Vol. 6 No. 2, 2025, pp. 178-187 **Jurnal Pepadun**

A Comparative Study of XGBoost, LightGBM, and CatBoost Models for Customer Churn Prediction in the Banking Industry

***1Rivaldo, 2Rahman Taufik, 3Igit Sabda ilman, and 4Ossy Dwi Endah Wulansari**

[1,2,3,4]Department of Computer Science, University of Lampung,
Jl. Prof. Dr. Ir. Sumantri Brojonegoro No. 1, Bandar Lampung, Lampung Province, Indonesia, 35145
e-mail: [*1]**rivaldo.21@students.unila.ac.id**, [2]rahman.taufik@fmipa.unila.ac.id,
[3]igit.sabda@fmipa.unila.ac.id, [4]ossy.dwiendah@fmipa.unila.ac.id

Abstract *- Customer churn is a critical issue in the banking industry, as retaining existing customers is more cost-effective than acquiring new ones. High churn rates can negatively affect profitability and long-term business sustainability, making churn prediction a key focus in customer relationship management. With the rise of digital banking and the availability of large-scale customer data, machine learning techniques have become valuable tools for identifying at-risk customers. In particular, gradient boosting algorithms have shown promising results in classification tasks involving structured data. This study compares the performance of three ensemble machine learning models XGBoost, LightGBM, and CatBoost in classifying churn using a publicly available banking customer dataset consisting of 10,127 records and 23 features. The evaluation is conducted using two data-splitting schemes (80:10:10 and 70:15:15), and four performance metrics: accuracy, precision, recall, and F1-score. The results indicate that XGBoost achieved the highest overall performance (98.3% accuracy in split 1 and 96.4% in split 2). LightGBM demonstrated competitive accuracy with significantly faster training time, while CatBoost offered strong predictive capability but required longer computation. These findings suggest that model selection in churn prediction depends on the trade-off between predictive performance and computational efficiency.*

Keywords: *Machine Learning; Churn Prediction; XGBoost; LightGBM; CatBoost.*

1. INTRODUCTION

In highly competitive business environment, retaining existing customers is far more cost-effective than acquiring new ones, as loyal customers not only provide recurring revenue but also act as brand advocates through word-of-mouth [1][2]. In the banking sector, customer churn when clients close accounts or cease using services can severely impact both revenue and institutional reputation [3]. High churn rates often point to underlying issues such as suboptimal service quality, inefficient processes, or a lack of competitive product offerings [4][5]. Consequently, banks must understand and manage churn to maintain service excellence and customer satisfaction [6].

In today's digital and data-driven era, the vast availability of customer data has made churn analysis increasingly important for modern businesses [7]. Customer churn is now recognized as a key factor affecting company profitability, especially amid rapid technological advancement and evolving business models [3]. The primary goal of churn analysis is to identify customers with a high risk of leaving the organization [8]. In the banking sector, while customer attrition is inevitable, machine learning has become essential for predicting churn using demographic and financial data to support retention strategies [6]. Nonetheless, challenges such as class imbalance and overfitting continue to hinder the effectiveness of machine learning models in churn prediction [9].

XGBoost (Extreme Gradient Boosting) is a machine learning algorithm developed by Dr. Tianqi Chen from the University of Washington in 2014 as an advancement of traditional gradient boosting methods [10]. It is known for its high predictive accuracy, interpretability, and flexibility in classification tasks [11]. XGBoost employs the Gradient Boosting Decision Tree (GBDT) technique, which combines multiple weak learners typically decision trees into a strong ensemble model [12]. During training, it updates model weights

 | doi: 10.23960/pepadun.v6i3.277

incrementally based on the gradient of the loss function to improve performance. A key feature is its level-wise tree growth strategy, which, while computationally intensive, enhances model stability and accuracy [13].

LightGBM is a gradient boosting framework developed by Microsoft in C++ that is widely used for building efficient and accurate machine learning models [14]. It uses a histogram-based algorithm that discretizes continuous values into bins, improving training speed and reducing memory usage. LightGBM supports two tree growth strategies: level-wise and leaf-wise. While level-wise growth enables multithreading, it may introduce redundant splits; the leaf-wise strategy focuses on the highest gain but requires depth limitation to prevent overfitting [15]. With this architecture, LightGBM offers superior computational efficiency, strong predictive performance, and the ability to scale to large datasets, outperforming many other boosting algorithms [16].

CatBoost is a gradient boosting model introduced by Yandex engineers in 2017 that excels in handling categorical features without the need for explicit preprocessing or normalization [17]. It captures feature interactions and uses gradient boosting to reduce noise, bias, and prediction errors [18]. A key characteristic is its use of oblivious decision trees, which enforce symmetric structure at each depth level as a form of regularization. This setup improves consistency and allows uncertainty estimation in predictions, similar to Gaussian Processes [19]. CatBoost also applies ordered boosting to avoid target leakage, computing transformation values only from previously seen data while iteratively adjusting predictions based on data distribution [20].

2. RESEARCH METHODOLOGY

This research aims to compare the performance of three machine learning models XGBoost, LightGBM, and CatBoost in predicting customer churn using a publicly available banking dataset. These models were selected due to their proven effectiveness in handling structured data and their widespread use in classification tasks. The objective is to evaluate which model provides the most accurate and efficient results in churn prediction. The methodology consists of several stages, including data acquisition, exploratory data analysis, preprocessing, data splitting, model development, and performance evaluation.

2.1. Data Acquisition

The dataset used in this study is the Credit Card Customers dataset obtained from Kaggle, originally provided by Sakshi Goyal. It contains 10,127 records and 23 features that represent various aspects of customer information. These features include demographic attributes such as age, income, and gender, as well as behavioral attributes like transaction counts and card type. The target label, Attrition_Flag, indicates whether a customer has churned or remained active.

2.1. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was carried out to gain an initial and comprehensive understanding of the dataset's structure, data quality, and feature characteristics before proceeding to the modeling stage. This step is essential to uncover underlying patterns, detect anomalies, and ensure that the data is suitable for analysis. The process involved several key activities, such as checking for missing or inconsistent values, identifying potential outliers that could skew the results, and exploring the distribution of both categorical and numerical variables to understand their behavior and scale. Additionally, correlation analysis was conducted to examine the relationships between numerical features, which helps in detecting multicollinearity and guiding feature selection. These insights provided a solid foundation for making informed preprocessing decisions and selecting appropriate modeling techniques.

© 2025 The Authors | doi: 10.23960/pepadun.v6i3.277

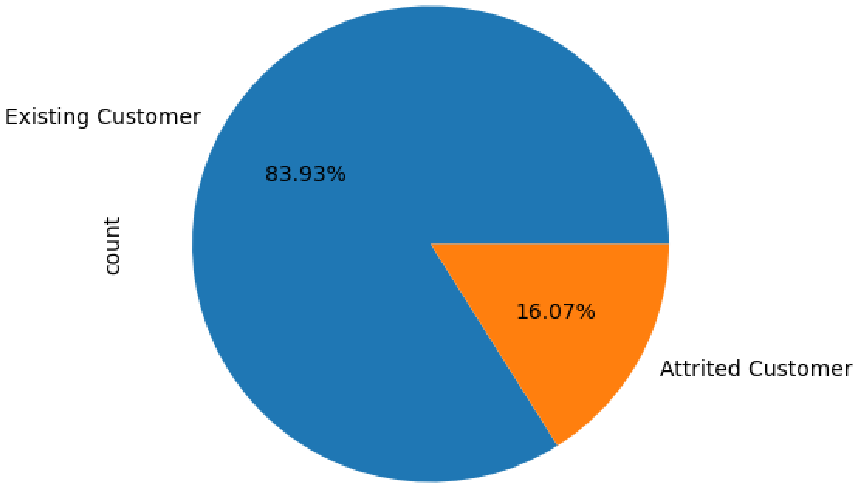

Figure 1. Class distribution of the target variable (Attrition_Flag).

Figure 1 presents the distribution of the target variable, Attrition_Flag, highlighting a significant class imbalance within the dataset. A large majority of the instances (83.93%) are categorized as Existing Customer, while only 16.07% are labeled as Attrited Customer. This imbalance may lead to biased model performance, where the classifier tends to favor the majority class unless appropriate balancing techniques are applied during preprocessing.

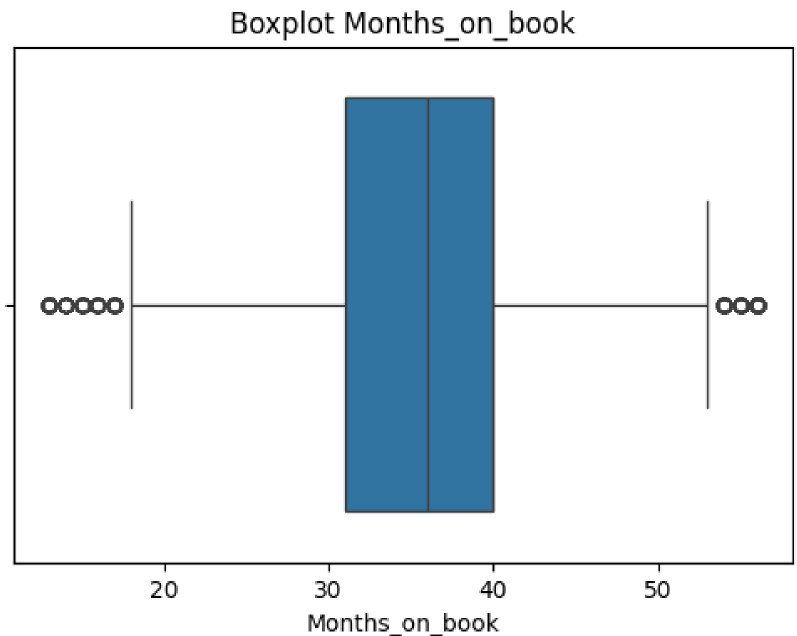

Figure 2. Boxplot of outliers in Months_on_book.

Figure 2 shows a boxplot of the Months_on_book variable, which represents how long a customer has been with the bank. In the plot, the blue box in the center represents the interquartile range (IQR), which contains the middle 50% of the data. The line inside the box indicates the median value of the variable. The whiskers extending from the box show the range of values within 1.5 times the IQR, while the small individual dots on either side represent outliers, or data points that fall significantly outside the typical range. These outliers are important to identify, as they may influence the performance of machine learning models if not handled appropriately during preprocessing.

© 2025 The Authors | doi: 10.23960/pepadun.v6i3.277

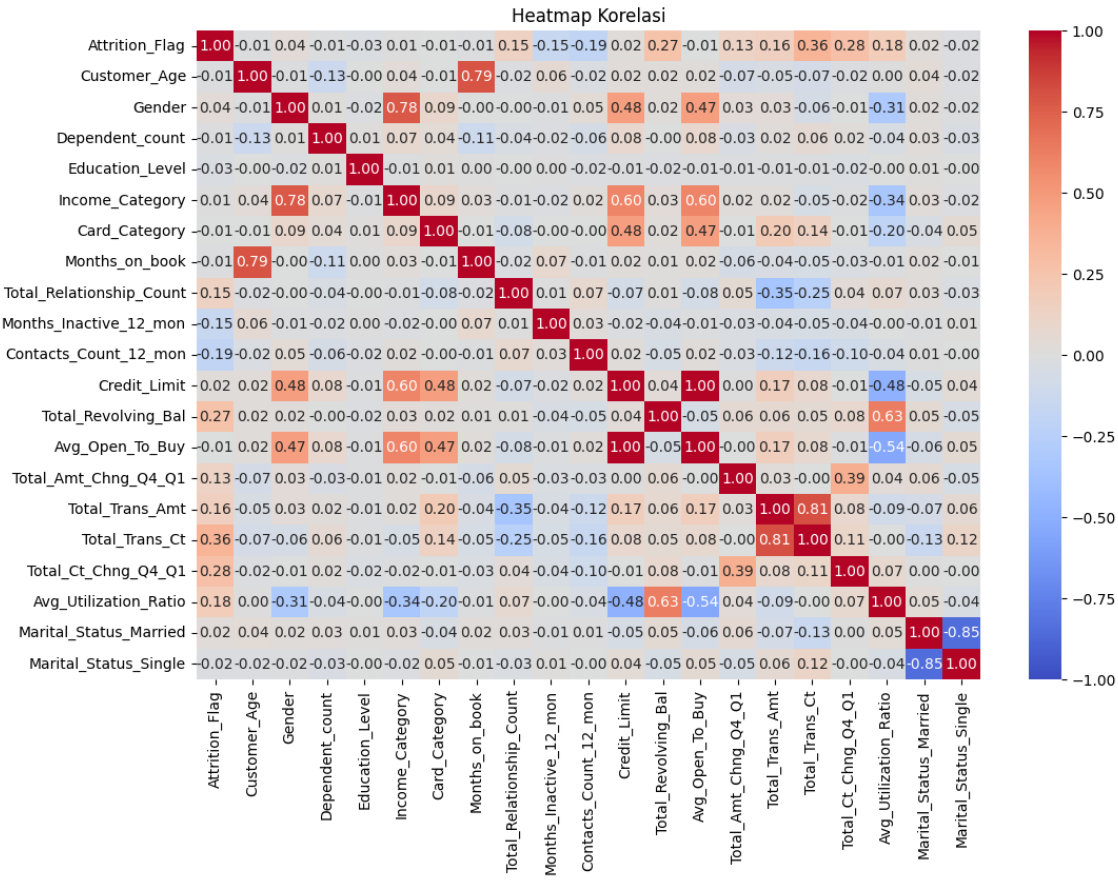

Figure 3. Correlation heatmap.

A correlation heatmap, as presented in Figure 3, illustrates the linear relationships among numerical features in the dataset. The color intensity represents the strength and direction of correlation between each pair of variables, where values closer to +1 or –1 indicate strong positive or negative relationships, respectively. This visualization is useful for identifying features that are strongly correlated with each other, which may lead to multicollinearity, as well as detecting variables that have a notable relationship with the target variable.

2.3. Data Preprocessing

The data preprocessing stage involved several essential steps to prepare the dataset for machine learning modeling. First, categorical columns were converted into numerical format because machine learning algorithms cannot process non-numeric data. Three columns were removed CLIENTNUM, which serves as a unique customer identifier and holds no predictive value, and two system-generated columns, Naive_Bayes_Classifier..._12_mon_1 and mon_2, which were deemed irrelevant for training. Rows containing unknown values in any categorical feature were also removed, resulting in a cleaned dataset with 20 columns and 7,081 rows, including 5,968 non-churned and 1,113 churned customers.

All numerical features were examined for the presence of outliers using the Interquartile Range (IQR) method, which identifies values that fall significantly outside the typical data spread. Instead of removing these outliers, normalization was applied using RobustScaler, a technique that scales data based on the IQR. This method is particularly suitable in the presence of outliers, as it is less sensitive to extreme values compared to standard normalization techniques. By doing so, the model is able to learn from the data without being skewed by unusually large or small values. To address class imbalance, a combined resampling approach was used, involving random undersampling to reduce the number of samples from the majority class and ADASYN (Adaptive Synthetic Sampling) to synthetically generate new instances for the minority class. This strategy allows the model to learn more effectively from both classes without being biased toward the dominant group.

2.4. Data Splitting

To evaluate the models under varying training proportions, the dataset was divided into three subsets: training, validation, and testing. The training set was used to build the models, The validation set was used to fine-tune hyperparameters and prevent overfitting, and the testing set to assess final performance on unseen data. This separation ensures that the evaluation process reflects the model's generalization capability. Table 1 presents the two data-splitting scenarios implemented in this study. These scenarios were designed to examine whether changes in data allocation impact model behavior and evaluation outcomes.

Table 1. Dataset splitting scenarios.

Scenario	Training Set (%)	Validation Set (%)	Testing Set (%)
1	80	10	10
2	70	15	15

2.5. Model Training

Three machine learning models XGBoost, LightGBM, and CatBoost were implemented to perform the churn classification task in this study. The development process for each model followed the same structure, including data preprocessing, hyperparameter tuning, and evaluation. To ensure a fair comparison, each model was trained using identical training data, features, and validation strategy. The evaluation was conducted under two different data-splitting scenarios to observe how varying data proportions affected model performance.

Hyperparameter tuning was carried out using GridSearchCV, which exhaustively explores predefined parameter combinations based on validation set scores. In this study, a total of 256 candidate parameter combinations were tested for each model. With 5-fold Stratified Cross-Validation, this resulted in 1,280 fitting processes per model, ensuring a robust and comprehensive search for optimal parameters. This setup was chosen to maintain balanced class distribution across folds and to enhance the reliability and generalizability of the evaluation process.

2.5. Model Evaluation

The performance of each model was evaluated using four commonly used classification metrics: Accuracy, Precision, Recall, and F1-Score. These metrics were calculated using the weighted average to account for the class imbalance in the target variable. To assess the consistency and robustness of each model, evaluations were conducted under two different data-splitting scenarios as previously described. The results for each model under both scenarios are summarized in the following tables.

Table 2. Evaluation metrics of XGBoost under Scenario 1 and 2.

Scenario	Precision	Recall	F1-Score	Accuracy
1	0.98	0.98	0.98	0.983
2	0.96	0.96	0.96	0.964

Table 3. Evaluation metrics of LightGBM under Scenario 1 and 2.

Scenario	Precision	Recall	F1-Score	Accuracy
1	0.98	0.98	0.98	0.983
2	0.96	0.96	0.96	0.964

Table 4. Evaluation metrics of CatBoost under Scenario 1 and 2.

Scenario	Precision	Recall	F1-Score	Accuracy
1	0.98	0.98	0.98	0.983
2	0.96	0.96	0.96	0.964

 | doi: 10.23960/pepadun.v6i3.277

As shown in Table 2, Table 3, and Table 4, the evaluation metrics for XGBoost, LightGBM, and CatBoost respectively were computed under two different data-splitting scenarios. These metrics include precision, recall, and F1-score, which provide a more balanced view of model performance, especially in imbalanced classification tasks such as churn prediction. The inclusion of multiple metrics ensures that the models are not only accurate in general but also capable of correctly identifying both classes, which is critical in this context. These results serve as the basis for comparative analysis in the following chapter.

3. RESULTS AND DISCUSSION

This section discusses the experimental results based on the performance of the machine learning models used in this study. The analysis focuses on evaluating and comparing the predictive capabilities of XGBoost, LightGBM, and CatBoost under different data-splitting scenarios. In addition, differences in training time due to hyperparameter tuning are also highlighted to provide insight into the computational efficiency of each model.

3.1. Comparative Evaluation under 80:10:10 Splitting

The performance of XGBoost, LightGBM, and CatBoost was evaluated using the first data-splitting scheme, where 80% of the dataset was used for training, 10% for validation, and 10% for testing. The comparison focuses on four evaluation metrics: accuracy, precision, recall, and F1-score. The results for each model under this scenario are presented in the following table.

Table 5. Evaluation metrics of each model (80:10:10 split).

Model	Precision	Recall	F1-Score	Accuracy
XGBoost	0.98	0.98	0.98	0.983
LightGBM	0.98	0.98	0.98	0.981
CatBoost	0.97	0.97	0.97	0.971

Based on the evaluation results shown in Table 5, all three models demonstrate strong predictive performance across all four metrics. XGBoost achieves the highest accuracy at 0.983, followed closely by LightGBM with 0.981, while CatBoost records a slightly lower accuracy of 0.971. Although XGBoost and LightGBM show identical values in precision, recall, and F1-score (0.98 each), XGBoost yields a slightly higher overall accuracy. This indicates that XGBoost was able to correctly classify a marginally greater number of instances compared to LightGBM, even though their per-class performance remains similar. CatBoost, while still performing well, trails slightly behind both models across all metrics under this splitting scheme.

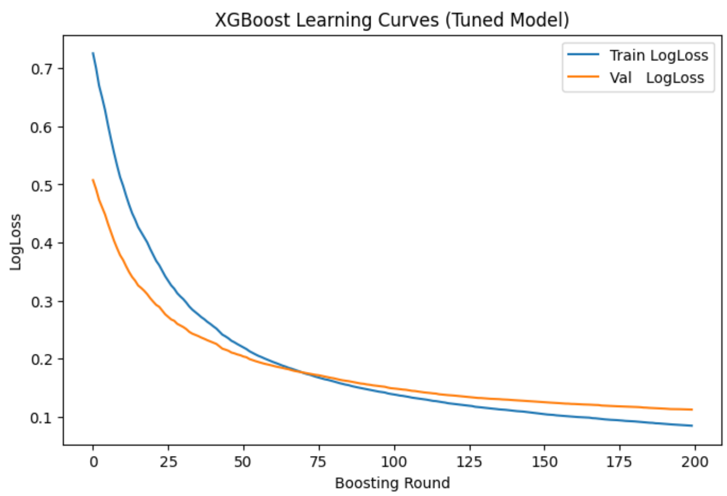

Figure 4. XGBoost learning curve scenario 1 (80:10:10 split).

 | doi: 10.23960/pepadun.v6i3.277

Figure 4 below illustrates the learning curve of the XGBoost model under Scenario 1 (80:10:10 data split), where it achieved the highest accuracy among all models, with a score of 0.983. The graph demonstrates a steady decrease in both training and validation log loss across boosting rounds, indicating effective learning throughout the training process. Furthermore, the small gap between the two curves suggests good generalization performance on unseen data, with no indication of overfitting.

3.2. Comparative Evaluation under 70:15:15 Splitting

To further evaluate the stability of each model's performance, a second data-splitting scheme was applied, using 70% of the dataset for training, 15% for validation, and 15% for testing. The evaluation metrics used remain the same as in the previous scenario. The results obtained from this configuration are presented in the following Table 6.

Table 6. Evaluation metrics of each model (70:15:15 split).

Model	Precision	Recall	F1-Score	Accuracy
XGBoost	0.96	0.96	0.96	0.964
LightGBM	0.96	0.95	0.96	0.953
CatBoost	0.96	0.96	0.96	0.964

As shown in Table 6, all three models experience a slight decline in performance compared to the previous scenario, which is expected due to the reduction in training data. XGBoost and CatBoost both achieve the highest accuracy at 0.964, while LightGBM records a slightly lower accuracy of 0.953. Although all models maintain consistent scores across precision, recall, and F1-score, the drop in LightGBM's recall to 0.95 may have contributed to its slightly lower overall accuracy. In contrast, CatBoost shows a relatively stable performance across all metrics and matches XGBoost in this scenario, suggesting improved robustness when trained on a smaller dataset.

In addition to the evaluation metrics presented in Table 6, the learning curve of the XGBoost model under Scenario 2 is shown in Figure 5 below. In this configuration, XGBoost achieved the highest accuracy (0.964), equal to CatBoost, making it one of the best-performing models under reduced training data. The graph demonstrates a smooth and consistent reduction in log loss on both training and validation sets, indicating that the model continues to learn effectively even with less data. The close alignment between the two curves also suggests that the model maintains good generalization performance and is not overfitting.

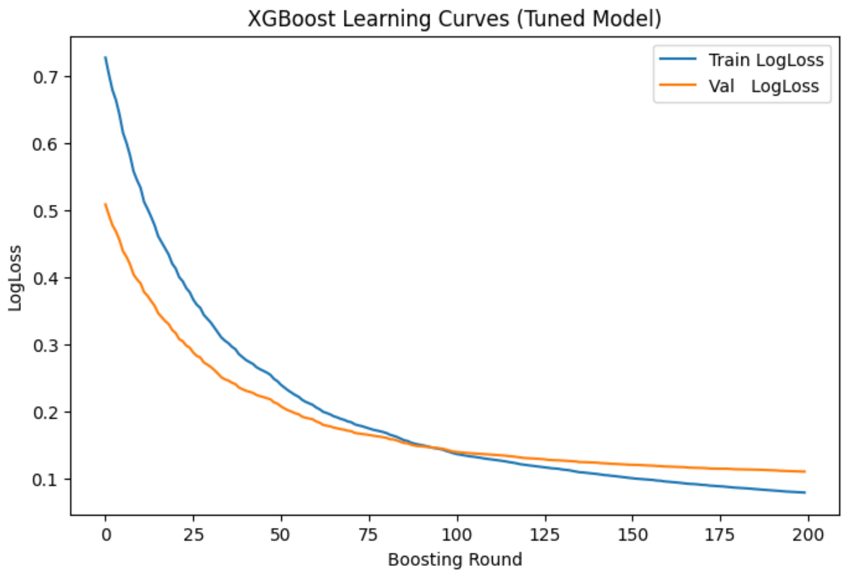

Figure 5. XGBoost learning curve scenario 2 (70:15:15 split).

© 2025 The Authors | doi: 10.23960/pepadun.v6i3.277

3.3. Training Time and Efficiency

Beyond accuracy and classification metrics, training time is an essential consideration when evaluating machine learning models, particularly in practical applications where computational resources and time constraints are critical. In this study, hyperparameter tuning was performed using GridSearchCV, which evaluated 256 combinations of candidate parameters for each model. With the use of 5-fold cross-validation, this process resulted in a total of 1,280 fitting iterations per model. The training time required to complete this tuning process varied between models and is summarized in the following Table 7.

Table 7. Tuning time of each model.

Model	Tuning Time (Seconds) Scheme 1	Tuning Time (Seconds) Scheme 2
XGBoost	291.33	377.20
LightGBM	173.86	182.81
CatBoost	664.08	617.52

As shown in Table 7, the tuning time for each model differs significantly. LightGBM consistently required the least amount of time to complete the hyperparameter search in both splitting schemes, reflecting its efficient training mechanism. XGBoost took moderately more time, while CatBoost required the longest tuning duration among the three models.

The increase in training time from Scheme 1 to Scheme 2 is observed in both XGBoost and LightGBM, which may be attributed to the larger validation and test sets in the second scheme. In contrast, CatBoost recorded slightly shorter tuning time in Scheme 2, though it remains the most time-consuming overall. These results highlight the trade-off between predictive performance and computational cost when selecting a model for deployment.

3.4. Overview of Model Evaluation

The evaluation of XGBoost, LightGBM, and CatBoost across both data-splitting schemes (80:10:10 and 70:15:15) showed that all three models performed consistently in predicting customer churn. Each model achieved high values across all evaluation metrics, demonstrating their capability to manage class imbalance and deliver reliable classification outcomes. However, closer analysis reveals meaningful differences in performance patterns and computational efficiency.

XGBoost produced the highest accuracy scores in both scenarios (0.983 and 0.964), supported by stable learning curves and low log loss throughout training and validation. This indicates strong learning behavior and minimal risk of overfitting. LightGBM, while achieving slightly lower accuracy, showed the fastest training times in both schemes. This efficiency may be advantageous in time-sensitive or resource-constrained environments. Meanwhile, CatBoost maintained stable precision, recall, and F1-scores across both scenarios, with accuracy identical to XGBoost under Scenario 2. Although it required the longest tuning time, CatBoost's performance suggests resilience when trained on smaller data proportions.

Overall, each model demonstrates distinct strengths XGBoost excels in accuracy and generalization, LightGBM in speed and efficiency, and CatBoost in stability across varying data conditions. These characteristics reflect the unique design philosophies behind each algorithm. These distinctions are essential for practitioners who must choose models based not only on accuracy, but also on training efficiency and robustness in real-world applications.

4. CONCLUSIONS

This study evaluated and compared the performance of three gradient boosting models XGBoost, LightGBM, and CatBoost for predicting customer churn in the banking sector. The models were tested under two different data-splitting schemes to assess their consistency and robustness. The results showed that all three models achieved high predictive accuracy, with XGBoost consistently performing well across both scenarios.

LightGBM demonstrated competitive accuracy while offering the fastest training time, making it suitable when computational efficiency is prioritized. CatBoost provided stable performance but required significantly more training time, especially during hyperparameter tuning. These findings highlight the importance of balancing predictive performance and computational cost when selecting machine learning models for churn prediction. Future research could explore additional feature engineering techniques, alternative resampling methods, or real-time implementation strategies to further enhance model effectiveness in practical.

LITERATURE

[1] B. Artha, I. Zahara, B. Bahri, and N. Sari, “Customer Retention: A Literature Review,” *Soc. Sci. Stud.*, vol. 2, pp. 30–45, Jan. 2022, doi: 10.47153/sss21.2952022.

[2] B. Prabadevi, R. Shalini, and B. R. Kavitha, “Customer churning analysis using machine learning algorithms,” *Int. J. Intell. Networks*, vol. 4, pp. 145–154, 2023, doi: https://doi.org/10.1016/j.ijin.2023.05.005.

[3] P. P. Singh, F. I. Anik, R. Senapati, A. Sinha, N. Sakib, and E. Hossain, “Investigating customer churn in banking: a machine learning approach and visualization app for data science and management,” *Data Sci. Manag.*, vol. 7, no. 1, pp. 7–16, 2024, doi: https://doi.org/10.1016/j.dsm.2023.09.002.

[4] H. A. Triyafebrianda and N. A. Windasari, “Factors Influence Customer Churn on Internet Service Providers in Indonesia,” *TIJAB (The Int. J. Appl. Business)*, vol. 6, no. 2, pp. 134–144, 2022.

[5] J. Lappeman, M. Franco, V. Warner, and L. Sierra-Rubia, “What social media sentiment tells us about why customers churn,” *J. Consum. Mark.*, vol. 39, no. 5, pp. 385–403, 2022.

[6] S. Akakpo, P. Dambra, R. Paz, T. Smyth, F. Torre, and C. Yu, “Optimization of the K-Nearest Neighbor Algorithm to Predict Bank Churn,” *Stat. Optim. Inf. Comput.*, vol. 12, no. 5, pp. 1397–1408, 2024.

[7] X. Liu, G. Xia, X. Zhang, W. Ma, and C. Yu, “Customer churn prediction model based on hybrid neural networks,” *Sci. Rep.*, vol. 14, no. 1, p. 30707, 2024.

[8] S. K. Hegde, R. Hegde, S. S. Nanda, G. Phatak, P. Hongal, and D. Gowda, “Customer Churn Analysis in Financial Domain using Deep Intelligence Network,” in *2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, IEEE, 2023, pp. 362–370.

[9] S. A. Fayaz, S. Kaul, M. Zaman, and M. A. Butt, “An adaptive gradient boosting model for the prediction of rainfall using ID3 as a base estimator,” *Rev. d’Intelligence Artif.*, vol. 36, no. 2, p. 241, 2022.

[10] S. E. H. Yulianti, O. Soesanto, and Y. Sukmawaty, “Penerapan Metode Extreme Gradient Boosting (XGBOOST) pada Klasifikasi Nasabah Kartu Kredit,” *J. Math. Theory Appl.*, pp. 21–26, 2022.

[11] Z. A. Ali, Z. H. Abduljabbar, H. A. Taher, A. B. Sallow, and S. M. Almufti, “Exploring the power of eXtreme gradient boosting algorithm in machine learning: A review,” *Acad. J. Nawroz Univ.*, vol. 12, no. 2, pp. 320–334, 2023.

[12] M. D. Maulana, A. I. Hadiana, and F. R. Umbara, “Algoritma Xgboost Untuk Klasifikasi Kualitas Air Minum,” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 5, pp. 3251–3256, 2023.

[13] R. Sibindi, R. Mwangi, and A. Waititu, “A boosting ensemble learning based hybrid light gradient boosting machine and extreme gradient boosting model for predicting house prices,” *Eng. Reports*, vol. 5, Nov. 2022, doi: 10.1002/eng2.12599.

[14] M. Alyami, M. Khan, A. W. A. Hammad, H. Alabduljabbar, R. Nawaz, M. Fawad, and Y. Gamil, “Estimating compressive strength of concrete containing rice husk ash using interpretable machine learning-based models,” *Case Stud. Constr. Mater.*, vol. 20, p. e02901, 2024.

[15] H. Zeng, C. Yang, H. Zhang, Z. Wu, J. Zhang, G. Dai, F. Babiloni, and W. Kong, “A LightGBM-Based EEG Analysis Method for Driver Mental States Classification,” *Comput. Intell. Neurosci.*, vol. 2019,

no. 1, p. 3761203, Jan. 2019, doi: https://doi.org/10.1155/2019/3761203.

[16] T. O. Omotehinwa, D. O. Oyewola, and E. G. Moung, "Optimizing the light gradient-boosting machine algorithm for an efficient early detection of coronary heart disease," *Informatics Heal.*, vol. 1, no. 2, pp. 70–81, 2024, doi: https://doi.org/10.1016/j.infoh.2024.06.001.

[17] J. Zheng, M. Hu, C. Wang, S. Wang, B. Han, and H. Wang, "Spatial patterns of residents' daily activity space and its influencing factors based on the CatBoost model: A case study of Nanjing, China," *Front. Archit. Res.*, vol. 11, no. 6, pp. 1193–1204, 2022, doi: https://doi.org/10.1016/j.foar.2022.04.003.

[18] X. Ren, H. Yu, X. Chen, Y. Tang, G. Wang, and X. Du, "Application of the CatBoost Model for Stirred Reactor State Monitoring Based on Vibration Signals," *C. - Comput. Model. Eng. Sci.*, vol. 140, no. 1, pp. 647–663, 2024, doi: https://doi.org/10.32604/cmes.2024.048782.

[19] D. Boldini, F. Grisoni, D. Kuhn, L. Friedrich, and S. A. Sieber, "Practical guidelines for the use of gradient boosting for molecular property prediction," *J. Cheminform.*, vol. 15, no. 1, p. 73, 2023, doi: 10.1186/s13321-023-00743-7.

[20] Y. Chen, B. Chen, and A. Shayilan, "Combining categorical boosting and Shapley additive explanations for building an interpretable ensemble classifier for identifying mineralization-related geochemical anomalies," *Ore Geol. Rev.*, vol. 173, p. 106263, 2024, doi: https://doi.org/10.1016/j.oregeorev.2024.106263.