ENSEMBLE ADABOOST IN CLASSIFICATION AND REGRESSION TREES TO OVERCOME CLASS IMBALANCE IN CREDIT STATUS OF BANK CUSTOMERS

[1*]ACHMAD EFENDI, [2]RAHMA FITRIANI, [3]HAFIZH IMAN NAUFAL, [4]BAYU RAHAYUDI

[1,2,3]Department of Statistics, Faculty of Sciences, Universitas Brawijaya, Malang, Indonesia

[4]Department of Information System, Faculty of Informatic Sciences, Universitas Brawijaya, Malang, Indonesia

[*]Corresponding Author E-mail: a_efendi@ub.ac.id

ABSTRACT

The Classification and Regression Trees (CART) is a popular classification method. Generally, at a bank, debtors who have delinquent loans (Non-performed Loan/NPL) have a small proportion compared to debtors who have smooth loan (Performed Loan/PL). Standard classification methods CART is not suitable for handling such cases as it is sensitive to classes with a high degree. Hence, additional methods are needed in order to improve classification accuracy in the case of class imbalance. This study aims at determining the results of the classification using the CART and Adaptive Boosting (Adaboost) CART methods on bank loan or credit collectability data where there is class imbalance. The data used for analysis are secondary data in the form of bank debtor credit collectability data with 9 predictor variables and one response variable. Simulations are also conducted to find out the consistency of the results of analysis and general performance of Adaboost CART. The results of this study indicate the accuracy of the classification on the Adaboost CART method can be increased compared to the CART method. This implies that Adaboost can add weights to classifiers which have small misclassifications and can reduce weights on the correctly classified objects. This research can be taken into consideration in choosing the right classification analysis in the case of data with class imbalance. Simulation results confirm that the classification accuracy of Adaboost CART is relatively large, 84.1%.

Keywords: *Adaboost, Classification and Regression Tree (CART), Class Imbalance, Credit, Bank*

1. INTRODUCTION

Classification and Regression Trees (CART) method is one of the supervised learning classification methods, a branch of the decision tree method. This CART classification method can be used for categorical and continuous scale predictors and it also can handle data with a very large size [1] such that in its application this method is generally used. In the real world, the main problem that becomes a challenge in the classification method is class imbalance, which has attracted the attention of academics and researchers in recent years [2]. Class imbalance is the condition where there are unequal amounts between classes contained in a data set (unequal distribution of data) [3]. Meanwhile, class imbalance is also defined as a condition in data set where there are classes that have a large size while the other classes are only represented by a few objects [4].

The standard classification method generally has poor performance in the case of class imbalance since it pays little attention to minority classes in unbalanced data sets [4]. Classification rules that predict minority classes tend to be weaker than rules in predicting majority classes. As a result, minority classes are more often misclassified than majority classes. This condition is often been in a bank that offers financial credit to customers. One of the main functions of banks is being to mobilize public savings funds properly.

Funds received by the bank from the community will be channeled back to the people who need it in the form of credit. Credit is the main factor and most dominant on bank income. Banks need to be careful in managing their credit such that risk can be controlled. At the moment, bank becomes one of the service providers in home ownership loans. The increase in population growth also impacts on increasing the desire of consumers to own a house. This then can provide benefits in channeling credit for a bank. However, an increase in consumers' desire to own a house also presents a risk for banks. The risk arises as the credit provided by the bank cannot be returned on time.

The type of data in financial research is often continuous as the object of the research is usually in the form of money. However, in the case of bank lending to customers, discrete data is often used. For example, the customer's credit status is PL (Performing Loan) or NPL (Non-Performing Loan). Several factors can influence the bank's decision to give credit or not to a prospective customer. The criteria given by banks are actually the same (from Indonesian Regulator). However, banks need to make selections to ensure that certain prospective loan customers do not have more dependents or loads and can pay the installments until they finish their loans.

The main problem that occurs in this study is the bank credit collectability data, where debtors who are categorized as PL and NPL have unbalanced class comparison. The class of NPL has a smaller proportion than the PL class, this will make the classification results using CART will be inaccurate. Decision trees have weaknesses in class imbalance because the criteria for sorting on decision trees use the Gini Index which is sensitive to classes that have a high degree [5]. Therefore, we need a method to overcome this class imbalance. The Adaboost ensemble method is the most effective method to improve classification accuracy in the cases of class imbalance [4]. The Adaboost method can improve the performance of a classification method in the case of class imbalance by giving more weights to misclassified objects (misclassified objects are often in the minority class). In short, the Adaboost method can improve classification performance in the minority class.

The debtor selection process includes collecting administrative requirements to the bank and then one of the managers at the bank is assigned to select the prospective debtor. This process requires quite a long time, on average the customer needs to wait a minimum of two weeks to get certainty whether or not their loan application is accepted. The manager and the selection team need to check all files of the customer. Apart from that, as many as 15% - 20% of these debtors on average experience delinquency (Bank X, Denpasar, Bali, Indonesia). The idea that emerged later was how the selection process was more suitable, faster and more appropriate for screening the prospective debtors.

Several studies discussed about Adaboost and its performances. Adaboost algorithm can be used with any other machine learning algorithm and it remains one of the successful boosting algorithms [6]. Adaboost ensemble classifier was discussed where improvement in performance over the single classifiers was obtained [7]. Meanwhile, Synthetic Minority Over-sampling Technique (SMOTE) and Rotation Forest algorithm can be effectively used to address the class imbalance problem [8]. Additionally, the boosting should be used like random forest technique [9]. Ensemble method to address class imbalance were also discussed [10]. Still, ensemble learning framework was used to predict seismic-induced liquefaction and was more effective than using single machine learning models [11]. Sudarto [12] discussed the handling of class imbalances using Density Based Feature Selection (DBFS) and Adaptive Boosting (Adaboost). The results of the study showed that the C4.5 algorithm performed boosting (with Adaboost) on various combinations, the model is said to be able to show a relatively better level of classification accuracy in handling class imbalances. This research will discuss in more detail regarding the ensemble method and logistic regression in the process of selecting bank loan prospective customers. Classification is done using CART and Adaboost CART methods. The output expected in the selection process is the estimated PL and NPL status of a bank customer or debtor. Factors that influence the status we enter from several factors or items that are determined by the bank are also important to know.

2. THE COMPREHENSIIVE THEORITICAL BASIS

Classification is a process to get a model or function that can distinguish classes in the data. Classification can also be used with the aim of predicting unknown classes on data [1]. Classification can be categorized as supervised learning and unsupervised learning. The supervised learning method has the purpose of identification, which means there are target variables specified previously. The algorithm of decision trees will automatically determine the most important variable based on the variable's ability to sort data into correct output categories/classes. This method has a relative advantage over the currently popular neural network method, which is able to understand the content of the model, and can find out why the prediction output is so by looking at the tree structure of Decision Trees directly [13]. In the decision tree, the decision of the attribute is stated in the branch until finally found the category or class of an object at the last node. The method used in this study is the CART method.

2.1 Classification and Regression Tree (CART) and Class Imbalance

Breiman, Friedman, Olshen and Stone [14] introduced one of the classification methods known as Classification and Regression Trees (CART). In CART there are two important steps that must be done to get a tree with optimal performance. The first step is the repetition of objects repeatedly based on certain attributes and the second one is pruning. This method continues to be done so that in a node as much as possible contains objects from the same group or class [13]. CART is called a binary partition method since the process of forming a model involves a collection of data that will be partitioned into two nodes based on the predictor variable criteria. CART is also referred to as a recursive method in the process of dividing data into two nodes. The main node is formed and then it is broken down again to produce two child nodes and so on to produce the terminal node which is the final branch of a tree [1]. There are three stages in forming a CART classification tree: sorting, class assignment, and pruning. Formation of classification trees uses training data. The training data will be sorted using the level of impurity $i(t)$, which is a measure of the heterogeneity of a particular node in the classification tree. There are several measures of impurity level that are popularly used, such as Information Gain, Gain Ratio, Gini Index and Entropy [13]. The measure of impurity level that often used is the Gini index, since it is easy to apply [14]. The following is the function of the heterogeneity of the Gini index,

$$i(t) = 1 - \sum_j p^2(j \mid t) \qquad (2.1)$$

$i(t)$: Index of heterogeneous function at the node t, $p(j \mid t)$: Class proportion j at node t, and $0 \le i(t) \le (k-1)/k$.

Class labeling is done to determine the most dominant class of a node. This is done to determine the characteristics of the classification of observations for each formed node. The class labeling for each node is based on the largest proportion of classes. The largest class proportion shows the class that dominates in classes. The following is an opportunity function of class marking,

$$p(j_0, t) = \max_j \, p(j \mid t) = \max_j \frac{Nj(t)}{N(t)} \qquad (2.2)$$

with $p(j|t)$: Proportion of class j in node t, $N_j(t)$: Number of observations of class j in node t; $N(t)$: Number of observations in node t; then the label for terminal node t is j_0 [14]. The classification tree that has been formed from the sorting process will produce a tree with a very large size structure (with many final nodes). This tree is commonly called the Maximal Tree (*Tmax*). Pruning steps are needed such that the overfitting does not occur. Overfitting is a condition where the ability of trees to classify training data is very good, but it is very bad in classifying new data (testing data). This happens as too much sorting is done by the node, so the node needs to be trimmed [13]. After pruning, an optimal classification tree will be obtained.

In the real world, the main problem that challenges the classification method is class imbalance. This has caught the attention of academics and researchers in recent years [2]. Class imbalance is a condition where there are classes that have a large size while the other classes are only represented by a few objects [4]. The standard classification method generally has poor performance in cases of class imbalance because the weak classification method pays little attention to minority classes in unbalanced data sets [4]. Therefore, classification rules that predict minority classes tend to be weaker than rules in predicting majority classes. As a result, minority classes are more often misclassified than majority classes.

In overcoming the problem of class imbalance, one approach is re-sampling methods. This is done by increasing the minority class and reducing the majority class from a given data. This method is the most commonly used method in overcoming class imbalance in a data set [15]. The performance of ensemble method increases the sensitivity as well as the accuracy of the classification. [16]. One of the classification methods known as Classification and Regression Trees (CART) [14], though class imbalance is still not accommodated well.

2.2 Ensemble and Adaptive Boosting (Adaboost)

The following explanation describes an ensemble learning method based on logistic regression which is expected to be a reliable model for classifying objects. The ensemble method is an algorithm in machine learning where it combines several models to achieve higher generalization performance than by a single model [17]. The basic ensemble regression functions are as follows:

$$f_B(x) \equiv \frac{1}{N}\sum_{i=1}^{N} f_i(x) = f(x) - \frac{1}{N}\sum_{i=1}^{N} m_i(x) \qquad (2.3)$$

If $m_i(x)$ is independent with zero centered value, then the mean squared error (MSE) of the basic ensemble regression function can be calculated with the following formula:

$$MSE[f_B] = E\left[\frac{1}{N}\sum_{i=1}^{N} m_i^2\right]$$

$$= \frac{1}{N^2} E\left[\sum_{i=1}^{N} m_i^2\right] + \frac{1}{N^2}\sum_{i \neq j} E[m_i]E[m_j] = \frac{1}{N^2} E\left[\sum_{i=1}^{N} m_i^2\right]$$

This implies that $MSE(f_B) = 1/N(\overline{MSE})$. The basic principle of the ensemble method is to develop a set of models from training data. Then, it combines a set of models to determine the final classification. The final classification is based on the largest collection of votes from the combined model. The basic ensemble regression function model above is very powerful as it can reduce MSE by a multiple of $1/N$.

Boosting is one of the popular methods used in machine learning. It is designed for problems related to classification and is applied to weak classifiers. Adaptive Boosting (Adaboost) is a boosting algorithm developed with classifiers [18]. Adaboost can improve the accuracy of various classification methods such as Decision stumps, Decision trees, Multi-Layer perceptron, and Support Vector Machines (SVM). Adaboost is a method that combines weak classifiers, which are iteratively made from weighted resampling samples, with weights adjusted adaptively at each step to provide weight gain in cases that have misclassified the previous step.

Adaboost works by giving more weight to objects that are not properly classified by weak classifiers, denoted by $h(x_i)$. Then these weak classifiers will be combined to form strong classifiers / final classifiers, which are denoted by $H(x_i)$. This method begins with the initial weighting of the initial training data $\{(x_i, y_i, \ldots, (x_N, y_N))\}$, where $Y \in \{-1,1\}$. Each object will be given the same weight $w_t(i)$. If the training data consist of N objects then the initial weight for each object is $1/N$. The weight is used for resampling data at each subsequent step, depending on the level of misclassification of the classifier made in the previous step. The following is the misclassified formula, denoted by ε_t,

$$\varepsilon_t = \sum_{i=1}^{n} w_t(i) I(y_i \neq h_t(x_i)).$$

Weights are denoted by w_t and $I(y_i \neq h_t(x_i))$ is an indicator function with a value of 1 if $y_i \neq h_t(x_i)$ and a value of 0 if $y_i = h_t(x_i)$. Then we calculate the weighting votes with the formula presented in $\alpha_t = 1/2(\ln((1-\varepsilon_t)/\varepsilon_t))$. Furthermore, weighting voting α_t is also calculated on weak classifiers $h_t(x)$. Then α_t is used to update the weights in the next step with the following formula

$$w_{t+1}(i) = \frac{w_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t} \qquad (2.4)$$

where, $\exp(-\alpha_t y_i h_t(x_i)) = \begin{cases} <1 \; y_i = h_t(x_i) \\ >1 \; y_i \neq h_t(x_i) \end{cases}$

$$Zt = \sum_{i=1}^{N} w_t(i)\exp(-\alpha_t y_i h_t(x_i))$$

From equation (2.4) it can be seen that weighting the wrongly classified training data will be worth more than one while weighting the correct classified data will be less than one. Also, in equation (2.4), Z_t is a normalized constant so $w_{t+1}(i)$ will be a distribution. The Adaboost algorithm will generate weak classifiers by practicing subsequent learning based on previously obtained errors. After the above stages continue until the T-iteration, a strong classifier/final classifier will be produced with the following formula

$$H(x_i) = \mathrm{sgn}\left(\sum_{t=1}^{T} \alpha_t h_t(x_i)\right). \qquad (2.5)$$

$H(x_i)$ is a combination of classifiers which is calculated as the sum of weighting voting signs as many as T steps. The Adaboost algorithm must be run for a long time of at least 1,000 steps, so an increasingly convergent error rate is obtained [6].

In this study the steps taken in the Adaboost algorithm are 1,000 steps [6]. A binary logistic regression model were used as a basic ensemble function. The α_i weighting value is obtained from the proportion of each $f(x)$ when predicting or classifying data appropriately. Big data about the consumers profile of credit products from a bank are used as a case study. Big data are divided into n_1, n_2, and n_3 groups. The grouping method used is deterministic grouping. And then boosting is used for classifying. In this case, boosting can be said to be very effective in reducing training errors [18].

2.3 Validation and Classification Evaluation

Validation is one of the most important techniques for the stability of learning models related to how well the model will be generalized to new data. The popular method for validation is cross-validation. It is a statistical technique partitioning data into subsets to evaluate a learning model. In this study, the K-fold cross-validation method was chosen with K = 10. This method is popularly used and has several advantages such as being able to reduce bias. Each K-partition on the data will be tested once and will be used in training K-1 times. The variety of produced estimates decreases with increasing *K*. And partitioning is done as much as *K = 10* [19]. Generally, it produces estimation models with low bias and simple variations. A value of *K* = 10 is also very commonly used in the field of machine learning.

The procedure for conducting a 10-fold cross-validation is that training data of size *N* will be randomly divided into *K* subsets of data of relatively equal size. *K-1* subsets data will be used as testing data. This process will be carried out *K* times using each subset of data such that each subset has at least once been testing data. The final result of validation is the average and standard error of misclassification value of 10 repetitions. The following is a given function of 10-fold cross-validation.

$$CV(T_t) = \frac{1}{K}\sum_{k=1}^{K} \mathrm{Re}(T_t^{(k)}) \;;$$

$$SD(T_t) = \sqrt{\mathrm{var}(\mathrm{Re}(T_t^{(k)}),\ldots,\mathrm{Re}(T_t^{(k)}))} \;;$$

$$SE(T_t) = \frac{SD_k(T_t)}{\sqrt{K}}$$

with $CV(T_t)$ is average of relative error on sub-tree T_t, $SD(T_t)$ is standard deviation on sub-tree T_t, $SE(T_t)$ is standard error on sub-tree T_t, and *K* is number of *K* subset of data.

From the classification model, it is necessary to evaluate the results of the classification. The model that has been formed from the training data will then be used to determine the class in the testing data. Then the actual results of the testing data are compared to the results obtained from the classification model. Good classification provides high classification accuracy values with low error rates. Some of the classification accuracy measurements are APER (*Apparent Error Rate*) and Hit Ratio (HR). APER is a value that is used to see the possibility for errors in classifying objects. The APER value gives information of the proportion of the sample that was incorrectly classified [20]. To simplify the APER calculation we can use a confusion matrix table. The matrix is a table that helps in evaluating how well the classifier can recognize patterns in each class [1].

Table 2.1: Confusion matrix

Prediction of membership	Actual membership	
	π_1	π_2
π_1	n11	n12
π_2	n21	n22

3. METHODOLOGY

The used data are secondary data in the form of bank debtor of credit collectability data from May 1, 1986 to May 31, 2018. The data were 6,961 debtors with 5,569 (80%) debtors were used as training data and 1,392 (20%) debtors were used as testing data. The distribution of training data and testing data is done randomly with a ratio of 80%: 20%. The rule is known as the Pareto principle which has been widely applied in various fields in the world. The response variable in this classification is the credit collectability of a debtor (COL) which has two classes; Performing Loan/PL (debtors who have no arrears) and Non-performing Loan/NPL (debtors who have arrears). Generally, a bank has a smaller proportion of delinquent loan or credit (NPL) compared to smooth credit (PL). Hence, the proportion between the PL and NPL classes has an unbalanced or popularly known as class imbalance. To overcome this problem, in this study two methods are used: CART and Adaboost CART.

CART Steps:

a) Dividing data into two parts, training data and testing data; randomly by 80%: 20%
b) Determining the best sorter that provide the highest level of impurity based on the goodness of split criteria
c) Class labeling and performing validation using *K*-fold cross-validation
d) Pruning CART classification trees
e) Evaluation of classification accuracy

Adaboost CART steps:

a) Dividing the data into two parts, training data dan testing data; randomly by 80%: 20%
b) Initialization of weight of training data $w_{t=1}(i) = 1/N$, for all $i = 1, \ldots, N$
c) Sampling of *N* data from training data with resampling bootstrap
d) Determining classification tree $h_t(x_i)$ with CART method
e) Calculating classification error ε_t and determining weighting vote α_t
f) Updating weight $w_{t+1}(i)$, (2.4)
g) Doing step *c* until *t* as many of *T* (*T*=1,000).
h) Determining final classifier of $H(x_i)$, (2.5)
i) Evaluation of classification accuracy.

Some steps and simulation scenarios:

1. Setting simulation based on sample size: 200, 500, 1000, 5000 observations.

2. Calculating the mean and standard errors of the classification accuracy of training and testing data.
3. Plotting the mean of classification accuracy based on different sample sizes

In the whole process of data analysis, statistical testing and simulation, R 3.5.3 software are used.

Table 3.1: Response and Predictor Variables

Variable	Information
COL	Credit collectability: 1) Performed Loan (PL); 2) Non-performed Loan (NPL)
TERM	Term of credit of a debtor (month)
PMTAT	Monthly credit installment paid by a debtor (million Rupiahs)
RATE	Bank Interest
AOC	Credit duration (year)
JOB	A = Permanent employee; B = Non-permanent employee; C = Paid per work employee; D = Others
STATUS	A = Married; B = Not yet; C = Divorced
GENDER	M = Male; F = Female
AGE	Age of the debtor (year)
EDU	Education: 1) <=Elementary school; 2) Junior High School; 3) Senior High School; 4) Diploma; 5) Bachelor; 6) Master; 7) Doctoral

4. RESULT AND DISCUSSION

In this section, the result of the analysis is explained and at the same time is given the comprehensive discussion. Results are presented in figures, graphs, tables and others that make the reader understand easily [2], [5]. The discussion is presented in several sub-chapters.

4.1. Data Description

Before analyzing CART and Adaboost CART, a description of bank credit collectability data is given. In this case the credit collectability data at the bank has a case of class imbalance, wherein the PL class is represented by a large sample while the NPL class is represented by a small sample. Generally, a debtor that has NPL has a smaller proportion compared to debtors with PL credit. CART analysis is included in the supervised learning classification method. The distribution is done randomly with the proportion of training data of 80% and testing data of 20%. The training data is used to form the classification tree, while the testing data is used to validate the classification tree, which is to see how much the ability of the classification tree in predicting new data. The following table is a description of the training and testing data.

Table 4.1: Data description

	NPL	PL	Total
Data	1,550 (78%)	5,411 (22%)	6,961 (100%)
Training data	1,238	4,331	5,569 (80%)
Testing data	312	1,080	1,392 (20%)

4.2. CART Analysis

The initial stage in forming a classification tree is doing all possible sorting that gives a measure of heterogeneity. The size of heterogeneity is expressed as the Gini index in equation (2.1). If the predictor variable is continuous then the sorting is determined by $X_j \le c$ where c is the average of the two observations of sequential variables X_j. While if the variable is a nominal category with p labels, then $2^{p-1}-1$ possible sorting will be obtained. In order to find out the best sorting of all possibilities, the value of goodness of split can be used. The greater the value of goodness of split the better the node's ability to sort observations. The process of calculating the goodness of split is determined by calculating the Gini index first. Table 4.2 shows the best sorting candidates to be used as the main node in the classification tree.

Table 4.2. Selection of main candidate nodes

Left node candidate	Right node candidate	*Goodness of split*
TERM ≥ 69	TERM < 69	0.0136
AOC < 2.5	AOC > 2.5	0.0112
AGE ≥ 33.5	AGE < 33.5	0.0052
PMTAT < 0.3329	PMTAT > 0.3329	0.0039
EDU (Dipl., Under graduate, Master, & Doctoral)	EDU (Elementary School, Junior, high school, Senior high school)	0.0038

Based on the goodness of split value in Table 4.2, the credit term variable (TERM) was chosen as the best sorter to sort the main node into the left node t_L the debtor who has a credit period of more than 69 months, and the right node t_R the debtor who has a credit period less than 69 months. The Gini index value and the proportion of observations go to the left node t_L and the right node t_R to each prospective candidate at the main node, with $\max_{sCS} \Delta i(s,t) = 0.0136$. The value of goodness of split at the main node is 0.0136, with credit term (TERM) selected as it has the greatest goodness of split value compared to other possible choices. This results that continuous splitting until a final node (terminal node) is obtained, a tree with a very large size structure (with many nodes). This tree is commonly called the Maximal tree ($T_{\max}$). The formed $T_{\max}$ classification tree produces a final node (terminal node) of 134 nodes with 133 splitting. The Maximal tree is then trimmed into smaller classification tree. Pruning steps are needed so that the phenomenon of overfitting does not occur.

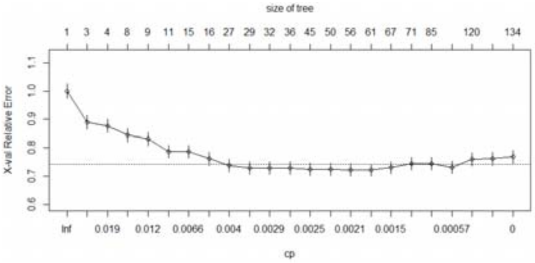

Figure 4.1. Plot of Complexity Parameter

In the pruning process, it results in a trade-off between relative error validations. The terminal nodes formed in pruned trees will produce trees that can capture the actual pattern, in other words not prone to overfitting (more general classification trees). The horizontal value in Figure 4.1 shows the value of the cross validation relative error in each size of formed tree. The dashed line in Figure 4.1 shows the value of "one standard error rule", which is the minimum Cross Validation (CV) value plus the standard error. The minimum CV value can be seen in Figure 4.1, which is a tree with a final node size of 61. Optimal classification tree will be selected at the cross validation relative error value that touches the "one standard error rule" line. The less the final node size will make it easier for the classification tree to interpret. If the classification tree size is too large, it will be difficult to interpret, given that one of the advantages of CART analysis is being able to find out the prediction output by looking at the classification tree structure directly [13].

4.3. Adaboost CART

The Adaboost was created from weighted training data, with weights adjusted adaptively at each step to provide weight gain in cases that had misclassified in the previous step. The classification used in Adaboost method is the CART classification tree. This method begins with the initial weighting of the training data, where each object is given the same weight, $w_t(i)$. If the training data consists of N objects, then the initial weight for each object is $1/N$. The training data used in this study is 5.569 observations, so the initial weight for each object is 0.00018. Then we resample for training data with returning. Resampling the Adaboost by weighting each object can produce a small amount of error [4].

The next step is classification with CART analysis using training data that has been done with resampling. The classification tree is formed with the classification error ε_t, weighting voting α_t, and normalization constant Z_t. After the first weighting is obtained, the initial weighting will be updated ($w_{t+1}(i)$). The resampling is then repeated until the weighting update is obtained as many as the specified iteration (T = 1,000). The result is presented in Table 4.3 which contains object weights for 3 iterations and in Table 4.4 which contains values of ε_t, α_t, and Z_t.

Table 4.3. Weight object $w_t(i)$ for 3 iterations

Iteration	1	2	I
$w_1(i)$	0.00018	0.00018	...
$w_2(i)$	0.000105	0.000105	---
$w_3(i)$	7.28×10-5	7.28×10-5	...
Iteration	5,565	5,566	5,569
$w_1(i)$	0.00018	0.00018	0.00018
$w_2(i)$	0.000105	0.000105	0.00062
$w_3(i)$	7.28×10-5	7.28×10-5	0.00043

From Table 4.3 it can be seen that in the 3rd iteration of the 5,569 objects (i = 5.569), the 3rd CART classification tree has misclassified the object, so that the weight w_3(5,569) has a value greater than $w_{3(1)}$, $w_{3(2)}$, $w_{3(3)}$, and $w_{3(4)}$. It can also be seen that the same weight value shows the 3rd CART classification tree with correct classifying objects. Furthermore, Adaboost CART makes a classification based on the weights that have been adapted to each iteration to provide weight increase in cases that have been misclassified.

Table 4.4 Values of $\varepsilon_t, \alpha_t, Z_t$

Iteration	ε_t	α_t	Z_t
1	0.1442	0.8905	0.7026
2	0.2795	0.4734	0.8976
3	0.3264	0.3623	0.9378
⋮	⋮	⋮	⋮
T	ε_T	α_T	Z_T

It can be seen in Table 4.4 that the weighting votes in the first iteration α_t have a greater value than the weighting votes in the iterations thereafter α_{t+1}. This shows that the first CART classification tree has a smaller classification error value than the iterations thereafter ε_{t+1}. Thus, when calculating the final classifier $H(x_i)$, the classification tree which has a classification error of close to 0.5 ($\varepsilon_t = 0.5$) will be given a small weighting vote. The following are a misclassified plot in Figure 4.2 and a weighting voting plot in Figure 4.3.

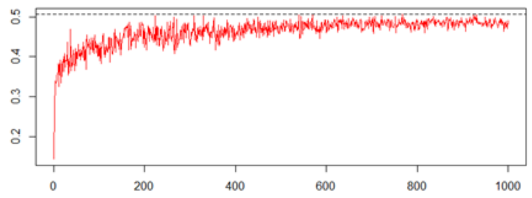

Figure 4.2. Plot of Classification Error

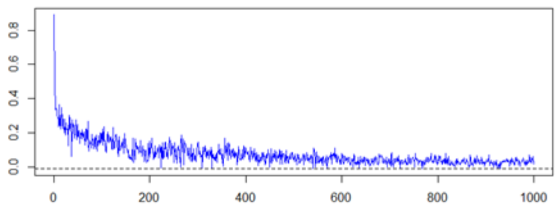

Figure 4.3. Plot of Weight Voting

Vertical axis in Figure 4.2 shows misclassification and the horizontal one indicates iteration. While in Figure 4.3, vertical axis shows weighting voting and the horizontal one for iteration. There is increasing misclassification ε_t that approaches 0.5 and the weighting value α_t decreases. The iteration step used in this study is as many as 1,000 iterations. The fact that Adaboost must be run in a long time of at least 1,000 steps to obtain increasingly convergent error rates [6].

The following is an illustration of the use of the final classification function $H(x_i)$ in determining the predictability of a debtor's collectability whether included in the PL or NPL. From 1,000 iteration steps used, a strong classifier / final classifier will be formed where a combination of 1,000 CART classification trees is calculated as the sum of the weighting voting sign of the number of T steps. The object used in this calculation is the 3rd object in the testing data, namely a debtor who has a 'credit age' (AOC) of 3 years, 'credit term' (TERM) of 180 months, 'installments credit' (PMTAMT) of 3 million Rupiah, 'bank interest rate (RATE) of 0.1275, 'debtor age' (AGE) 51 years, 'last debtor education' (EDU), high school / vocational school, 'debtor work' (JOB) of permanent employee, 'married status'' (STATUS), and 'male sex' (GENDER). From the results of calculation of the combined classifier / final $H(x_i)$, the 3rd object in the testing data is classified as a positive class (positive class is PL and negative class is NPL). The debtor can be considered by the bank as PL hence the credit application can be approved, since the prediction results indicate that the debtor has PL collectability. The same will be done for each debtor to know their credit collectability predictions.

4.4. Classification Accuracy

Calculation of classification accuracy needs to be done to find out how good a set of objects can be categorized appropriately in a class. The following are interpretations of each evaluation value of classification accuracy. The total level of classification accuracy in the testing data is 82.04% (*HR*). This indicates that the classification trees CART is able to classify the testing data correctly as much as 82.04%. The total level of misclassification in the testing data is 17.96% (*APER*), meaning that the classification tree CART incorrectly classifies the class in the testing data by 17.96%. And then, sensitivity is 96.30%, which means that the PL class on the actual membership that is correctly classified into indeed the PL class with the CART is 96.30%. Furthermore, the accuracy of classification in the CART classification tree that has been done by Adaboost of 1,000 iterations in the testing data is presented in Table 4.5.

Table 4.5. Confusion Matrix on testing data (Adaboost CART)

Membership Prediction	Actual Membership		Total
	PL	NPL	
PL	1027	137	1164
NPL	53	175	228

The following are interpretations of each evaluation value of classification accuracy. The total level of classification accuracy in the testing data is 86.35% (HR), meaning that the Adaboost CART with 1,000 iterations is able to classify the class in the testing data exactly 86.35%. The total misclassification rate in the testing data is 13.65% (*APER*), meaning that Adaboost CART misclassifying the class in the testing data by 13.65%. And then, sensitivity is 95.09% that means the PL class in the actual membership is classified correctly into the PL class is 95.09%.

From the results of the calculation of classification accuracy (*HR*), it is found that the training data on the classification accuracy generated in the CART method is 85.13% while the accuracy generated in the Adaboost CART method is 98.17%. The classification accuracy for testing data generated in the CART method is 82.04% and the accuracy with Adaboost CART is 86.35%. It can be seen that there is an increase in the accuracy of the classification of the CART method with Adaboost both in the training data and in the testing data.

In the training data there was an increase in classification accuracy of 13.04% between the CART and the Adaboost CART methods. While in the testing data there was an increase in classification by 4.31%. So, it can be concluded that the Adaboost CART with 1,000 iterations is able to classify the whole class appropriately better than the CART classification tree. Adaboost CART with 1,000 iterations proved to be able to overcome the class imbalance problem. The classification accuracy can be increased since, at each iteration on Adaboost CART, it adds weight to the classifier which has a small misclassification and reduces the weight on objects that are classified correctly. Hence, it can change the distribution of data. The iteration used in Adaboost CART is 1,000 iterations [6], in order to obtain an increasingly convergent error rate. This opinion is proven by plotting misclassification for each iteration; with a proof in Figure 4.4.

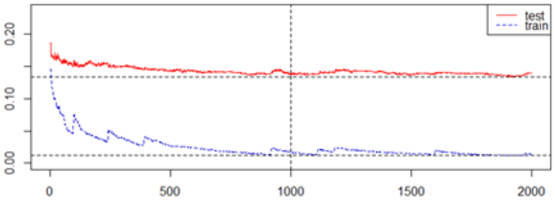

Fig. 4.4. Plot of Misclassification of Adaboost CART

It can be seen in Figure 4.4, before 1,000 iterations in the testing data and training data, that the value of misclassification is fluctuating for each iteration. The value of misclassification reached convergence at the time of iteration of more than 1,000. This confirms the opinion of Mease and Wyner [6]. At a bank, debtors with NPL have a smaller proportion compared to debtors who have PL status. Hence, the proportion between the PL class and the NPL class has an unbalanced class. The results of the evaluation of the accuracy of the classification method Adaboost CART can be used as an appropriate method in classifying credit collectability in banks. This also indicates that ensemble Adaboost is proven to improve classification accuracy in class imbalance cases. The CART method was chosen because of capability of handling predictors of both categorical and continuous scaled predictors; given the bank's credit collectability data has predictor variables with various data scales. The ability of the CART method in handling data that has a very large size is one of the advantages why this method is popularly used today. The recommendation that can be given to banks related to the results of the analysis is that the classification analysis using Adaboost CART can be used as a tool in supporting the decision making whether a prospective debtor can be approved by the bank or not. With the hardware specifications in this analysis, the time needed to run an Adaboost with 1,000 iterations takes on average of 3 minutes and 30 seconds. This proves that the number of iterations performed does not significantly affect the time of running the analysis.

4.5. Simulation Study

The simulation is carried out with several possible settings, iterations of 100, 30 replications, with sample sizes of 200, 500, 1000, and 5000. From the simulation study, the results are obtained and presented in Table 4.6.

Table 4.6: Mean accuracy from simulation results

Sample Size	Mean (Training)	SD (Training)	Mean (Testing)	SD (Testing)
200	0.9118	0.00755	0.8414	0.00513
500	0.8948	0.00678	0.8425	0.00722
1000	0.8833	0.00742	0.8368	0.00614
5000	0.8633	0.00523	0.8453	0.00435

The average accuracy value is 84.15.% for testing data, while for training data is approximately 88.8%. The percentage change in classification accuracy between testing and training data by 4% is considered to be good enough, in the sense that with data outside the model (data testing), it can be obtained that classification accuracy is quite good, 84%. Then the percentage change in classification accuracy, mainly from testing data, as the sample size increases is also in line with the principle of stability, where the greater the sample size, the classification accuracy becomes relatively fixed. This is evidenced by the classification accuracy with a sample size of 500 of 84.2%, 83.6% for a sample size of 1000, and 84.5% for a sample size of 5000.

5. CONCLUSION

There are several conclusions from this paper. The result of cross-validation on the CART is a tree with 27 final nodes. It is chosen based on the "one standard error rule" and also the final node size. The smaller the size of the final node will facilitate interpretation. The accuracy of the Adaboost CART can increase as, at each iteration, it can add weight to the classification tree that has a small misclassification and reduce the weight on objects that are classified correctly. Simulation results confirm that the classification accuracy of Adaboost CART is relatively large, 84.1%.

ACKNOWLEDGEMENT

We would like to show our gratitude to Universitas Brawijaya, Malang, Indonesia, for providing us with the research fund in 2019.

REFRENCES:

[1] Han, J. and Kamber, M. *Data Mining Concepts and Techniques, 2nd ed*. Morgan Kaufmann, pp. 15, 2006.

[2] Pant, H. and Reena, S. Computer Engineering and Applications. *A Survey on Feature Selection Methods for Imbalanced Datasets*, 9 (2), pp. 197-198, 2015.

[3] Weiss, G. *Foundation of Imbalanced Learning*. John Wiley and Sons, pp. 4-5, 2012.

[4] Sun, Y., Kamel, M.S., Wong, A.K.C., and Wang, Y. Cost-Sensitive Boosting for Classification of Imbalanced Data. *Pattern Recognition*, 40 (12), pp. 3358-3360, 2007.

[5] Cieslak, D.A., Hoens, T.R., Chawla, N.V., and Kegelmeyer, W.P. Hellinger distance decision trees are robust and skew-insensitive. *Data Mining and Knowledge Discovery*, 24 (1), pp. 137-138, 2012.

[6] Mease, D and Wyner, A. Evidence Contrary to the Statistical View of Boosting. *Journal of Machine Learning Research*, 9, pp. 131-156, 2008.

[7] Ramakrishnan, S., Mirzaei, M., and Bekri, M. Adaboost Ensemble Classifiers for Corporate Default Prediction. *Research Journal of Applied Sciences, Engineering and Technology*, 9 (3), pp. 224-230, 2015.

[8] Fattahi, S., Othman, Z., and Othman, Z.A. New Approach with Ensemble Method to Address Class Imbalance Problem. *Journal of Theoretical and Applied Information Technology*, 72 (1), pp. 23-33, 2015.

[9] Wyner, A.J., Olson, M., and Bleich, J. Explaining the Success of Adaboost and Random Forest as Interpolating Classifiers. *Journal of Machine Learning Research*, 18, pp. 1-33, 2017.

[10] Khuat, T.T. and Le, M.H. Ensemble learning for software fault prediction problem with imbalanced data. *International Journal of Electrical and Computer Engineering*, 9 (4), pp. 3241-3246, 2019.

[11] Alobaidi, M.H., Meguid, M.A., and Chebana, F. Predicting Seismic-induced Liquefaction Through Ensemble Learning Frameworks. *Scientific Report - Nature Research*, 2019.

[12] Sudarto, S. Analysis for Solving Class Imbalance using DBFS and Adaboost, unpublished, pp. 67, 2016.

[13] Budi, S. and Ardian, U. *Data Mining dan Big Data Analytics. 2nd ed*. Yogyakarta: Penebar Media Pustaka, pp. 81, 2018.

[14] Breiman, L., Friedman, J., Stone, C.J., and Olshen, R.A. *Classification and Regression Tree*. New York: Chapman & Hall/CRC, pp. 17, 1984.

[15] Hulse, J.V. and Khosghoftar, T. Knowledge Discovery from Imbalanced and Noisy Data. *Data and Knowledge Engineering*, 68 (12), pp. 1513-1542, 2009.

[16] Makhtar, M., Awang, M.K., Rahman, M.D.A., Fadzli, S.A., and Mohamad, M. Optimizing Sensitivity and Specificity of Ensemble Classifiers for Diabetic Patients. *Journal of Theoretical and Applied Information Technology*, 82 (2), pp. 230-236, 2015.

[17] Wittek, P. *Quantum Machine Learning*. Sweden: Elseiver. 2014.

[18] Schapire, R.E. and Freund, Y. *Boosting: Foundation and Algorithm*. The MIT Press, pp.54, 2012.

[19] Kuhn, M. and Johnson, K. *Applied Predictive Modelling*. New York: Springer, pp. 277-281, 2013.

[20] Hosmer, D.W. and Lemeshow. S. *Applied Logistic Regression*. New York: John Wiley and Sons Inc, pp. 223, 2000.