Int. J. Advance Soft Compu. Appl, Vol. 13, No. 1, March 2021
Print ISSN: 2710-1274, Online ISSN: 2074-8523

Performance Evaluation of Backpropagation, Extreme Gradient Boosting, Feedforward Network for Classification of Customer Deposits Subscription

Tris Dianasari[1], Dedi Rosadi[2]*

Department of Mathematics, Universitas Gadjah Mada, Indonesia

[1]e-mail: tris.dianasari@mail.ugm.ac.id
[2]*Corresponding author, e-mail: dedirosadi@gadjahmada.edu

Abstract

Neural Network is a method often used to predict. The most popular technique is the Neural Network Backpropagation algorithm. However, the Backpropagation algorithm has some weaknesses. It took too long to be convergent and it has minimum local problems that make artificial Neural Networks often get stuck at the local minimum. Deep Neural Network is an Artificial Neural Network that has many layers, generally more than 3 layers (input layer, N hidden layers, output layer). Mxnet is one of the developed algorithms from deep Neural Networks that have the advantage of producing better accuracy. Boosting is an ensemble family that includes many algorithms. Xgboost is a more efficient and scalable version of the Gradient Boosting Machine. This case study using data is related to direct marketing campaigns of a Portuguese banking institution in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed. From the results of the analysis with 3 different methods, it can be concluded that the method with the best accuracy in the case study of additional Bank data is the Extreme Gradient Boosting Method, followed by the Deep Learning Feedforward Network method, and finally the Neural Network.

Keywords: *Backpropagation, Neural Network, Extreme Gradient Boosting, Deep Learning, Accuracy, Classification*

1 Introduction

In the current era, the term industry 4.0 refers to the fourth industrial revolution marked by growing trends in the field of automation, one of which is big data. In general, big data can be defined as a collection of data that is very large, very fast-changing, comes in various formats, and has a certain value, provided that it comes from an accurate source. The main thing that distinguishes big data from conventional data sets lies in the management mechanism. Machine learning is considered as a discipline that studies computer algorithms that can learn without being explicitly programmed.

In the journal by Chatzis, et al. [1] discusses crisis prediction for the next day or 20 days using Logistic Regression, Decision Trees, Random Forest, Support Vector Machines, Neural Networks, Extreme Gradient Boosting, Deep Learning Feedforward Network, and Forecast Combination methods. From the methods mentioned above, the MXNET (Deep Learning) method gives the most optimal results. The author will compare these methods using other datasets because it allows a method to produce the best predictions for certain data.

Currently, in a competitive global world, positive responses to mass campaigns are usually very low, less than 1%, according to the same study [2]. Otherwise, the marketing focus is on targets who will be assumed to be interested in a particular product and service, making this type of campaign more attractive due to its efficiency [3]. The problem is that one of the strategies adopted is a long-term offering that attracts deposit applications at good interest rates, particularly by using targeted marketing campaigns [4].

At this time there is a very large amount of data available, one of them is the data of clients who purchased time deposits. Someone will be classified whether to buy a deposit product or not based on the identity information held by the consumer. Of the 20 indicator variables, that the financial product provider can predict someone to buy the financial product using a machine learning model for classification is Neural Networks, Extreme Gradient Boosting and the development of Neural Networks, namely Deep Learning.

In this study, The Authors used general gbtree parameters in the Extreme Gradient Boosting method, booster parameters with different eta values, task parameters on the objective used multi: softprob and the maximum boosting number of iterations was 1000. Meanwhile, the Deep Learning Feedforward Network method uses the tanh activation function and additional momentum parameters.

This case study using data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed. One way to improve business is Marketing selling campaign, one way is to contact potential customers and offer products to them. From this analysis, it will make it easier for the company to obtain specific and unambiguous targets, therefore this data is important to study.

Neural Network is a knowledge engineering concept in the field of machine learning designed by adopting the human nervous system, whose main processing is in the brain. The arrangement of neurons in the layers and their connection patterns within and between layers is called a Network Architecture. The types that are often used are single layered networks, multiple layered networks, and opportunity neural networks. Backpropagation Neural Network is a method for training Multilayer Neural Networks. The development of the method is getting faster so that the development of the Backpropagation method which is still conventional has become a breakthrough, namely the Feedforward Neural Network, which is a Multilayer method with a more complex architecture. Meanwhile, Xgboost can perform various functions such as Regression, Classification and Ranking. Xgboost is a tree ensembles algorithm that consists of a collection of several Classification and Regression Trees (CART).

2 Related Work

Ramadhani [5] discussed the amount of expenditure for each customer contained in the purchase variable which will be divided into 2 classes, namely customers with low and high expenses. Based on the results of the analysis carried out, the best method for Black Friday classification analysis is Xgboost Classifier with a full dataset, Xgboost with 3 variable results from feature selection and Random Forest Classifier with 3 variables resulting from feature selection, all of which produce the highest accuracy value of 91%. Using preprocessing methods and classification methods using Convolutional Neural Networks are reliable enough to determine the correctness of the object image classification with an accuracy of 20% -50%. Izah [6] in his research identified the nominal value of 2017 emission rupiah banknotes by applying the Convolutional Neural Network algorithm using Mxnet.

Widodo, et al. [7]. In their research determined the accuracy of the backpropagation prediction model using a combination of hidden neurons and alpha, it was concluded that the lowest MSE value in the prediction model was at the alpha training variable value 0.7, with 7 hidden neurons, 0.7 momentum, maximum epoch. 10000 with an error tolerance of 0.001 with the resulting predictive model accuracy of 95%. Ghofur [8], in his final assignment, forecast currency exchange rates using the Stocastic Gradient Descent Multilayer Perspectron model and obtained the best model with a learning rate of 0.01, sliding window 3 input and the number of hidden units is 1.

Handayani, et al. [9] found the best model from the Fine Needle Aspiration (FNA) test data to classify malignant tumors and benign breast tumors by evaluating the Area Under the Curve (AUC) value from the Extreme Gradient Boosting (XGBoost) algorithm, Support Vector Machine Kernel Radial Basic Function (SVM-RBF), and Multilayer Perceptron (MLP). The results show that the AUC value and the lowest cost score or the best algorithm is SVM-RBF in the data set which eliminates missing values.

3 Problem Formulations or Methodology

This study aims to determine the biner classification method to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed in machine learning namely Neural Network, Extreme gradient boosting and Deep Learning Feedforward Network. Next is applying Neural Network, Extreme Gradient Boosting and Deep Learning Feedforward Network to the data. And finally to find out which Regression Classification method has the best performance by looking at its accuracy value. In this study, the authors did not calculate the time of the training process explicitly, because to see the best method prioritizes the level of accuracy.

4 Method

4.1 Backpropagation

The human brain consists of hundreds to millions of neurons. Each neuron has a simple design. Neural Networks are composed of nodes that combine their inputs (variables from the database or output from other nodes). The nodes are connected via a link. The idea of this Neural Network mimics the workings of the human brain, which has the characteristics of parallel processing, large amounts of processing elements and fault tolerance. Neural Network can perform Classification Regression for various types of data, both numerical and categorical and can be used for various data formats, namely text, image, video and audio data.

These nodes can be classified into three simple layers. Input layer, output layer and hidden layer. Each node in each layer has an error rate, which will be used for the training process.

4.2 Extreme Gradient Boosting

Gradient Tree Boosting or Gradient Boosted Regression Trees are generalizations of boosting to distinguish arbitrary loss functions. Gradient Boosted Regression Trees are accurate and effective 'off-the-shelf' procedures for solving regression and classification problems. The Gradient Tree Boosting model is used in a variety of areas including web search rankings and ecology.

Extreme Gradient Boosting was first introduced by Friedman [10]. The advantage of the Xgboost Algorithm is that it can use storage memory efficiently. Extreme Gradient Boosting is a technique in Machine Learning for Regression and Classification problems that produces a predictive model in the form of a weak ensemble prediction model and can be used for various types of data, both numeric and categorical and can be used for various data formats, namely text data, images, video and audio. Model development is done by using the boosting method, namely by creating a new model to predict the error / residual from the previous model. New models are added until no more error fixes can be made. This algorithm is called gradient descent to minimize errors when creating a new model.

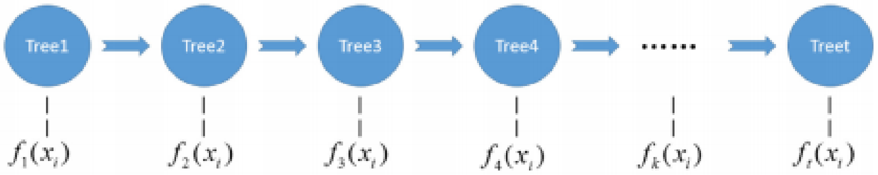

Fig. 1. Systematic Diagram of Xgboost Algorithm

where $\hat{y}_i^{\,t} = \sum_{k=1}^{t} f_k(x_i)$ and $f_k(x_i)$ describe the tree model and y$_i$ is obtained from the following calculations:

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) \tag{1}$$

where $\hat{y}_i^{\,t}$ is the final tree model; $\hat{y}_i^{\,t-1}$ is the previously generated tree model; $f_t(x_i)$is the newly created model, and *t* is the total number of models from the base tree model.

To find the optimal algorithm can be replaced by finding a new classifier that can reduce the loss function, with the target loss function shown in the following equation:

$$\mathcal{L}(\phi) = \sum_{i=1}^{t} l\left(y_i\, ,\hat{y}_i^{\,t}\right) + \sum_{i=i}^{t} \Omega(f_i) \tag{2}$$

where y_i is the actual value; $\hat{y}_i^{\,t}$ is the predicted value; $l\left(y_i\ \hat{y}_i^{\,t}\right)$is a lost function and $\Omega(f_i) = \gamma T + \frac{1}{2}\lambda\left|\left|w\right|\right|^2$ is regularization.

The ensemble tree model in equation 2 includes a function as a parameter and cannot be optimized using the optimization method in Euclidean space. So that the model is trained additively. $\hat{y}_i^{\,t}$is used in the iteration i and t iteration, to minimize the loss function, it is necessary to add f_t to obtain the following equation:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) \tag{3}$$

To optimize the objective, the second order approach is generally used and the following equation is obtained using the Taylor series approach:

$$f(a+h) = f(a) + f'(a)h + \frac{1}{2}f''(a)h^2 + \cdots + f^n(a)\frac{h^n}{n!} \tag{4}$$

where in the above equation:

$$a = \hat{y}_i^{(t-1)}$$

$$h = f_t(x_i)$$

$$f(a) = l\left(y_i, \hat{y}_i^{(t-1)}\right)$$

Then we substitute it to equation (3), so that the following equation is obtained:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)}\right) + \left(\frac{\partial l\left(y_i, \hat{y}_i^{(t-1)}\right)}{\partial\left(\hat{y}_i^{(t-1)}\right)}\right) f_t(x_i) + \left(\frac{\partial^2 l\left(y_i, \hat{y}_i^{(t-1)}\right)}{\partial^2\left(\hat{y}_i^{(t-1)^2}\right)}\right) f_t(x_i)^2 + \cdots \tag{5}$$

It can be seen that $l\left(y_i, \hat{y}_i^{(t-1)}\right)$is constant.

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} \left(g_i f_t(x_i) + h_i f_t(x_i)\right) + \Omega(f_t) \tag{6}$$

by substituting the values for g_i and h_i

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} \left[l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \Omega(f_t) \tag{7}$$

where $g_i = \partial_{\hat{y}(t-1)} l(y_i, \hat{y}_i^{(t-1)})$ and $h_i = \partial^2_{\hat{y}(t-1)} l(y_i, \hat{y}_i^{(t-1)})$ then removing the constant the following equation is obtained:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \tag{8}$$

Suppose f_t has as many leaf nodes as K, I_j is the jth node, and w_j is the prediction for node j.

$$\Omega(f_t) = \gamma K + \frac{1}{2} \lambda \sum_{j=1}^{K} w_j^2 \tag{9}$$

$$\mathcal{L}^{(t)} = \sum_{j=1}^{K} \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \lambda \tag{10}$$

For each leaf j, $\frac{d\, l^t}{d w_j^*} = 0$

$$w_j^* = \frac{-\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \tag{11}$$

Assume the repair base learner f_t has K leaf nodes. For example I_j being the set of node j, and w_j being the prediction for that node.

$$\mathcal{L}^{(t)} = -\frac{1}{2} \sum_{j=1}^{K} \frac{\left(\sum_{i \in I_j} g_i \right)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma K \tag{12}$$

Is the best loss function to fix base learner with K nodes.

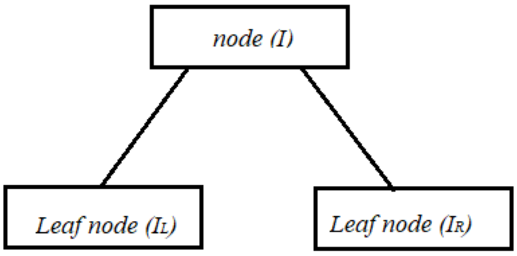

Fig. 2. Formation of the Xgboost leaf

At node (I) towards the leaf node (I_L) the loss values are obtained as follows:

$$\mathcal{L}^{(t)} = -\frac{1}{2} \frac{\left(\sum_{i \in I_L} g_i \right)^2}{\sum_{i \in I_L} h_i + \lambda} + \gamma(1) \tag{13}$$

At node (I) towards the leaf node (I_R) the loss values are obtained as follows:

$$\mathcal{L}^{(t)} = -\frac{1}{2} \frac{\left(\sum_{i \in I_R} g_i \right)^2}{\sum_{i \in I_R} h_i + \lambda} + \gamma(1) \tag{14}$$

So that the loss reduction value is obtained as follows:

$$\mathcal{L}_{split} = \frac{1}{2} \left[\frac{\left(\sum_{i \in I_L} g_i \right)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{\left(\sum_{i \in I_R} g_i \right)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{\left(\sum_{i \in I} g_i \right)^2}{\sum_{i \in I} h_i + \lambda} \right] - \gamma(1) \tag{15}$$

Extreme Gradient Boosting method algorithm for binary classification:

1. Initialize the model with the value:

$$F_0(x) = argmin \sum_{i=I}^{n} l(y_i, \hat{y}_i) \tag{16}$$

   or initialize using value 0.

$$\hat{y}^0 = 0$$

2. For the 1st iteration,

   a. Compute the value of the first derivative and second derivative of the statistical gradient, with the logistics loss function as follows:

$$l(y_i, \hat{y}_i) = y_i \ln\left(1 + e^{-\hat{y}_i}\right) + (1 - y_i)\ln(1 + e^{\hat{y}_i}) \tag{17}$$

$$g_i^{(1)} = -\frac{(y_i - 1)e^{\hat{y}_i^{(0)}} + y_i}{e^{\hat{y}_i^{(0)}} + 1} \tag{18}$$

   And value

$$h_i^{(1)} = \frac{e^{\hat{y}_i^{(0)}}}{(e^{\hat{y}_i^{(0)}} + 1)^2} \tag{19}$$

   b. Determine the best split that will increase the prediction the most or reduce the loss function value the most. For example, the separation for leaf = 2 with the first leaf I_L and the second leaf L_R uses the $\mathcal{L}_{split}$ equation or called gain.

   c. Perform point b for observing each leaf. The algorithm will calculate the gain scores for all potential splits on each leaf separately to see if any new branches with positive gain can be added to either of the two or both leaves.

   d. Calculate the weight value for each leaf w_j^*

   e. Perform points a through d for the next iteration.

3. Combine the resulting tree model. To predict new data, add up all the weights from each tree then transform it into a probability form.

Table 1: Symbols in Extreme Gradient Boosting Algorithms

No	Symbol	Meaning
1.	$f_k(x_i)$	Tree Model
2.	$\hat{y}_i^{t}$	Final tree (Sum of the tree model)
3.	$\hat{y}_i^{t-1}$	Previously generated tree model
4.	t	Total number of models from the base tree model
5.	$\mathcal{L}(\phi)$	Target Loss Function
6.	y_i	Actual Value
7.	$\Omega(f_i)$	Regularization
8.	I_j	Node j-th
9.	w_j	Prediction for Node to j-th

4.3 Deep Learning Feedforward Network

Deep Machine Learning (Deep Learning) refers to machine learning using a Neural Network model, which has a hidden layer, usually more than one.

The smallest form of an Artificial Neural Network is a Single Perceptron which only consists of a Neuron. Furthermore, for Multilayer Perceptron (MLP) it is known as a Feedforward Neural Network. Multilayer Perceptron literally has several layers. There are generally three layers: input, hidden, and output layers. The Input Layer accepts input without performing any operations, then the input values without being passed to the activation function are assigned to hidden units. In hidden units, the input is processed and the results of the activation function are calculated for each neuron, then the results are given to the next layer. The output from the input layer will be accepted as input for the hidden layer. Likewise, the hidden layer will send the results to the output layer. This activity is called Feedforward. The same is true for artificial Neural Networks with more than three layers. Neuron parameters can be optimized using gradient-based optimization. Multilayer Perceptron is a combination of many non-linear functions.

In general, a deep Neural Network has more than 3 layers (input layer, N hidden layers, output layer) or in other words, a Multilayer Perceptron with more layers. Because there are relatively many layers, it is called deep.

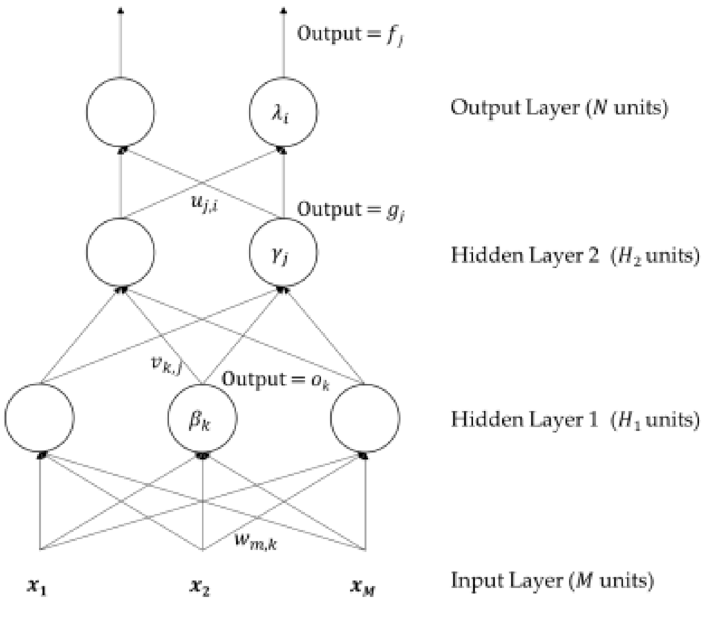

Fig. 3. Deep-Neural Network

In Figure 3 the Neural Network has 4 layers, namely 1 output layer, 2 hidden layers, and 1 input layer. How to calculate the final output based on the following equation:

$$f_i = \sigma\left(\sum_{j=1}^{H_2} u_{j,i}\,\sigma\left(\sum_{k=1}^{H_1} v_{k,j}\,\sigma\left(\sum_{m=1}^{M} x_m w_{m+k} + \beta_k\right) + \gamma_j\right) + \lambda_i\right) \tag{18}$$

Where β, γ, λ are noise or bias. The training process uses Backpropagation. A Deep Network consists of many layers and synapse weights; therefore parameter estimation is more difficult to do to determine the relationship between input and output. Deep Learning can find a "hidden" relationship between input and output, which cannot be solved using a Multilayer Perceptron (3 layers). Feedforward has an analogy with transformation, meaning that the input is transformed non-linearly to the output.

Table 2: Symbols in Deep Learning Feedforward Network Algorithms

No	Symbol	Meaning
1.	f_i	Output
2.	$u_{j,i}$	Network input hidden layer N
3.	$v_{k,j}$	Network input hidden layer 2
4.	w_{m+k}	Network input hidden layer 1
5.	x_m	Training vector input $x_i(1,2,3,\dots,m)$
6.	β,γ,λ	Bias in hidden layers

5 Results, Analysis and Discussions

In this case study, the authors used secondary data from the UCI (University of California Irvine) Machine Learning Repository. This data has 20 variables and 41188 observations on dataset A and dataset B constitutes 10% of the data randomly selected from dataset A which has 4119 observations. This data relates to direct marketing in the form of telephone calls from Portuguese banking institutions to find out whether clients will subscribe to time deposits.

Table 3: Feature Data

No	Feature	Meaning
1.	Age	Customer Age (numeric)
2.	Job	Type of job: Admin (1), Blue-collar (2), Entrepreneur (3), Housemaid (4), Management (5), Retired (6), Self-employed (7), Services (8), Student (9), Technician (10), Unemployed (11), Unknown (12).
3.	Marrital	Marital Status: Divorced (1), Married (2), Single (3), Unknown (4).
4.	Education	Basic.4y (1), Basic.6y (2), Basic.9y (3), High.school (4), Illiterate (5), Professional.course (6), University.degree (7), Unknown (8).

5.	Default	has credit in default? No (1), Unknown (2), Yes (3).
6.	Housing	has housing loan? No (1), Unknown (2), Yes (3).
7.	Loan	has personal loan? No (1), Unknown (2), Yes (3).
8.	Contact	contact communication type : Cellular (1), Telephone (2).
9.	Month	last contact month of year : Jan (1) - Dec (12).
10.	Day_of_week	last contact day of the week : Mon (1) – Fri (5).
11.	Duration	last contact duration, in seconds (numeric).
12.	Campaign	number of contacts performed during this campaign and for this client (numeric, includes last contact).
13.	Pday	number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted).
14.	Previous	number of contacts performed before this campaign and for this client (numeric).
15.	Poutcome	outcome of the previous marketing campaign : Failure (1), Not existent (2), Success (3).
16.	Emp.var.rate	employment variation rate - quarterly indicator (numeric).
17.	Cons.conf.idx	consumer price index - monthly indicator (numeric).
18.	Cons.price.idx	consumer confidence index - monthly indicator (numeric).
19.	Euribor3m	euribor 3 month rate - daily indicator (numeric).
20.	Nr.Employed	number of employees - quarterly indicator (numeric).
21.	y	Has the clinet subcribed a term deposit? Yes(1), No(0).

Prior to data analysis, preprocessing is required. Data preprocessing describes the types of processes that carry out raw data to prepare other process procedures for good results.

The first is to test for missing data. The data to be analyzed is checked whether it contains missing values or not. However, in the dataset used, there are no missing values, so there is no need for handling.

The second is data transformation. The transformation is performed when the attributes are measured in different units. In this case study, the min-max transformation will be used for all attributes of the numeric data type.

The third is oversampling. The number of data classes (instances) which are one is less than the other classes need to be handled so that the data becomes balanced. If working on unbalanced data almost any classification algorithm will produce much higher accuracy for the majority class than for the minority class.

Each class in the data set must be represented in the correct proportion between the training data and the testing data. The data were divided randomly in each class with different comparisons. Details of the size of the training data and testing data for each data can be seen in Table 3.

Table 3: Details of the size of the training data and testing data

Data	**Proportion**		**Data Size**	
	Training Data	*Testing* Data	*Training* Data	*Testing* Data
A	0,8	0,2	32950	8238
	0,7	0,3	28832	12356

	0,6	0,4	24713	16475
B	0,8	0,2	3295	824
	0,7	0,3	2883	1236
	0,6	0,4	2471	1647

After data preprocessing has been performed, the next step is to create a data partition and one hot encoding. Each class in the data group must be represented in the right proportion between the training data and the testing data. Some machine learning algorithms cannot operate on label data directly. So it is necessary to convert from categorical data to numeric form. In this study, one-hot encoding was used by encoding 1 or 0.

5.1 Backpropagation

In this study, an activation function for classification is used, namely the sigmoid logistic with a learning rate of 0.00001 and the error function of cross-entropy or log loss. The training process and data set testing will be divided into two parts, where 80% of the data set will be used as training data and the other 20% will be used for testing. The iteration procedure will be repeated until a convergent value is obtained with a minimum error value.

Table 4: Accuracy Value of Neural Network Method

	Hidden Layer 1	
	Node 1	Node 2
Accurancy	0.8801939	0.9009695
Error	1.60E+03	1.45E+03
Reached.treshold	3.00E-01	9.00E-01
steps	5.48E+05	6.03E+05

Then the backpropagation method with 2 nodes was chosen because it provides the best accuracy value.

Table 5: Network Weight Value Hidden layer 1 node 2

Feature	Weight
Intercept.to.1layhid1	1.39E+00
age.to.1layhid1	6.25E+00
job2.to.1layhid1	-2.70E+00
marital4.to.1layhid1	2.47E-01
⋮	⋮
nr.employed.to.1layhid1	-3.05E+00
Intercept.to.1layhid2	4.99E-01
age.to.1layhid2	-2.35E-01
job2.to.1layhid2	-4.22E-02

marital4.to.1layhid2	-3.23E-01
⋮	⋮
nr.employed.to.1layhid2	2.75E+00
Intercept.to.y	-1.28E+01
1layhid1.to.y	4.16E+00
1layhid2.to.y	1.51E+01

In Table 5 the Intercept.to.1layhid2 feature shows the bias value in the hidden layer 1 node 2 of 0.499. The age.to.1layhid2 feature shows the weight value of the age feature in hidden layer 1 node 2 and so does the other features with a large number of input features. Intercept.to.y is the bias value of the hidden layer that is connected to the output layer (y), which is -0.128 and 1layhid1.to.y shows the weight value of the hidden layer with the output layer (y) of 4.16.

In the neural network 1 hidden layer 2 nodes in this case study, there are 108 weights with 3 biases.

Table 6: Confusion Matrix Neural Network Method

Actual	**Prediction**	
	0	1
0	613	107
1	36	688

From the above calculations, the accuracy is 90.09%, which means that the Neural Network method can classify the data correctly by 90.09%. The sensitivity value is 94.45%, which means that the probability of correct classification results in fact is correct at 94.45%. The specificity value is 86.54%, which means that the probability of the wrong classification result is 86.54% in fact and the accuracy value is 85.13%.

5.2 Extreme Gradient Boosting

In Machine Learning, usually as a rule of thumb, the proportion of the test set is 80% and the train set is 20%. But some determine 75%: 25%. In this study, the proportion experiment used for the training set and, the testing set was 80%: 20%, 70%: 30% and 60%: 40%, respectively. The best accuracy results are obtained, namely the proportion of training set and testing set of 80% and 20%. Where the accuracy value with eta parameter 10-5 is 88.67312%, accuracy with eta parameter 0.1 is 95.64923%, accuracy with eta parameter 0.5 is 96.75215% in data set A. for data set B the value accuracy with eta parameter 10-5 is 89.05817%, accuracy with eta parameter 0.1 is 97.29917%, accuracy with eta parameter 0.5 is 97.43767%.

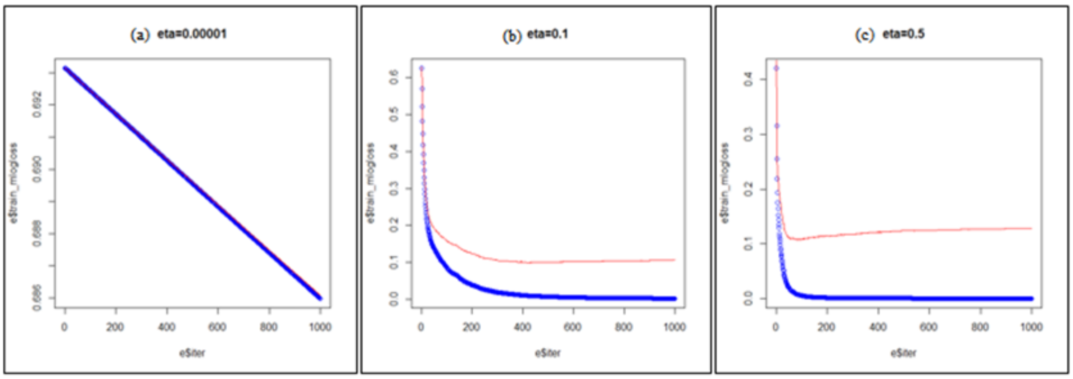

Fig. 4. log loss of training data and testing data

In Figure 4 (a), the logarithmic loss for training data is shown in blue and the logarithmic loss for testing data is shown in red until the 1000th iteration with eta parameters of 10-5. The minimum logarithmic loss value is 0.686085. Evidence of underfitting models is that the training and testing sets have large errors, and the error curves are close to each other.

Table 7: Value of Loss with eta parameter 0.1

Iteration	Train-mlogloss	Test-mlogloss
[1]	0.62384	0,6256
[2]	0.56728	0.57093
[3]	0.51992	0.52503
[4]	0.47991	0.48694
[5]	0.44511	0.45377
[6]	0.41525	0.42503

[7]	0.38873	0.3997
[8]	0.36542	0.37848
[9]	0.34467	0.3589
⋮	⋮	⋮
[992]	0.00266	0.10529
[993]	0.00266	0.10529
[994]	0.00266	0.10531
[995]	0.00265	0.10533
[996]	0.00265	0.10537
[997]	0.00265	0.10544
[998]	0.00264	0.10557
[999]	0.00264	0.10555
[1000]	0.00264	0.10555

Using dataset B with 80% training data and eta parameter of 0.1, confusion matrix is obtained for data testing in Table 8.

Table 8: Confusion Matrix Xgboost Method

Actual	**Prediction**	
	0	1
0	681	39
1	0	724

From the above calculations, the accuracy of 97.29% is obtained, which means that the Extreme Gradient Boosting method can classify data correctly (total prediction accuracy) of 97.29%. The Sensitivity value is 100%, which means that the probability of correct classification results is true at 100%. The specificity value is 94.88%, which means that the probability of the wrong classification result is wrong at 94.88% and the accuracy is 94.58%.

Xgboost has a variable important function that can be used to view important variables. Each variable used in the analysis has a different effect on the given modeling. In the Extreme gradient boosting method, it can show the amount of contribution of each variable to the formed model.

Table 9**:** Feature Importart

Feature	Gain	Cover	Frequency
duration	5.43E-01	0.36823	0.25509
nr.employed	2.03E-01	0.11603	0.04448
euribor3m	1.01E-01	0.11203	0.06672
cons.conf.idx	6.14E-02	0.08226	0.04448
age	1.52E-02	0.02935	0.12154
month4	1.03E-02	0.0057	0.02224
month10	1.01E-02	0.05288	0.02224
pdays	9.42E-03	0.10097	0.06672

job10	8.99E-03	0.02264	0.04448
cons.price.idx	6.60E-03	0.01629	0.04448
campaign	6.23E-03	0.00375	0.03413
job5	5.30E-03	0.01115	0.02224
month7	4.37E-03	0.0124	0.02224
day_of_week4	2.90E-03	0.00518	0.02224
job2	2.85E-03	0.01211	0.02224
marital2	2.52E-03	0.00126	0.0436
job3	2.20E-03	0.01253	0.02224
month12	1.95E-03	0.00393	0.02224
job1	1.67E-03	0.00043	0.02224
contact2	1.01E-03	0.02261	0.02224
job9	1.74E-04	0.00809	0.01112
loan3	4.53E-05	0.000167	0.000778

In Table 9 there are features used in the model. The gain column shows the contribution of each feature to the model based on the total gain. The higher presentation is that feature duration is the most important predictive feature which contributes 53.3% to the prediction model. The Cover column shows the number of observations related to this feature. The frequency column represents the relative number of features that have been used in the tree.

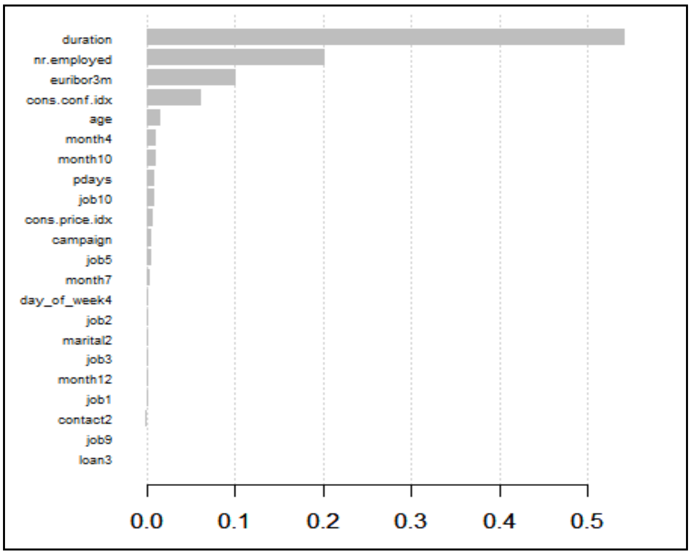

Fig. 5. Contribution of Each feature in Model Building

The amount of contribution for each parameter is shown in Figure 5. with the results obtained following Table 9.

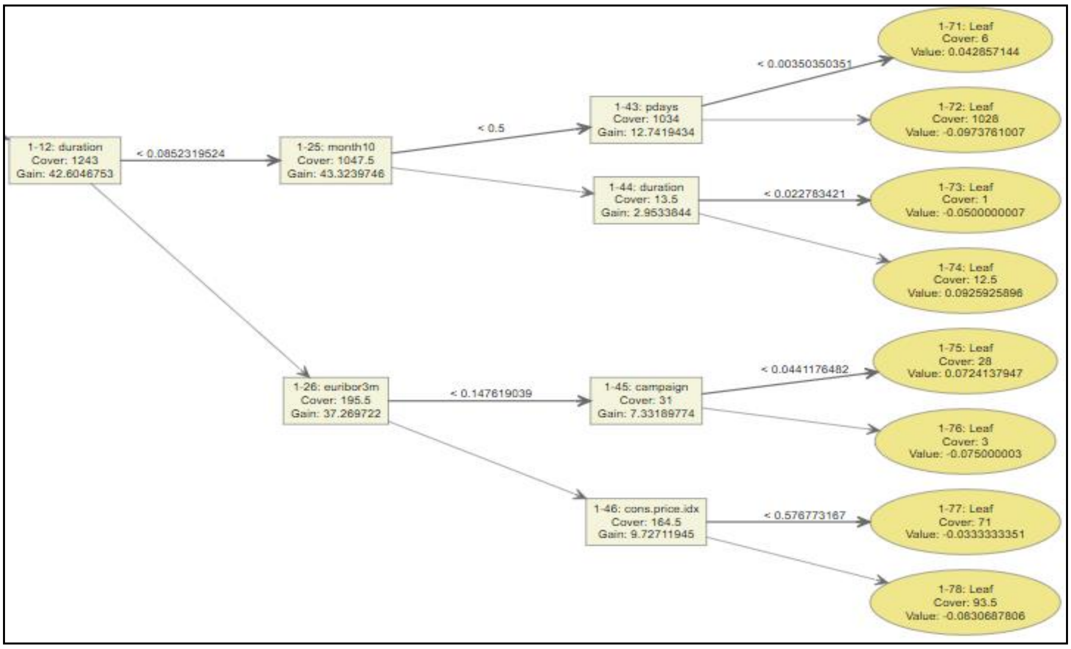

Fig. 6. The structure of the first tree with a depth of 4

In the leaf formed in Figure 6, an example of modeling for the first tree will be carried out. A client has a duration variable less than 0.085. Then the last contact month this year (month) is March (moth10 = 0) so the value of month10 will be less than 0.5. The number of days that pass after the client was contacted the last time in a promotional event has a value of less than 0.0035, so the value or weight of the client is 0.04285. So that the following opportunities are obtained:

$$Odds = \frac{e^{0.04285}}{1 + e^{0.04285}} = 0.517$$

Clients with the criteria as mentioned, get a chance of 0.517 greater than 0.5 so that it can be predicted that the client will buy the offered time deposit.

5.3 Deep Learning Feedforward Network

In this study, the package "mxnet" in R is used as a package that supports the Deep Learning Feedforward Network training process. Mxnet is a simple library of deep learning that requires less code in its programming but the resulting output is very representative, namely the results of classification and the level of accuracy of the classification model. In this study, researchers used 10 hidden node parameters, 2 nodes in the output layer, the activation function used was tanh. As with the Neural Network method, the iteration procedure will be repeated until a convergent value is obtained with a minimum error value.

The use of the learning level parameter has an important effect on the time needed to achieve the desired target. If the learning rate used is too small, there will be too many epochs needed to reach the desired target value, thus causing the training process takes a long time. The greater the value of the learning rate used, the network training process will be faster, but if too large it will cause the network to become unstable and cause value. The error is repeated back and forth between certain values, thus preventing the error from reaching the expected target. Therefore, choosing the value of the learning rate variable must be as optimal as possible in order to get a fast training process.

Giving momentum to the weight change causes a fairly large change in the processing time, the greater the value of the momentum given, the faster the processing time is needed. This means that if you want the prediction process time to be faster then use a large momentum value, but preferably less than 0.9. The computation will be performed using data set B with a data partition of 80:20. Obtained in Table 10.

Table 10: Accuracy Value of Momentum Comparison

Momentum	Learning Rate	Accuracy
0.9	0.1	0.9556787
	0.5	0.8981994
0.5	0.1	0.9452909
	0.5	0.950831

Using dataset B with 80% training data. the learning rate of 0.1 and momentum of 0.9. confusion matrix is obtained for data testing in Table 11.

Table 11: Confusion Matrix Deep Learning Method

Actual	Prediction	
	0	1
0	656	64
1	0	724

From the above calculations, it is obtained the accuracy of 95.56%, which means that the Deep Learning Feedforward Network method can classify data correctly at 95.56%. The sensitivity value is 100%, which means that the probability of correct classification results is at 100%. The specificity value is 89.86%, which means that the probability of the wrong classification result is 89.86% in fact and the accuracy is 91.87%.

When the architecture of the network is finally created, Mxnet provides a simple way to graphically inspect it using the following function call:

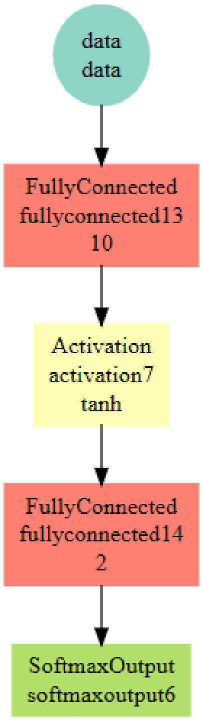

Fig. 7. Computation Graph Mxnet

In Figure 7 the parameter is the trained model represented by the symbol. The first network is constructed by mx.mlp() and the second using the symbol system.
A Deep Learning Feedforward Network can be visualized as a computation graph with an input node where computation begins and an output node where the results can be interpreted.

6 Conclusion

From the case study on the Bank Additional data, the best accuracy value for each method:

Table 12: Accuracy for each Method

Method	Accuracy
Neural Network	90.09%
Extreme Gradient Boosting	97.43%
Deep Learning Feedforward Network	95.56%

From the results obtained, it can be seen that the Extreme Gradient Boosting method has a better accuracy value than the Neural Network and Deep Learning Feedforward Network methods. So it can be concluded that the Extreme Gradient Boosting method has the best performance than the Neural Network and Deep Learning Feedforward Network methods.

ACKNOWLEDGEMENTS
The authors would like to thank the Department of Mathematics, Universitas Gadjah Mada, Indonesia, for their support of this research.

References

[1] Chatzis. SP..Siakoulis.V.Petropoulos.A. (2018). *Forecasting Stock Market Crisis Event Using Deep and Statostical Machine Learning Techniques*. Expert Systems With Applications 112(2018) 353-371.

[2] Ling, X. and Li, C. (1998). *Data Mining for Direct Marketing: Problems and Solutions*. In Proceedings of the 4th KDD conference, AAAI Press, 73–79.

[3] Ou, C., Liu, C., Huang, J. and Zhong, N. (2003). *On Data Mining for Direct Marketing*. In Proceedings of the 9th RSFDGrC conference, 2639, 491–498.

[4] Moro,S,. Cortez and Rita, P. (2014). *A Data-Driven Approach to Predict the Success of Bank Telemarketing*. Decision Support Systems, Elsevier, 62:22-31.

[5] Ramadhani. DI. (2018). *Klasifikasi jumlah Expense Customer pada Data Black Friday dengan Pendekatan Machine Learning*. 06211540000110 (in Bahasa Indonesia).

[6] Izah. RN. (2018) *Klasifikasi Nominal Uang Kertas Rupiah tahun Emisi 2017 dengan Algoritma Convolutional Neural Network menggunakan MXNET* (Undergraduate mini-thesis.Universitas Islam Indonesia, in Bahasa Indonesia).

[7] Widodo. AP., Suhartono. Sarwok. EA dan Firdaus. Z,. (2017). *Akurasi Model Prediksi Metode Backpropagation dengan Kombinasi Hidden Neuron dengan Alpha.*Vol.20 no 2 PP.79-84. (in Bahasa Indonesia).

[8] Ghofur. MM. (2018).*Stochastic Gradient Descent- Multilayer Perceptron untuk Peramalan Nilai Tukar Mata Uang Rupiah Terhadap Dolar Amerika* (Undergraduate mini-thesis. Universitas Gadjah Mada, in Bahasa Indonesia).

[9] Handayani.A. Jamal.A. dan Septiandri. AA. (2017) *Evaluasi Tiga Jenis Algoritma Berbasis Pembelajaran Mesin untuk klasifikasi Jenis Tumor Payudara*. JNTETI.Vol.6. No.4. (in Bahasa Indonesia).

[10] Friedman. J.H. (2001). *Greedy Function Approximation: A Gradient Boosting Machine.*, The Analysis of Statistics. *Vol. 29. No.5. 1189-1232*.

[11] Cengiz, Z. (2019). *How Does Extreme Gradient Boosting (Xgboost) Work*. https://sites.education.miami.edu/zopluoglu/2019/01/15/how-does-extreme-gradient-boosting-xgboost-work/. Accessed December 1, 2020

[12] Darmanto. E. (2019). *Model ABC-PM Boost: Model Klasifikasi Persediaan Obat Menggunakan Gabungan Analisis ABC. Profile Maching dan Adaboost*. (Disertation. Universitas Gadjah Mada, in Bahasa Indonesia).

[13] Machine Learning Repository (2014). *Bank Marketing Data Set*. https://archive.ics.uci.edu/ml/datasets/bank+marketing. Accessed December 1, 2020

[14] Salsabila. (2018). *Penerapan Deep Learning menggunakan Convolutional Neural Network untuk Klasifikasi Citra Wayang Punakawan* (Undergraduate mini-thesis. Universitas Islam Indonesia, in Bahasa Indonesia).

[15] Kanan T. and Fox, EA. (2016), *Automated arabic text classification with P-Stemmer, machine learning, and a tailored news article taxonomy*, Journal of the Association for Information Science and Technology, vol. 67, (11), pp. 2667-2683.