BAREKENG: Journal of Mathematics and Its Applications
September 2025 Volume 19 Issue 3 Page 1957-1972
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol19iss3pp1957-1972

APPLICATION OF THE SUPPORT VECTOR MACHINE, LIGHT GRADIENT BOOSTING MACHINE, ADAPTIVE BOOSTING, AND HYBRID ADABOOST-SVM MODEL ON CUSTOMERS CHURN DATA

Felice Elena[1], Robyn Irawan[2*], Benny Yong[3]

[1,2,3]*Center for Mathematics and Society, Faculty of Science, Universitas Katolik Parahyangan*
Jln. Ciumbuleuit No. 94, Bandung, 40141, Indonesia

*Corresponding author's e-mail: * **robynirawan.tjia@unpar.ac.id***

Article History:

Received: 15th January 2025
Revised: 3rd March 2025
Accepted: 8th April 2025
Published: 1st July 2025

Keywords:

Customers churn;
Hybrid AdaBoost-SVM;
LightGBM;
Machine learning;
SVM.

ABSTRACT

A service provider is a business that provides services or the expertise of an individual in a certain sector. A service provider's customer flow could be very dynamic, with both new and churning customers. For the purpose of minimizing the number of churning customers, the company should perform a customer churn analysis. Customer churn analysis is the process of identifying a pattern or trend in churning customers. In order to classify and predict churning customers, machine learning techniques are required to build the classifier model. This paper will use the Support Vector Machine (SVM), Light Gradient Boosting Machine (LightGBM), and hybrid Adaptive Boosting-SVM (AdaBoost-SVM) model. The hybrid AdaBoost-SVM model is a boosting model which uses SVM as its basis classifier instead of a decision tree. The models will be implemented using airlines and telecommunication customers churn data. The usage of oversampling technique is required to balance the number of observations in both classes of training data. Furthermore, a model comparison will be conducted using the F1-Score and the AUC score as the evaluation metric. The analysis shows that LightGBM performs the best result in both dataset with the highest F1-Score and the shortest computational time. In addition, the boosting model AdaBoost-SVM has a better performance than the SVM model due to the boosting algorithm which always minimizes the model error in each iteration. Despite having a better result, AdaBoost-SVM performs in the longest computational time, making it computationally expensive for large datasets. Additionally, the imbalanced nature of the datasets presents challenges in model performance, requiring the application of oversampling techniques to mitigate bias towards the majority class. In conclusion, LightGBM is the best model to classify churning customers based on the higher F1-Score, AUC score, and the shortest computational time.

How to cite this article:

F. Elena, R. Irawan and B. Yong., "APPLICATION OF THE SUPPORT VECTOR MACHINE, LIGHT GRADIENT BOOSTING MACHINE, ADAPTIVE BOOSTING, AND HYBRID ADABOOST-SVM MODEL ON CUSTOMERS CHURN DATA," *BAREKENG: J. Math. & App.*, vol. 19, no. 3, pp. 1957-1972, September, 2025.

Journal homepage: *https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*

Research Article · Open Access

1. INTRODUCTION

A service provider is a company that offers products in the form of people's skills or experience in a certain field. A service provider's primary goal is to help customers through a process. Given the significance of services to businesses, it is essential for companies to provide excellent customer service. When a business offers poor service, customers may leave negative feedback. Customers may decide to switch service providers if the poor services continue to occur over an extended period of time. This is known as customer churn [1].

According to the Cambridge Dictionary, churn rate is the percentage of customers who discontinue using a company's goods or services during a particular time period. The data from one of the telecommunication companies in the United States shows that customer churn rate increased from 2.69% in the first half of 2023 to 2.78% in the second half of the same year, as stated by Statista. For the main purpose of minimizing churn rate, a company should perform a customer churn analysis. This analysis has resulted in the identification of churning consumer characteristics [2]. Furthermore, to recognize this pattern, machine learning techniques are required to create the classifier model [3].

Machine learning is a subset of Artificial Intelligence (AI) which focuses on systems that can learn from data, identify pattern, and create logical decisions with little to no human intervention [4]. The machine learning models used in this paper are Support Vector Machine (SVM), Light Gradient Boosting Machine (LightGBM), and Adaptive Boosting (AdaBoost) due to its great performance as a classifier model [5]. Research on model predictions in bank customer churn resulted in SVM achieved a F1-Score of 86% while AdaBoost model had 63% in its F1-Score [6]. A different case of predicting bank customer churn rate showed LightGBM reached 91.04% in F1-Score [7].

In their applications, those models have a good performance as a classifier model. However, the combination of single learner model (ensemble model) shows a better performance instead of a single learner model [6]. One of the ensemble models is Adaptive Boosting (AdaBoost). This model is an ensemble model which is a combination of weak learners. Commonly, AdaBoost utilize decision tree as its base learner, but AdaBoost could be a hybrid model by combining with the other single learner. A hybrid model AdaBoost-SVM which uses SVM as its base classifier will be implemented and evaluated in this research. In the previous bank customer churn case, AdaBoost-SVM showed a higher F1-Score than its single learner, SVM [6].

The paper will use airlines and telecommunication datasets with binary target variables. While the telecommunication dataset has its binary target variable, the airlines dataset does not have a predefined target variable, requiring the establishment of a churn criterion. Both datasets are classified as imbalanced, with a churn rate of 34% in the airlines dataset and 27% in the telecommunication dataset. Hence, the random oversampling technique is applied to balance the number of churning and non-churning customers. Despite the effectiveness of machine learning in customer churn prediction, selecting the best model remains a challenge due to varying dataset characteristics and model limitations. This study aims to compare the performance of SVM, LightGBM, and AdaBoost-SVM in classifying customer churn, considering both prediction accuracy and computational efficiency, particularly addressing the challenges posed by data imbalance and model runtime constraints.

2. RESEARCH METHODS

There are two datasets used. The first dataset (dataset 1) is a customer churn data in travel industry from Kaggle website. There are 3 sub-datasets which are flights, users, and hotels dataset. As part of this study, we only used the users and flights dataset from 2019 to 2024. This dataset did not come with a target variable (churn) and therefore, the churn criterion needed to be set. Churn is a prolonged period of inactivity [8]. Therefore, from 2019 to 2024, if a customer does not have any transaction in two years or more, the customer is considered to be churning customer. The desired churn criterion resulting in 34% customers churns rate.

The second dataset (dataset 2) is a telecommunication customer churn data from Kaggle website. This dataset already comes with its target variable (churn). The second dataset is added as a comparison to evaluate the model performance in dataset which comes with or without target variable. Both datasets have a binary

target variable with the value 1 is for churning customers and the value -1 is for customers who do not churn. As a result, this dataset has 27% telecommunication customers churn rate.

2.1 Data Description

The airlines customers churn dataset consists of 13 input variables and 271,888 rows. The variables are described in the **Table 1** below.

Table 1. Airlines Customers Churn Variables Description

Variable Name	Value	Description
userCode	[0, 1339]	Passenger ID
company	"4You", "Acme Factory", "Umbrella LTDA", "Wonka Company", "Monster CYA"	Passenger's company name
name	1,400 passengers name	Passenger's name
gender	"male", "female", "none"	Passenger's gender
age	[21, 65]	Passenger's age
travelCode	[0, 135943]	Flight code
from	"SC", "SE", "MS", "DF", "PE", "RN", "SP"	Abbreviations of origin city
to	SC", "SE", "MS", "DF", "PE", "RN", "SP"	Abbreviations of destination city
flightType	"economic", "premium", "firstClass"	Type of seat
price	[301.51, 1754.17]	Price per flight (R$)
time	[0.44, 2.44]	Duration per flight (hours)
distance	[168.22, 937.77]	Distance per flight (km)
date	[26/09/2019, 24/07/2023]	Date of flight

***Data source**: kaggle.com/code/suelin/customer-churn-in-travel-industry/notebook*

With 271,888 rows in the airline's customer churn data, every row describes a single trip flight. Therefore, the rows are merged into round trip flight and the data will only have 135,944 rows. With another process of pre-processing data, the final number of rows in dataset 1 became 135,741. Since every passenger in dataset 1 has more than one round trip flights, the data become repetitive and it will affect the process of building model. Thus, another data processing method is performed to combine all flights in each of the customers, resulting in each row only describe a single passenger information. After performing the data pre-processing, the number of passengers reduced from 1,340 to 1,333 passenger. As a result, the modified airlines customer churn data (dataset 1b) will also be used in this paper.

The telecommunication customers churn dataset consists of 20 input variables and 7,032 rows. The variables are described in the **Table 2** below.

Table 2. Telecommunication Customers Churn Variables Description

Variable Name	Value	Description
CustomerID	7,032 unique ID	Customer ID
Gender	"Male", "Female"	Customers' gender
SeniorCitizen	{0, 1}	Senior citizen yes/no

Variable Name	Value	Description
Partner	"Yes", "No"	Has partner yes/no
Dependents	"Yes", "No"	Has dependents yes/no
Tenure	[0, 72]	Duration of subscription (months)
PhoneService	"Yes", "No"	Has phone service yes/no
MultipleLines	"Yes", "No", "No internet service"	Has multiple lines yes/no
InternetService	"No", "Fiber optic", "DSL"	Type of internet service
OnlineSecurity	"Yes", "No", "No internet service"	Has online security yes/no
OnlineBackup	"Yes", "No", "No internet service"	Has online backup yes/no
DeviceProtection	"Yes", "No", "No internet service"	Has device protection yes/no
TechSupport	"Yes", "No", "No internet service"	Has tech support yes/no
StreamingTV	"Yes", "No", "No internet service"	Has a TV streaming yes/no
StreamingMovies	"Yes", "No", "No internet service"	Has movies streaming yes/no
Contract	"Month-to-month", "One year", "Two year"	Type of subscription contract
PaperlessBilling	"Yes", "No"	Use paperless billing yes/no
PaymentMethod	"Bank transfer", "Credit card", "Electronic check", "Mailed check"	Type of payment method
MonthlyCharges	[18.3, 119]	Amount of monthly charges
TotalCharges	[18.8, 8964.8]	Amount of total charges during subscription
Churn	"Yes", "No"	Churn yes/no

***Data source**: kaggle.com/datasets/blastchar/telco-customer-churn/data*

Both datasets are divided into 70% and 30% train-test data with detail amount of each data is reported in **Table 3**. Furthermore, the train and test data are standardized separately in order to prevent any data leakage. Data leakage is a term used when the data from outside (not a part of training dataset) is used for the learning process of the model **[9]**.

Since both datasets are considered as an imbalance dataset, the sampling technique is required to balance the number of data in both classes. Research from Jordan University of Science and Technology shows that oversampling is a better sampling technique than undersampling **[10]**. Oversampling works in a way where it duplicates the data from the minority class to balance the number of data in the majority class.

Table 3. Airlines Customers Churn Variables Description

Dataset	Number of Data	Training Data	Test Data
1	135,741	89,488	46,253
1b	133,33	933	400
2	7,032	4,922	2,110

Using all three datasets, the models SVM, LightGBM, AdaBoost, and hybrid AdaBoost-SVM will be implemented. The evaluation metrics for model performance are the F1-Score and Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) and Precision-Recall (PR) curves. The F1-Score is particularly important in imbalanced classification problems, as it represents the essential mean of precision and recall **[11]**. A high F1-Score indicates a good balance between precision, which implies the proportion of correctly predicted churn cases out of all predicted churn cases, and recall that implies the proportion of correctly predicted churn cases out of all actual churn cases. Therefore, F1-Score is considered as a more reliable metric than accuracy when dealing with datasets where one class is significantly larger than the other.

2.2 Support Vector Machine (SVM)

Support Vector Machine is a widely use machine learning model used for classification analysis. The model works by finding the optimal hyperplane which could identify and separate data in two classes. The optimal hyperplane is the hyperplane which maximize the margin between support vectors. Support vectors are the closest data points to the hyperplane.

Define $\vec{w}$ as a weight vector, b as a bias, and $\vec{x}$ is the data points vector. The optimal hyperplane is defined as

$$y = \vec{w} \cdot \vec{x} + b \tag{1}$$

To find the weight vector and bias to maximize the margin, SVM works in solving optimization problem written as

$$\begin{gathered} \min_{\vec{w},b} \frac{1}{2}\|\vec{w}\|^2 \\ subject\ to\ f_i \geq 1,\ \ i = 1, \ldots, n \end{gathered} \tag{2}$$

and $f_i = y_i(\vec{w} \cdot \vec{x}_i + b)$ is a functional margin of the SVM.

As a default, SVM use the hard margin approach which means data points has to be linearly separable and completely avoiding any misclassification. However, outliers, empty value, or any data can easily be found in the real application. Those noisy data can hinder the process of finding the optimal hyperplane. Therefore, the usage of soft margin approach is needed to allow any misclassification within a certain tolerance limit. With the regularization parameter C and slack variable τ, the soft margin formulation can be written as

$$\begin{gathered} \min_{\vec{w},b} \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{n}\tau_i \\ subject\ to\ y_i(\vec{w} \cdot \vec{x_i} + b) \geq 1 - \tau_i, \\ \tau_i \geq 0,\ \ i = 1, \ldots, n. \end{gathered} \tag{3}$$

One of the methods of solving the soft margin formulation is using the Lagrange Multiplier. Define $\vec{\lambda}$ as a Lagrange Multiplier for n constrains. The optimization problem is defined as

$$\begin{gathered} \max_{\vec{\lambda}}\ \sum_{i=1}^{n}\lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j\vec{x}_i\vec{x}_j y_i y_j \\ subject\ to\ 0 \leq \lambda_i \leq C, i = 1, \ldots, n \\ \sum_{i=1}^{n}\lambda_i \cdot y_i = 0\,. \end{gathered} \tag{4}$$

After calculating the Lagrange multiplier and substituting the $\vec{\lambda}$ to $\vec{w} = \sum_{i=1}^{n}\lambda_i \cdot \vec{x_i} \cdot y_i$ and $b = y_i - \vec{w} \cdot \vec{x}_i$, the optimal hyperplane which maximize the margin can be obtained [12].

2.2.1 Kernel

SVM is essentially worked by classifying data which is linearly separable. For non-linearly separable, SVM used a function called kernel function [13]. Kernel is used to transform data points into higher dimension in order for the data to be linearly separable. There are three types of kernel functions: [14]

1. **Linear kernel** is the most common and simplest form of kernel function which defined as

$$k(\vec{x}_i, \vec{x}_j) = \vec{x}_i \cdot \vec{x}_j \tag{5}$$

with

$$\vec{x}_i \cdot \vec{x}_j = \sum_{m=1}^{n}(x_{im} \cdot x_{jm}) \tag{6}$$

Polynomial kernel uses the constant r, a gamma parameter γ, and polynomial degree d which defined as

$$k(\vec{x}_i, \vec{x}_j) = (\gamma \cdot \vec{x}_i \cdot \vec{x}_j + r)^d \tag{7}$$

The gamma parameter defines the width or slope of the kernel function. When the gamma value is low, the decision region becomes very broad and make the classification very general [15].

2. A more complex kernel function, a **Radial Basis Function** (RBF) is a kernel function which its value depends on the distance from the origin. RBF Kernel is defined as

$$k(\vec{x}_i, \vec{x}_j) = e^{\left(-\gamma \|\vec{x}_i - \vec{x}_j\|^2\right)} \tag{8}$$

with

$$\|\vec{x}_i - \vec{x}_j\|^2 = \sum_{m=1}^{n} (x_{im} - x_{jm})^2 \tag{9}$$

2.3 Adaptive Boosting (AdaBoost)

Boosting is one of the algorithms in machine learning which combines weak learner to enhance the performance of the learner [16]. For every iteration, the errors from weak learner classifiers are minimized by the boosting algorithm. AdaBoost is one of the boosting algorithms which needs a base classifier. By default, the base classifier for AdaBoost is decision tree [7].

Initially, all data points are given the equal weight and are applied to the weak learner. AdaBoost minimized errors by increasing the weight for misclassified data points, and reducing weight for data points which is correctly classified. Therefore, the model will emphasize more on data points with higher weight, preventing them to get misclassified. The procedure of AdaBoost is shown in Algorithm 1 [17].

Algorithm 1. Adaptive Boosting (AdaBoost) Algorithm

Input: Training data $\mathcal{D} = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_n, y_n)\}$ where $y_i \in \{-1, +1\}$, number of iteration (T)

1: Initialize the weights $D_1(\vec{x}_i) = \frac{1}{n}$ for $i = 1, \dots, n$.
2: **For** $t = 1, \dots, T$ **do**:
3: Input training data $\mathcal{D}$ on weak learner using weights $D_t(\vec{x}_i)$.
4: Define $h_t(\vec{x}_i)$ as the weak learner classifier results on data $\vec{x}_i$ at the t-th iteration.
5: Compute $\varepsilon_t = \sum_{i=1}^{n} D_t(\vec{x}_i) \cdot I\left(y_i \neq h_t(\vec{x}_i)\right)$.
6: Compute $\alpha_t = \frac{1}{2}\log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$.
7: Update weights $D_{t+1}(\vec{x}_i) = D_t(\vec{x}_i) \cdot \exp\left(-\alpha_t \cdot y_i \cdot h_t(\vec{x}_i)\right)$.
8: Renormalize the weights $D'_{t+1}(\vec{x}_i) = \frac{D_t(\vec{x}_i)}{\sum_{i=1}^{n} D_{t+1}(\vec{x}_i)}$.
9: Increment the iteration counter $t = t + 1$.
10: **End for**
11: **Output**: $H(\vec{x}) = sign\left(\sum_{t=1}^{T} \alpha_t \cdot h_t(\vec{x})\right)$.

The function $I : \mathbb{R} \to \{0,1\}$ in line 5 is a Heaviside function which the value is 1 if $y_i \neq h_t(\vec{x}_i)$ and the value is 0 if $y_i = h_t(\vec{x}_i)$. In further explanation, the error value $\varepsilon_t = 1$ if the original target variable value is different from the weak learner result (misclassified) and $\varepsilon_t = 0$ if the original target variable value is the same as the weak learner result (correctly classified).

2.4 Light Gradient Boosting Machine (LightGBM)

Gradient Boosting Machine (GBM) is an algorithm to construct a model to be maximally correlated with the negative gradient of the loss function. In GBM, the most used base learner are linear models, smooth models, and decision tree. In this paper, the decision tree GBM will be used [18] [19].

Gradient Boosting Decision Tree (GBDT) is an ensemble model of decision trees which are trained in an iteration process. In each iteration, GBDT learns the decision tree model by fitting the negative gradients (residual error) [20]. During the learning process, one of the inefficient parts of GBDT is the long computational process in determining split points for decision tree. Therefore, the techniques Gradient-based One Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) are introduced [21].

Both techniques are required to deal with the large number of data which could be a problem during the process in decision tree. GOSS technique only keeps all the data points with large gradients (misclassified

data points) and only choose a random sampling on data with small gradients. On the other hand, EFB algorithm can bundle several similar variables to reduce the number of variables. This algorithm of GBDT using GOSS and EFB is called the Light Gradient Boosting Machine (LightGBM). The algorithm of LightGBM is shown in Algorithm 2 [22].

Algorithm 2. Light Gradient Boosting Machine (LightGBM) Algorithm

Input: Training data $\mathcal{D} = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_n, y_n)\}$ where $y_i \in \{-1, +1\}$, number of iteration (M), loss function $\mathcal{L}(y_i, p)$, sampling ratio of large gradient data (a), sampling ratio of small gradient data (b).

1: Merge mutually exclusive feature using Exclusive Feature Bundling (EFB) technique.
2: Initialize $F_0(\vec{x}) = arg\min_p \mathcal{L}(y_i, p)$.
3: **For** $m = 1, \dots, M$ **do**:
4: Compute gradients $g_i = -\left[\frac{\partial \mathcal{L}(y_i, F(\vec{x}_i))}{\partial F(\vec{x}_i)}\right]_{F(\vec{x})=F_{m-1}(\vec{x}).}$
5: Resampling data using GOSS technique.
6: Compute information gains
$$V_j(d) = \frac{1}{n}\left(\frac{1}{n_l^j(d)}\left(\sum_{\vec{x}_i \in A_l} g_i + \frac{1-a}{b}\sum_{\vec{x}_i \in B_l} g_i\right)^2 + \frac{1}{n_r^j(d)}\left(\sum_{\vec{x}_i \in A_r} g_i + \frac{1-a}{b}\sum_{\vec{x}_i \in B_r} g_i\right)^2\right)$$
7: Construct a new decision tree $F_m{}'(\vec{x})$ on resampled dataset.
8: Update model $F_m(\vec{x}) = F_{m-1}(\vec{x}) + F_m{}'(\vec{x})$
9: **End for**
10: **Output**: $F_M(\vec{x})$ after M iterations.

The information gains on line 6 is an important calculation to determine which nodes to split in decision tree model. Based on the information gains value, decision tree model will split each node at the most informative feature (largest information gain) [20].

2.5 Hybrid Model AdaBoost-SVM

AdaBoost is one of the ensembles boosting techniques. Yoav Freund and Robert Schapire created this approach where it combines weak learners to form a strong predictive model. The default weak learner for AdaBoost is decision tree, however any other weak learners are allowed to be combined with AdaBoost. The proposed algorithm for this paper is using Support Vector Machine (SVM) as the weak learner for AdaBoost. Several researches have applied the hybrid model AdaBoost-SVM and successfully resulting in better performance compared to SVM model as a single learner and AdaBoost model, as cited in [6] and [7].

The algorithm of this hybrid model is similar to the AdaBoost algorithm. However, when calculating the result from the predictive model, instead of using decision tree, the model uses SVM. At first, each data is given the same weight. Next, the data is applied to the SVM model and the result for every data point is compared to the original target variable value. Continued by calculating the error, importance score, updating new weights to the misclassified data (reweighted), and reshuffle the data to redo the first iteration. The whole process of this hybrid model is explained in Algorithm 3 below.

Algorithm 3. Adaptive Boosting – Support Vector Machine (AdaBoost-SVM) Algorithm

Input: Training data $\mathcal{D} = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_n, y_n)\}$ where $y_i \in \{-1, +1\}$, number of iteration (T), kernel function $(k(\vec{x}_i, \vec{x}_j))$, stopping error percentage (ε).

1: Set $t = 0$.
2: Initialize the weights $D_1(\vec{x}_i) = \frac{1}{n}$ for $i = 1, \dots, n$.
3: **While** $t \leq T$ **and** $\varepsilon_t \leq \varepsilon$ **do**:
4: Solving SVM optimization problem to obtain the Lagrange multiplier (λ).
5: Compute the optimal hyperplane $y = \vec{w} \cdot \vec{x} + b$.
6: Define $h_t(\vec{x}_i)$ as the SVM classifier results on data $\vec{x}_i$ at the t-th iteration.
7: Compute $\varepsilon_t = \sum_{i=1}^{n} D_t(\vec{x}_i) \cdot I\left(y_i \neq h_t(\vec{x}_i)\right)$.

Algorithm 3. Adaptive Boosting – Support Vector Machine (AdaBoost-SVM) Algorithm

Compute $\alpha_t = \frac{1}{2}\log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$.

Update weights $D_{t+1}(\vec{x}_i) = D_t(\vec{x}_i) \cdot \exp\left(-\alpha_t \cdot y_i \cdot h_t(\vec{x}_i)\right)$.

Renormalize the weights $D'_{t+1}(\vec{x}_i) = \frac{D_t(\vec{x}_i)}{\sum_{i=1}^{n} D_{t+1}(\vec{x}_i)}$.

Increment the iteration counter $t = t + 1$.

End while

Output: $H(\vec{x}) = sign\left(\sum_{t=1}^{T} \alpha_t \cdot h_t(\vec{x})\right)$.

3. RESULTS AND DISCUSSION

This section will cover the three models result and parameter chosen on repetitive airlines customer churn data (dataset 1), modified airlines customer churn data (dataset 1b), and telecommunication customer churn data (dataset 2). Furthermore, the discussion regarding which criteria affects the most in customer churn using variable importance will also be covered.

3.1 Case Study Result on Repetitive Airlines Customer Churn Data (Dataset 1)

On dataset 1, the parameter chosen for each of the model is based on the performance of F1-Score, AUC ROC curve score, and AUC PR curve score. The usage of these three metrics is because these metrics are commonly used to evaluate the model performance on imbalance data. The parameter chosen on this dataset is provided in **Table 4**. Hyperparameters for all models, including learning rates, number of estimators, types of kernels, and gamma, have been tuned in an iterative manner. This means that we applied a grid search strategy where different ranges of values were applied for each parameter. The models were then trained and tested on those sets of parameters, and we chose the optimal combinations with the highest F1-score and AUC values. This method ensures that all models run under the best conditions for meaningful comparison.

Table 4. Parameter Chosen of Each Model using Dataset 1

Model	Learning Rate	n_estimators	Kernel	C	Gamma (γ)
LightGBM	0.1	100	-	-	-
AdaBoost	0.01	20	-	-	-
AdaBoost-SVM	0.1	50	Polynomial	0.01	10
SVM	-	-	Polynomial	0.1	0.1

Using the parameters above, the model result and its computational time are stated in the **Table 5** below.

Table 5. Model Results using Dataset 1

Model	Accuracy	F1-Score	Precision	Recall	AUC-ROC	AUC-PR	Time
LightGBM	0.732	0.6381	0.5912	0.6929	0.81	0.72	2.5 s
AdaBoost	0.7158	0.5712	0.5881	0.5553	0.74	0.59	21 s
AdaBoost-SVM	0.5552	0.4432	0.3865	0.3865	0.57	0.4	99.6 h
SVM	0.559	0.4443	0.3894	0.5172	0.54	0.39	3 h

One remarkable finding on **Table 5** is the much greater computational time of AdaBoost-SVM (99.6 hours) compared to LightGBM (2.5 seconds). This is primarily due to the iterative algorithm of AdaBoost, which iteratively trains a number of SVM classifiers on reweighted data. Unlike LightGBM, where it speeds up the selection of most important features by employing gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB), AdaBoost-SVM is required to train one SVM model for each boosting

iteration. Because SVM itself is computationally expensive, particularly when there are non-linear kernels and huge data, repeated training makes the runtime significantly high.

Results on **Table 5** concludes that all four models did not perform well to predict and classify the data. Despite the low F1-Score, the AUC score from ROC in LightGBM model successfully reaches 0.8. This means that the LightGBM model has a well performance in classifying the data into its positive or negative class. However, with the low score in AUC score in PR curve, it shows that LightGBM could not balance its precision and recall score.

Some of the reasons for these results are the imbalanced data and the repetitive dataset. Even after using the oversampling technique for training dataset, models seem to not be able to classify the data well. Data exploration process is used to check the data pattern to identify the repetitive data. The number of flights for each passenger are shown in the histogram below.

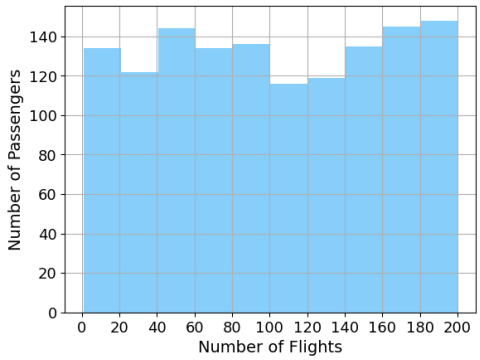
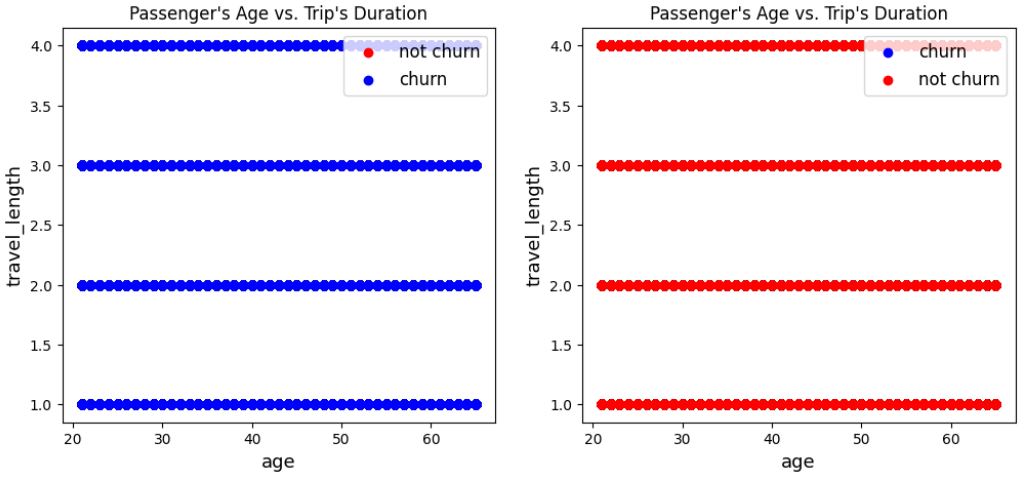


Figure 1. Histogram of number of flights

Figure 1 shows that every passenger has at least booked one flight during 2019-2024 which caused the data to be repetitive. For example, if one passenger has booked 130 flights in the time period, it means there will be 130 rows of same passenger characteristics with all 130 different flights information. Another data exploration in **Figure 2** shows a scatterplot of two variables to identify if there is a certain characteristic for churning and not churning customer.



(a) **(b)**

Figure 2. Scatterplot Showing the Relationship between Passenger Age and Trip Duration for Churning (Blue) and Non-Churning (Red) Customers
(a) Churning Customers, (b) Not Churning Customers

Every round-trip flight has a trip duration between 1 day to 4 days which explains the scatterplot shape. Both scatterplots show the relationship between passenger's age and trip's duration. However, the scatterplots are shown side by side because the points are both overlap to each other. On the left, the non- churning customers which shown in red dots are exactly behind the blue dots. This situation also applies to the right scatterplot which the blue dots are behind the red dots. The scatterplot has shown a case of repetitive dataset.

These scatterplots explain why all the four models are not able to classify the data well enough. Every model is struggling to classify between churning and non-churning customer due to very similar characteristics. Therefore, another pre-processing data technique is required to summarize the data for every customer, not for every flight.

3.2 Case Study Result on Modified Airlines Customer Churn Data (Dataset 1b)

Despite using the same dataset, the parameter used in each model are different from the previous parameter in Dataset 1. This is due to the different process in data pre-processing. The parameter used in modified airlines customer churn data is described in **Table 6**.

Table 6. Parameter Chosen of Each Model using Dataset 1b

Model	Learning Rate	n_estimators	Kernel	C	Gamma (γ)
LightGBM	0.3	50	-	-	-
AdaBoost	0.3	50	-	-	-
AdaBoost-SVM	0.3	50	Polynomial	0.1	0.1
SVM	-	-	Polynomial	0.1	0.1

Using the parameters above, the model result and its computational time are stated in the **Table 7** below.

Table 7. Model Results using Dataset 1b

Model	Accuracy	F1-Score	Precision	Recall	AUC-ROC	AUC-PR	Time
LightGBM	0.98	0.9827	0.966	1	0.976	0.966	0.06 s
AdaBoost	0.98	0.9827	0.966	1	0.976	0.966	0.04 s
AdaBoost-SVM	0.9725	0.9762	0.9617	0.99	0.998	0.999	12.5 s
SVM	0.8975	0.9175	0.8475	1	0.997	0.998	0.5 s

From the **Table 7**, we can conclude that the models have a great performance in identifying and predicting churning and non-churning customers. Different from the result in the previous dataset, this dataset which concludes every customer information in every row, does not have a repetitive information. This also supported by the scatterplot shown in **Figure 3**.

The scatterplot shows several relationships between two numeric variables in the data. It shows that there is a linear relationship and a distinct difference between two types of passengers. Therefore, the models could easily classify the passengers into two classes.

Comparing the AdaBoost-SVM and SVM model, there is a clear increase in all score except the recall score. This shows that the boosted SVM model (AdaBoost-SVM) produces a better result. However, considering the computational time, LightGBM is the fastest. This is due to the GOSS (Gradient-based One Side Sampling) used to only uses data with large gradients and randomly sampling the small gradients data as a decision tree split criterion. With only choosing the large gradient data, model will process the data more efficiently since not all data is used.

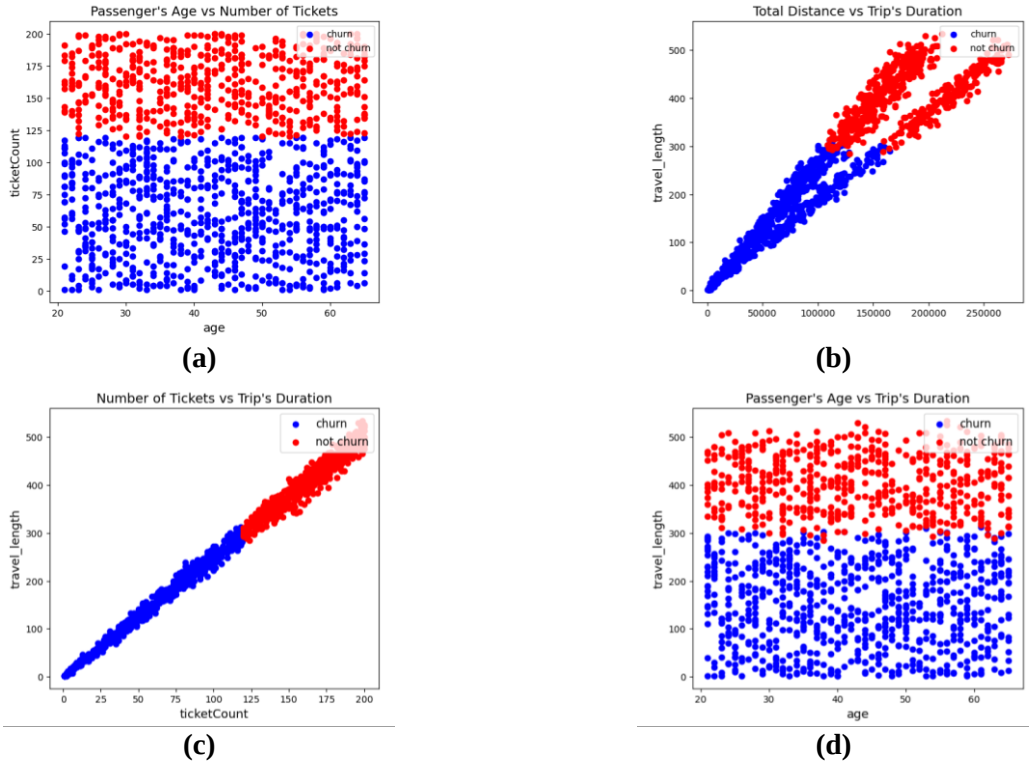


Figure 3. Scatterplots Showing Relationships between Various Passenger and Travel-Related Variables in the Modified Airline Customer Churn Dataset
(a) Number of Tickets vs Passenger's Age, (b) Travel's Duration vs Total of Distance,
(c) Travel's Duration vs Number of Tickets, (d) Travel's Duration vs Passenger's Age

3.3 Case Study Result on Telecommunication Customer Churn Data (Dataset 2)

This dataset is not a repetitive dataset because each row represents a different customer and it makes it has a more random data pattern than the previous dataset. Furthermore, this telecommunication dataset has already come with its target variable (churn) which does not need to create a churn criterion as what the flight dataset has. The parameter chosen on this dataset is provided in **Table 8**.

Table 8. Parameter Chosen of Each Model using Dataset 2

Model	Learning Rate	n_estimators	Kernel	C	Gamma (γ)
LightGBM	0.1	50	-	-	-
AdaBoost	0.05	50	-	-	-
AdaBoost-SVM	0.01	70	Polynomial	0.01	1
SVM	-	-	Polynomial	0.1	0.1

Using the parameters above, the model result and its computational time are stated in the **Table 9** below.

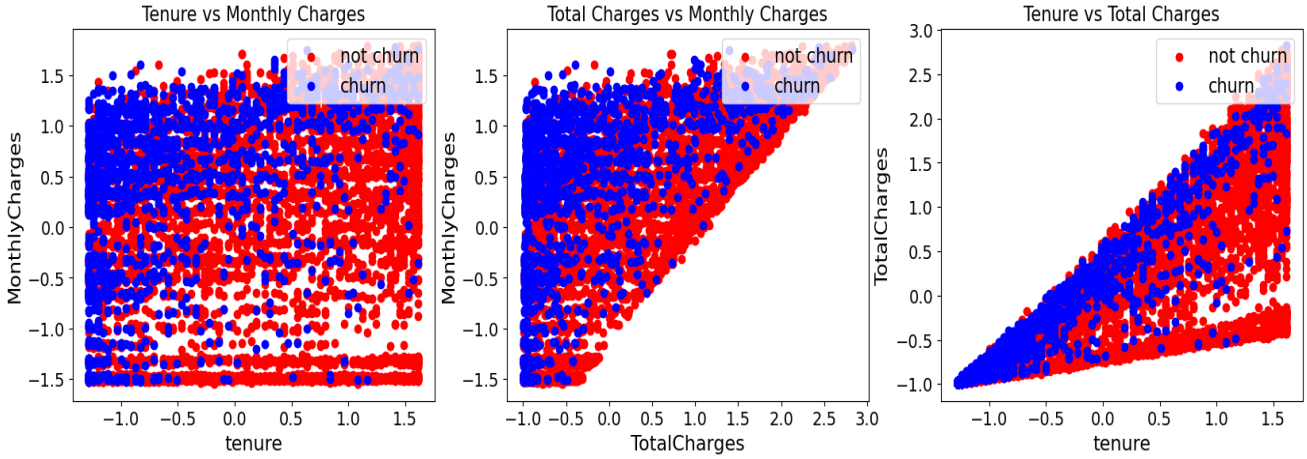
Table 9. Model Results using Dataset 2

Model	Accuracy	F1-Score	Precision	Recall	AUC-ROC	AUC-PR	Time
LightGBM	0.6459	0.5714	0.417	0.9071	0.83	0.654	1.2 s
AdaBoost	0.7653	0.5349	0.5477	0.5227	0.76	0.51	3 s
AdaBoost-SVM	0.7462	0.6167	0.5034	0.7959	0.83	0.653	20 m
SVM	0.7322	0.6116	0.4911	0.8105	0.82	0.63	20 s

Using a more scattered and more random data as shown in **Figure 4**, model results are shown in various score. A high score in AUC-ROC shows that model could classify well in both classes. On the other hand, a low PR score shows the model inability to balance a high precision and recall score.

A high precision score is achieved if model correctly classify the churning customers (increasing true positive). This could cause model to miss several churning customers and classify it into not churn (increasing false negative resulting in low recall). The same case also applies when trying to increase the recall score, usually the precision is going low. In an imbalance dataset, precision and recall are among the most important scores in model evaluation. Model with a high precision score could perform well in predicting churning customers and it helps the company to lower down their cost, for example, by not giving out discount to the customers that has not potential on becoming churn.

On the other hand, a high recall score assures the model to detect potential churning customers. This could help the company to create a strategy for those potential churning customers. Therefore, based on the score and computational time, the best model for imbalanced dataset is LightGBM.



(a) **(b)** **(c)**

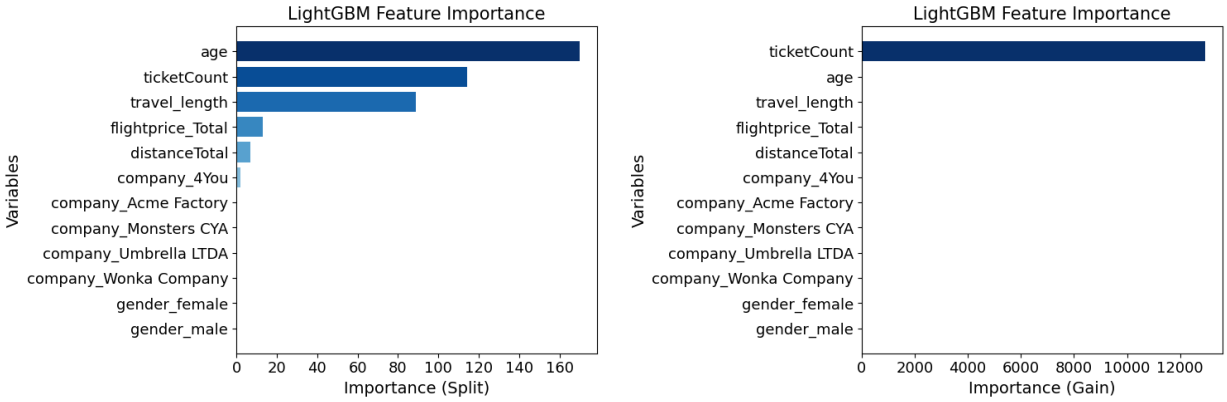
Figure 4. Scatterplot of Various Variable Relationship in Telecommunication Data
(a) Monthly Charges vs Tenure, (b) Monthly Charges vs Total Charges, (c) Total Charges vs Tenure

3.4 LightGBM Feature Importance

In machine learning, feature importance or variable importance is one of the most essential parts in evaluating variable. By calculating feature importance, one can determine which variables should be used in the model to increase the accuracy and model performance. Furthermore, this technique also helps in minimizing the risk of the usage of noisy variables which could interfere the model building process.

In LightGBM, two metrics are used to measure feature importance: split importance and gain importance scores. The split importance score counts how many times a variable is utilized to split the data in the decision tree model. This helps identify which variables are most frequently part of the decision-making process.

The gain importance score measures the improvement in the model's accuracy when a specific variable is used for splitting in the decision tree. This metric offers more valuable insight, as it reflects the quality of the split. **Figure 5** shows the LightGBM feature importance in flight customer churn data (dataset 1b).

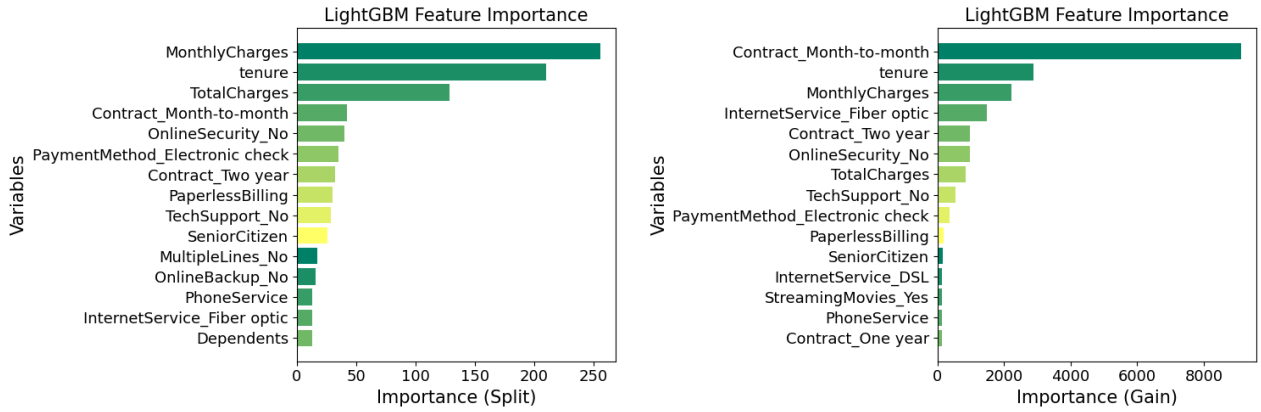


(a) **(b)**

Figure 5. LightGBM Feature Importance in Dataset 1b
(a) Split importance Measurement, (b) Gain Importance Scores Measurement

Based on both split and gain score, the variable “ticketCount” is the most important variable for the target variable. Furthermore, most variables do not have a gain importance score. This shows not every variable that is chosen as a splitting variable in decision tree, could contribute to the improvement of the model performance. Despite the variable “age” has been used for over 160 times as a splitting variable, the variable itself does not contribute to the improvement of the model performance. As for the interpretations, the "ticketCount" feature indicating how many tickets a customer has purchased previously. The higher the tickets, the more it suggests customer loyalty; the lower the tickets, the more it suggests an inactive customer who will churn. Other features, such as "travel duration" and "distance", may also help predict churn by indicating customer preference which short-distance frequent travelers may be less loyal than long-distance travelers.

The feature importance in telecommunication dataset (dataset 2) is shown in **Figure 6**. The figure shows that the condition whether a customer has a month-to-month contract or not is the most influential variable in determining churn customers. Same result also achieved by the variable “tenure” and “MonthlyCharges”. The most influential attribute is "Contract_Month-to-month" states that the customers who have monthly contracts are more likely to churn compared to customers who have annual or multi-year contracts. This makes sense because long-term contracts are less volatile, whereas monthly contracts allow for frequent switching among providers. As for "Tenure" (subscription duration) feature is also a key variable which states customers with low tenure are more likely to churn, and longer-tenured customers are more likely to stay. Lastly, "MonthlyCharges" is also a contributor where more charges will indicate more likely dissatisfaction and churn will occur, especially if customers don't feel they receive enough value for the cost.



(a) **(b)**

Figure 6. LightGBM Feature Importance in Dataset 2
(a) Split Importance Measurement, (b) Gain Importance Scores Measurement

3.5 AdaBoost Feature Importance

Other than utilizing LightGBM, another model AdaBoost can also be used to calculate the feature importance. This is due to both of the models are a tree-based model. Therefore, others model which is not a tree-based model such as SVM and AdaBoost-SVM cannot be used in feature importance. The result of AdaBoost feature importance in dataset 1b is shown in **Figure 7**.

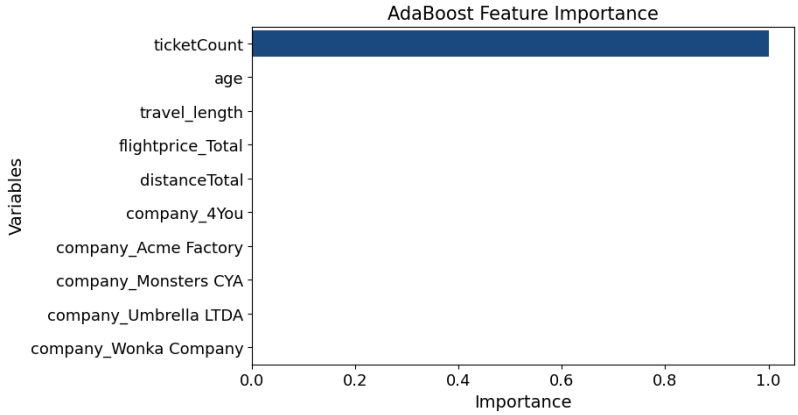


Figure 7. AdaBoost Feature Importance in Dataset 1b

Using AdaBoost feature importance, the same result is obtained as the LightGBM feature importance. The variable "ticketCount" which has the information of number of tickets that has been bought by the passenger during the period is the most importance variable to the customer churn. Furthermore, the AdaBoost feature importance result in telecommunication dataset (dataset 2) is also the same as the LightGBM. The variable "Contract_Month-to-month" has the highest importance score to the target variable. The result is shown in **Figure 8**.

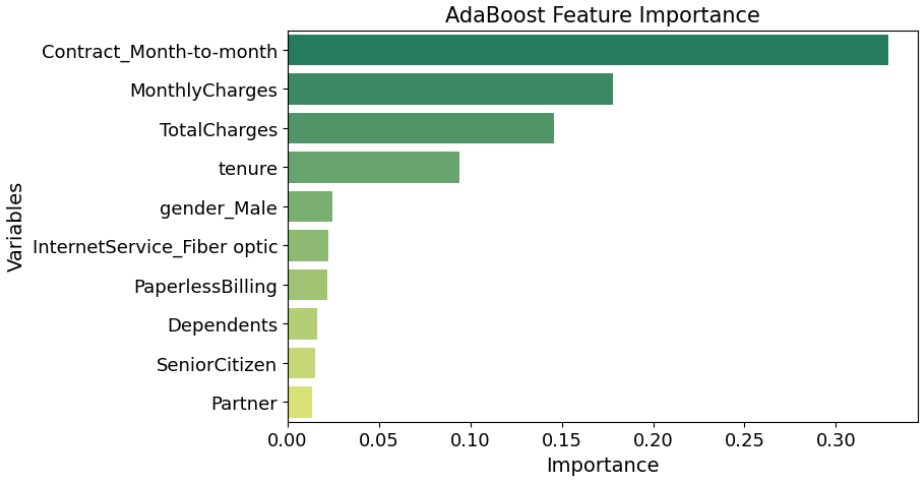


Figure 8. AdaBoost Feature Importance in Dataset 2

4. CONCLUSIONS

Based on the findings in this study, we obtained several conclusions as following.

1. We proposed the AdaBoost-SVM model and evaluated the performance among other common models such as SVM, LightGBM, and AdaBoost. Based on the results obtained, LightGBM performed the best by showing a high F1-Score and the shortest computational time. The proposed model AdaBoost-SVM could not perform better than LightGBM due to its long computational time. However, since AdaBoost-SVM is the boosting model of SVM, it showed a better result than SVM despite having a longer processing time;

2. Based on the telecommunication data, the criteria whether a customer have a month-to-month contract or not, the customer's tenure, and the amount of monthly charges, are three of the top variables to affect the customer churn rate. On the other hand, the number of tickets bought by each passenger is the most important variable in airlines customer's churn data.

REFERENCES

[1] B. Huang, M. T. Kechadi and B. Buckley, "CUSTOMER CHURN PREDICTION IN TELECOMMUNICATIONS," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1414-1425, 2012. doi: https://doi.org/10.1016/j.eswa.2011.08.024

[2] S. Nurhaliza, K. Sadik and A. Saefuddin, "A COMPARISON OF COX PROPORTIONAL HAZARD AND RANDOM SURVIVAL FOREST MODELS IN PREDICTING CHURN OF THE TELECOMMUNICATION INDUSTRY CUSTOMER," *BAREKENG: Jurnal Ilmu Matematika dan Terapan*, vol. 16, no. 4, pp. 1433-1440, 2022. doi: https://doi.org/10.30598/barekengvol16iss4pp1433-1440

[3] A. Shi, C. Y. Lim and S. L. Ang, "CUSTOMER CHURN ANALYSIS FOR LIVE STREAM E-COMMERCE PLATFORMS BY USING DECISION TREE METHOD," in *International Conference on Advanced Communication and Intelligent Systems*, Warsaw, 2023. doi: https://doi.org/10.1007/978-3-031-45124-9_13

[4] M. Wazid, A. K. Das, V. Chamola and Y. Park, "UNITING CYBER SECURITY AND MACHINE LEARNING: ADVANTAGES, CHALLENGES AND FUTURE RESEARCH," *ICT express*, vol. 8, no. 3, pp. 313-321, 2022. doi: https://doi.org/10.1016/j.icte.2022.04.007

[5] D. S. Sisodia, S. Vishwakarma and A. Pujahari, "EVALUATION OF MACHINE LEARNING MODELS FOR EMPLOYEE CHURN PREDICTION," in *2017 international conference on inventive computing and informatics (icici)*, Coimbatore, 2017.doi: https://doi.org/10.1109/ICICI.2017.8365293

[6] S. Dutta, P. Bose, S. Bandyopadhyay and M. Janarthanan, "A HYBRID MACHINE LEARNING MODEL FOR BANK CUSTOMER CHURN PREDICTION," *International Journal of Engineering Trends and Technology*, vol. 70, pp. 13-23, 2022.doi: https://doi.org/10.14445/22315381/IJETT-V70I6P202

[7] N. T. M. Sagala and S. D. Permai, "ENHANCED CHURN PREDICTION MODEL WITH BOOSTED TREES ALGORITHMS IN THE BANKING SECTOR," in *2021 International Conference on Data Science and Its Applications (ICoDSA)*, Bandung, 2021. doi: https://doi.org/10.1109/ICoDSA53588.2021.9617503

[8] Á. Periáñez, A. Saas, A. Guitart and C. Magne, "CHURN PREDICTION IN MOBILE SOCIAL GAMES: TOWARDS A COMPLETE ASSESSMENT USING SURVIVAL ENSEMBLES," in *2016 IEEE international conference on data science and advanced analytics (DSAA)*, Montreal, 2016. doi: https://doi.org/10.1109/DSAA.2016.84

[9] A. Apicella, F. Isgrò and R. Prevete, "Don't Push the Button! Exploring Data Leakage Risks in Machine Learning and Transfer Learning," *arXiv preprint arXiv:2401.13796*, 2024. doi: https://doi.org/10.2139/ssrn.4733889

[10] R. Mohammed, J. Rawashdeh and M. Abdullah, "MACHINE LEARNING WITH OVERSAMPLING AND UNDERSAMPLING TECHNIQUES: OVERVIEW STUDY AND EXPERIMENTAL RESULTS," in *2020 11th international conference on information and communication systems (ICICS)*, Irbid, 2020. doi: https://doi.org/10.1109/ICICS49469.2020.239556

[11] M. Grandini, E. Bagli and G. Visani, "METRICS FOR MULTI-CLASS CLASSIFICATION: AN OVERVIEW," *arXiv preprint arXiv:2008.05756*, 2020.

[12] A. Kowalczyk, "SUPPORT VECTOR MACHINES SUCCINTCTLY, SYNCFUSION," *Succinctly E-Book Series*, vol. 114, 2017.

[13] S. Anam, A. N. Guci, F. Widhiatmoko, M. Kurniawaty and K. A. A. Wijaya, "DEVELOPMENT OF HEALTH INSURANCE CLAIM PREDICTION METHOD BASED ON SUPPORT VECTOR MACHINE AND BAT ALGORITHM," *BAREKENG: Jurnal Ilmu Matematika dan Terapan*, vol. 17, no. 4, pp. 2281-2292, 2023. doi: https://doi.org/10.30598/barekengvol17iss4pp2281-2292

[14] Y. Ren, F. Hu and H. Miao, "THE OPTIMIZATION OF KERNEL FUNCTION AND ITS PARAMETERS FOR SVM IN WELL-LOGGING," in *2016 13th International Conference on Service Systems and Service Management (ICSSSM)*, Kunming, 2016.

[15] I. S. Al-Mejibli, J. K. Alwan and D. H. Abd, "THE EFFECT OF GAMMA VALUE ON SUPPORT VECTOR MACHINE PERFORMANCE WITH DIFFERENT KERNELS," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 5, pp. 5497-5506, 2020. doi: https://doi.org/10.11591/ijece.v10i5.pp5497-5506

[16] J. Wang, P. Li, R. Ran, Y. Che and Y. Zhou, "A short-term photovoltaic power prediction model based on the gradient boost decision tree," *A short-term photovoltaic power prediction model based on the gradient boost decision tree*, vol. 8, no. 5, 2018.

[17] B. Schölkopf, Z. Luo and V. Vovk, EMPIRICAL INFERENCE: FESTSCHRIFT IN HONOR OF VLADIMIR N. VAPNIK, Berlin: Springer Science & Business Media, 2013.

[18] J. H. Friedman, "GREEDY FUNCTION APPROXIMATION: A GRADIENT BOOSTING MACHINE," *Annals of statistics*, vol. 29, no. 5, pp. 1189-1232, 2001. doi: https://doi.org/10.1214/aos/1013203451

[19] A. Kadiyala and A. Kumar, "APPLICATIONS OF PYTHON TO EVALUATE THE PERFORMANCE OF DECISION TREE-BASED BOOSTING ALGORITHMS," *Environmental Progress & Sustainable Energy*, vol. 37, no. 2, pp. 618-623, 2018. doi: https://doi.org/10.1002/ep.12888

[20] E. Kristiani, Y.-T. Tsan, P.-Y. Liu, N. Y. Yen and C.-T. Yang, "BINARY AND MULTI-CLASS ASSESSMENT OF FACE MASK CLASSIFICATION ON EDGE AI USING CNN AND TRANSFER LEARNING," *Human-centric Computing and Information Sciences*, vol. 12, no. 53, 2022.

[21] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye and T.-Y. Liu, "LIGHTGBM: A HIGHLY EFFICIENT GRADIENT BOOSTING DECISION TREE," *Advances in neural information processing systems*, vol. 30, 2017.

[22] M. A. Mohammed, S. M. Kadhem and A. A. Maisa'a, "INSIDER ATTACKER DETECTION USING LIGHT GRADIENT BOOSTING MACHINE," *Tech-Knowledge*, vol. 1, no. 1, pp. 67-76, 2021.

[23] C. Zhang and Y. Ma, ENSEMBLE MACHINE LEARNINg, New York: Springer, 2012.