Bayesian Additive Regression Trees for Classification of Unbalanced Class of Credit Collectability Data

ABSTRACT

Aims: This study aims at determining the classification results using the Bayesian Additive Regression Trees (BART) method on bank credit collectability data, where there is a class imbalance in the data.
Study design: Quantitative Design.
Place and Duration of Study: Sample: The used data are secondary data in the form of bank debtor's credit collectability data with nine predictor variables and one response variable in the form of credit collectability. They are collected from Banks in East Java, Indonesia, from the date of 01 May 1986 to 31 May 2018.
Methodology: The Bayesian approach is one of the estimation methods in statistics that is currently being popularly used, this is because the rapid development of technology makes computational challenges no longer a problem. The Bayesian estimation continues to develop and can be used in various statistical methods, for instance both for regression and classification. The Classification and Regression Trees (CART) method is one of the most popular classification methods used. Debtors, in a bank, who have delinquent credit have a small proportion compared to debtors who have current credit. Standard classifier methods such as CART are not suitable for handling this case, because CART is sensitive to classes that have a high degree. Hence, additional methods such as ensemble BART (Bayesian Additive Regression Trees), are needed in order to increase the accuracy of classification in cases of class imbalance.
Results: The results of the cross-validation on the BART show a high consistency of classification accuracy, 83.49%. This indicates that the BART method can work consistently even though there is a class imbalance. The results of this study indicate that the classification accuracy of the training data is 84.53%, while the accuracy in the testing data is 85.48%. These results also show that the BART method has ability to overcome overfitting in the classification method, where overfitting often occurs in most of the classification methods that have very good classification abilities.
Conclusion: The testing data show that the accuracy is relatively similar to the one of the training data, this indicates that the BART method has been able to capture patterns in the data.

Keywords: BART, CART, Ensemble, Class Imbalance, Credit Collectability

1. INTRODUCTION

Provide a factual background, Rapid technological developments make computational challenges no longer a problem. This has an impact on the widespread use of Bayesian methods for estimation in several current statistical models. The conclusion from the Bayesian estimation method that is clear and direct (utilizing all available information) makes this method fundamentally very powerful and flexible to use compared to the classical or likelihood approach [1]. "Although the Bayesian approach requires determining the prior

distribution (viewed as a Bayesian weakness), on the other hand, if it is precise in determining the prior distribution, this method will be much stronger than the classical approach. The Bayesian approach has now been developed and can be used in various statistical methods, both for regression and classification. Classification and Regression Trees (CART) is a supervised learning classification method which is a branch of the decision tree method. The advantage of this classification method is that it can be used for predictor variables on a categorical or continuous scale and it is also capable of handling very large data [2], hence in practice this method is so popular".

"In the real world, the main problem that poses a challenge in classification methods is class imbalance, which has attracted the attention of academicians and researchers in recent years" [3]. "Class imbalance occurs if there is an unequal number between classes contained in a data set (unbalanced data distribution)" [4]. "Class imbalance can be defined as well as a condition in a data set where there are classes that have a large size while other classes are only represented by a few objects" [5]. "The standard classification methods generally have poor performance in cases of class imbalance, as the methods pays little attention to minority classes in unbalanced data sets" [5]. Therefore, the classification rules that predict the minority class tend to be weaker than the rules that predict the majority class. As a result, the minority class is more frequently misclassified than the majority class.

Banking is one of the financial institutions that have an important role in the economy of a country. Banks act as financial intermediaries and are also the center of the economy in a country. One of the main functions of a bank is to properly mobilize public savings funds. The funds received by the bank from the community will be channeled back to the community in need in the form of credit. Credit is the main factor that is most dominant in bank income. Furthermore, banks need to be careful in managing their credit services, so that credit risk can be controlled. The bank is also one of the service providers in channeling home ownership loans. With increasing population growth, this will also have an impact on increasing the desire of the public or consumers to own a home. Increased consumer desire to own a home can provide benefits for the bank. However, the increase in consumer desire to own a home also poses a risk to banks. This risk arises if the credit provided by the bank is not returned on time. In this study, a classification of bank credit collectability was carried out. This is necessary and important to do so that in the future the bank can predict prospective debtors who have the potential to have delinquent loans.

"The main problem that occurs in this study is bank credit collectability data, where debtors who are categorized as current class and delinquent class have an unequal class comparison. Delinquent class debtors have a very small proportion compared to current class debtors. This will potentially make the results of classification using CART have inaccurate predictions. Decision trees have a weakness in class imbalance because the sorting criteria in decision trees use the Gini Index, that is sensitive to classes with a high degree" [6]. Therefore, we need a method to overcome the class imbalance.

"The ensemble method is an effective method for increasing classification accuracy in class imbalance cases" [5]. One of the ensemble methods that recently emerged is BART. The ensemble BART (Bayesian Additive Regression Trees) method can improve the performance of a classification method in class imbalance cases by giving more weights to misclassified objects (misclassified objects are often in a minority class). This method also makes it possible to build flexible empirical models to help exploring the data. Modeling with BART uses Bayesian estimation techniques to compute and positively aggregate the regression tree.

Previous research [7] discussed the use of ensemble boosting CART on unbalanced bank credit collectability data. In that study, the obtained results were that the performance of the ensemble boosting CART had very good results in classifying unbalanced data. However, in that study the ensemble boosting CART method had a weakness although the training data had very good classification results of up to 98% while using testing data produced 84% accuracy. The results of this accuracy are indeed good but also show that there is overfitting in the boosting CART method, shown in quite large difference in accuracy between the training and testing data. Hence, in this study we want to know whether class imbalance can be overcome by the Bayesian approach, more specifically Bayesian Additive Regression Trees (BART).

2. METHODOLOGY

2.1 Classification and Regression Trees (CART)

"A classification method known as Classification and Regression Trees (CART) was introduced in 1984 by Breiman, Friedman, Olshen and Stone" [8]. In CART there are two important steps that must be taken to get a tree with optimal performance, namely the first is sorting objects repeatedly based on certain attributes, the second is pruning. The pruning step is carried out to avoid overfitting, which is a condition where the results of classification accuracy are good for training data but poor for testing data. Pruning works by turning a node into a final node (terminal node), meaning that no more node sorting is done after that. So that a classification tree will be formed with fewer node sizes.

2.1.1 Splitting.

"Formation of a classification tree using training data. The training data will be disaggregated using the measure of impurity level $i(t)$, which is a measure of the degree of heterogeneity of a particular node in the classification tree. There are several measures of impurity that are popularly used, namely Information Gain, Gain Ratio, Gini Index and Entropy" [9]. "The measure of impurity that is often used is the Gini index, as it is easy to apply" [8]. In the following, the heterogeneity function of the Gini index is presented

$$i(t) = 1 - \sum_j p^2(j \mid t) \tag{1}$$

where $i(t)$: Gini impurity/Gini index of node t; $p(j|t)$: probability of an object being classified to a particular node or class t.

Splitting begins by looking for all possible splitting on all predictor variables. Goodness of split is an evaluation of splitting by classifier s at node t which is defined as a decrease in heterogeneity [8]. The heterogeneity measure of Goodness of split is the reduction of the gini index value at the nodes before splitting and after splitting into the left node t_L and the right node t_R. The Goodness of split function is presented in the following equation.

$$\begin{aligned} \phi(s,t) = \Delta i(s,t) &= i(t) - p_L i(t_L) - p_R i(t_R) \\ &= 2 p_L p_R \sum_{j=1} | p(j \mid t_L) - p(j \mid t_R) | \end{aligned} \tag{2}$$

Where $\phi(s,t) = \Delta i(s,t)$: The goodness of split heterogeneity measure on splitting objects s and t nodes; where s is a candidate split at node t; $i(t)$: Gini index value at the t^{th} node; p_L: Probability of splitting to left node; p_R: Probability of splitting to right node; $i(t_L)$: Gini index value at left node t_L; $i(t_R)$: Gini index value at right node t_R; t_L and t_R are the left and the right children of note t using split s; p_L and p_R are the proportions of records in node t t_L and t_R, respectively; $p(j|t_L)$ and $p(j|t_R)$ are the proportions of class j records in t_L and t_R respectively.

Formation of a tree structure is done by looking for all possible splitting at each node starting from the main node. Then look for a disaggregation *s** that is capable of producing the highest heterogeneity value among all disaggregations *s* presented in the form of the equation below.

$$\Delta i(s^*,t) = \max_{sCS} \Delta i(s,t) \quad (3)$$

Where $\Delta i(s^*,t)$: The goodness of split heterogeneity measure on the object of selecting *s** and *t* nodes which is capable of producing the highest heterogeneity.

2.1.2 Class Assignment

Class label marking is done to determine the most dominant class of a node. This is done to find out the characteristics of the observation classification for each node that is formed. Class labeling for each node is created based on the proportion of the largest class. The proportion of the largest class shows the class that dominates in that class. The following is the probability function of class marking which is presented in the below equation.

$$p(j_0,t) = \max_j p(j,t) = \max_j \frac{N_j(t)}{N(t)} \quad (4)$$

Where $p(j_0,t)$: the proportion of class j_0 node t which is the highest proportion; $p(j,t)$: proportion in class j node t; $N_j(t)$: the number of observations at the j^{th} class t node; $N(t)$: number of observations at node t.

2.1.3 Tree Pruning Process

"A classification tree that has been designed from the splitting process will produce a tree with a very large size structure (having many endpoints), this tree is commonly called the Maximal tree (Tmax). The tree pruning steps are needed such that the overfitting phenomenon does not occur. Overfitting is a condition where the tree's ability to classify training data is very good, but it is very bad at classifying new data (testing data), this occurs since too much splitting is done by the node, so the node needs to be trimmed" [9]. After pruning is done, an optimal classification tree will be formed. There is a relative value that can be used to assist in selecting the optimal classification tree, namely the relative error. It is a value obtained from the division between the resubtition estimate and the classification error at the main node (root node error). Resubtition estimate is a misclassification obtained from a classifier (in this case a classification tree). The following is the probability function of the resubtition estimate.

$$R(T_t) = \frac{1}{N} \sum_{i=1}^{N} I(T_t(x_n) \neq y_n) \quad (5)$$

Where $R(T_t)$: resubstitution estimate (proportion of errors in subtrees) in tree T node t; $I(T_t(x_n) \neq y_n)$:Gini index heterogeneity function. The relative error value can be defined into an equation which will be presented below.

$$R\,e(T_t) = \frac{R(T_t)}{R(T_1)} \quad (6)$$

Where $R\,e(T_t)$: resubstitution estimate of a tree T node t at complexity e; $R(T_t)$: resubstitution estimate (proportion of errors in subtrees) in tree T node t; $R(T_1)$: resubstituting estimate (proportion of errors in sub-trees) in the initial node T tree.

Pruning works by turning child nodes into terminal nodes, such that no further splitting is done after that. Thus the size of trees will be reduced. The pruning process will result in a

trade-off between relative error validations (the used validation is *K*-fold cross-validation). The number of terminal nodes formed on a pruned tree will produce a tree that can capture the real pattern, in other words not prone to overfitting [9].

2.2 ENSEMBLE METHOD

"The standard classification method generally has poor performance in cases of class imbalance because this classification method pays little attention to minority classes in unbalanced data sets" [5]. "Therefore, the classification rules that predict the minority class tend to be weaker than the rules that predict the majority class. As a result, the minority class is more frequently misclassified than the majority class. Decision trees have a weakness in class imbalance because the splitting criteria in decision trees using the Gini Index are sensitive to classes that have a high degree" [6].

"The basic principle of ensemble method is to develop a set of models from training data. Then, combining a set of models to determine the final classification. The final classification is based on the largest pool of votes from a combined set of models. The most frequently used examples of ensemble methods are Bagging and Boosting" [2].

2.3 BAYESIAN ADDITIVE REGRESSION TREES (BART)

Bayesian Additive Regression Trees (BART) was first introduced by Chipman, George, and McCulloch [10]. It is a Bayesian approach used for estimating nonparametric functions using regression trees. Regression trees rely on recursive binary partitioning of the predictor space into a set of hyperrectangles to estimate some unknown function *f*. The predictor space has the same dimensions as the number of variables. Tree-based regression models have the ability to flexibly adjust for interactions and nonlinearities. Models consisting of multiple regression trees have an even greater ability than single trees to capture interactions and non-linearity as well as additive effects in unknown function *f*. BART can be thought of as an ensemble of tree counts, with a novel estimation approach that relies entirely on Bayesian probability models. Specifically, the BART model can be expressed as:

$$Y = f(X) + \varepsilon \approx T_1^M(X) + T_2^M(X) + \ldots + T_m^M(X) + \varepsilon$$
$$\varepsilon \sim N_n(0, \sigma^2 I_n) \tag{7}$$

The prior of the BART method has three components, namely: the tree structure itself, the leaf parameters given by the tree structure, and the error variance σ^2 or independent errors of the tree structure and leaf parameters. The following gives the prior function of BART:

$$\begin{aligned} P\left(T_1^M, \ldots, T_m^M, \sigma^2\right) &= \left[\prod_t P\left(T_m^M\right)\right] P\left(\sigma^2\right) = \left[\prod_t P\left(M_t | T_t\right) P\left(T_t\right)\right] P\left(\sigma^2\right) \\ &= \left[\prod_t \prod_l P\left(\mu_{t,l} | T_t\right) P\left(T_t\right)\right] P\left(\sigma^2\right) \end{aligned} \tag{8}$$

where in the last equation follows the additional assumption of conditional independence from the leaf parameter considering it is a tree structure. Where, $P(T_t)$ is the previous component that affects the location of the nodes in the tree. The node depth is defined as the distance from the root. So, root itself has a depth of 0, its first child node has a depth of 1, the second child node, and so forth. Nodes at the d^{th} depth are nonterminal with prior probabilities of $\alpha\,(1 + d)^{-\beta}$ where $\alpha \in (0, 1)$ and $\beta \in [0, \infty]$. The former tree structure component had the ability to enforce shallow tree structures, thereby limiting the complexity of each single tree and resulting in more regularization models. The default values for the

hyperparameters α = 0.95 and β = 2 (using the Beta distribution) are recommended by Chipman, George, and McCulloch [10].

The final prior is on the error variance and is chosen to be $\sigma^2 \sim Inv.Gamma\left(\nu/2, \nu\lambda/2\right)$, statistic λ is determined from the data such that there is a q = 90% a priori chance (by default) that the BART model will improve at Root Mean Square Error (RMSE) from ordinary least squares regression. Therefore, the majority of the prior probability mass lies below the RMSE of the least squares regression. Additionally, it limits the probability mass to be placed at small values σ^2 to prevent overfitting. Thus, the higher the q value, the larger the sample value of σ^2, and resulting in more regularization models.

The Metropolis-in-Gibbs sampler was used to generate the retrieving from the posterior distribution $P\left(T_1^M,...,T_m^M,\sigma^2|y\right)$. The main feature of this sampler for BART is that it uses a form of "Bayesian backfitting" [11] in which the j trees that fit iteratively hold all other m - *1* trees constant by exposing only the remaining responses that still do not fit, where partial residual defined as:

$$
\begin{aligned}
&R_{-j} := y - \sum_{t \neq j} T_t^M\left(X\right) \\
&1: T_1 \left| R_{-1}, \sigma^2 \right. \\
&2: M_1 \left| T_1, R_{-1}, \sigma^2 \right. \\
&3: T_2 \left| R_{-2}, \sigma^2 \right. \\
&4: M_2 \left| T_2, R_{-2}, \sigma^2 \right. \\
&... \\
&2m-1: T_m \left| R_{-m}, \sigma^2 \right. \\
&2m: M_m \left| T_m, R_{-m}, \sigma^2 \right. \\
&2m+1: \sigma^2 \left| T_1, M_1, ..., T_m, M_m, \varepsilon \right.
\end{aligned}
\qquad (9)
$$

The next process is changes to the structure of the first T tree that are accepted or rejected via the Metropolis-Hastings step. The sampling of the posterior tree structure is independent of the leaf parameters, as they can be integrated analytically from the calculations. With an existing tree structure, samples from the posterior leaf parameter, b M_1: = $\{\mu_1, \ldots, \mu_b\}$ is drawn. This procedure expands iteratively for each tree, using an updated set of partial residuals R_{-j}. Finally, depending on the updated tree structure set and leaf parameters, the retrieval from the posterior σ^2 is made based on the full model residuals, $\varepsilon := y - \sum_{t=1}^{m} T_t^M\left(X\right)$. Row *1; 3,. . . , 2m − 1* above relying on the Metropolis-Hastings drawing from the posterior distribution tree. This involves introducing minor perturbations to the tree structure, involving the growing of a terminal node by adding two child nodes, pruning two child nodes (rendering the terminal of its parent node), or changing division rules. This step is shown with three possible tree changes, such as growing, pruning, and changing.

2.4 VALIDATION

Validation is one of the most important techniques for the stability of the learning model regarding how well the model will generalize to new data. The popular validation method used is cross-validation. It is a statistical technique that partitioning data into subsets to evaluate a learning model. The following will give the function of the 10-fold cross-validation (CV) presented.

$$CV(T_t) = \frac{1}{k}\sum_{k=1}^{K} \mathrm{Re}(T_t^{(k)})$$
$$SD(T_t) = \sqrt{\mathrm{var}(\mathrm{Re}(T_t^{(k)}),...,\mathrm{Re}(T_t^{(K)}))}$$
$$SE(T_t) = \frac{SD_k(T_t)}{\sqrt{K}} \quad (10)$$

Selection of the best classification tree uses the "one standard error rule" rule, namely the rule for selecting a classification tree, where the first classification tree will be selected which has a cross validation (CV) value less than or equal to the minimum value of CV plus a standard error [11]. The following is the formula for "one standard error rule" .

$$\hat{T}_t = \underset{T_t \in \{T_t,...,T_{\max}\}}{\arg\min} CV_k(T_t)$$
$$CV(T_t) \le CV(\hat{T}_t) + SE(\hat{T}_t) \quad (11)$$

The first step in determining the optimal classification tree is to find the minimum relative error value in the sub-tree T_t ($CV(\hat{T}_t)$), then using the "one standard error rule" to obtain an optimal classification tree.

2.5 CLASSIFICATION ACCURACY EVALUATION

"From the classification model that has been formed, it is necessary to evaluate the results of the classification. The classification model that has been formed from the training data will then be used to determine the class of the testing data, then the actual results of the testing data will be compared with the results obtained from the classification model. A good classification is one that is able to provide a high classification accuracy value with a low error rate. Some of the measures are used to measure the accuracy of classification are APER (Apparent Error Rate) and Hit Ratio. APER is used to see opportunities for errors in classifying objects. The APER value represents the proportion of samples that are incorrectly classified" [12]. "To make it easier to calculate APER, a confusion matrix table can be used. The matrix is a table that helps evaluate how well the classifier can recognize patterns in each class" [2]. It is presented in the following table.

Table 1. Classification accuracy

Predicted Membership	Actual Membership	
	π_1	π_2
π_1	n_{11}	n_{12}
π_2	n_{21}	n_{22}

Classification accuracy can be calculated using the Hit Ratio formula which is presented in the following equation, Hit Ratio (HR).

$$HR(100\%) = \frac{n_{11} + n_{22} + ... + n_{gg}}{N} \quad (12)$$

Meanwhile, the *APER* value can be calculated using the formula of $APER(100\%) = 1 - HR(100\%)$.

3. DATA AND METHODOLOGY

3.1 CREDIT PROCESS

According to the Banking Act No. 10 of 1998, credit is the provision of money or equivalent claims, based on a loan agreement or agreement between a bank and another party that requires the borrower to repay the debt after a certain period of time with interest. Loan collectibility is a classification of the status of the state of payment of interest installments or principal installments and credit interest by the debtor as well as the level of possibility of receiving funds invested in securities or other investments. Before giving credit, the bank must conduct a credit analysis using the 5C and 7P principles. Principle 5C namely, Character, Capacity, Capital, Collateral, Condition of economy. While the 7Ps are Personality, Party, Purpose, Prospect, Payment, Profitability, and Protection [13].

3.2. CREDIT DATA AND ANALYSIS METHOD

The data used in this analysis is secondary data in the form of bank debtor credit collectibility data from the date of 01 May 1986 to 31 May 2018. The response variable used in this classification is the credit collectibility of a debtor (COL) which has two classes, namely current (debtors who do not have arrears) and arrears (debtors who have arrears). Generally, in a bank, debtors who have delinquent credit have a smaller proportion than debtors who have current credit. Data were collected from 6,960 debtors, 5,569 (80%) debtors were used as training data and 1,392 (20%) debtors were used as testing data.

Distribution of training data and testing data was done randomly with a ratio of 80%:20% based on the Pareto principle. This study uses the Rstudio software and computer hardware specifications used as tools in conducting the analysis are: Processor: Intel Core i7-7700HQ Processor (6M Cache, 3.80 GHz); Memory:16GB; and Graphic card: NVidia GeForce GTX 1050.

4. RESULTS AND DISCUSSION

Calculation of classification accuracy needs to be done to find out how much a set of objects can be categorized correctly in a class. The following are the results of the analysis presented in Table 2 in the form of a confusion matrix.

$$HR(100\%) = \frac{4367 + 616}{5569} x100\% = 83.49\%$$

$$APER(100\%) = 1 - 83.49\% = 16.51\%$$

From the results of the 10-cross validation it was found that the results of the classification accuracy were 83.49%. This shows quite high accuracy results, consistency and the ability of the BART method is very good in various distributions of collectibility data. Then a calculation of the accuracy of the classification on the BART training data was carried out for 1,250 iterations on the training data which can be seen in Table 3.

Table 2. Confusion matrix for learning data for cross-validation

Predicted Membership	Actual Membership		
	Current	Delinquent	Total
Current	4367	137	4504
Delinquent	821	479	1300
Total	5188	616	5804

Table 3. Confusion matrix learning data with BART

Predicted Membership	Actual Membership		
	Current	Delinquent	Total
Current	4401	103	4504
Delinquent	795	505	1300
Total	5196	608	5804

$$HR(100\%) = \frac{4401+505}{5804} x100\% = 84.53\%$$
$$APER(100\%) = 1-84.53\% = 15.47\%$$

From the results of BART with 1,250 iterations, the accuracy of the overall classification result (Hit ratio) was 84.53%. This shows that BART's ability is quite good in classifying credit collectibility. Classification accuracy that is too high can cause overfitting so that the ability of BART can be said to be good enough. In the following, Table 4., the results of the accuracy of the BART method of classification are presented in the testing data.

Table 4. Confusion matrix testing data with BART

Predicted Membership	Actual Membership		
	Current	Delinquent	Total
Current	885	22	907
Delinquent	146	104	250
Total	1031	126	5804

$$HR(100\%) = \frac{885+104}{1157} x100\% = 85.48\%$$
$$APER(100\%) = 1-85.48\% = 14.52\%$$

From the aforementioned results it can be concluded that the ability of the BART method to predict classification is very good, namely 85.48%. These results are relatively the same as

the classification accuracy of the testing data which shows that there is no overfitting in the BART method. With the hardware specifications described in Sub-chapter 3.2, the time needed to run BART with 1,250 iterations in this study takes 7.39 seconds.

The percentage of classification result is quite large, 85%. However, this still needs improvement as for the application of the prediction, in banking or other implementation as well, requires high precision (with minimum error). Other implementation as well as simulation to confirm the goodness of fit of this method are highly expected for future research. Furthermore, variable selection, as also stated in [14], would challenging to investigate.

5. CONCLUSION

We have the conclusion as the following:

1. The results of the cross-validation on the BART show a high consistency of classification accuracy, 83.49%, indicating that the BART method can work consistently even though there is a class imbalance.
2. The classification accuracy is 84.53% in the training data while in the testing data it is 85.48%. These results indicate that the BART method has ability to overcome overfitting in the classification method, where overfitting often occurs in most of the classification methods with very good classification abilities.
3. The results of the testing data show that the accuracy is relatively similar to the one of the training data, this confirms that the BART method has been able to capture patterns in the data.

ETHICAL APPROVAL

All authors hereby declare that all experiments have been examined and approved by the appropriate ethics committee and have therefore been performed in accordance with the ethical standards laid down in the 1964 Declaration of Helsinki.

REFERENCES

1. Conigliani C, Castro JI, O'Hagan A. Bayesian Assessment of Goodness of Fit Against Nonparametric Alternatives. The Canadian Journal of Statistics. 2000;28(2): 327–342.
2. Han J, Kamber M, Pei J. Data Mining: Concepts and Techniques, 3rd ed. San Francisco: Morgan Kauffman; 2012.
3. Pant H, Srivastava R. A Survey on Feature Selection Methods for Imbalanced Datasets. International Journal of Computer Engineering and Applications. 2015;9(2): 197--204.
4. Weiss GM. Foundations of Imbalanced Learning. In Imbalanced Learning: Foundations, Algorithms, and Applications. Wiley Online Library; 2013. pp. 13–42.
5. Sun Y, Kamel MS, Wong AKC, Wang Y. Cost-Sensitive Boosting for Classification of Imbalanced Data. Pattern Recognition. 2007;40(12): 3358–3378.
6. Cieslak DA, Hoens TR, Chawla NV, Kegelmeyer WP. Hellinger Distance Decision Trees Are Robust and Skew-Insensitive. Data Mining and Knowledge Discovery. 2012;24(1): 136–58.
7. Efendi A, Fitriani R, Naufal HI, Rahayudi B. Ensemble Adaboost in classification and regression trees to overcome class imbalance in credit status of bank customer. Journal of Theoretical and Applied Information Technology. 2020;98(17):3428--3437.
8. Breiman L, Friedman J, Stone CJ, Olshen, RA. Classification and Regression Trees, 1st ed. New York: Chapman and Hall; 1984.

9. Santosa B, Umam A. Data Mining dan Big Data Analytics: Teori dan Implementasi Menggunakan Python & Apache Spark, 2nd ed. Jakarta: Penebar Media Pustaka; 2018.
10. Chipman HA, George EI, McCulloch, RE. Bayesian CART Model Search. Journal of the American Statistical Association. 1998;93(443): 935–48.
11. Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. California: Springer; 2009.
12. Lemeshow S, Hosmer Jr DW. A Review of Goodness of Fit Statistics for Use in the Development of Logistic Regression Models. American journal of epidemiology. 1982;115(1): 92–106.
13. Hasibuan MSP. Manajemen Sumber Daya Manusia. Jakarta: Bumi Aksara; 2016.
14. Prado EB, Moral, RA, Parnell, AC. Bayesian additive regression trees with model trees. Statistics and Computing. 2021;31(20):

APPENDIX

Appendix A: R Software Codes for analysis:

```
dataL <- datax %>%filter(COL =='L')
dataM<- datax %>%filter(COL =='M')
ada.generate=function(formula,latih,uji,iterasi,ulangan,syarat){
  ##inisiasi awal
  bas=all.vars(formula)
  y<-latih %>% select(bas[1])
  y_kat<-levels(y[,1])
  n<-length(y_kat)
  hasil<-matrix(0,ulangan,2);   nums=array(0,ulangan)
  ## looping_1
  N_data<-matrix(0,nrow(hasil)+2,2)
  colnames(N_data)<-c("N_training","N_Uji")
  jum=list()
  b=1;  i=1
  repeat{
    fit<-boosting(formula,latih,mfinal = iterasi)
    fit2<-predict.boosting(fit,uji)
    prob<-fit$prob
    prob2<-fit2$prob
    peluang<-matrix(0,length(prob[,1]),1)
    peluang2<-matrix(0,length(prob2[,1]),1)
    ###training
    for ( t in 1:length(prob[,1])){
      ifelse(prob[t,1]>=syarat,peluang[t,1]<-prob[t,1],{ifelse(prob[t,2]>=syarat,{peluang[t,1]<-
prob[t,2]},hafizh<-1)})    }
    hastie<-(peluang>=syarat)
    proba<-peluang[hastie,]
    f1<-as.factor(fit$class[hastie])
    levels(f1)<-c(levels(f1),y_kat[2])
    #testing
    for ( t in 1:length(prob2[,1])){
      ifelse(prob2[t,1]>=syarat,peluang2[t,1]<-prob2[t,1],{ifelse(prob2[t,2]>=syarat,{peluang2[t,1]<-
prob2[t,2]},hafizh<-1)})    }
    hastie2<-(peluang2>=syarat)
    proba2<-peluang2[hastie2,]
    f2<-as.factor(fit2$class[hastie2])
    levels(f2)<-c(levels(f2),y_kat[2])
    ###
    binom<-rbinom(length(f1),n-1,prob =proba)
    binom2<-rbinom(length(f2),n-1,prob =proba2)
    generate<-matrix(0,length(proba),1)
    generate2<-matrix(0,length(proba2),1)
    ## looping_2
    for (j in 1:length(proba)){
      ifelse(binom[j]==1,generate[j,1]<-y_kat[1],generate[j,1]<-y_kat[2])
    }
    for ( j in 1:length(proba2)){
      ifelse(binom2[j]==1,generate2[j,1]<-y_kat[1],generate2[j,1]<-y_kat[2])     }
    ###
    aktual<-as.data.frame(f1)
    generate<-as.factor(generate)
    levels(generate)<-c(levels(generate),y_kat[2])
    ifelse( length(levels(aktual[,1]))>1 && length(levels(generate))>1,{
      cfm<-confusionMatrix(generate,aktual[,1])
      akurasi<-cfm$overall },{akurasi[1]<-0})
    ###
    aktual2<-as.data.frame(f2)
    generate2<-as.factor(generate2)
```

```
    levels(generate2)<-c(levels(generate2),y_kat[2])
    ifelse( length(levels(aktual2[,1]))>1 && length(levels(generate2))>1 ,{
      cfm2<-confusionMatrix(generate2,aktual2[,1])
      akurasi2<-cfm2$overall
    },{akurasi2[1]<-0})
    ###
    N_data[i,1]<-length(f1)
    N_data[i,2]<-length(f2)
    ###
    ifelse(akurasi[1]>akurasi2[1] && akurasi2[1]>=0.8,{hasil[i,1]<-akurasi[1]
    hasil[i,2]<-akurasi2[1]
    jum[[b]]<-1
    b=b+1; i=i+1
    },{jum[[b]]<-0
    b=b+1})
    if(i==ulangan+1){break
    } }
  # looping_3
  for ( k in 1:ulangan){
    nums[k] <- (paste("ulangan ke-",k))   }
  ###
  rata_rata=mean(hasil[,1])
  rata_rata2=mean(hasil[,2])
  stder=sd(hasil[,1])/sqrt(ulangan)
  stder2=sd(hasil[,2])/sqrt(ulangan)
  ovv<-matrix(c(stder,rata_rata,stder2,rata_rata2),2,2)
  row.names(ovv)<-c("std_err","mean")
  ###
  sas<-matrix(0,length(jum),1)
  for ( p in 1:length(jum)){
    sas[p,1]<-jum[[p]]  }
  row.names(hasil)<-nums
  colnames(hasil)<-c("akurasi_latih","akurasi_uji")
  hasil<-rbind(hasil,ovv)
  hasil<-cbind(hasil,N_data)
  value=list(akurasi=hasil,sukses_gagal=sas)
  return(value) }
hasil_akhir<-matrix(data=NA,nrow = 2,ncol = 4)
```