Implementing Extreme Gradient Boosting (XGBoost) Classifier to Improve Customer Churn Prediction

Iqbal Hanif[1]
{iqbal.hanif@telkom.co.id[1]}

Media & Digital Department, PT. Telkom Indonesia, Jakarta, 10110, Indonesia[1]

Abstract. As a part of Customer Relationship Management (CRM), Churn Prediction is very important to predict customers who are most likely to churn and need to be retained with caring programs to prevent them to churn. Among machine learning algorithms, Extreme Gradient Boosting (XGBoost) is a recently popular prediction algorithm in many machine learning challenges as a part of ensemble method which is expected to give better predictions with imbalanced-classes data, a common characteristic of customers churn data. This research is aimed to prove or disprove that XGBoost algorithm gives better prediction compared with logistic regression algorithm as an existing algorithm. This research was conducted by using customer's data sample (both churned and stayed customers) and their behaviors recorded for 6 months from October 2017 to March 2018. There were four phases in this research: data preparation phase, feature selection phase, modelling phase, and evaluation phase. The results show that XGBoost algorithm gives a better prediction than LogReg algorithm does based on its prediction accuracy, specificity, sensitivity and ROC curve. XGBoost model also has a better capability to separate churned customers from not-churned customers than LogReg model does according to KS chart and Gains-Lift charts produced by each algorithm.

Keywords: churn prediction, classification, extreme gradient boosting, imbalanced-classes data, logistic regression.

1 Introduction

According to the survey conducted by Indonesia Internet Service Provider Association (APJII), the growth of Indonesia population positively correlates with the growth of Indonesian internet user proportion compared to national population in 2016-2018: 51.80% from 256.2 million people (APJII, 2016)[1], 54.68% from 262 million people (APJII, 2017)[2] and 64.80% from 264.2 million people (APJII, 2018)[3]. This fact shows that people communication and information needs have increased, creating large opportunity for communication service provider, including internet line service. As a result, a tougher competition between communication service companies is unavoidable. The rivalry atmosphere in this sector is also perceived by Telkom Indonesia, as the one and only State-owned Enterprise providing communication service.

One of the competition effects is churn, a condition when the customers move from one provider to another competitor provider. The higher churn rates the lower revenue company will be. On the other hand, the cost to get new subscribers is more expensive than keeping the existing subscribers (Babu and Ananthanarayanan, 2014)[4]. Therefore, companies tend to protect their customers from churn by improving customers' loyalty through Customer Relationship Management (CRM), including churn prediction analytics.

Churn prediction is aimed to predict how likely a customer becomes churn before it actually happens; so they can be treated with caring programs to prevent them to churn. Predicting process can be done by

ICSA 2019, August 02-03, Bogor, Indonesia

DOI 10.4108/eai.2-8-2019.2290338

machine learning; one of the machine learning algorithms that have been implemented is Logistic Regression (LogReg), a supervised learning algorithm for classifying the customers whether or not they are likely to churn. Unfortunately, this algorithm does not work well for imbalanced-class data; while the majority class in training data (not-churn) is much bigger than the minority class (churn). This condition will lead the classifier to ignoring the minority class and hence, the induced classifier might lose its classification ability (Galar et. al, ,2011)[5].

Extreme Gradient Boosting (XGBoost) is a newly popular algorithm in machine learning scientist in Kaggle competition in which among 29 winning solutions in 2015, 17 of them used XGBoost as a part of their algorithm combination (Chen and Guestrin, 2016)[6]. XGBoost applies ensemble method (boosting) expected to give better predictions with imbalanced-classes data. Therefore, this research aims to compare XGBoost algorithm with LogReg algorithm in predicting churn with imbalanced-classes data. This was conducted with Telkom Indonesia customers dataset sample which have been categorized as churned customers and not-churned customers. The better method was selected by comparing each method metric evaluations based on its accuracy, sensitivity, specificity, Kolmogorov-Smirnov chart, Gains-Lift charts, and Receiver Operating Characteristic (ROC) curve, calculated for each training phase and testing phase.

2 Materials

2.1 Dataset

This research was conducted by using Telkom Indonesia customers sample dataset in March 2018, with customer's lifetime of not more than 12 months as the criteria. This dataset contained customer's profiles recorded from October 2017 to March 2018, with 105,694 rows and 132 columns in total. All predictor features are continuous feature, and the target feature is binary feature, with 1 is defined as churn and 0 is defined as not-churn. Each predictor category was distinguished as 10 different features with attribute as following:

- Mi : The ith month value (start from October 2017 to March 2018, i=1, 2, …, 6)
- MIN6 : The minimum value between 6 months
- MAX6 : The maximum value between 6 months
- SUM6 : The total value in 6 months
- AVG6 : The average value in 6 months

The detail of feature categories is shown in Table 1.

Table 1. Feature categories

Name	Unit	Description
LOS_NCLI	Month	Customer's Length of Stay
SPEED	Mbps	Customer's Internet Access Speed
R2BB_ANGKA	Qty	Count of Total Network Trouble Based on R2BB Measures
ALL_REV	Rp	Customer's Total Revenue
ALL_TRB	Qty	Customer's Total Trouble
ALL_MTTR	Hours	Customer's Mean Time to Repair
INT_PAY	Rp	Customer's Internet Payment (for Internet Connection and UseeTV Service)
INT_REV_TOT	Rp	Customer's Total Revenue from Internet (Internet Connection

Name	Unit	Description
		and UseeTV Service)
INT_REV_INT	Rp	Customer's Revenue from Internet Connection Only
INT_REV_USE	Rp	Customer's Revenue from UseeTV Service Only
INT_TRB	Qty	Customer's Internet Trouble Number
INT_MTTR	Hours	Customer's Internet Mean Time to Repair
INT_USG	MB	Customer's Internet Usage
INT_UPLD	MB	Customer's Internet Upload
INT_DWNL	MB	Customer's Internet Download
INT_REVMINPAY	Rp	Customer's Internet Revenue Minus Payment
ALL_REVMINPAY	Rp	Customer's Total Revenue Minus Payment

2.2Dataset

This research was conducted using Python 3 software with Jupyter Lab as Graphical User Interface (GUI) and some open-source python packages as listed below in Table 2.

Table 2. List of open-source python packages

Name	Purpose
pyspark	Loading data
pandas	Data frame operation
numpy	Mathematic calculation
scipy	Generating statistical distribution
matplotlib	Creating data visualization
scikit-learn	Data preparation and LogReg modelling
xgboost	XGBoost modelling
pickle	Saving/loading the model

2.3 Algorithm

2.3.1 Logistic Regression (LogReg)

Suppose that X as explanatory variables and y as binary response, logistic regression is a classifier which predicts the probability of successful condition $\pi(x)$ at value x. Agresti (2007)[7] states that the logistic regression model has a linear form for the logit that shown in equation (1).

$$\text{logit}[\pi(x)] = \log\left(\frac{\pi(x)}{1-\pi(x)}\right) = \alpha + \beta x \qquad \textbf{(1)}$$

This logit function has a range from 0 to 1, so the label *y* is predicted by default cut-off value 0.5. The $\pi(x) < 0.5$ will be labelled as 0, while the $\pi(x) > 0.5$ will be labelled as 1 (Hanifa et. al, 2017)[8].

2.3.2 Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting is started from a combination between gradient descent and boosting, called Gradient Boosting Machine (GBM). Boosting is an ensemble-learning algorithm that gives different weight

for training data distribution for each iteration. Each boosting iteration adds weight for miss-classified error sample and subtract weight for correct-classified sample, so it changes the training data distribution effectively (Bisri and Wahono, 2015)[9]. GBM uses second order gradient statistics to minimize following regularized objectives that shown in equation (2).

$$\mathcal{L}(\phi) = \sum_{i} l(\hat{y}_i, y_i) + \sum_{k} \Omega(f_k)$$

$$\text{where } \Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2 \qquad \textbf{(2)}$$

with l is a differentiable convex loss function that measures the difference between the prediction $\hat{y}_i$ and the target y_i, and Ω penalizes the complexity of the model (Chen and Guestrin, 2016)[10].

As a tree-based algorithm, GBM is purposed to find the best candidate split points, which is non-trivial for large dataset. Chen and Guestrin (2016)[11] purpose a novel distributed weighted quantile sketch algorithm that can handle weighted data with a provable theoretical guarantee, resulting a new scalable and efficient algorithm called Extreme Gradient Boosting (XGBoost). XGBoost is also provided in many programming languages such as R, Julia and Python.

3 Materials

3.1 Data Preparation

Before the modelling phase, the dataset was needed to be cleaned and prepared, in order to minimize prediction error caused by missing values or outliers. Data preparation consists of four main steps: splitting predictors&target, checking and imputing missing values, handling outlier, and splitting dataset into training and testing dataset, with flow is shown in **Figure 1**.

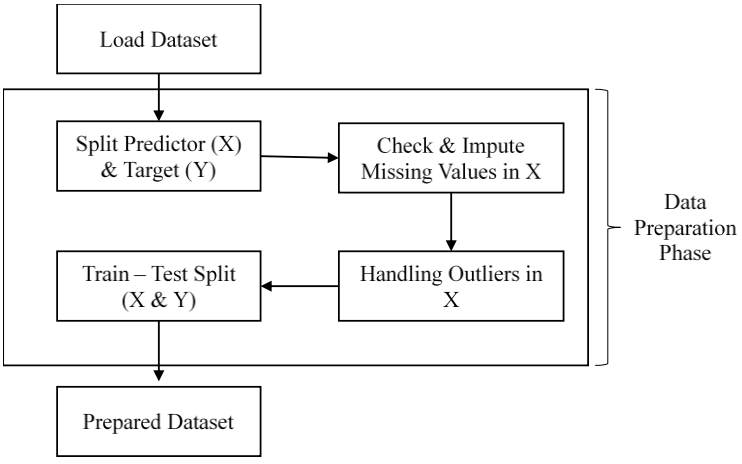

Fig. 1. Data preparation phase flow chart.

Split Predictor Features (X) and Target Feature (Y). Dataset wassplit into two parts: the predictor features (X), consisting of observation features that have been identified and approved in data understanding processes, and the target feature (Y) as customer binary label (1 as churn and 0 as not-churn). The predictor features was used to predict the target feature by using machine learning algorithm.

Checking and Imputing Missing Value (Null). Checking and imputing missing value in predictor features are parts of data cleansing process for creating valid and reliable dataset. Each feature containing missing values was imputed to keep existing feature distribution without deleting any records or information.Imputing missing value was kept in full sample size, which can be advantageous for bias and precision (Gelman and Hill, 2006)[12]. Every missing value was replaced by feature's median value based on its robustness characteristic.

Handling Outliers. Outlier is an object that has different characteristics froma common object or an object that lies in an abnormal distance from other objects in a feature. As a part of data cleansing, outlier can be detected and capped by statistical method. Based on normal distribution characteristics, Hekimoglu and Koch (2000)[13] state that an object can be defined as an outlier if its value is larger than$\mu+3\sigma$ as the upper bound or smaller than $\mu-3\sigma$ as the lower bound (with μ as distribution mean and σ as distribution standard deviation). The simple way of handling an outlier in the predictor features is by replacing its value with its upper bound value and lower bound value.

Split Dataset into Training Dataset & Testing Dataset. Splitting the dataset into two parts (training dataset and testing dataset) is a preparation step to make a valid classification model with reliable accuracy in modelling phase. Training dataset was used to generate classification model through machine learning algorithm, while the testing dataset was purposed to measure the classification model capability for extrapolating new label for new observation which is not included in training dataset; so it can be known whether or not the algorithm is reliable to produce accurate prediction. The training dataset proportion is bigger than test dataset due to construct classification model with low variance. In this research, dataset wassplit with proportion of 70% for training dataset and 30% for testing dataset.

3.2 Feature Selection

This phase is aimed to minimize the feature numbers used in modelling phase, so it consumed low computer resources and simplified the model interpretation. LogReg algorithm used multicollinearity checking for features selection, while XGBoost algorithm eliminated its features with Recursive Feature Elimination (RFE) process.

Multicollinearity Checking. Multicollinearity checking eliminates predictor features by checking its Variance Inflation Factors (VIF), which assesses how far the variance of an estimated regression coefficient increases when predictors are correlated (Akinwande et. al, 2015)[14]. VIF are calculated based on equation (3):

$$VIF = \frac{1}{1-R^2} \quad \textbf{(3)}$$

whereR^2can be obtained by doing a linear regression of that predictor on all other predictors. Some researchers define 10 as the VIF threshold for eliminating predictors causing multicollinearity problem (Haier et. al, 1995)[15]. On the other hand, VIF threshold can be roughly calculated using equation (3) by defining the maximum level of R^2 between predictors. Adeboye et. al. (2014)[16] and Allison (2012)[17] define 2.5 as VIF threshold which corresponds to an $R^2$of 0.60 with other predictors to determine which

variables they want to eliminate or they want to focus on. This research defines 2 as the VIF threshold with 0.50 as the maximum R^2 between predictors, assuming that this threshold would give the best model with the desired lower levels of VIF. Multicollinearity checking has a recursive step as shown in **Figure 2**, with process detailed in the following list:

1. All predictor features were standardized/normalized for scaling all values in all features with same scale.
2. All standardized features were used for LogReg modeling; each features VIF score was calculated after modelling.
3. The feature with maximum VIF score were checked. If VIF score > 2, the feature was eliminated, and the process was returned to the step one.
4. This iteration process was stopped until all remaining features have VIF score less than 2. All remaining features were considered as the best features that pass into the modelling phase.

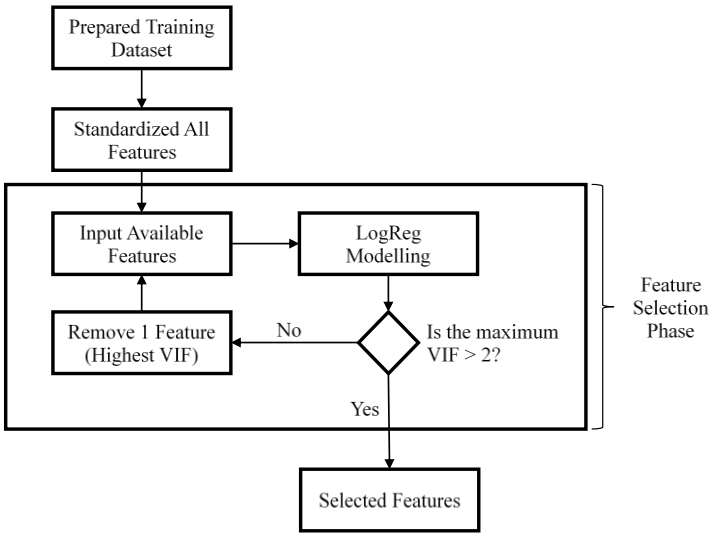

Fig. 2. Multicollinearity checking flow chart.

Recursive Feature Elimination (RFE). Recursive Feature Elimination (RFE) is a recursive step to reduce the dimensionality of dataset based on feature importance (Omar, 2018)[18]. RFE flowchart is shown in **Figure 3**, with process details is shown in this following list:

1. All predictors were used to generate a XGBoost model. Feature important score for each predictor was measured.
2. The AUC score of the model was measured and recorded on a table. The lowest important feature was eliminated, and the process was returned to the step one.
3. This iteration process was stopped until the remaining features are only 2 features. The selected features are the features that produce highest score model with minimum number of features as possible. All selected features were considered as the best features that pass into the modelling phase.

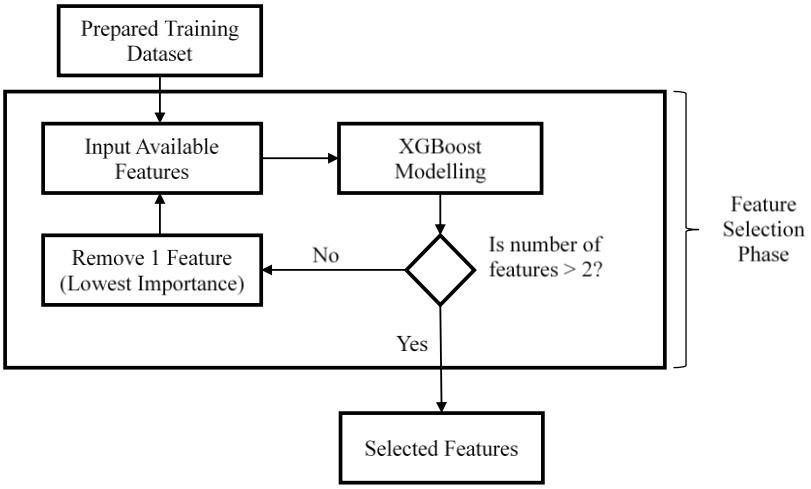

Fig. 3. Recursive Feature Elimination (RFE) flow chart

3.3 Modelling Phase

In modelling phase, two algorithms were used to produce best classification model for predicting churn. Best model for each algorithm can only be determined by cross-validation, a technique to measure or validate the accuracy of a model generated from the training dataset for predicting other data that have not been included in training dataset (prevent overfitting prediction). Cross-validation randomly splits up the data into *k* different groups, also called folds. For each fold, a model was trained on the data not in the fold and then evaluated on the data in the fold. The best set of hyperparameters is the one given by the model with the lowest overall error as computed by averaging the errors from each of the folds (Bruce and Bruce, 2017)[19].

LogReg Modelling with Cross Validation. In LogReg algorithm, there are two types of hyperparameter that are tuned in cross-validation process. Those hyperparameters are listed in Table 3.

Table 3. Logistic Regression hyperparameters

Hyperparameter	Value	Description
Penalty	[l1, l2]	Type of regularization
C	uniform (0,4)	Inverse of regularization strength

LogReg modelling phase flow is explained in **Figure 4** with detail explanation are in the list below:

1. Define a set of LogReg hyperparameters configuration,
2. By using selected features in training dataset, do LogReg modelling 100 times with random selected combinations of hyperparameters and 10-fold cross validation. The best hyperparameters that produce highest ROC score will be saved as the best model configuration.
3. Evaluate the best model by showing all of the model evaluation metrics and scores based on training dataset.

4. Use the best model to predict the label of customers in testing dataset, then evaluate the prediction result by using model evaluation metrics and scores.

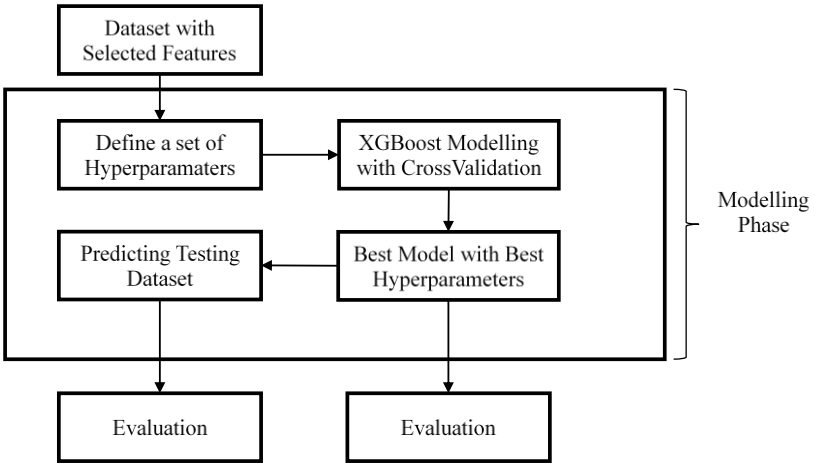

Fig. 4. Logistic Regression modelling phase flow chart

XGBoost Modelling with Cross Validation. There are six types of hyperparameters in XGBoost that need to be tuned by cross-validation technique. Hyperparameter lists are shown in Table 4.

Table 4. Extreme Gradient Boosting hyperparameters.

Hyperparameter	Value	Description
max_depth	[3, 5, 7, 9]	Maximum depth of the tree
min_child_weight	[1, 3, 5]	Minimum sum of instance weight (hessian) needed in a child
gamma	[0.0, 0.33333, 0.25, 0.5, 0.66667, 0.75]	Minimum loss reduction
reg_alpha	[1e-5, 1e-2, 0.1, 1, 100]	L1 regularization term on weights
n_estimators	[100, 200, 300, 500, 750, 1000]	Number of trees
learning_rate	[0.01, 0.015, 0.02, 0.05, 0.08, 0.1]	Step size shrinkage used in update to prevent overfitting

XGBoost modeling phase flow is explained in**Figure 5**. The detail of XGBoost modeling phase steps are listed below:

1. Defining a set of XGBoost hyperparameters configuration,
2. Using selected features in training dataset, do XGBoost modeling 100 times with random selected combinations of hyperparameters and 10-fold cross validation. The best hyperparameters that produce highest ROC score was saved as the best model configuration.
3. Evaluating the best model by showing all of the model evaluation metrics and scores based on training dataset.
4. Using the best model to predict the label of customers in testing dataset, then evaluating the prediction result by using model evaluation metrics and scores.

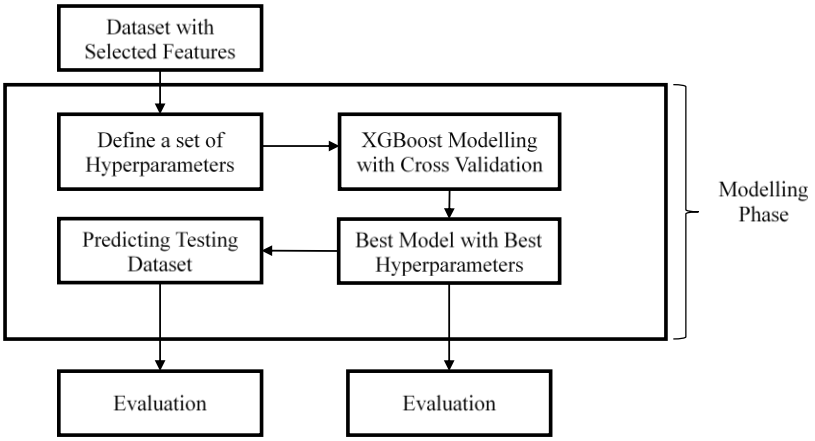

Fig. 5. Extreme Gradient Boosting modelling phase flowchart

3.4 Model Evaluation

Classification model was evaluated by calculating confusion matrix, accuracy, sensitivity, specificity, ROC curve, Kolmogorov-Smirnov chart, and Gains–Lift charts.

Confusion matrix, accuracy, sensitivity, and specificity. Bisri and Wahono (2015)[20] state that classification model is evaluated by its accuracy. Model's accuracy can be measured by a confusion matrix, as can be seen in Table 5. The results of confusion matrix are used to calculate accuracy, sensitivity (true positive rate), and specificity (true negative rate) of prediction resulted by classification model with the formulations shown in equation (4).

Table 5. Confusion Matrix.

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

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}$$

$$\text{Sensitivity} = \text{TP Rate} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Spesificity} = \text{TN Rate} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad \textbf{(4)}$$

Receiver Operating Character (ROC) Curve. Technically ROC curve evaluation is a two-dimensional graphic with true positive level (TP) in the Y axis and false positive level (FP) in the X axis. ROC shows a trade-off between TP and FP. The closer the curve edge to the (0,1) coordinate, the better the model is. The classifier accuracy of diagnostic test is measured by Area Under Curve (AUC) value of ROC that categorized in Table 6 with formulas shown in equation (5) (Saifudin and Romi, 2015)[21]

$$\text{AUC} = \frac{1 + \text{TP}_{\text{rate}} + \text{FP}_{\text{rate}}}{2}, \text{with } \text{FP}_{\text{rate}} = \frac{\text{FP}}{\text{TN} + \text{FP}} \qquad \textbf{(5)}$$

Table 6. Area Under Curve Value Categorization.

AUC	Description
0.90 - 1.00	Excellent Classification
0.80 - 0.90	Good Classification
0.70 - 0.80	Fair Classification
0.60 - 0.70	Poor Classification
< 0.60	Failure

Kolmogorov – Smirnov (KS) Chart. Kolmogorov-Smirnov (KS) is a score that measures how distinct the prediction between two classes is. The KS score is 100 if the model can separate the data between two class accurately, while the KS score is 0 if the model cannot produce distinctive prediction between two classes of data. This score has also been visualized by a chart with two lines presenting each class in the dataset (Stava, 2017)[22].

Gains – Lift Charts. Gains and Lift charts effectively evaluate the classification model by calculating the prediction ratio between the model and without the model (baseline). Gains and Lift charts are mainly concerned with checking the rank ordering of the probabilities by following the following steps:

1. Calculating probability for each observation
2. Ranking these probabilities in decreasing order.
3. Building deciles with each group having almost 10% of the observations.
4. Calculating the response rate at each decile for churn, not-churn, and the total.

Cumulative Gains chart compares cumulative %right (churn) with cumulative %sample, while Lift chart compares between total lift and %sample (Stava, 2017)[23].

4 Results and Discussion

4.1 Data Preparation Result

Data preparation phase reveals two datasets: training dataset and testing dataset with different uses. The details of dataset composition for each label (churn and not-churn) are shown in Table 7. In training dataset, the proportion of not-churned customers compared with churned customers is 9.38 : 1, while the label proportion in testing dataset is 9.49 : 1. These facts confirm that the datasets contain imbalanced-classes problem.

Table 7. Training and testing dataset composition

Dataset	Not-Churn (Y=0)	Churn (Y=1)
Training	66,935	7,050
Testing	28,653	3,056

4.2 Feature Selection Result

Multicollinearity checking processes have selected 31 features with VIF score less than 2. Selected features presented in Table 8 were used for LogReg modeling phase.

Table 8. Selected variable in multicollinearity checking phase

Attribute	Variance Inflation	Status
ALL_REVMINPAY_M6	1.0262	Selected
INT_REVMINPAY_M1	1.0269	Selected
ALL_REVMINPAY_M4	1.0451	Selected
INT_REVMINPAY_M3	1.0547	Selected
ALL_REVMINPAY_M2	1.0632	Selected
ALL_MTTR_M1	1.1296	Selected
INT_PAY_M6	1.147	Selected
ALL_MTTR_M3	1.1516	Selected
ALL_MTTR_M6	1.1849	Selected
ALL_MTTR_M2	1.1853	Selected
ALL_TRB_M6	1.1977	Selected
ALL_TRB_M4	1.2019	Selected
ALL_TRB_M5	1.2027	Selected
ALL_TRB_M1	1.2118	Selected
ALL_TRB_M2	1.2173	Selected
ALL_TRB_M3	1.2562	Selected
ALL_MTTR_M5	1.2791	Selected
ALL_MTTR_M4	1.3229	Selected
LOS_NCLI	1.3606	Selected
R2BB_ANGKA	1.3753	Selected
ALL_REVMINPAY_M5	1.5049	Selected
INT_UPLD_M6	1.5147	Selected

Attribute	Variance Inflation	Status
INT_UPLD_M1	1.5156	Selected
INT_REVMINPAY_M5	1.5339	Selected
INT_PAY_M1	1.5868	Selected
INT_PAY_M3	1.592	Selected
ALL_TRB_MIN6	1.6621	Selected
INT_REV_USE_M1	1.7667	Selected
INT_REV_USE_M6	1.8495	Selected
INT_PAY_M2	1.8727	Selected
ALL_MTTR_MIN6	2	Selected

The RFE process chose 58 best features producing a model with 94.06% accuracy and 0.7319 ROC-AUC score. All important features are shown in **Figure 6**. These selected features are used in XGBoost modeling phase.

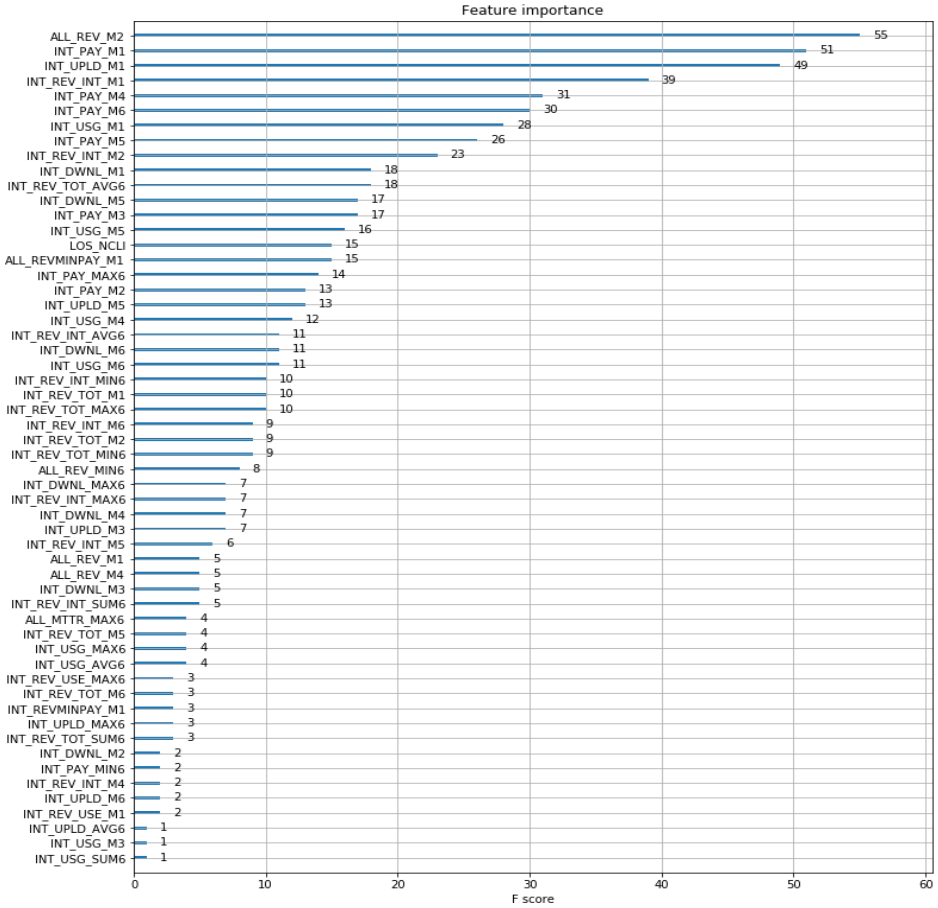

Fig. 6. Selected features in RFE phase

4.3 Modelling Phase Result

Best hyperparameters for LogReg algorithm is shown in Table 9, with ROC-AUC cross validation score is 0.8133. Meanwhile, best hyperparameters for XGBoost algorithm is shown in Table 10, with ROC-AUC cross validation score is 0.9919. All of these hyperparameters configurations are used in evaluation phase both in training dataset and testing dataset.

Table 9. Logistic regression best hyperparameters

Hyperparameter	Value
Penalty	l1
C	3.895
ROC-AUC Score CV	0.8133

Table 10. Extreme gradient boosting best hyperparameters

Hyperparameter	Value
max_depth	9
min_child_weight	5
Gamma	0.25
reg_alpha	0.00001
n_estimators	750
learning_rate	0.02
ROC-AUC Score CV	0.9919

4.4 Model Evaluation (Training Dataset)

4.4.1 Confusion Matrix, Accuracy, Sensitivity, and Specificity

Confusion Matrix, Accuracy, Sensitivity, and specificity score evaluation in training dataset for LogReg algorithm (LR) and XGBoost algorithm (XGB) are shown in Table 11 and Table 12. According to evaluation table, XGBoost algorithm produces classification model with slightly better result than LogReg algorithm based on its accuracy and specificity. However, XGBoost algorithm sensitivity is significantly higher than LogReg; this implies that XGB algorithm can predict churn customers (Y=1) more precisely than LogReg does. It also proves that XGBoost prediction is not affected by imbalanced-class condition in training dataset, while the LogReg algorithm is only focused on majority class, resulting very high specificity but very low sensitivity to which the churn prediction focused on.

Table 11. LR & XGB confusion matrix in training dataset

Algorithm	TP	TN	FP	FN
LR	469	66,634	301	6,581
XGB	5,651	66,829	106	1,399

Table 12. LR & XGB accuracy, sensitivity, and specificity in training dataset

Metrics	LR	XGB
Accuracy	0.9069	0.9797
Sensitivity	0.0665	0.8016
Specificity	0.9955	0.9984

4.4.2 ROC curve, KS Chart, Gains-Lift Charts.

Figure 7 shows the ROC curve and AUC-ROC score for each algorithm. XGBoost has ROC curves with the edge near (1,0) coordinate and larger area under curve, both for two classes rather than LogReg curve; this indicates that XGBoost model is better with higher TP-FP trade-off rate than with LogReg model. The ROC-AUC score for XGBoost (0.9919) is even better than LogReg algorithm ROC-AUC score (0.8133).

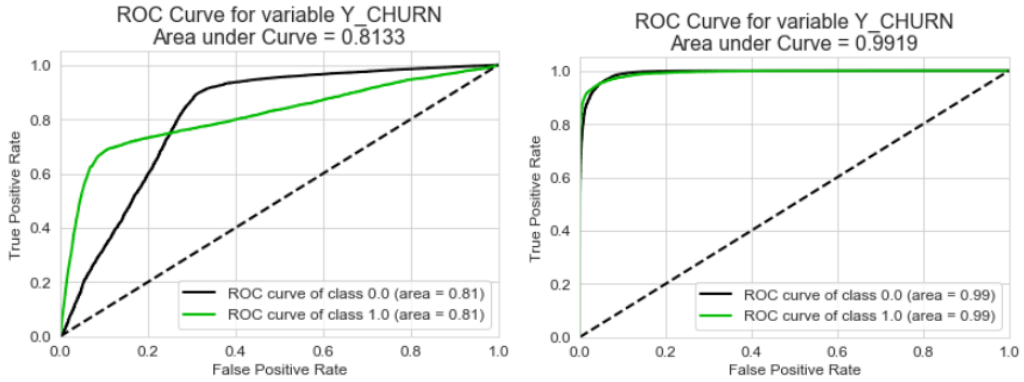

Fig. 7.ROC curve of LR model (left) and XGB model (right) in training dataset

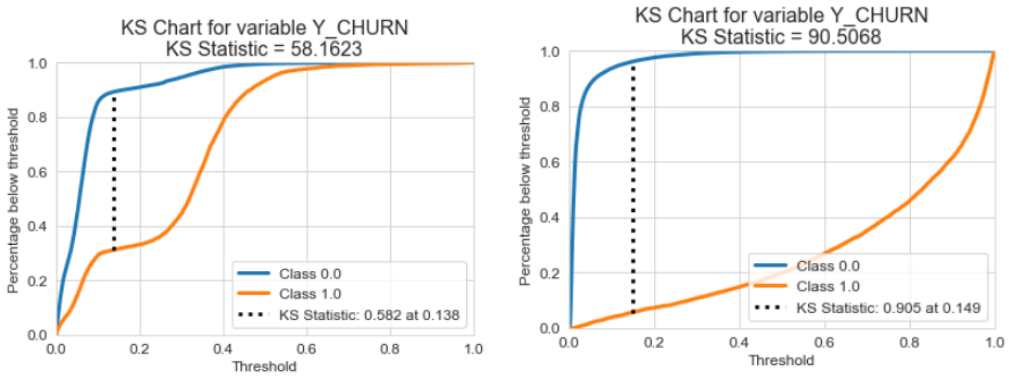

Fig. 8. KS Chart of LR model (left) and XGB model (right) in training dataset

KS charts for both algorithms are shown in *figure 8*, with dashed lines in each figure showing the largest deviation between two class distributions. XGBoost's KS chart has a larger deviation (with score 90.5068) than LogReg's KS (with score 58.1623) indicating that XGBoost model distinguishes the two classes better than LogReg model does.

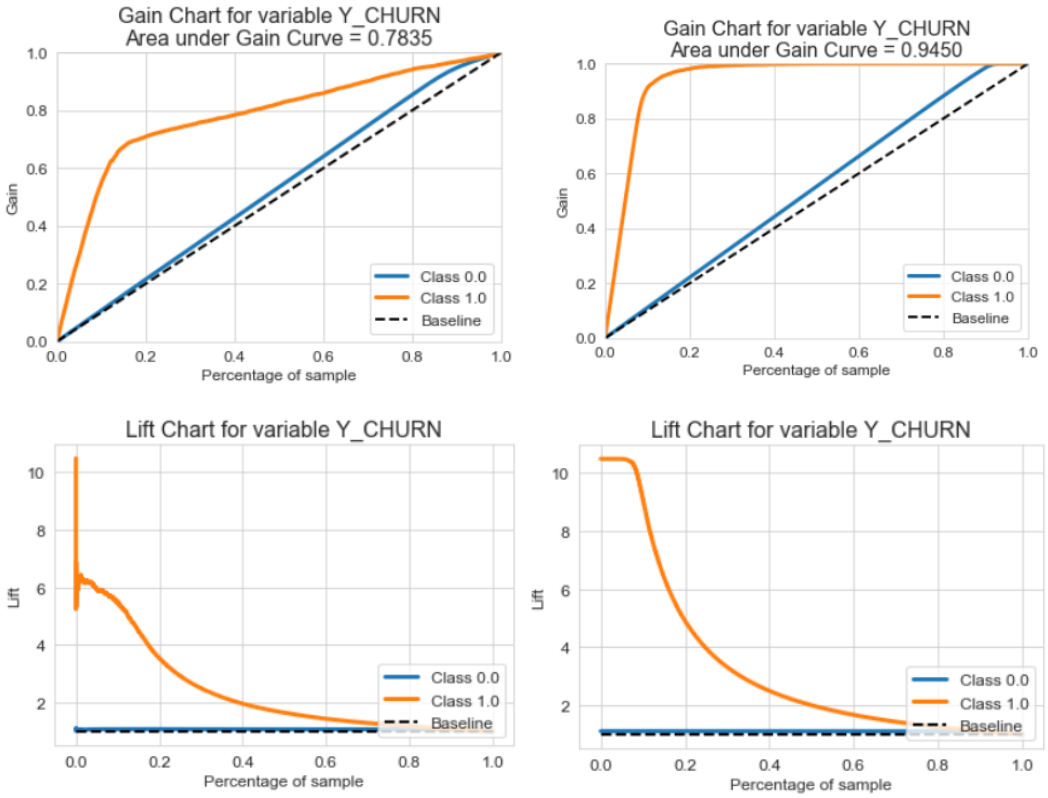

Fig. 9. Gains-Lift charts for LR model (left) and XGB model (right) in training dataset

Gains-Lift charts for both algorithms are shown in **Figure 9**. Based on Gains charts, XGB model has better capabilities to separate two-class efficiently than LogReg model does. For example, top 10% of XGBoost prediction sample (based on its probabilities) contains almost 90% of churn-class sample (Y=1), while top 10% of LogReg model contains less than 70% of churn-class sample. The Lift charts tells that the total lift at top 10% in XGBoost model is higher than LogReg model.

4.5 Model Evaluation (Testing Dataset)

4.5.1 Confusion Matrix, Accuracy, Sensitivity, Specificity, AUC-ROC Score

Evaluation metric values in testing dataset for each model resulted by each algorithm are shown in Table 13 and Table 14. XGBoost model also reveals prediction with higher accuracy, specificity, and specially, sensitivity than LogReg model does, telling that XGBoost model can handle imbalanced-classes problem in testing dataset, yet LogReg is still distracted by majority class and gives very low sensitivity (low TP rate). Meanwhile, XGBoost specificity in testing dataset is decreased to 0.5998 compared with

sensitivity in training dataset (0.8016). It means that real XGBoost model capability to predict the churned customer has been validated not as high as model capability with training dataset used in modeling phase.

Table 13. LR & XGB confusion matrix in testing dataset

Algorithm	TP	TN	FP	FN
LR	254	28,457	196	2,802
XGB	1,833	28,305	348	1,223

Table 14. LR & XGB accuracy, sensitivity, and specificity in testing dataset

Metrics	LR	XGB
Accuracy	0.9055	0.9505
Sensitivity	0.0831	0.5998
Specificity	0.9932	0.9879

4.5.2ROC curve, KS Chart, Gains-Lift Charts.

Figure 10 shows the ROC curve and AUC-ROC score for each algorithm, with XGBoost model gives better prediction with ROC-AUC score (0. 9388) higher than LogReg model (0.8178). The ROC-AUC score is slightly corrected from 0.9919 in training phase to 0.9388 in testing phase.

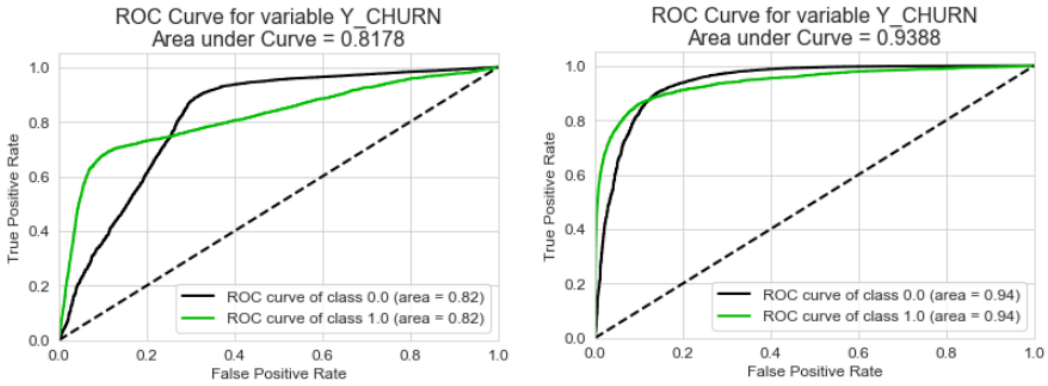

Fig. 10.ROC curve of LR model (left) and XGB model (right) in testing dataset

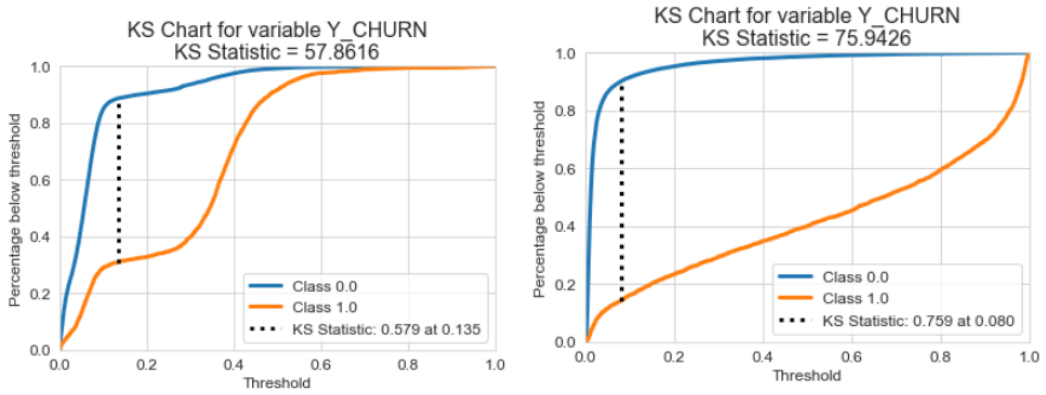

Fig. 11.KS Chart of LR model (left) and XGB model (right) in testing dataset

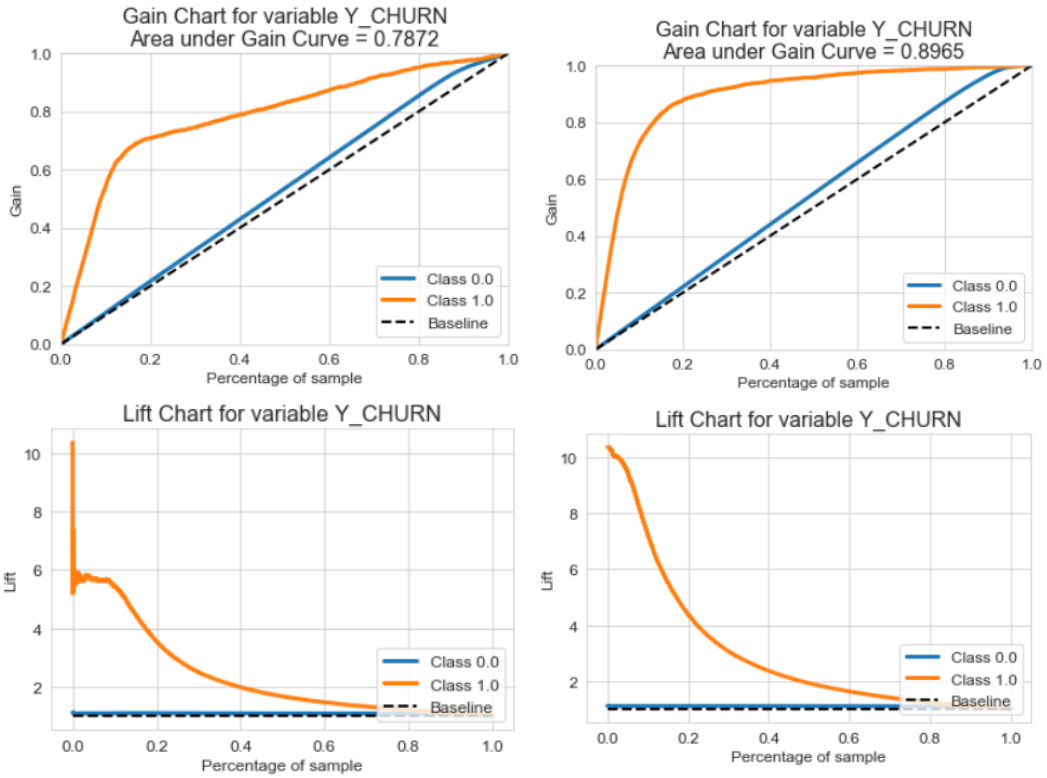

Fig. 12. Gains-Lift charts for LR model (left) and XGB model (right) in testing dataset

KS chart, Gains chart, and Lift chart are shown respectively in **Figure 11** and **Figure 12**. In KS chart, XGBoost model still has a larger deviation between curves (75.9426) than LogReg model (57.8616) does. In Gains chart, 10% top samples of XGBoost model prediction consist of almost 80% of churn-class sample (Y = 10), while 10% top samples of LogReg model contain almost 70% of churn-class sample. In Lift chart, that total lift at top 10% in XGBoost model is also higher than LogReg model. Based on those charts, in can be concluded that XGBoost model capabilities to distinct two classes in testing dataset is better than LogReg model. Compared to training evaluation, XGBoost model capabilities to separate two classes are corrected based on KS statistic (from 90.5068 in training dataset to 75.9426 in testing dataset) and area under curve of Gains chart (0.9450 in training dataset to 0.8965 in testing dataset).

4.6 Conclusion

XGBoost algorithm has been proven to give better prediction compared with LogReg algorithm based on its prediction accuracy, specificity, sensitivity and ROC curve. XGBoost model sensitivity remains high, while LogReg model sensitivity remains very low, indicating that XGBoost can handle imbalanced-classes data better than LogReg does. XGBoost model also has a better capability to separate churn class and not-churn class than LogReg model does according to KS chart and Gains-Lift charts produced by each algorithm. All model evaluations are validated by testing dataset with XGBoost gives a better result than LogReg algorithm does. The decreasing of XGBoost sensitivity from training phase to testing phase indicates that perhaps the XGBoost model is overfitting.

References

[1]Asosiasi Penyedia Jasa Internet Indonesia (APJII): Penetrasi & Perilaku Pengguna Internet Indonesia: Survei 2016 (2016).

[2]Asosiasi Penyedia Jasa Internet Indonesia (APJII): Penetrasi & Perilaku Pengguna Internet Indonesia: Survei 2017 (2017).

[3]Asosiasi Penyedia Jasa Internet Indonesia (APJII): Penetrasi & Profil Perilaku Pengguna Internet Indonesia: Survei 2018 (2018).

[4]Babu S., Ananthanarayanan N. R.: A Review on Customer Churn Prediction in Telecommunication Using Data Mining Techniques. International Journal of Scientific Engineering and Research (IJSER). Vol. 4, no. 1, pp. 35-40 (2016).

[5]Galar M., Fernández A., Barrenechea E., Bustince A: A Reviwew on Ensambles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches. IEEE Transactions on Systems, Man, and Cybernetics: Applications and Reviews, pp. 1-22 (2016).

[6][10][11] Chen T., Guestrin C.: XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August (2016).

[7]Agresti A.: An Introduction to Categorical Analysis, 2nd ed. New Jersey: Wiley (2007).

[8]Hanifa T. T., Adiwijaya, Al-Faraby S.: Analisis Churn Prediction pada Data Pelanggan PT. Telekomunikasi dengan Logistic Regression dan Underbagging. E-Proceeding of Engineering. Vol 4, no. 2, pp. 3210-3224 (2017).

[9][20] Bisri A., Wahono R. S.: Penerapan Adaboost untuk Penyelesaian Ketidakseimbangan Kelas pada Penentuan Kelulusan Mahasiswa dnegan Metode Decision Tree. Journal of Intelligent Systems. Vol 1, no. 1, pp. 27-32 (2015).

[12]Gelman A., Hill J.: Data Analaysis Using Regression and Multilevel/Hierarchial Models. Cambridge: Cambridge University Press (2007).

[13]Hekimoglu S., Koch K.: How can reliability of the robust methods be measured. In: M. Altan (Ed.), Gründige L (eds) Third Turkish-German joint geodetic days (pp. 179–196). Istanbul: Istanbul Technical University (1999).

[14]Akinwande M. O., Dikko H. G., Samson A.: Variance Inflation Factor: As a Condition for the Inclusion of Suppressor Variable(s) in Regression Analysis. Open Journal of Statistics, Vol. 5, pp. 754-767 (2015).
[15]Hair J. F., Jr. AndersonR. E., Tatham R. L., Black W. C.: Multivariate Data Analysis, 3rd ed. , New York: Macmillan Publishing Company (1995).
[16]Adeboye N.O., Fagoyinbo I. S., and Olatayo T.O.: Estimation of the Effect of Multicollinearity on the Standard Error for Regression Coefficients. IOSR Journal of Mathematics, Vol. 10, no. 4, pp. 16-20 (2014).
[17]Allison P.: When Can You Safely Ignore Multicollinearity?. 2012. https://statisticalhorizons.com/ multicollinearity (Acessed 2019-11-03)
[18]Omar K. B. A.: XGBoost and LGBM for Porto Seguro's Kaggle challenge: A comparison. Semester Project. Distributed Computing Group Computer Engineering and Networks Laboratory ETH Zurich. URL: https://pub.tik.ee.ethz.ch/students/2017-HS/SA-2017-98.pdf (Acessed 07/07/2019).
[19]Bruce P., Bruce A.: Practical Statistics for Data Scientists: 50 Essential Concepts. Sebastopool: O'Reilly Media (2017).
[21]Saifudin A., Romi S. W.: Penerapan Teknik Ensemble untuk Menangani Ketidakseimbangan Kelas pada Prediksi Cacat Software. Journal of Software Engineering, Vol. 1, no. 1, pp. 28-37 (2015).
[22][23]Stava T.S.: 7 Important Model Evaluation Error Metrics Everyone should know. 2016. https://www.analyticsvidhya.com/blog/2016/02/7-important-model-evaluation-error-metrics (Acessed 2019-07-09).