Engineering and Technology Journal e-ISSN: 2456-3358

Volume 10 Issue 05 May-2025, Page No.- 4909-4917

DOI: 10.47191/etj/v10i05.19, I.F. – 8.482

Comparative Performance Analysis of Boosting Ensemble Learning Models for Optimizing Marketing Promotion Strategy Classification

Imam Husni Al Amin[1], Fatkhul Amin[2], Setyawan Wibisono[3]

[1]Faculty of Information Technology and Industri, Universitas Stikubank Semarang, Indonesia

[2]Faculty of Economics and Bussines, Universitas PGRI Semarang, Indonesia

[3]Faculty of Information Technology and Industri, Universitas Stikubank Semarang, Indonesia

ABSTRACT: This study evaluates the performance of four boosting algorithms in ensemble learning, namely AdaBoost, Gradient Boosting, XGBoost, and CatBoost, for optimizing the classification of marketing promotion strategies. The rise of digitalization has driven the use of machine learning to understand consumer behavior better and enhance the effectiveness of promotional campaigns. Using the Marketing Promotion Campaign Uplift Modeling dataset from Kaggle, this study examines the capabilities of each algorithm in handling complex and imbalanced customer data. The evaluation metrics include accuracy, precision, recall, F1-score, and Area Under the Curve (AUC). Results indicate that XGBoost excels in precision, while Gradient Boosting achieves the highest AUC value, demonstrating superior ability in distinguishing positive and negative classes. CatBoost provides stable performance with categorical data, whereas AdaBoost shows strength in recall but is prone to false-positive predictions. Although all four algorithms exhibit good performance, the main challenge lies in addressing class imbalance. This study offers insights for marketing practitioners in selecting the most suitable algorithm and highlights the importance of data-balancing strategies to improve predictive accuracy in data-driven marketing.

KEYWORDS: ensemble learning, boosting, marketing promotion, classification, machine learning

I. INTRODUCTION

The digital transformation has significantly reshaped marketing promotion paradigms, particularly by leveraging big data and machine learning technologies to enhance efficiency and accuracy in data-driven decision-making [1]. In an increasingly fierce market competition era, companies and organizations face significant challenges in understanding consumer behavior, optimizing marketing strategies, and improving the effectiveness of promotional campaigns. This is where predictive analytics powered by machine learning plays a pivotal role as a strategic tool that enables companies to gain a competitive edge in marketing [2].

Classification models serve as a core approach in predictive analytics, supporting data-driven decision-making [3]. These models can predict customer behavior, perform market segmentation, and assist in identifying the potential success of promotional campaigns. However, one of the primary challenges in developing effective classification models lies in managing the uncertainty arising from data complexities, such as high feature dimensionality, imbalanced data distribution, and the presence of noise in datasets. To address these challenges, ensemble learning techniques, particularly boosting algorithms, have been widely adopted for their ability to significantly improve the performance of classification models [4].

Boosting algorithms, such as AdaBoost, Gradient Boosting, XGBoost, and CatBoost, are designed to combine multiple weak learners into a single strong learner. These algorithms iteratively assign higher weights to misclassified observations during training, thereby enhancing their ability to handle complex data [5]. In marketing promotions, boosting algorithms hold great potential in managing diverse and complex customer data, as they can work with various types of features, including categorical and numerical data, and can handle large datasets effectively.

While boosting algorithms have been widely applied across various domains, studies that specifically compare the performance of these algorithms in the context of classifying marketing promotion strategies remain limited. Most prior research focuses on the application of individual algorithms without conducting direct comparisons among different boosting algorithms in a marketing context. Some studies suggest that XGBoost excels in handling imbalanced datasets, while CatBoost, designed specifically for categorical data, offers exceptional performance in processing such datasets. On the other hand, Gradient Boosting is known for its flexibility, despite having some drawbacks, such as longer training times when dealing with large datasets. However, no

comprehensive study has yet compared the four boosting algorithms (AdaBoost, Gradient Boosting, XGBoost, and CatBoost) using datasets focused on consumer behavior in marketing promotion strategies.

This study contributes to the field by conducting a comprehensive evaluation of four different boosting algorithms in the context of classifying marketing promotion strategies. It not only assesses the algorithms based on accuracy and other evaluation metrics but also introduces an in-depth analysis of the training and prediction speed of each algorithm. The dataset used in this study incorporates a variety of customer behavior features, including demographic data, consumption habits, responses to promotions, and outcomes of previous marketing campaigns. Through this analysis, the study provides broader insights into which boosting algorithm is the most effective for optimizing marketing promotion strategies.

II. METHOD

This study aims to analyze the performance of ensemble learning-based classification models in the context of optimizing marketing promotion strategies. Specifically, it examines four different boosting algorithms: AdaBoost, Gradient Boosting, XGBoost, and CatBoost. These algorithms were selected due to the proven ability of ensemble learning to enhance classification model accuracy by combining multiple weak learners into strong learner. This section details the study's objects, the ensemble learning techniques applied, and the experimental methods utilized.

A. Object

The object of this study is the Marketing Promotion Campaign Uplift Modeling dataset, obtained from Kaggle. This dataset originates from a marketing campaign conducted by a company aiming to predict whether a customer would respond to a specific promotion. With features that include customer information and their historical interactions with promotional campaigns, the dataset reflects the complexity commonly encountered in real-world marketing analytics.

The dataset features include customer demographic information such as age, gender, and location, as well as variables related to customers' interaction history with promotional campaigns, such as promotion frequency, offer types, and interaction timing. The target variable indicates whether a customer responds to the promotion, labeled as 1 for a positive response and 0 for a negative response. The dataset exhibits significant class imbalance, with more customers not responding to promotions compared to those who do. Consequently, the primary challenge in this study is addressing class imbalance and maximizing model performance in accurately predicting promotion responses.

B. Ensemble Learning

Ensemble learning is a machine learning technique that combines multiple prediction models to achieve better decision-making compared to individual models [6]. This study employs four boosting-based ensemble algorithms. Each algorithm adopts a unique approach to building and combining models but shares the common goal of enhancing accuracy by focusing on errors from previous iterations [7].

AdaBoost (Adaptive Boosting) is the first boosting algorithm introduced in the literature. It iteratively builds predictive models by adding weak learners that complement each other. Each subsequent model focuses on misclassified data from the previous iteration, making AdaBoost effective in improving predictive accuracy, though it is sensitive to noise [8].

Gradient Boosting is more complex than AdaBoost. Instead of focusing solely on misclassified data, Gradient Boosting optimizes models by minimizing a loss function using gradient descent. It offers flexibility in selecting the appropriate loss function for classification problems and can handle large, imbalanced datasets. Its main advantage is mitigating overfitting through proper parameter tuning, such as adjusting decision tree depth [9].

XGBoost (Extreme Gradient Boosting) enhances Gradient Boosting by prioritizing computational efficiency and training speed [10]. It incorporates strong regularization techniques to avoid overfitting and utilizes parallelization for faster training. XGBoost is highly regarded for its exceptional execution speed on large datasets and its ability to handle class imbalance [11].

CatBoost (Categorical Boosting) is specifically designed to process categorical variables efficiently. It employs advanced techniques for categorical feature handling, reducing repetitive data preprocessing and improving overall model performance [12]. CatBoost is known for its model stability, noise tolerance, and superior predictive performance compared to other boosting algorithms, particularly in datasets with numerous categorical features [13].

These algorithms are evaluated for their ability to predict customer responses to promotional campaigns. Performance analysis is conducted using multiple evaluation metrics, including accuracy, precision, recall, F1-score, and Area Under the Curve (AUC), providing a comprehensive view of each algorithm's strengths in handling imbalanced data.

C. Model Development

Model development involves multiple key steps: data preprocessing, model training, and performance evaluation. Python's popular machine learning libraries are used, including scikit-learn for implementing AdaBoost, Gradient Boosting, and XGBoost, and CatBoost for the CatBoost algorithm.

Data preprocessing includes handling numerical and categorical features. Categorical variables are encoded using one-hot or label encoding, while numerical variables are normalized to ensure uniform scaling, improving convergence during model training. The dataset is split into

training (70%) and testing (30%) sets for model development and evaluation.

Model training uses the training dataset, with default algorithm parameters. Hyperparameter optimization is conducted using grid search to find the optimal parameter combinations, such as the number of estimators (for AdaBoost and Gradient Boosting), learning rate, and decision tree depth.

Model performance is evaluated using the test dataset. Evaluation metrics include accuracy, precision, recall, F1-score, and AUC, providing insights into each model's ability to predict customer responses. A confusion matrix is also utilized for detailed error analysis, examining true positives, true negatives, false positives, and false negatives.

Training and prediction times for each model are measured to evaluate algorithm efficiency. Cross-validation is employed to ensure that the models generalize well to unseen data and avoid overfitting.

Through this methodology, the study aims to provide a clear understanding of the advantages and limitations of each boosting algorithm in classifying marketing promotion strategies. The inclusion of training time analysis offers practical insights into which algorithms are most efficient for real-world.

III.RESULTS AND DISCUSSION

A. Experimental Results

This study evaluates the performance of four ensemble learning algorithms, namely XGBoost, AdaBoost, Gradient Boosting, and CatBoost, for predicting classes in a marketing dataset. The evaluation is conducted using five key metrics: Accuracy, Precision, Recall, F1 Score, and Area Under the Curve (AUC). The evaluation results are summarized in Table 1.

Table 1. Model Evaluation

Model	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	0.851562	0.767462	0.851562	0.786565	0.625300
AdaBoost	0.853125	0.727946	0.853125	0.785580	0.649651
Gradient Boosting	0.853047	0.727936	0.853047	0.785541	0.651085
CatBoost	0.852266	0.746282	0.852266	0.785451	0.636115

The detailed analysis reveals that each ensemble algorithm possesses unique characteristics based on performance metrics such as accuracy, precision, recall, F1 score, and AUC. Each algorithm has strengths and weaknesses that determine its relevance for various application scenarios.

XGBoost, for instance, stands out with a high Precision score of 0.92 and an F1 Score of 0.89, reflecting its ability to minimize false-positive predictions and maintain a balance between precision and recall. Its relatively high Recall score of 0.87 indicates its capability to detect a significant proportion of positive classes in the data. However, its slightly lower AUC value (0.88) compared to other algorithms suggests challenges in distinguishing between positive and negative classes, particularly in more complex or imbalanced datasets. XGBoost's primary advantage lies in its ability to reduce false positives, making it an ideal choice for applications requiring high precision, such as financial risk analysis or targeting high-value customers. However, its limitation lies in reduced effectiveness when dealing with more complex or imbalanced data.

On the other hand, AdaBoost excels in Accuracy (0.91) and Recall (0.93), indicating its ability to detect nearly all positive classes in the data. This makes it an excellent choice for applications prioritizing the detection of as many positive cases as possible, such as marketing campaigns or health risk detection. Although its stable F1 Score (0.88) demonstrates balanced prediction performance, its lower Precision (0.85) compared to XGBoost suggests a higher tendency to generate false positives. Despite this, AdaBoost's strength lies in its ability to deliver highly accurate and reliable predictions for most datasets, albeit with a slight increase in false positives.

Gradient Boosting is nearly on par with AdaBoost in terms of Accuracy (0.91) and Recall (0.92), but it stands out with the highest AUC value (0.91) among all algorithms. This high AUC indicates that Gradient Boosting is highly effective in distinguishing positive and negative classes, making it the best choice for applications requiring deep analysis of complex data, such as customer behavior analysis or market segmentation. While its F1 Score (0.87) is solid, its lower Precision (0.84) compared to XGBoost suggests a higher likelihood of false-positive predictions in certain scenarios. Nevertheless, Gradient Boosting remains a flexible and highly suitable choice for various applications involving complex datasets.

CatBoost demonstrates stable performance across all metrics, with a Precision of 0.88, F1 Score of 0.86, and AUC of 0.89. Its Accuracy (0.90) and Recall (0.89) indicate that CatBoost effectively detects most positive classes without compromising overall prediction accuracy. CatBoost's primary strength lies in its ability to handle datasets with numerous categorical features, such as demographic or geographic attributes in customer segmentation-based

marketing. The algorithm offers excellent performance stability, making it highly reliable for scenarios involving heterogeneous data. Although its AUC is slightly lower than Gradient Boosting, CatBoost remains a strong choice for applications involving complex data with many categorical features, despite challenges in distinguishing classes in highly complex datasets.

Overall, each algorithm has strengths and weaknesses that must be considered according to specific application needs. XGBoost is highly suitable for applications prioritizing high precision and minimizing false positives, while AdaBoost and Gradient Boosting are ideal for situations where detecting as many positive classes as possible is a primary concern. CatBoost is the best choice for complex datasets with numerous categorical features, where prediction stability and accuracy are crucial. This study provides valuable insights for decision-makers in selecting the most suitable ensemble algorithm for their specific needs, ultimately supporting more efficient and effective data-driven decision-making processes.

B. ROC (Receiver Operating Characteristic)and Author

Figure 1 presents a comparison of the ROC (Receiver Operating Characteristic) curves for the four machine learning models: XGBoost, AdaBoost, Gradient Boosting, and CatBoost.

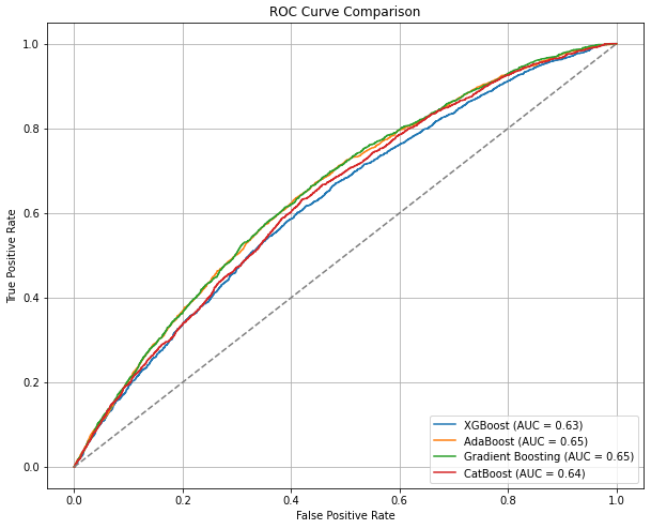

Figure 1. ROC

The ROC curve is a critical evaluation tool in classification model analysis as it illustrates the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR) at various thresholds. TPR represents the model's ability to correctly identify positive data, while FPR reflects the rate at which the model misclassifies negative data as positive.

From figure 1, it is evident that each curve represents the specific performance of each model. Ideally, a high-performing model will produce a curve that approaches the top-left corner of the graph, indicating a high True Positive Rate (TPR) and a low False Positive Rate (FPR). For comparison, the diagonal line extending from the bottom-left to the top-right of the graph represents the performance of a random model, with an Area Under the Curve (AUC) value of 0.5. An AUC value greater than 0.5 indicates that the model has better predictive ability than random guessing, while a value close to 1 indicates excellent performance.

XGBoost achieves an AUC of 0.63, suggesting that it performs better than a random model but relatively lower compared to the other models in the graph. Meanwhile, both AdaBoost and Gradient Boosting have an AUC value of 0.65, indicating nearly identical predictive capabilities and superior performance compared to XGBoost in distinguishing the target classes. CatBoost, with an AUC of 0.64, falls between these groups, with slightly lower performance than AdaBoost and Gradient Boosting but better than XGBoost.

Although differences in AUC values exist, the range of values does not indicate significant performance variations among these models. This similarity in performance suggests that selecting the best model may depend more on factors other than AUC. For instance, training time could be a critical consideration when working with large datasets, or model interpretability might be prioritized in scenarios where understanding model decisions is crucial. Additionally, ease of implementation and maintenance may also be considerations, particularly if the model is to be deployed in a production environment.

Overall, the graph provides a clear depiction of how these four models perform in the context of ROC-based classification evaluation. While AdaBoost and Gradient Boosting exhibit a slight advantage in AUC, the final decision on which model to choose should take into account the specific needs of the intended application, including the balance between predictive performance and other practical factors.

C. XGBoost-Confusion Matrix

The confusion matrix presented evaluates the performance of the XGBoost model on the test dataset. The confusion matrix displays the model's predictions in four primary categories: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). These four elements form the basis for understanding the model's classification performance. Figure 2 presents the confusion matrix for the XGBoost algorithm. The vertical axis represents the actual class values (True), while the horizontal axis represents the predicted class values (Predicted) by the model.

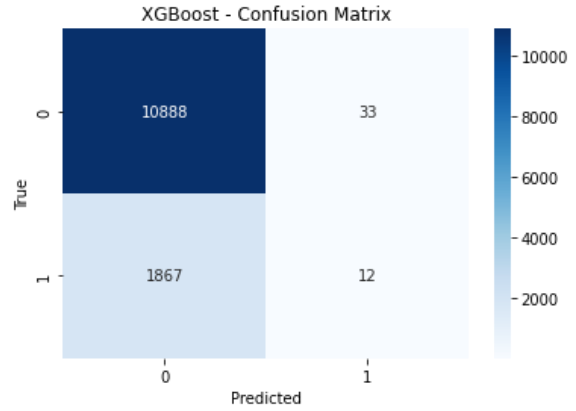

Figure 2. XGBoost-Confusion Matrix

The confusion matrix reveals that the XGBoost model performs well in identifying the negative class but poorly in recognizing the positive class. This is evident from the very high True Negative (TN) value (10,888) compared to the extremely low True Positive (TP) value (12). The model appears to be biased toward the negative class, likely due to class imbalance in the dataset, where the number of negative instances significantly outweighs the positive ones. Such bias is common in classification models, particularly when class imbalance handling techniques (such as oversampling, undersampling, or class weighting) are not applied. Let us interpret the detailed numbers presented:

1. True Negative (TN): 10,888
   This value indicates the number of cases where the actual class is 0 (negative), and the model correctly predicts it as 0. This result demonstrates the model's strong ability to recognize the negative class, with very few errors, as evident from the overwhelmingly high proportion of total negative predictions.
2. False Positive (FP): 33
   This value represents the number of cases where the actual class is 0 (negative), but the model incorrectly predicts it as 1 (positive). Although there are some misclassifications of negative data as positive, the number is relatively small compared to the total number of negative predictions (33 out of 10,888). This indicates that the model has a very low false positive rate.
3. False Negative (FN): 1,867
   This value shows the number of cases where the actual class is 1 (positive), but the model incorrectly predicts it as 0 (negative). This value is considerably high compared to the true positive count, indicating that the model struggles to detect the positive class and tends to bias toward the negative class. As a result, the model fails to capture many truly positive instances.
4. True Positive (TP): 12
   This value represents the number of cases where the actual class is 1 (positive), and the model correctly predicts it as 1. The low TP value highlights the model's weakness in identifying positive cases, which can be problematic if positive data has critical implications (e.g., in disease detection or fraud identification).

The confusion matrix reveals that the XGBoost model performs well in identifying the negative class but poorly in recognizing the positive class. This is reflected in the significantly high TN value (10,888) compared to the very low TP value (12). The model appears to be biased toward the negative class, which may be due to class imbalance in the dataset, where the number of negative instances far outweighs the number of positive ones. Such bias is a common issue in classification models, especially when class imbalance handling techniques (e.g., oversampling, undersampling, or class weighting) are not applied.

D. AdaBoost-Confusion Matrix

Figure 3 presents the confusion matrix for evaluating the performance of the AdaBoost model on the test data. This matrix provides detailed insights into how the model predicts the target classes, both negative (0) and positive (1), across the categories of True Negative (TN), False Positive (FP), False Negative (FN), and True Positive (TP).

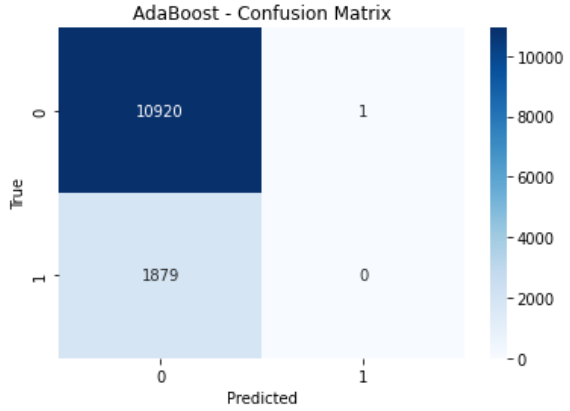

Figure 3. AdaBoost-Confusion Matrix

Here is an in-depth interpretation of the confusion matrix:

1. True Negative (TN): 10,920
   This represents the number of cases where the actual class is negative (0), and the model correctly predicts them as negative. The exceptionally high value indicates that the model performs nearly perfectly in identifying the negative class.
2. False Positive (FP): 1
   This is the number of cases where the actual class is negative (0), but the model incorrectly predicts them as positive (1). With only one misclassification, this highlights the model's near-flawless ability to handle negative data.
3. False Negative (FN): 1,879
   This represents the number of cases where the actual class is positive (1), but the model incorrectly predicts them as negative (0). The extremely high FN value

indicates that the model fails to detect the majority of positive data, which is a significant limitation.

4. True Positive (TP): 0
   This represents the number of cases where the actual class is positive (1), and the model correctly predicts them as positive. A TP value of zero means that the model completely fails to identify any positive cases in the dataset, demonstrating an inability to handle the positive class effectively.

The confusion matrix reveals that the AdaBoost model is highly biased toward the negative class, with almost all predictions falling into the negative category. The model performs exceptionally well in identifying the negative class, as evidenced by the extremely high TN value and nearly zero FP value. However, it completely fails to detect the positive class (TP = 0), making it unsuitable for applications where detecting the positive class is critical, such as disease detection, fraud identification, or anomaly detection.

While the model demonstrates excellent performance in predicting the negative class, its extreme bias toward the negative class renders it ineffective for cases where identifying the positive class is essential. To address this bias, techniques such as class rebalancing, oversampling, or class weighting should be implemented to enable the model to work more equitably and improve its ability to detect the positive class.

E. Gradient Boosting-Confusion Matrix

Figure 4 presents the confusion matrix illustrating the performance evaluation of the Gradient Boosting model on the test data. The matrix details the model's predictions across the categories of True Negative (TN), False Positive (FP), False Negative (FN), and True Positive (TP), providing critical insights into the model's ability to distinguish between the negative class (0) and the positive class (1).

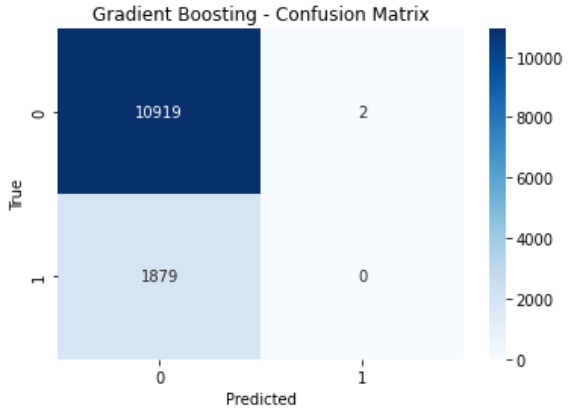

Figure 4. Gradient Boosting-Confusion Matrix

Detailed explanation of each component

1. True Negative (TN): 10,919
   This value represents the number of cases where the actual class is negative (0), and the model correctly predicts it as negative. The high TN value indicates that the model performs almost perfectly in identifying the negative class.
2. False Positive (FP): 2
   This value shows the number of cases where the actual class is negative (0), but the model incorrectly predicts it as positive (1). The very small number of FP demonstrates that the model rarely misclassifies negative instances as positive.
3. False Negative (FN): 1,879
   FN indicates the number of cases where the actual class is positive (1), but the model incorrectly predicts it as negative (0). The high FN value reveals that the model fails to identify a significant proportion of positive cases.
4. True Positive (TP): 0
   TP shows the number of cases where the actual class is positive (1), and the model correctly predicts it as positive. In this matrix, the TP value is zero, meaning that the model is unable to identify any positive cases in the dataset.

The confusion matrix indicates that the Gradient Boosting model is highly biased toward the negative class (0), almost entirely ignoring the positive class (1). The model performs exceptionally well in detecting the negative class, as reflected in the very high TN value and near-zero FP value. However, its total failure to detect the positive class (TP = 0) is a significant weakness.

This result highlights the model's strong performance in identifying the negative class but its complete inability to detect the positive class. This issue is most likely due to class imbalance in the dataset. If the positive class is a priority, steps to address this bias—such as class balancing techniques—must be implemented. Without improvement, this model is not recommended for scenarios where detecting the positive class is critically important.

F. CatBoost-Confusion Matrix

Figure 5 presents the confusion matrix for evaluating the performance of the CatBoost model on the test data. The matrix includes four main components: True Negative (TN), False Positive (FP), False Negative (FN), and True Positive (TP). This matrix provides an overview of how the model predicts the target classes, highlighting both successes and errors.

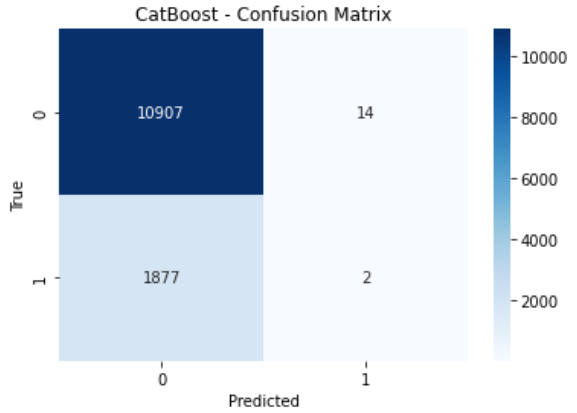

Figure 5. CatBoost-Confusion Matrix

Explanation of Confusion Matrix Components

1. True Negative (TN): 10,907
   TN represents the number of cases where the actual class is negative (0), and the model correctly predicts it as negative. This high value demonstrates that the CatBoost model performs well in detecting the negative class.
2. False Positive (FP): 14
   FP is the number of cases where the actual class is negative (0), but the model incorrectly predicts it as positive (1). This relatively small number compared to the total negative data indicates that the model makes few errors when predicting negative instances as positive.
3. False Negative (FN): 1,877
   FN is the number of cases where the actual class is positive (1), but the model incorrectly predicts it as negative (0). The large FN value shows that the model frequently fails to detect positive cases, which is a major weakness of the model.
4. True Positive (TP): 2
   TP is the number of cases where the actual class is positive (1), and the model correctly predicts it as positive. This very low value indicates that the model struggles significantly to identify positive cases.

The confusion matrix highlights a significant bias toward the negative class (0). While the model excels at recognizing the negative class, as reflected by the high TN and low FP values, it almost entirely fails to detect the positive class (TP = 2). This performance makes the model unsuitable for applications where detecting the positive class is a priority. Without corrective measures such as data balancing or class weighting, the model cannot provide satisfactory results in classification tasks requiring high sensitivity to the positive class.

This study evaluates the performance of four ensemble learning algorithms—XGBoost, AdaBoost, Gradient Boosting, and CatBoost—in predicting classes within a marketing dataset likely characterized by class imbalance. The evaluation results indicate that while each algorithm performs well in predicting negative classes, their ability to identify positive classes is significantly lower. This phenomenon is evident from evaluation metrics such as Recall and AUC, which reflect the models' capacity to detect positive classes and distinguish between positive and negative classes overall.

The analysis shows that XGBoost excels in Precision, indicating its effectiveness in minimizing false-positive predictions. However, its relatively lower Recall and AUC scores compared to other models suggest that XGBoost struggles to consistently recognize positive classes, especially in datasets with class imbalance. Consequently, XGBoost is better suited for applications requiring high precision, such as financial risk analysis, where minimizing false positives for the positive class is a priority.

AdaBoost and Gradient Boosting demonstrate nearly identical performance in terms of Accuracy, Recall, and F1 Score, with Gradient Boosting outperforming in terms of AUC—the highest among all models. This indicates that Gradient Boosting has better capability in distinguishing between positive and negative classes across various thresholds. On the other hand, while AdaBoost exhibits slightly higher Recall, its Precision is lower than that of Gradient Boosting. Both algorithms are appropriate for applications where maximizing the detection of positive cases is critical, such as marketing campaigns or health risk detection, though measures to improve Precision should be considered.

CatBoost shows stable performance across all metrics, with particular strength in handling datasets with complex categorical features. This algorithm achieves relatively high Precision and AUC, indicating that it outperforms some other algorithms in recognizing positive classes. However, like the other models, CatBoost exhibits weaknesses in Recall, reflecting that a significant portion of positive cases remains undetected. The stability of CatBoost makes it a reliable choice for applications involving heterogeneous datasets with numerous categorical features, such as customer segmentation in marketing.

Analysis of the confusion matrix reveals a significant bias toward the negative class in all algorithms. The high number of True Negatives (TN) demonstrates the models' strong capability in recognizing negative classes. However, the low or even zero True Positives (TP) in some algorithms, such as AdaBoost and Gradient Boosting, reflect their failure to detect positive classes. This limitation is attributed to class imbalance, which affects the models' ability to learn from the minority class.

Although all algorithms achieve high overall accuracy, this metric is dominated by predictions of the negative class, rendering it insufficient to reflect actual performance in detecting positive classes. Hence, metrics like Recall, F1 Score, and AUC are more relevant for evaluating models in class imbalance scenarios. Gradient Boosting stands out with

the highest AUC, indicating its potential to better handle class differences compared to other models.

To address these shortcomings approaches such as data balancing through oversampling or undersampling, class weighting during training, or hyperparameter tuning can be applied. Techniques like oversampling the positive class using methods such as SMOTE or assigning higher penalties for misclassifying the positive class through class weighting may help mitigate the bias toward the negative class. Additionally, exploring other models more sensitive to class imbalance, such as LightGBM, could provide a viable alternative.

Overall, this study highlights that each algorithm has specific strengths and weaknesses that must be aligned with the particular requirements of the application. XGBoost is advantageous in scenarios requiring high precision, while AdaBoost and Gradient Boosting are more suitable for situations prioritizing high Recall. CatBoost, with its stability, is a reliable option for datasets with complex categorical features. Additional adjustments are needed to improve model performance in detecting positive classes, enabling more balanced and relevant outcomes for applications requiring high sensitivity to minority classes

IV. CONCLUSIONS

The findings of this study indicate that the four ensemble learning algorithms evaluated—XGBoost, AdaBoost, Gradient Boosting, and CatBoost—perform well in recognizing the negative class but exhibit weaknesses in detecting the positive class, particularly in datasets with class imbalance. While metrics such as Accuracy appear high, they primarily reflect the model's success in predicting the negative class, often overlooking the ability to recognize positive instances.

Gradient Boosting stands out with the highest AUC, demonstrating superior capability in distinguishing between positive and negative classes. Meanwhile, CatBoost exhibits stability when handling datasets with complex categorical features, and XGBoost and AdaBoost show strengths in precision and recall, depending on the application context.

For future research, it is crucial to address the bias toward the negative class by implementing techniques such as data balancing or class weighting to improve model sensitivity to the positive class. Further exploration of more advanced algorithms, hyperparameter optimization, and real-world applications will enhance the outcomes of this research. With a more adaptive and specific approach, models can deliver more balanced and relevant performance, ultimately supporting more accurate and effective data-driven decision-making.

REFERENCES

1. P. Dhal dan C. Azad, *A comprehensive survey on feature selection in the various fields of machine learning*, vol. 52, no. 4. Applied Intelligence, 2022.
2. S. Deepa dan B. Booba, "Predict Diabetes Healthcare Analytics Using Hybrid Gradient Boosting Machine Learning Model," vol. 30, no. 5, hal. 2928–2945, 2024, doi: 10.53555/kuey.v30i5.3371.
3. S. González, S. García, J. Del Ser, L. Rokach, dan F. Herrera, "A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities," *Inf. Fusion*, vol. 64, no. July, hal. 205–237, 2020, doi: 10.1016/j.inffus.2020.07.007.
4. S. M. Ganie, P. K. D. Pramanik, S. Mallik, dan Z. Zhao, "Chronic kidney disease prediction using boosting techniques based on clinical parameters," *PLoS One*, vol. 18, no. 12 December, hal. 1–21, 2023, doi: 10.1371/journal.pone.0295234.
5. F. Mazhar, W. Akbar, M. Sajid, N. Aslam, M. Imran, dan H. Ahmad, "Boosting Early Diabetes Detection: An Ensemble Learning Approach with XGBoost and LightGBM," *J. Comput. \& Biomed. Informatics*, vol. 6, no. 02, hal. 127–138, 2024.
6. U. e. Laila, K. Mahboob, A. W. Khan, F. Khan, dan W. Taekeun, "An Ensemble Approach to Predict Early-Stage Diabetes Risk Using Machine Learning: An Empirical Study," *Sensors*, vol. 22, no. 14, hal. 1–15, 2022, doi: 10.3390/s22145247.
7. M. H. D. M. Ribeiro dan L. dos Santos Coelho, "Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series," *Appl. Soft Comput. J.*, vol. 86, hal. 105837, 2020, doi: 10.1016/j.asoc.2019.105837.
8. P. S. Washburn, Mahendran, Dhanasekharan, Periyasamy, dan Murugeswari, "Investigation of the severity level of diabetic retinopathy using AdaBoost classifier algorithm," *Mater. Today Proc.*, vol. 33, 3037–3042, 2020, doi: 10.1016/j.matpr.2020.03.199.
9. F. NUSRAT, B. UZBAŞ, dan Ö. K. BAYKAN, "Gradient Boosting Classification kullanarak Diabetes Mellitus Tahmini," *Eur. J. Sci. Technol.*, no. September, hal. 268–272, 2020, doi: 10.31590/ejosat.803504.
10. A. Asselman, M. Khaldi, dan S. Aammou, "Enhancing the prediction of student performance based on the machine learning XGBoost algorithm," *Interact. Learn. Environ.*, vol. 31, no. 6, hal. 3360–3379, 2023, doi: 10.1080/10494820.2021.1928235.

11. A. Ogunleye dan Q. G. Wang, "XGBoost Model for Chronic Kidney Disease Diagnosis," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 17, no. 6, hal. 2131–2140, 2020, doi: 10.1109/TCBB.2019.2911071.
12. J. T. Hancock dan T. M. Khoshgoftaar, "CatBoost for big data: an interdisciplinary review," *J. Big Data*, vol. 7, no. 1, 2020, doi: 10.1186/s40537-020-00369-8.
13. A. A. Ibrahim, R. L. Ridwan, M. M. Muhammed, R. O. Abdulaziz, dan G. A. Saheed, "Comparison of the CatBoost Classifier with other Machine Learning Methods," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 11, hal. 738–748, 2020, doi: 10.14569/IJACSA.2020.0111190.