loan unique code	Integer
2.	No Of Dependents	Number of dependents of the applicant	Integer
3.	Education	Education of the applicant	Graduate Not Graduate
4.	Self Employed	Employment status of the applicant	No Yes
5.	Income Annum	Applicant’s annual income	Integer
6.	Loan Amount	Applicant’s loan amount	Integer
7.	Loan Term	Loan term (Year)	Integer
8.	Cibil Score	Credit score	Integer

No	Variable	Description	Information
9.	Residential Assets Value	Residential asset value	Integer
10.	Commercial Assets Value	Commercial asset value	Integer
11.	Luxury Assets Value	Luxury asset value	Integer
12.	Bank Assets Value	Bank asset value	Integer
13.	Loan Status	Loan approval status	Approved Rejected

2.2. Method of Analysis

The data analysis procedure that was carried out in this study consists of several stages.

a. Collecting loan approval data obtained from the Kaggle website.
b. Exploring the data to see an overview of the data. At this stage, some of the objectives were checking duplicate data and missing data, checking the proportion of response variables, knowing the descriptive statistics of each variable, knowing the relationship between explanatory variables and responses, calculating the correlation between explanatory variables.
c. Preprocessing the data. At this stage, some data preprocessing was carried out including:
   1) Handling outliers using the Inter Quartile Range (IQR) method.
   2) Performing data normalization to change the data scale so that each variable in the dataset has the same range of values.
   3) Convert category variables into binary form using one-hot encoding.
   4) Handling imbalanced data using oversampling method, namely SMOTE.
d. Dividing the data into two parts, namely 80% training data and 20% test data.
e. Next, define the random forest and XGBoost models with scenarios without handling imbalanced data and with handling imbalanced data.
f. Analyzing the accuracy, sensitivity, specificity, and F1-score of each developed model to determine how well it performs in terms of categorization.
g. Selecting the best model using 10-fold cross validation.
h. After obtaining the best model, it can be seen which explanatory variables contribute most to the response variable using the feature importance of all explanatory variables.
i. Interpretation of results and conclusions.

2.3. Ensemble Learning

Ensemble models synthesize results from different learning algorithms to obtain better results than individual algorithms. These models improve predictions and reduce variance and bias if used correctly [10]. Bagging and boosting are the most well-known ensemble classifier techniques. When performing bagging, the original training set is divided into N subsets of equal size, and each subset is used to create a classifier. By combining several specific classifiers, an overall classification model is built. In contrast, the boosting algorithm creates a poor model, and after a few uses, the boosting algorithm combines these poor learners into a prediction model that will be much more accurate than the poor learners [11].

2.4. Random Forest

This method is well-suited for regression and classification tasks because of its broad application across multiple domains and robust results. Based on statistical learning theory, the random forest algorithm uses the bootstrap resampling method to create additional samples from the original dataset. Every bootstrap sample is converted into a decision tree. The outputs of these decision trees are aggregated, usually by average, to get the final prediction. The technique increases the diversity of the decision trees by utilizing resampled data and arbitrarily altering the set of predictors while building multiple trees [6].

Random forest creates predictive values by combining predictions from individual trees through a process called regression. This puts an end to mismatches [12]. For example, a random forest model can be described as (1).

$$g(x) = f_0(x) + f_1(x) + f_2(x) + \cdots + f_n(x) \tag{1}$$

where every model $f(x)$ is a decision tree and g is the final model, which is the sum of all models.

2.5. Extreme Gradient Boosting (XGBoost)

XGBoost is a cutting-edge ensemble model based on decision trees, utilizing boosting techniques for weak learners. It is designed for superior performance and speed compared to other tree-based models. Key advantages of the XGBoost method include regularization to prevent overfitting, built-in cross-validation capabilities, efficient handling of missing data, awareness of data capture, parallel tree building, and effective tree pruning [13].

To control overfitting, the objective function (minimization) in XGBoost consists of two parts: the loss function and the regularization term [11]. The objective function (minimization) manages the complexity of the model, as illustrated in (2).

$$Obj = \sum_{i=1}^{n} l(y_1, f_i) + \sum_{m=1}^{M} \Omega(f^m) \tag{2}$$

where $\Omega(f^m)$ is the regularization term. In XGBoost, a second-order approach is used to optimize the objective function. Consequently, at each iteration, the best tree is selected using (3).

$$Obj = -\frac{1}{2}\sum_{j=1}^{T} \frac{G_j^2}{H_j+\lambda} + \gamma T \tag{3}$$

Here, T represents the number of leaf nodes, while G_j and H_j denote the statistical sums of the first- and second-order gradients of the loss function for the samples in the jth leaf node, respectively. The terms λ and γ are regularization coefficients. Consequently, this method allows for the tree complexity to be chosen and controlled independently in each iteration, meaning the number of leaves can vary between iterations. Once the optimal tree is selected in an iteration, the values of the leaf nodes are computed using the gradient statistics for each leaf, as illustrated in the given (4).

$$w_j^* = \frac{G_j^2}{H_j+\lambda} \tag{4}$$

2.6. Synthetic Minority Over-sampling Technique (SMOTE)

The SMOTE algorithm uses an oversampling technique to restore balance to the initial training set. SMOTE adds synthetic examples instead of just copying the minority class examples. Interpolating between many minority class samples within a given neighborhood yields this additional data. As a result, the process concentrates on the "feature space" as opposed to the "data space," meaning that the algorithm is built on feature values and their correlations rather than taking into account all of the data points. This method requires a thorough examination of the dimensions of the data as well as the theoretical link between the original and synthetic cases [14].

2.7. Model Evaluation

Model evaluation is useful for assessing the model's ability to predict or classify. It allows us to measure model performance and select the best model. Accuracy, sensitivity, specificity, and F1-score are used to evaluate the performance of ensemble learning models [15]. The confusion matrix for model evaluation is shown in Table 2.

Table 2. Confusion Matrix

		Actual Class	
		True (1)	***False (0)***
Predicted Class	Positive (1)	True Positive (TP)	False Positive (FP)
	Negative (0)	False Negative (FN)	True Negative (TN)

2.7.1. Accuracy

Accuracy is calculated by dividing the sum of true positive (TP) and true negative (TN) predictions by the total number of data points (P + N). The ideal accuracy is 1, while the poorest accuracy is 0.

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP} \qquad (5)$$

2.7.2. Sensitivity

Sensitivity is calculated by dividing the number of true positive predictions (TP) by the total number of actual positive cases (P). The maximum possible TP rate is 1, while the minimum is 0.

$$Sensitivity = \frac{TP}{TP+FN} \qquad (6)$$

2.7.3. Specificity

Specificity is determined by dividing the number of true negative predictions (TN) by the total number of actual negative cases (N). The highest possible specificity is 1, and the lowest is 0.

$$Spesificity = \frac{TN}{TN+FN} \qquad (7)$$

2.7.4. F1-Score

The F1-score, which measures the accuracy of the test, is calculated using (8) based on precision and recall:

$$F1 - Score = \frac{2 \times precision \times recall}{precision \times recall} \qquad (8)$$

3. Results and Discussion

3.1. Data Exploration

The data were examined so that they could be applied efficiently to machine learning. The response variable in this study was a variable containing two classes, namely approved and rejected loan status. Fig. 1 shows the distribution of approved and rejected classes for the response variable.

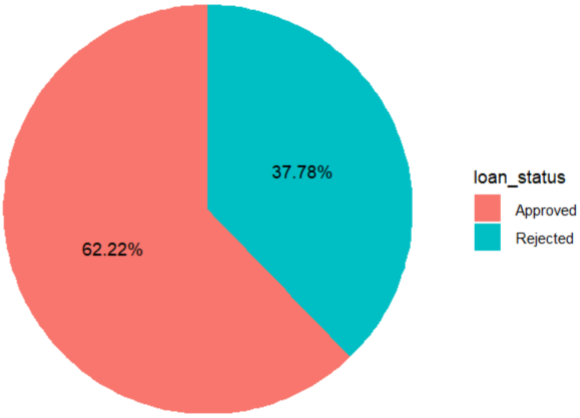

Fig. 1 Pie chart of response variable distribution.

Fig. 1 shows that the response variables are highly imbalanced, with the approved class having 62.22% observations and the rejected class having only 37.78% observations. Machine learning algorithms do not work efficiently for imbalanced data. Therefore, it is necessary to handle imbalanced data. Next, look at the relationship between the explanatory variables and the response. The explanatory variables used were numerical explanatory variables. For this exploration, the boxplot visualization presented in Fig. 2 was used.

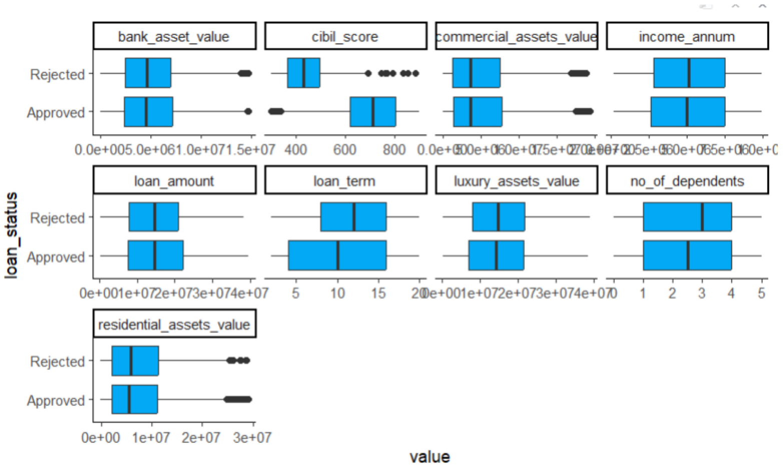

Fig. 2 Boxplot of the relationship between explanatory variables and response.

Based on Fig. 2, it can be seen that some variables have quite a lot of outliers. Therefore, it is necessary to handle outliers. Based on the boxplot, it can also be seen that there is scale non-uniformity in each variable, needing to be handled at the data preprocessing stage. Based on the boxplot, it can be seen graphically that there is a difference in average and distribution between the approved and rejected groups for each explanatory variable.

Checking how the correlation between explanatory variables to be used in the model was done using the Spearman correlation coefficient. The variables used should be those that were not correlated with each other. The visualization of the correlation results is shown in Fig. 3.

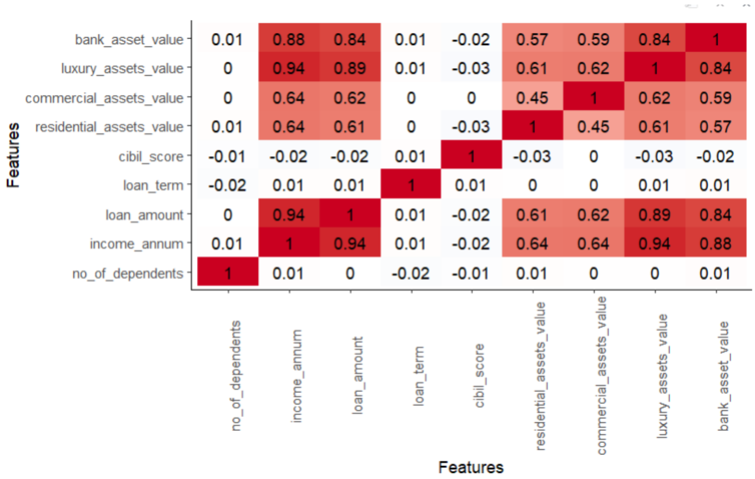

Fig. 3 Heatmap of the relationship between variables.

Based on Fig. 3, some explanatory variables had correlation values above 0.8, indicating that there was multicollinearity between them. However, the high correlation is not overcome because ensemble methods such as random forest and XGBoost, which are more resistant to multicollinearity, can help reduce the impact of high correlation.

3.2. Data Preprocessing

The data were first preprocessed before implementing the classification process with ensemble learning. The preprocessing carried out on the data in this analysis overcame the outliers in some variables. The Interquartile Range (IQR) method was used to resolve this. Data treated with outliers is shown in the visualization in Fig. 4.

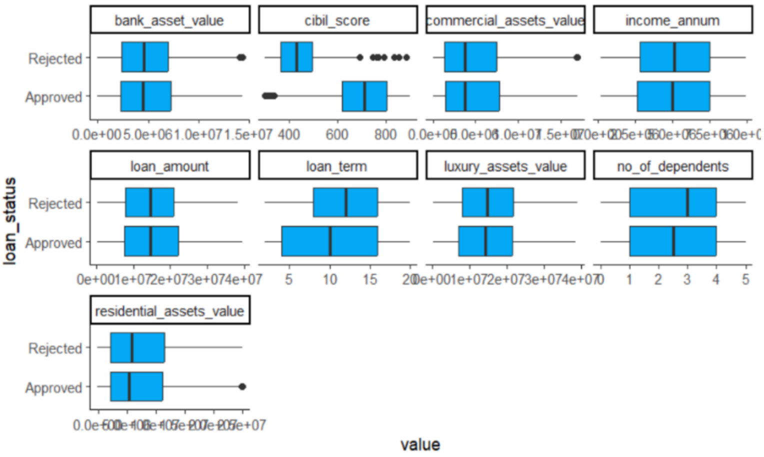

Fig. 4 Boxplot after outlier handling.

The IQR is a widely used method for measuring data dispersion, especially in cases of non-normal distributions. IQR is calculated by taking the difference between the third quartile (Q3) and the first quartile (Q1), representing the middle 50% of the data [16]. In outlier detection, values below $Q1 - 1.5 * IQR$ or above $Q3 + 1.5 * IQR$ are considered outliers. IQR is popular due to its robustness against outliers, unlike standard deviation, which is sensitive to the influence of extreme values. The results of research conducted [17] has shown that the approach to detecting outliers with the proposed IQR has better accuracy than other competitive approaches. Based on Fig. 4, after the IQR method was performed, there was a decrease in the number of outliers so that these results were used in further analysis.

Imbalanced response variables will avoid bias in one of the classes, so imbalanced data are handled using SMOTE. Fig. 5 shows the distribution of response variables after handling imbalanced classes. The SMOTE algorithm rebalances the initial training set through an oversampling approach. Unlike merely duplicating minority class examples, SMOTE generates synthetic examples. These new data are generated by interpolating between different instances of the minority class within the existing dataset [14].

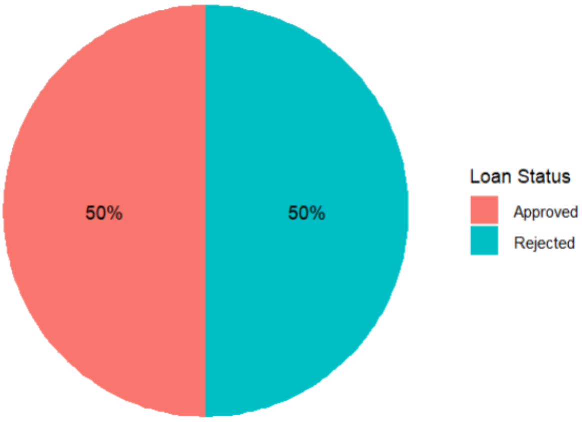

Fig. 5 Pie chart after imbalanced data handling.

3.3. Modeling with Random Forest and XGBoost

This research used random forest and XGBoost methods for classification. The classification technique results were compared with two ways of handling data, namely, without handling imbalanced data and with SMOTE. The 10-fold cross-validation method was employed as an evaluation metric to mitigate the bias linked with random sampling. This technique minimized bias by dividing the loan approval data into ten roughly equal parts. Using the R application, the accuracy, sensitivity, specificity, and F1-score for the loan approval data were obtained with the random forest and XGBoost models, as shown in Table 3 and Table 4.

Table 3. Treatment Comparison with Random Forest Model

Treatment	Accuracy (%)	Sensitivity (%)	Specificity (%)	F1-Score (%)
Without handling imbalanced data	98.10	98.90	96.77	98.48
With handling imbalanced data	97.72	98.04	97.21	98.17

Based on Table 3, the treatment given to the random forest model shows a very high level of accuracy, where the model without unbalanced data handling (98.10%) is slightly higher than the model with handling (97.72%). In addition, the model without imbalanced data handling was also slightly superior in sensitivity and F1-score. It is different in specificity, where the model with imbalanced data handling was slightly higher than the model without handling. In the context of loan approval classification using ensemble learning on imbalanced data, the interpretation of the results showed several trade-offs. First, the model without imbalanced data handling had a slight edge in sensitivity (98.90%) and F1-score (98.48%), indicating that it was better at detecting loan applications deserving approval. Second, the model with imbalanced data handling had a higher specificity (97.21%), indicating that it was better at detecting ineligible loan applications.

Table 4. Treatment Comparison with XGBoost Model

Treatment	Accuracy (%)	Sensitivity (%)	Specificity (%)	F1-Score (%)
Without handling imbalanced data	98.47	99.24	97.21	98.78
With handling imbalanced data	98.52	98.83	98.01	98.81

Based on Table 4, both XGBoost model treatments show a very high level of accuracy, where the model with unbalanced data handling (98.52%) is slightly higher than the model without handling (98.47%). In addition, the model with imbalanced data handling was also slightly superior in specificity and F1-score. It is different in sensitivity, where the model without imbalanced data handling was slightly higher than the model with handling. This showed that the model without imbalanced data handling had a slight advantage in sensitivity (99.24%), meaning that it was better at detecting loan applications that deserved approval. Meanwhile, the model with handling had an advantage in specificity (98.01%) and was slightly better in F1-score (98.81%). It was better at detecting ineligible loan applications and provided a better balance between specificity and sensitivity.

3.4. Model Comparison

The assessment of each ensemble learning classifier employed in this investigation will be given in the next part. The model with the best accurate predictions was determined by comparing the outcomes. A comparison of the accuracy, sensitivity, specificity, and F1-score values of each evaluated model is shown in the visualization in Fig. 6.

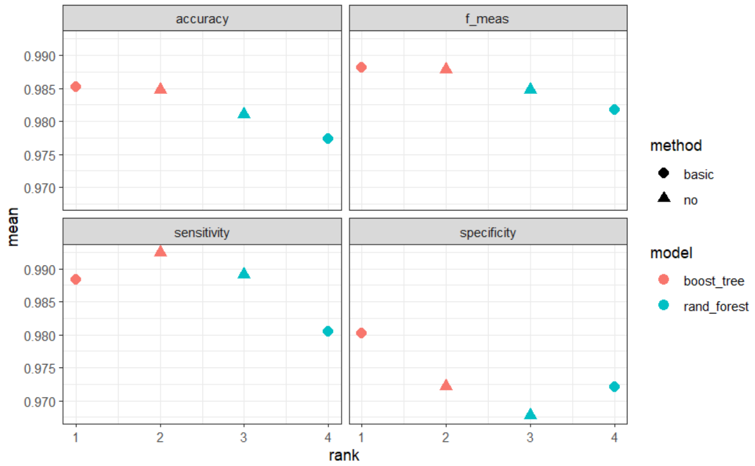

Fig. 6 Comparison of model classification performance.

Based on Table 3 and Table 4 and the visualization in Fig. 6, it can be concluded that all models have good performance in classifying loan approval data. Applying SMOTE to handle imbalanced data yielded an accuracy of 98.52%, making XGBoost the most accurate model. Random forest without handling, random forest with handling, and XGBoost without handling come next. The same ranking also occurred in F1-score performance. However, the sensitivity showed that the XGBoost model without imbalanced data handling yielded the highest value of 99.24%, meaning that this model increased the ability to classify loan applications that deserved approval. At the same time, the specificity showed that the XGBoost model with imbalanced data handling provided the highest value of 98.01%, meaning that this model increased the ability to classify loan applications that were not feasible, so they must be rejected.

3.5. Important Variables

The best model obtained previously was used to generate the important variables. The variable importance calculated for each feature indicated the contribution of each independent variable to predicting the outcome. Fig. 7 shows the variable importance, starting with the variable with the largest influence and ending with the variable with the smallest influence [18].

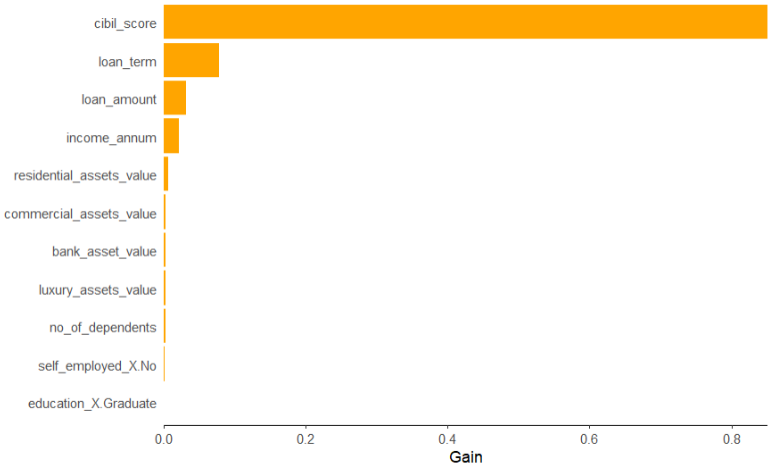

Fig. 7 Important variable.

According to the results on Fig. 7, the variable with the highest level of importance is the credit score variable, followed by the loan term, loan amount, and annual income. The variables with the lowest importance were employment status and education. These variables did not contribute much to the XGBoost classification modeling. The probability of getting a loan increased with a higher credit score, indicating a better loan history. Since shorter loan terms reduce the risk of repayment uncertainty in the long run, shorter loan terms are preferred. The decision is also affected by the loan amount requested, as larger loan amounts pose a higher risk for the lender. Therefore, banks may concentrate on customers with good credit scores, grant shorter loan terms, and apply loan scoring.

4. Conclusion

The classification modeling analysis with random forest and XGBoost on loan approval data showed that the XGBoost model with imbalanced data handling using SMOTE achieved the highest accuracy of 98.52%, with a good balance between sensitivity of 98.83% and specificity of 98.01%. The model without imbalanced data handling also performed well, with an accuracy of 98.47%, sensitivity of 99.24%, and specificity of 97.21%. However, there was no significant difference in overall accuracy. However, handling imbalanced data with SMOTE improved the model's ability to detect loan applications that should be rejected, as indicated by the higher specificity. This suggests that handling imbalanced data can be more beneficial in cases where the ability to detect unqualified loan applications is crucial.

The most important variables in determining loan approval are credit score, loan term, loan amount, and annual income. At the same time, education and employment status have less influence. These results emphasize that handling imbalanced data and utilizing key features in loan decision-making are critical to improving the accuracy and effectiveness of loan assessment systems.

References

[1] M.C. Aniceto, F. Barboza, and H. Kimura, "Machine Learning Predictivity Applied to Consumer Creditworthiness," *Future Business Journal*, Vol. 6, No. 1, pp. 1–14, 2020, doi: 10.1186/s43093-020-00041-w.

[2] A.A. Ibrahim, R.L. Ridwan, M.M. Muhammed, R.O. Abdulaziz, and G.A. Saheed, "Comparison of the CatBoost Classifier with other Machine Learning Methods," *International Journal of Advanced Computer Science and Applications*, Vol. 11, No. 11, pp. 738–748, 2020, doi: 10.14569/IJACSA.2020.0111190.

[3] K. Gupta, B. Chakrabarti, A.A. Ansari, S.S. Rautaray, and M. Pandey, "Loanification - Loan Approval Classification using Machine Learning Algorithms," in *Proceedings of the International Conference on Innovative Computing & Communication (ICICC) 2021*, 2021, pp. 1–4, doi: 10.2139/ssrn.3833303.

[4] M. Hanafy and R. Ming, "Machine Learning Approaches for Auto Insurance Big Data," *Risks*, Vol. 9, No. 2, pp. 1–23, 2021, doi: 10.3390/risks9020042.

[5] K.A. Nguyen, W. Chen, and B. Lin, "Comparison of Ensemble Machine Learning Methods for Soil Erosion Pin Measurements," *SPRS International Journal of Geo-Information*, Vol. 10, No. 1, pp. 1–17, 2021, doi: 10.3390/ijgi10010042.

[6] W. Zhang, C. Wu, H. Zhong, Y. Li, and L. Wang, "Prediction of Undrained Shear Strength Using Extreme Gradient Boosting And Random Forest Based On Bayesian Optimization," *Geoscience Frontiers*, Vol. 12, No. 1, pp. 469–477, 2021, doi: 10.1016/j.gsf.2020.03.007.

[7] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2026, pp. 785–794, doi: 10.1145/2939672.2939785.

[8] J.M.A.S. Dachi and P. Sitompul, "Comparative Analysis of XGBoost Algorithm and Random Forest Ensemble Learning Algorithm on Credit Decision Classification," *Jurnal Riset Rumpun Matematika dan Ilmu Pengetahuan Alam*, Vol. 2, No. 2, pp. 87–103, 2023, doi: 10.55606/jurrimipa.v2i2.1470.

[9] N.V. Chawla, "Data Mining for Imbalanced Datasets: An Overview," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. Boston, MA, USA: Springer, 2009, pp. 875–886.

[10] A. Dinh, S. Miertschin, A. Young, and S.D. Mohanty, "A Data-Driven Approach to Predicting Diabetes and Cardiovascular Disease With Machine Learning," *BMC Medical Informatics and Decision Making.*, Vol. 19, No. 1, pp. 1–15, 2019, doi: 10.1186/s12911-019-0918-5.

[11] J. Tanha, Y. Abdi, N. Samadi, N. Razzaghi, and M. Asadpour, “Boosting Methods for Multi-Class Imbalanced Data Classification: An Experimental Review,” *Journal of Big Data*, Vol. 7, No. 1, 2020, Art. No. 70, doi: 10.1186/s40537-020-00349-y.

[12] M. Kayri, I. Kayri and M. T. Gencoglu, “The Performance Comparison of Multiple Linear Regression, Random Forest and Artificial Neural Network by Using Photovoltaic and Atmospheric Data,” in *14th International Conference on Engineering of Modern Electric Systems (EMES)*, 2017, pp. 1–4, doi: 10.1109/EMES.2017.7980368.

[13] M. Nabipour, P. Nayyeri, H. Jabani, S. Shahab, and A. Mosavi, “Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; A Comparative Analysis,” *IEEE Access*, Vol. 8, pp. 150199–150212, 2020, doi: 10.1109/ACCESS.2020.3015966.

[14] A. Fernández, S. García, F. Herrera, and N.V. Chawla, “SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary,” *Journal of Artificial Intelligence Research*, Vol. 61, pp. 863–905, 2018, doi: 10.1613/jair.1.11192.

[15] Ž. Đ. Vujović, “Classification Model Evaluation Metrics,” *International Journal of Advanced Computer Science and Applications*, Vol. 12, No. 6, pp. 599–606, 2021, doi: 10.14569/IJACSA.2021.0120670.

[16] L. Sullivan and W.W. LaMorte, “InterQuartile Range (IQR).” Accessed: 5 September 2024. [Online]. Available: https://sphweb.bumc.bu.edu/otlt/mphmodules/bs/bs704_summarizingdata/bs704_summarizingdata7.html

[17] C.S.K. Dash, A.K. Behera, S. Dehuri, and A. Ghosh, “An Outliers Detection and Elimination Framework in Classification Task of Data Mining,” *Decision Analytics Journal*, Vol. 6, No. January, 2023, doi: 10.1016/j.dajour.2023.100164.

[18] M. Kuhn and K. Johnson, *Applied Predictive Modeling*. New York, NY, USA: Springer, 2013.