

Prediction of After-Sales Behavior in E-Commerce Using Machine Learning Models

Yulin Cai^{1*}, Fengqing Chen¹, Jingyang Zhang²

¹Department of Mathematics, Hanshan Normal University, Chaozhou, China

²School of Business, Macau University of Science and Technology, Taipa, Macau, China

Email: *caiy197@163.com

How to cite this paper: Cai, Y.L., Chen, F.Q. and Zhang, J.Y. (2024) Prediction of After-Sales Behavior in E-Commerce Using Machine Learning Models. *Open Journal of Statistics*, 14, 757-774.

<https://doi.org/10.4236/ojs.2024.146035>

Received: November 30, 2024

Accepted: December 21, 2024

Published: December 24, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

With the rapid growth of e-commerce and online transactions, e-commerce platforms face a critical challenge: predicting consumer behavior after purchase. This study aimed to forecast such after-sales behavior within the digital retail environment. We utilized four machine learning models: logistic regression, decision tree, random forest, and XGBoost, employing SMOTE oversampling and class weighting techniques to address class imbalance. To bolster the models' predictive capabilities, we executed pivotal data processing steps, including feature derivation and one-hot encoding. Upon rigorous evaluation of the models' performance through the 5-fold cross-validation method, the random forest model was identified as the superior performer, excelling in accuracy, F1 score, and AUC value, and was thus deemed the most effective model for anticipating consumer after-sales behavior. The findings from this research offer actionable strategies for e-commerce platforms to refine their after-sales services and enhance customer satisfaction.

Keywords

E-Commerce, Consumer Behavior, Machine Learning Algorithms, Random Forest

1. Introduction

The proliferation of e-commerce and the surge in online transactions have made predicting post-purchase behaviors, such as returns, exchanges, and refunds, a critical challenge for e-commerce platforms. According to the China Internet Network Information Center's 54th Statistical Report on Internet Development in China, as of June 2024, China's internet user base reached nearly 1.1 billion (1099.67 million), with an increase of 7.42 million from December 2023, and the

internet penetration rate reached 78.0% [1]. Online shopping has become an integral part of daily life for many consumers. However, as e-commerce continues to grow, issues such as product returns and refunds have also become more prevalent. These after-sales behaviors not only waste merchants' human, material, and financial resources but also pose significant challenges to their operations. Consequently, reducing the rate of returns and refunds and minimizing the occurrence of after-sales behaviors have become critical priorities for most e-commerce platforms.

Traditional methods of after-sales service management often rely on human experience and heuristic rules, which can lead to inefficiencies and a lack of personalization. Moreover, with the growth of big data, it is becoming increasingly difficult to process and analyze consumer behavior manually. In this context, leveraging machine learning models to predict after-sales behaviors has become a promising solution. Machine learning models, which can analyze large volumes of consumer data, hold the potential to accurately predict after-sales needs and optimize service strategies.

This study aims to construct a machine learning-based predictive system for after-sales behavior. By applying various machine learning algorithms, such as decision trees, random forests, and XGBoost, the study will predict consumers' after-sales needs based on historical purchase and service data. The objectives of this research include collecting and analyzing consumer behavior data with a focus on after-sales actions, developing machine learning models to predict after-sales behaviors (e.g., returns, exchanges, complaints), evaluating the performance of these models and comparing the effectiveness of different algorithms, and proposing targeted strategies to improve after-sales services and customer satisfaction. Merchants can dynamically adjust shipping and insurance rates based on machine learning predictions, improving customer satisfaction while minimizing the impact of after-sales activities on operations.

The innovation of this study lies in the development of an automated model that uses machine learning to predict consumer after-sales behavior. The application of the model can reduce the additional costs of online shopping merchants, increase the profit margin of manufacturers, improve the efficiency of after-sales service, reduce the probability of consumers initiating after-sales behavior, and promote effective economic growth.

2. Related Work

In the realm of e-commerce, post-purchase behaviors—including returns, exchanges, and refunds—pose significant operational challenges. As e-commerce platforms seek to optimize after-sales services and mitigate the costs associated with returns, there has been a growing interest in employing machine learning techniques to predict these behaviors. This section reviews relevant studies that highlight the potential of predictive models not only to enhance customer satisfaction but also to inform strategic decision-making in return management.

Yang *et al.* [2] delved into the role of artificial intelligence in shaping online

return policies, particularly in forecasting return volumes through machine learning. Their work underscored the efficacy of machine learning in anticipating return behaviors, thereby enabling businesses to refine their strategies and streamline after-sales services. Building on this, Chen and Ma [3] investigated the determinants of customer satisfaction within online return services. They utilized machine learning to forecast the propensity for returns, underscoring the criticality of comprehending post-purchase consumer dynamics for enhancing satisfaction and optimizing service delivery.

Da Veiga *et al.* [4] provided a comprehensive review of after-sales attributes in e-commerce, with a focus on returns, exchanges, and refunds. Their study illuminated how AI and machine learning can be leveraged to streamline these processes, bolster customer service, and strengthen retention strategies. Liu [5] explored the application of deep learning to forecast customer satisfaction in cross-border e-commerce, with an emphasis on enhancing return services. This research demonstrated the accuracy of machine learning models in predicting return behaviors, thus amplifying the efficiency of after-sales service processes.

Guan *et al.* [6] examined the factors influencing consumer satisfaction in the context of fresh produce e-commerce. They adopted a hybrid approach, combining LDA-SEM and XGBoost, to predict consumer satisfaction and return behaviors, thereby showcasing the efficacy of machine learning in enriching post-purchase consumer experiences in the e-commerce sector. Duong *et al.* [7] conducted a systematic literature review on the application of machine learning techniques to predict product returns in e-commerce. Their study identified various models, including decision trees and random forests, which have been shown to enhance the accuracy of predicting post-purchase behaviors such as returns and exchanges.

Sunarya *et al.* [8] compared the performance of two prevalent machine learning algorithms, Logistic Regression and Random Forest, in predicting e-commerce customer behavior, with a specific focus on customer churn. Their study utilized an E-commerce Customer dataset to delve into the intricacies of customer interactions and behaviors, assessing the predictive capabilities of each model in forecasting post-purchase behaviors like returns, exchanges, and complaints.

These studies collectively contribute to a burgeoning body of knowledge on consumer behavior prediction in e-commerce. They not only provide a foundation for our research but also highlight the need for further exploration, particularly in the context of diverse consumer behaviors and the evolving landscape of e-commerce. Our study aims to build upon these insights by developing a robust predictive model that can accurately forecast post-purchase behaviors, thereby offering valuable strategies for e-commerce platforms to enhance customer service and operational efficiency.

3. Data Sources and Processing

3.1. Data Introduction

The dataset for this analysis is sourced from the customer shopping order infor-

mation of the “Damon Home Flagship Store” on TikTok Mall, which has been anonymized to ensure privacy and exported into an Excel table via the merchant interface. This selection, while providing a rich dataset, inherently limits the generalizability of our findings to other e-commerce platforms or product categories due to the unique characteristics of a single store.

We have defined a target column named “After-sales Behavior” within our dataset to categorize orders. An entry of “1” indicates the presence of after-sales behavior, which includes activities such as returns, exchanges, and refunds. An entry of “0” signifies the absence of such behavior, suggesting that the order was completed without any subsequent actions, and the product was received as intended.

We acknowledge that the specific attributes of the “Damon Home Flagship Store,” including its marketing strategies and customer demographics, may introduce biases into our dataset and affect the after-sales behaviors of consumers. To address these potential biases, we have taken measures to ensure that the dataset is as representative as possible within the scope of our study. Despite these efforts, we recognize that our findings may not be entirely transferable to other e-commerce contexts, and we encourage further research to explore the variability of after-sales behaviors across different platforms and product types.

The data contains a total of 200,270 order data from the store from January 1, 2023 to June 6, 2023, of which some order information is shown in **Figure 1**:

Order Number	Area	Payment Methods	Purchase Time	Number of Products	Product Amount	Freight	Discount Amount	After-sales Behavior
0	1 Guangdong	Tiktok payment	\t2023-06-04 18:24:41	1	9.9	0.0	0.00	1
1	2 Guizhou	Tiktok payment	\t2023-06-04 18:15:43	1	9.9	0.0	5.00	1
2	3 Henan	Alipay	\t2023-06-04 17:31:57	1	19.9	0.0	0.00	1
3	4 Shandong	Alipay	\t2023-06-04 17:18:08	1	14.9	0.0	2.41	1
4	5 Zhejiang	Tiktok payment	\t2023-06-04 17:17:51	2	33.8	0.0	0.00	1
...
200265	200266 Henan	Alipay	\t2023-01-01 00:06:49	1	9.9	0.0	0.00	0
200266	200267 Jiangsu	Tiktok payment	\t2023-01-01 00:04:09	1	29.9	0.0	0.00	0
200267	200268 Henan	Tiktok monthly payment	\t2023-01-01 00:03:44	1	19.9	0.0	0.01	0
200268	200269 Jiangsu	Tiktok monthly payment	\t2023-01-01 00:03:31	1	29.9	0.0	0.00	1
200269	200270 Liaoning	WeChat Pay	\t2023-01-01 00:00:06	1	29.9	0.0	0.00	0

Figure 1. Partial order information diagram.

The original data contains 12 characteristic variables, including order number, product category, region, payment method, order submission time, payment completion time, product quantity, product amount, shipping cost, total discount

amount, order status, and after-sales status. Among them, the after-sales status is the target variable, which is divided into after-sales application and no after-sales application. It is necessary to use the machine learning model to predict the after-sales status based on the existing sample data.

3.2. Data Preprocessing

Data preprocessing is a critical step in machine learning that involves transforming raw data into a format suitable for modeling. This process includes several techniques, such as feature derivation and one-hot encoding, which are essential for converting categorical variables into numerical data [9]. Feature derivation is an important technique in machine learning data preprocessing, as it can improve the performance of the model by converting or combining some variables of the original data into new features so that the model can better capture the information in the data [10]. One-hot encoding, in particular, is a common method used to convert categorical variables into numerical data, which is essential for machine learning models that require numerical input [11].

3.2.1. Feature Derivation

Feature derivation is a critical step in machine learning data preprocessing that can significantly enhance model performance by transforming or combining original data variables into new features that better capture the underlying patterns and information relevant to the prediction task [10]. In this study, we derived new features from the original dataset based on domain knowledge and statistical analysis to improve the predictive power of our machine learning models.

Here we derive the original features and obtain the following three new features as shown in **Table 1**, which improve the accuracy of model prediction.

Table 1. Derived features and descriptions.

Original Features	Derived Features	Derived Feature Description
Area	Regional Development Level	Divided into 5 levels according to the region's GDP in 2023: 5: Shanghai, Beijing 4: Guangdong, Jiangsu, Zhejiang, Fujian 3: Shandong, Hubei, Shaanxi, Chongqing, Inner Mongolia 2: Hunan, Anhui, Jiangxi, Shanxi 1: Other areas
Purchase Time	Shopping time	Divide shopping time periods by the order submission time: 1: 00:00-06:00 2: 06:00-12:00 3: 12:00-18:00 4: 18:00-24:00
Product Amount, Discount Amount	Discount Ratio	The ratio of the total discount amount to the product amount

We carefully selected features for derivation based on their potential to reveal hidden patterns in consumer behavior that could influence after-sales actions. Each feature was chosen to provide unique insights into consumer decision-making processes. For instance, the “Regional Development Level” feature was derived from the “Area” variable, categorizing regions into different levels based on their GDP in 2023. This feature was included because economic development levels have been shown in studies such as Lichtenstein, Drumwright, and Braig [12], as well as de Mooij and Hofstede [13], to correlate with consumer behavior and financial capabilities, which in turn can affect the likelihood of returns or exchanges. Additionally, the “Shopping Time” feature was derived to capture the time of day when orders were placed, as purchase timing can indicate consumer urgency or convenience-driven behavior, potentially impacting the likelihood of after-sales interactions. Prior research suggests that the time of day when consumers make purchases can be indicative of their decision-making processes and may influence post-purchase actions, such as the propensity for returns or exchanges.

Each derived feature was carefully selected to ensure it added unique information to the model and improved its ability to distinguish between customers who would and would not engage in after-sales behaviors. The inclusion of these features allowed our models to capture more nuanced patterns in the data, leading to improved accuracy, F1 score, and AUC compared to using only the original features.

3.2.2. One-Hot Encoding

One-Hot Encoding is a common method for converting categorical variables into numerical data, especially in machine learning models [11]. The core idea is to represent the value of each category as a binary vector with a length equal to the total number of categories, where only one position is 1 and all other positions are 0. Each category corresponds to a unique position. Here, we apply one-hot encoding to the payment method. (Figure 2)

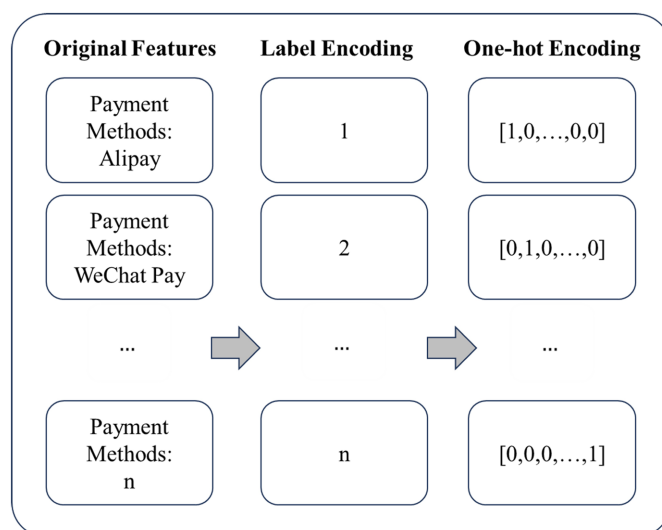


Figure 2. Payment method One-Hot Encoding.

After one-hot encoding, the categorical features can be digitized and then added to the machine learning model for training.

3.2.3. Outlier Processing

The Z-Score method, a standard score measure, is widely used to identify outliers in a dataset. It calculates the number of standard deviations an element is from the mean, facilitating the detection of data points that deviate significantly from the norm [14].

$$Z = \frac{X - \mu}{\sigma}$$

After calculating the Z value, we delete the samples whose absolute value is greater than 3. By processing the outliers, we deleted 1300 outliers and obtained 198,890 sample data.

3.2.4. Data Normalization

The Min-Max Normalization method is employed to scale numerical data within the interval [0, 1], which is particularly beneficial for features with varying scales. This method ensures that all features contribute equally to the model, regardless of their original scale or range. For a feature column *xx*, the normalization formula is given by:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

This process is crucial as it eliminates the impact of inconsistent dimensions across features, thereby improving the effectiveness of the model [15]. By normalizing the data, we ensure that no single feature dominates the model due to its scale, which is especially important in machine learning algorithms where feature scaling can significantly affect performance.

In summary, after feature derivation, encoding, outlier processing and data standardization, we obtain the data required for preliminary analysis as shown in the following **Figure 3**:

	Number of Products	Product Amount	Freight	Discount Amount	Regional Development Level	Shopping Time	Payment Methods_Alipay	Payment Methods_Tiktok monthly payment	Payment Methods_Tiktok payment	Payment Methods_WeChat Pay	Discount Ratio	After-sales Behavior
0	0.000000	0.242647	0.0	0.000000	0.75	1.000000	0	0	1	0	0.000000	1
1	0.000000	0.242647	0.0	0.162338	0.00	1.000000	0	0	1	0	0.420521	1
2	0.000000	0.487745	0.0	0.000000	0.00	0.666667	1	0	0	0	0.000000	1
3	0.000000	0.365196	0.0	0.078247	0.50	0.666667	1	0	0	0	0.134674	1
4	0.333333	0.828431	0.0	0.000000	0.75	0.666667	0	0	1	0	0.000000	1
...
198885	0.000000	0.242647	0.0	0.000000	0.00	0.000000	1	0	0	0	0.000000	0
198886	0.000000	0.732843	0.0	0.000000	0.75	0.000000	0	0	1	0	0.000000	0
198887	0.000000	0.487745	0.0	0.000325	0.00	0.000000	0	1	0	0	0.000418	0
198888	0.000000	0.732843	0.0	0.000000	0.75	0.000000	0	1	0	0	0.000000	1
198889	0.000000	0.732843	0.0	0.000000	0.00	0.000000	0	0	0	1	0.000000	0

Figure 3. Processed data.

4. Models and Methods

4.1. Logistic Regression

Logistic Regression (LR) is a fundamental model used for binary classification problems. It estimates the probabilities of the outcomes and is based on the maximum likelihood method. In their work, Agresti detailed the statistical foundations of logistic regression, making it a suitable choice for our model due to its simplicity and interpretability [16]. LR predicts the probability that an individual belongs to a certain category, such as whether or not they have an after-sales behavior, by converting the linear combination of features into a probability value through a logit function.

The mathematical expression of this function is as follows:

$$P(Y = 1 | X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n)}}$$

where X_1, X_2, \dots, X_n are input features, $\beta_1, \beta_2, \dots, \beta_n$ are model parameters, and e is the base of the natural logarithm. $P(Y = 1 | X)$ is the probability that the target variable Y is equal to 1 given the features X_1, X_2, \dots, X_n .

4.2. XGBoost

eXtreme Gradient Boosting (XGBoost) is an efficient and scalable implementation of the gradient boosting framework. Chen and Guestrin developed XGBoost, which is known for its ability to handle large datasets and provide state-of-the-art performance in many machine learning tasks [17]. The key advantage of XGBoost lies in its ability to model complex, non-linear relationships and handle large datasets efficiently. The algorithm minimizes a combination of loss and regularization terms, effectively preventing overfitting and improving model generalization.

The objective function of XGBoost is given by:

$$L(\theta) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k)$$

where $l(y_i, \hat{y}_i)$ is the loss function and $\Omega(f_k)$ is the regularization term. The loss function is typically cross-entropy for classification problems, and the regularization term penalizes large models to avoid overfitting.

By integrating features such as parallel computation, tree pruning, and regularization, XGBoost can achieve high predictive accuracy, making it a powerful tool in a wide range of machine learning applications, including e-commerce predictions, customer churn analysis, and medical diagnostics [18] [19].

4.3. Decision Tree

The basic idea of a Decision Tree (DT) is to classify or predict data by “splitting” features. When building a decision tree, first select a feature to split so that the data set becomes more “pure” after the split (that is, samples of the same category are concentrated together). This splitting continues until a certain stopping condition is met (such as reaching the maximum tree depth, the number of samples

contained in the leaf node is less than a certain threshold, etc.). Since the data set used is large, the number of features is large, and higher computational efficiency is required. There is an imbalance in the categories, so this article uses the Gini coefficient as the splitting criterion for the decision tree model.

The Gini Index is also a criterion for selecting the best features, especially used in CART models. For the data set D , the calculation formula for the Gini Index is:

$$Gini(D) = 1 - \sum_{i=1}^n p_i^2$$

where p_i is the proportion of the i -th class in the dataset D . For each possible value v of a feature A , its Gini index is:

$$Gini(D, A) = \sum_{v \in \text{values}(A)} \frac{|D_v|}{|D|} * Gini(D_v)$$

where D_v is the subset of the data where feature A takes the value v , D is the total number of samples in the dataset. The feature and threshold that result in the lowest Gini Index are selected for the split at each node of the tree.

Pseudocode for the number of decisions implemented is as follows:

Algorithm: Decision Tree.

Input: Training set $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$;

Attribute set $A = \{a_1, a_2, \dots, a_d\}$.

Procedure: Function TreeGenerate (D, A)

- 1: Create a node node;
- 2: **if** all samples in D belong to the same class C **then**
- 3: Label node as class C leaf node; **return**
- 4: **end if**
- 5: **if** $A = \emptyset$ **OR** all samples in D have the same value for attributes in A **then**
- 6: Label node as leaf node with the most common class in D ; **return**
- 7: **end if**
- 8: Select the best splitting attribute a^* from A ;
- 9: **for** each value a_v^* of attribute a^* **do**
- 10: Create a branch for node; D_v is the subset of D where attribute a^* has value a_v^* ;
- 11: **if** D_v is empty **then**
- 12: Label the branch node as leaf node with the most common class in D ; **return**
- 13: **else**
- 14: Use TreeGenerate ($D_v, A \setminus \{a^*\}$) as the branch node;
- 15: **end if**
- 16: **end for**

Output: A decision tree with node as the root.

The Gini Index is computationally efficient and easy to interpret. It is particularly suitable for large datasets and is commonly used in binary classification problems. Unlike entropy-based measures, the Gini Index tends to be faster because it doesn't involve logarithmic calculations. Moreover, it is robust to class imbalances, which makes it a good fit for many real-world classification tasks, including e-commerce behavior prediction [20] [21].

4.4. Random Forest

Random Forest (RF) is an ensemble learning method that improves prediction accuracy and controls overfitting by building multiple decision trees and outputting average results. Random Forest performs well for problems of various data types and sizes, especially when dealing with high-dimensional data and features with complex interactions [22].

Random Forest uses the bootstrap self-service resampling technique to randomly extract n samples from the training set of the original data with replacement to form a new training set. For each sample set, a decision tree is built based on the characteristics and the best split point is determined. This process is repeated many times to generate m decision trees. Finally, the test set is predicted using the bagging strategy and a majority voting mechanism. The learning process of the random forest algorithm is shown in **Figure 4** below:

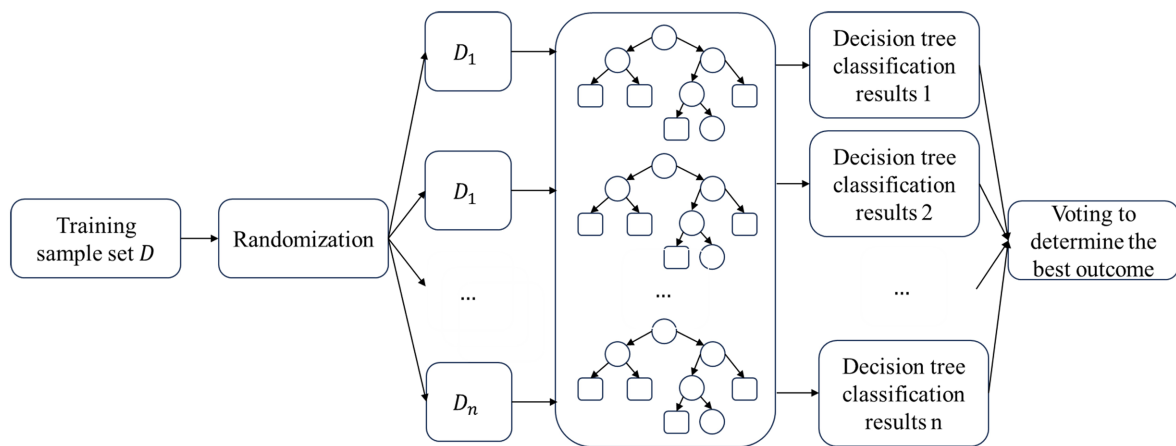


Figure 4. Random forest flowchart.

4.5. Class Imbalance

Class imbalance is a significant issue in many predictive models, particularly when one class of the target variable, such as after-sales behaviors (e.g., returns and refunds), occurs less frequently than the other class (no after-sales behavior). By counting the feature columns, we found that the number of orders with after-sales behavior accounted for 80.6%, indicating a class imbalance. (**Figure 5**)

This often leads to decreased model performance, as the model tends to be biased towards the majority class. To address this issue and improve the accuracy of our predictions, we apply various techniques to balance the dataset, ensuring that

the model can better learn to predict both classes effectively. These methods help mitigate the negative impact of class imbalance and enhance the overall performance of the model.

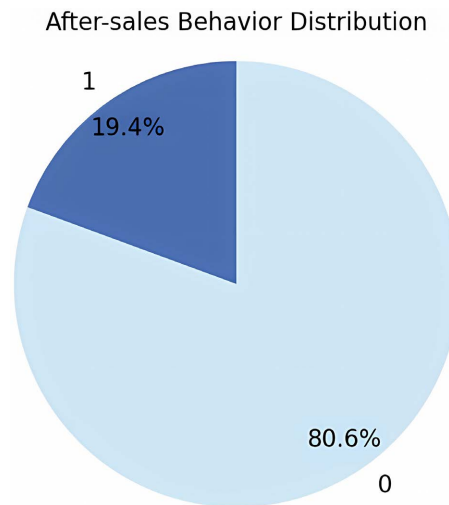


Figure 5. Proportion of orders with after-sales behavior.

1) Oversampling: Increase the sample size of the minority class by duplicating the minority class samples. We can use the SMOTE (Synthetic Minority Over-sampling Technique) method to generate new minority class samples.

2) Undersampling: Reduce the number of majority class samples to balance class distribution.

3) Class weighting: When training logistic regression, decision tree, random forest, XGBoost models, the model pays more attention to the minority class by assigning a larger weight to the minority class.

4.6. Evaluation

We divide the dataset into training and testing sets to train and evaluate the models. Using the selected features, we train logistic regression, decision tree, random forest, and XGBoost models respectively. We make predictions for each model and evaluate its performance using metrics such as Accuracy, F1 Score, and AUC.

1) Accuracy: Accuracy is one of the simplest evaluation indicators, which indicates the proportion of correctly classified samples to the total samples.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Samples}}$$

2) Precision: Precision represents the proportion of samples that are actually positive among all samples predicted to be positive. It can help us evaluate the accuracy of the model in predicting positive samples.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

3) Recall: The recall rate indicates the proportion of samples that are correctly

predicted as positive among all samples that are actually positive.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

4) F1-Score: The F1 score is the harmonic mean of precision and recall, which takes into account the balance between precision and recall. The F1 score ranges from 0 to 1, and the larger the value, the better the model performs in predicting the positive class.

$$\text{F1 Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

5) Area Under Curve (AUC): AUC is the area under the ROC curve, which indicates the ability of the model to distinguish between different categories. The value of AUC ranges from 0 to 1, and the higher the value, the stronger the model's ability to distinguish. When the AUC value is 0.5, it means that the model is no different from random guessing; when the AUC value is 1.0, it means that the model can perfectly distinguish between the two categories.

6) Receiver Operating Characteristic Curve (ROC): The ROC curve plots the relationship between the false positive rate (FPR) and the true positive rate (TPR), and the AUC value is the area under the ROC curve.

$$\text{AUC} = \int_0^1 \text{ROC Curve} dx$$

5. Main Results

In this study, we evaluated four machine learning models—Logistic Regression, Decision Tree, Random Forest, and XGBoost—to predict after-sales behaviors in online shopping, such as returns, exchanges, and refunds. The dataset, which is imbalanced with a higher frequency of non-return behaviors, poses a significant challenge to model performance. To address this, we applied class balancing techniques, including SMOTE oversampling and the use of class weights. The models were rigorously evaluated using five-fold cross-validation, with performance assessed through Accuracy, F1-Score, and ROC curve analysis.

5.1. Model Performance Summary

After rigorous testing, the performance metrics of the four models are encapsulated in the confusion matrices and AUC values presented in **Figure 6** and **Figure 7**. These figures provide a visual representation of each model's ability to distinguish between after-sales behaviors and non-after-sales behaviors.

This figure offers a detailed look at the true positives, false positives, true negatives, and false negatives for each model, serving as a critical tool for assessing classification model performance.

Building on the insights from the confusion matrices, **Figure 7** displays the ROC curves for our models, measuring their ability to discriminate between different classes. The Area Under the Curve (AUC) is a key metric that quantifies the overall performance of the models, especially in the context of imbalanced datasets.

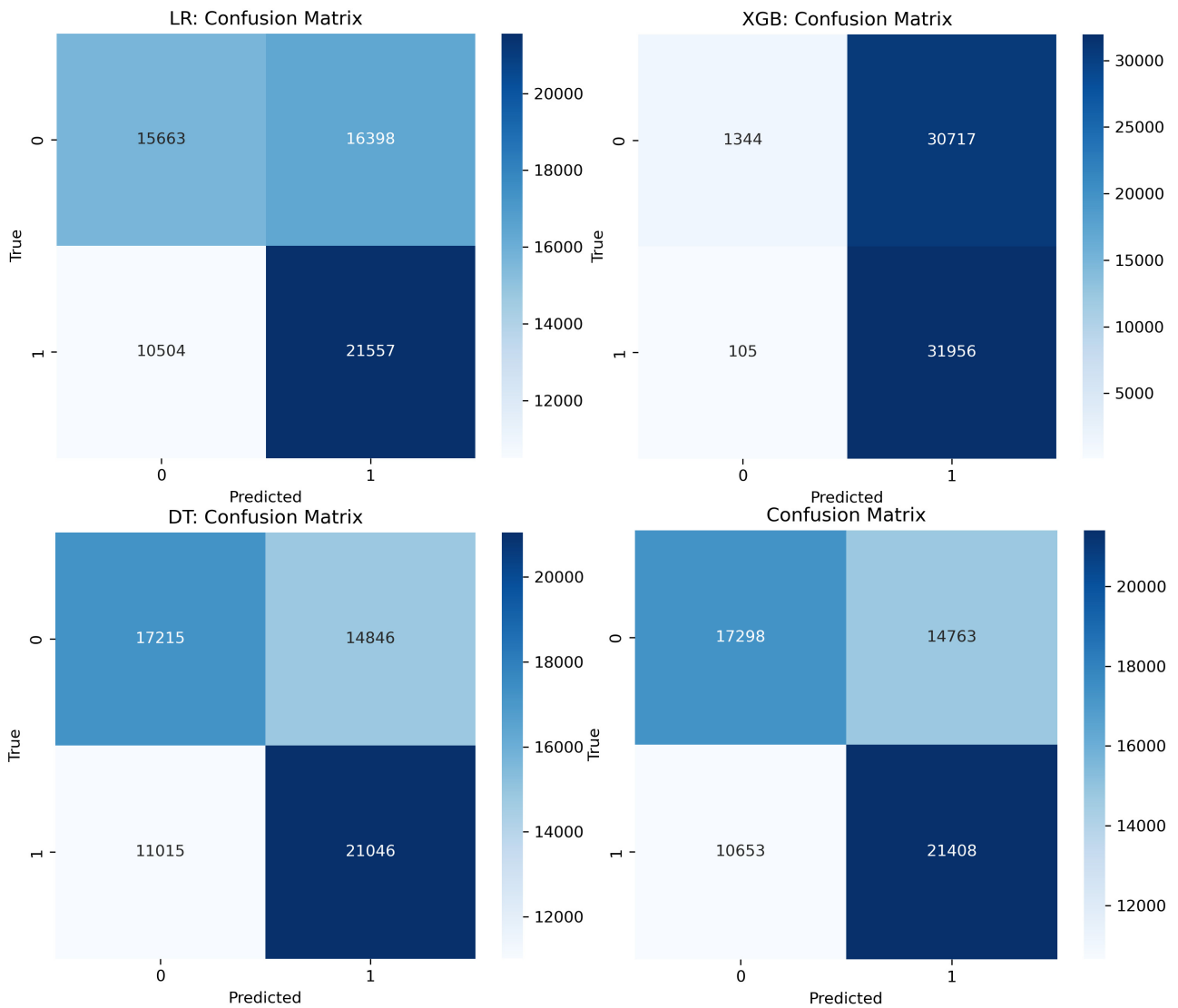
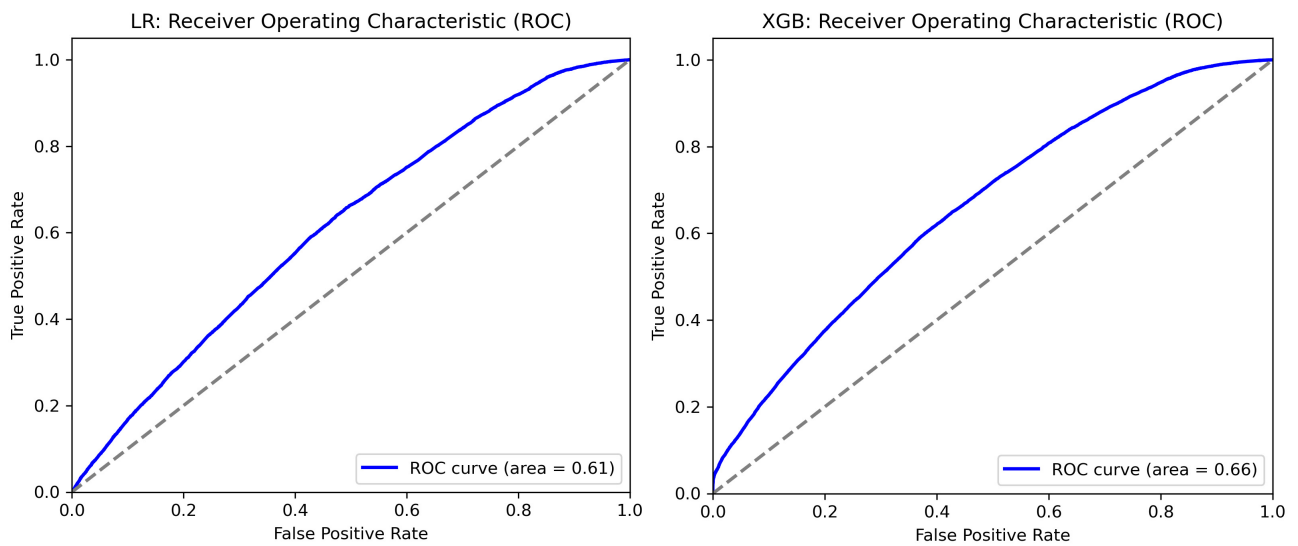


Figure 6. Confusion matrix diagram of four models.



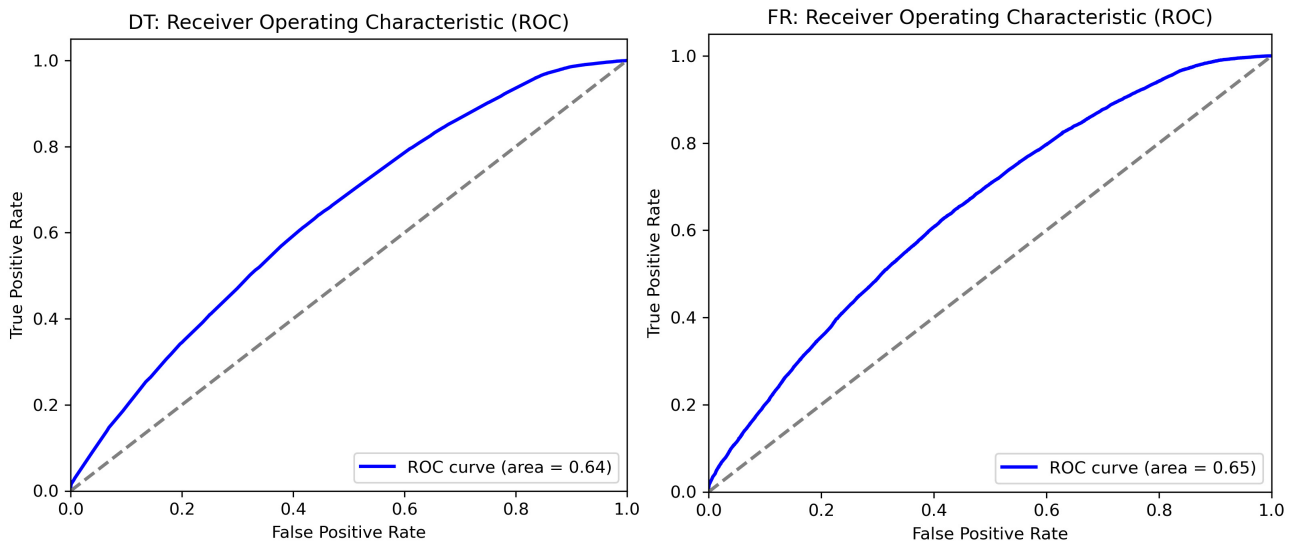


Figure 7. ROC curve of machine learning prediction results.

Summarizing the data from these visual representations, we present in **Table 2**, which includes the accuracy, R^2 , F1 Score, and AUC for each model.

Table 2. Prediction results including accuracy, R^2 , F1 Score, and AUC.

Model	Mean Accuracy	Mean R^2	Mean F1 Score	Mean AUC
Logistic Regression	0.5794	0.5794	0.6152	0.6126
XGBoost	0.5184	0.5184	0.6743	0.6582
Decision Tree	0.5956	0.5956	0.6166	0.6376
Random Forest	0.6025	0.6025	0.6262	0.6467

To provide a comprehensive overview of the models' performance, we have included a radar chart (**Figure 8**) that visualizes the mean accuracy, mean R^2 , mean F1 score, and mean AUC for each model. This chart offers a clear and concise comparison of the models' performance across different metrics.

The radar chart (**Figure 8**) reinforces our findings that Random Forest and XGBoost outperform Logistic Regression and Decision Tree in terms of accuracy, F1 score, and AUC. While XGBoost excels in F1 score and AUC, it is less suitable for large-scale data analysis due to its high complexity and computational cost. Random Forest, on the other hand, offers a consistent performance across all metrics and is more efficient in terms of speed and training time, making it the optimal choice for predicting post-purchase behavior in e-commerce contexts.

In summary, Random Forest performed the best overall, achieving a mean accuracy of 0.6025, F1 score of 0.6262, and AUC of 0.6467. Random Forest's robust performance across all metrics suggests it is the most reliable choice for this problem, demonstrating strong predictive power and stability without overfitting. XGBoost, while having a lower accuracy (0.5184), excelled in F1 score (0.6743) and AUC (0.6582), showing its ability to effectively balance precision and recall,

especially in class-imbalanced scenarios. Decision Tree showed comparable results to Logistic Regression, with a slight improvement in accuracy (0.5956) and AUC (0.6376), indicating that it could be useful in cases where model interpretability is critical. Logistic Regression performed reasonably well but had relatively lower AUC and F1 scores compared to the more complex models, with a mean accuracy of 0.5794, suggesting that while it is suitable for simpler tasks, it may not be the best choice for imbalanced datasets or non-linear relationships.

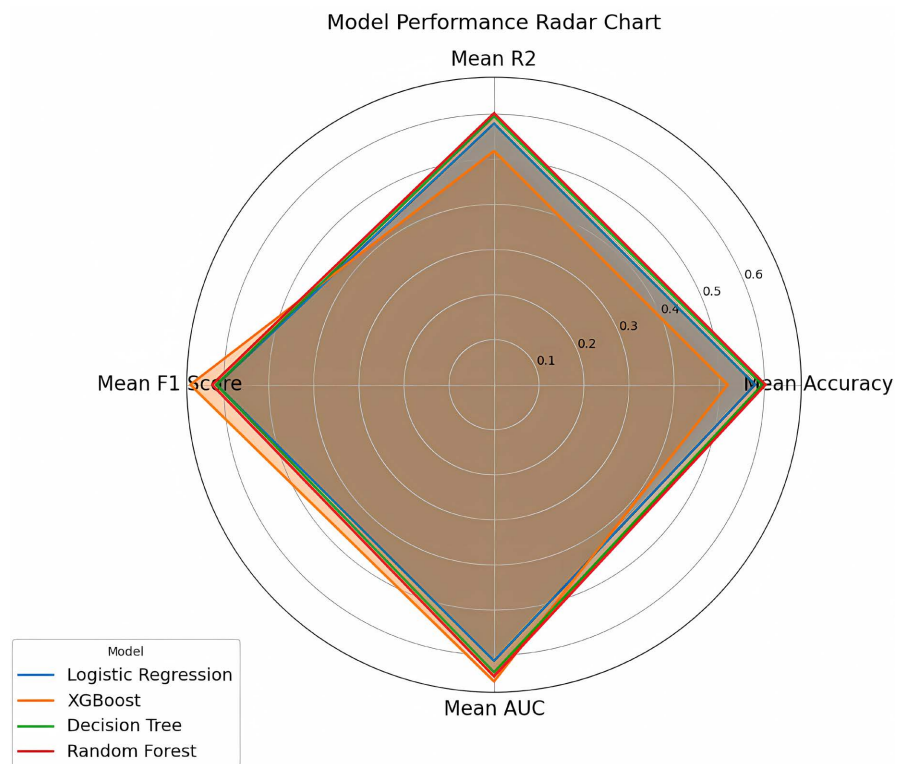


Figure 8. Model performance radar chart.

5.2. Insights and Recommendations

In conclusion, our analysis shows that XGBoost and Random Forest both outperform Logistic Regression and Decision Tree in terms of accuracy, F1 score, and AUC. While XGBoost excels in F1 score and AUC, indicating the best balance between accuracy and recall, especially in the presence of class imbalance, it is less suitable for large-scale data analysis like e-commerce or online shopping due to its high complexity and computational cost. On the other hand, Random Forest consistently performs well across all metrics and is more efficient in terms of speed and training time, offering a lower computational cost. Overall, Random Forest provides the best combination of performance, speed, and efficiency, making it the optimal choice for predicting post-purchase behavior in e-commerce contexts.

6. Conclusions

In this paper, we apply four machine learning models, logistic regression, decision

tree, random forest, and XGBoost, to predict post-purchase behavior for online shopping, focusing on returns, exchanges, and refunds. Results show that while logistic regression provides a useful baseline, it performs poorly in handling class imbalance compared to more complex models. XGBoost performs well in terms of F1 score and AUC, and excels in scenarios that require high precision and recall, but have a high computational cost. Decision tree models, while effective, are more suitable for applications that prioritize model interpretability. The random forest model outperforms the other models with the highest accuracy and stability, making it the most reliable choice for handling complex datasets with class imbalance.

While this study provides valuable insights, there are several limitations to consider. The current dataset is limited in size and may not fully capture the complexities of consumer behavior, especially in the context of more personalized factors. Additionally, the use of oversampling methods like SMOTE may introduce noise or lead to overfitting in some cases. Future research could focus on exploring alternative oversampling strategies, integrating more advanced features (such as customer sentiment or browsing behavior), and refining the machine learning models. Incorporating ensemble methods or hybrid models could further enhance predictive accuracy. Moreover, improving model efficiency, particularly for computationally intensive algorithms like XGBoost, would allow for more scalable real-time applications.

The findings of this study have significant practical implications for e-commerce platforms aiming to optimize their after-sales services. By leveraging machine learning models, merchants can better predict post-purchase behaviors, such as returns and exchanges, and develop targeted strategies to address these behaviors. For example, merchants can dynamically adjust shipping fees and insurance rates based on predicted after-sales behavior, minimizing operational costs while improving customer satisfaction. These strategies can lead to more efficient resource allocation, better customer retention, and overall enhanced service quality.

In conclusion, the application of machine learning models, particularly Random Forest and XGBoost, offers an effective approach to predicting and managing after-sales behavior in online shopping. These models provide valuable insights that can help e-commerce platforms improve their customer service strategies, reduce operational inefficiencies, and ultimately enhance the consumer experience.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] China Internet Network Information Center (CNNIC) (2024) The 54th Statistical Report on China's Internet Development.

- [2] Yang, G., Ji, G. and Tan, K.H. (2020) Impact of Artificial Intelligence Adoption on Online Returns Policies. *Annals of Operations Research*, **308**, 703-726. <https://doi.org/10.1007/s10479-020-03602-y>
- [3] Chen, J. and Ma, Y. (2020) What Constitutes Excellent User Experience in Online Consumers' Return Services. *International Journal of Services Technology and Management*, **26**, 291-304. <https://doi.org/10.1504/ijstm.2020.107438>
- [4] Ferraz, R.M., da Veiga, C.P., da Veiga, C.R.P., Furquim, T.S.G. and da Silva, W.V. (2023) After-Sales Attributes in E-Commerce: A Systematic Literature Review and Future Research Agenda. *Journal of Theoretical and Applied Electronic Commerce Research*, **18**, 475-500. <https://doi.org/10.3390/jtaer18010025>
- [5] Liu, H. (2024) Return Avoidance in Online Shopping: The Role of Return Credits and Purchase-Risk Notices.
- [6] Guan, G., Liu, D. and Zhai, J. (2022) Factors Influencing Consumer Satisfaction of Fresh Produce E-Commerce in the Background of COVID-19—A Hybrid Approach Based on LDA-SEM-XGBoost. *Sustainability*, **14**, Article No. 16392. <https://doi.org/10.3390/su142416392>
- [7] Duong, Q.H., Zhou, L., Meng, M., Nguyen, T.V., Ieromonachou, P. and Nguyen, D.T. (2022) Understanding Product Returns: A Systematic Literature Review Using Machine Learning and Bibliometric Analysis. *International Journal of Production Economics*, **243**, Article ID: 108340. <https://doi.org/10.1016/j.ijpe.2021.108340>
- [8] Sunarya, P.A., Rahardja, U., Chen, S.C., *et al.* (2024) Deciphering Digital Social Dynamics: A Comparative Study of Logistic Regression and Random Forest in Predicting e-Commerce Customer Behavior. *Journal of Applied Data Sciences*, **5**, 100-113. <https://doi.org/10.47738/jads.v5i1.155>
- [9] Maharana, K., Mondal, S. and Nemade, B. (2022) A Review: Data Pre-Processing and Data Augmentation Techniques. *Global Transitions Proceedings*, **3**, 91-99. <https://doi.org/10.1016/j.gltp.2022.04.020>
- [10] Chandrashekar, G. and Sahin, F. (2014) A Survey on Feature Selection Methods. *Computers & Electrical Engineering*, **40**, 16-28. <https://doi.org/10.1016/j.compeleceng.2013.11.024>
- [11] Rodríguez, P., Bautista, M.A., González, J. and Escalera, S. (2018) Beyond One-Hot Encoding: Lower Dimensional Target Embedding. *Image and Vision Computing*, **75**, 21-31. <https://doi.org/10.1016/j.imavis.2018.04.004>
- [12] Lichtenstein, D.R., Drumwright, M.E. and Braig, B.M. (2004) The Effect of Corporate Social Responsibility on Customer Donations to Corporate-Supported Nonprofits. *Journal of Marketing*, **68**, 16-32. <https://doi.org/10.1509/jmkg.68.4.16.42726>
- [13] De Mooij, M. and Hofstede, G. (2011) Cross-Cultural Consumer Behavior: A Review of Research Findings. *Journal of International Consumer Marketing*, **23**, 181-192. <http://dx.doi.org/10.1080/08961530.2011.578057>
- [14] Box, G.E.P. (1976) Science and Statistics. *Journal of the American Statistical Association*, **71**, 791-799. <https://doi.org/10.1080/01621459.1976.10480949>
- [15] Garcia, S., Derrac, J., Cano, J.R. and Herrera, F. (2012) Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34**, 417-435. <https://doi.org/10.1109/tpami.2011.142>
- [16] LaValley, M.P. (2008) Logistic Regression. *Circulation*, **117**, 2395-2399. <https://doi.org/10.1161/circulationaha.106.682658>
- [17] Chen, T. and Guestrin, C. (2016) XGBoost: A Scalable Tree Boosting System. *Pro-*

ceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, 13-17 August 2016, 785-794.

<https://doi.org/10.1145/2939672.2939785>

- [18] Song, P. and Liu, Y. (2020) An XGBoost Algorithm for Predicting Purchasing Behaviour on E-Commerce Platforms. *Tehnički Vjesnik*, **27**, 1467-1471.
<https://doi.org/10.17559/TV-20200808113807>
- [19] Gan, L. (2022) XGBoost-Based E-Commerce Customer Loss Prediction. *Computational Intelligence and Neuroscience*, **2022**, Article ID: 1858300.
<https://doi.org/10.1155/2022/1858300>
- [20] Kim, S. and Lee, H. (2022) Customer Churn Prediction in Influencer Commerce: An Application of Decision Trees. *Procedia Computer Science*, **199**, 1332-1339.
<https://doi.org/10.1016/j.procs.2022.01.169>
- [21] Vanneschi, L., Horn, D.M., Castelli, M. and Popovič, A. (2018) An Artificial Intelligence System for Predicting Customer Default in E-Commerce. *Expert Systems with Applications*, **104**, 1-21. <https://doi.org/10.1016/j.eswa.2018.03.025>
- [22] Biau, G. and Scornet, E. (2016) A Random Forest Guided Tour. *Test*, **25**, 197-227.
<https://doi.org/10.1007/s11749-016-0481-7>