Journal of Information Technology Management
ISSN-OnLine: 2980-7972
University of Tehran

Predicting Customer Lifetime Value Using Machine Learning Algorithms with Cross-Validation

Zahra Muhammed Al-Hashimy*

*Corresponding author, Department of Economics of Bank Management, College of Business Economics, Al-Nahrain University, Baghdad, Iraq. E-mail: st14@ced.nahrainuniv.edu.iq

Ali Ibrahim

Department of Economics of Bank Management, College of Business Economics, Al-Nahrain University, Baghdad, Iraq. E-mail: Prof.ali@nahrainuniv.edu.iq

Feryal Musharraf Eidan

Department of Economics of Bank Management, College of Business Economics, Al-Nahrain University, Baghdad, Iraq. E-mail: Fryal@nahrainuniv.edu.iq

Journal of Information Technology Management, 2026, Vol. 18, Special Issue, pp. 1-22.
Published by the University of Tehran, College of Management
doi: https://doi.org/10.22059/jitm.2026.109971
Article Type: Research Paper
© Authors
Received: August 04, 2026
Received in revised form: October 16, 2026
Accepted: November 12, 2026
Published online: December 01, 2026

Abstract

This study highlights the importance of using machine learning (ML) algorithms to predict customer lifetime value (CLV) while ensuring fair and realistic treatment of datasets during the training process without favoring specific feature classes. The research applies several ML algorithms, including Gradient Boosting (GB), Logistic Regression (LR), Random Forest (RF), Linear Support Vector Machine (SVM), and Neural Network (NN), using a five-fold cross-validation technique for repeated training and testing. The RandomizedSearchCV method is used for hyperparameter tuning, along with data shuffling and fixed random seeds to ensure reproducibility. Model performance is evaluated using accuracy, precision, recall, F1-score, and ROC-AUC metrics through Python programming. The approach aims to achieve better behavioral understanding because of ML's ability to handle large, mixed, and diversified datasets with reduced error after preprocessing and training, allowing models to generalize effectively to new data. The comparison results show that the Gradient Boosting algorithm achieved the statistically highest mean ROC-AUC among the evaluated algorithms, while NN achieved the highest holdout ROC-AUC; GB consistently outperformed NN based on the statistical comparison across the cross-validation folds.

Keywords: Logistic Regression, Gradient Boosting, Random Forest, Linear SVM, Neural Network, Python, Customer Lifetime Value

Introduction

VahanBima is one of the most prominent motor vehicle insurance institutions in India, prioritizing customer satisfaction and service quality. The company provides various types of insurance policies that allow customers to claim compensation for vehicle loss or damage according to the level of damage, provided that customers pay the required premiums (Insurance Authority, 2016).

This study investigates the versatility of machine learning algorithms using the Python platform, which is a programming language that consists of multiple libraries such as NumPy, pandas, and scikit-learn (Ibrahim, 2026), to predict customer lifetime value (CLV) and achieve better behavioral understanding because of their ability to handle large, mixed, and diverse datasets with lower error rates (Abidar et al., 2023). The company aims to enhance its services by providing personalized solutions to customers in order to maintain loyalty, increase retention, and reduce future churn by identifying customers with high CLV.

The research gap addressed in this study uses the dataset's naturally occurring distribution and realistic conditions when using ML algorithms with the RandomizedSearchCV method to perform cross-validation across the selected algorithms with optimized hyperparameters over five training and testing folds to achieve stable and representative outcomes. This combination with statistical significance provides more bias when deciding the best model that goes beyond the evaluation metrics. Many previous studies achieved high prediction accuracy by relying on different resampling techniques that alter the original customer distribution, which may reduce the representativeness of the sample to the population.

Literature Review

This research demonstrates the effectiveness of machine learning algorithms in predicting Customer Lifetime Value with reduced overfitting and data leakage while maintaining dataset stability. Logistic Regression, Random Forest, Gradient Boosting, Neural Networks, and Support Vector Machines are used to perform CLV prediction with cross-validation techniques; here is a detailed description of each:

- Logistic regression – Logistic regression is a supervised machine learning algorithm that estimates the probability of an observation belonging to a particular class, with predicted probabilities ranging from 0 to 1. It was selected as the baseline linear classifier (Osisanwo et al., 2017).

- Random forest – Random forest is a supervised machine learning algorithm that combines multiple decision trees, with each tree trained using randomly selected subsets of the data and features. It was selected because of its ability to handle large datasets and capture nonlinear patterns (Alnuaimi & Albaldawi, 2024).

- Gradient boosting – Gradient boosting is a supervised machine learning algorithm that combines multiple decision trees sequentially, with each subsequent tree designed to improve the predictions of the previous trees through an optimization process. It was selected because of its effectiveness in modeling structured and complex data patterns (Ke et al., 2017; Natekin & Knoll, 2013).

- Neural network – A neural network is a supervised machine learning algorithm consisting of multiple interconnected layers of computational nodes (neurons) that transform input variables into predictions or outputs. It was selected because of its ability to capture complex nonlinear patterns in data (Shanbharkar et al., 2024).

- Support vector machine – A support vector machine (SVM) is a supervised machine learning algorithm that identifies an optimal hyperplane for separating observations into different classes while maximizing the margin between the hyperplane and the nearest data points. It was selected because of its effectiveness in handling high-dimensional data and complex classification patterns (Ali & Mashwani, 2023).

Over the years, different methods have been proposed for predicting Customer Lifetime Value (CLV). To highlight a few, Ibrahim and Kaml (2024) used a different modeling approach to predict customer behavior through fuzzy multiple linear regression (FMLR), applying it to data collected from Kaggle datasets with more than 200 customers and evaluating the model using the mean squared error (MSE) metric. Mabel et al. (2025) scored customers from 1 to 5 for each RFM variable and divided them into segments using machine learning models such as regression, decision trees, and random forest. These models were evaluated using RMSE (root mean square error) and MAE (mean absolute error) metrics with Python libraries, including pandas, scikit-learn, and Matplotlib, and were applied to a dataset of 10,000 customers based on their historical transactions.

Akter et al. (2025) combined RFM analysis with machine learning algorithms to improve the accuracy of CLV prediction. Customer-related data were collected through electronic platforms and financial transaction records, then segmented into different groups using K-means clustering along with regression, random forest, and neural network algorithms. Similarly, Haddadi and Hamidi (2025) used a combination of four key measures: expected purchases using the BG-NBD model, average expected value using the Gamma-Gamma model, customer classification using K-means clustering, and a customer status index using the Markov chain model. These measures were integrated with the dataset, and neural network, random forest, SVM, and XGBoost models were applied to increase prediction accuracy and reduce errors.

However, many previous studies achieved high prediction accuracy by relying on different resampling techniques that alter the original customer distribution, which may lead to overfitting and reduce the representativeness of the sample to the population. Therefore, to

address this gap, the present study uses the dataset's naturally occurring distribution and realistic conditions when using ML algorithms, with the RandomizedSearchCV method to perform cross-validation across the selected algorithms with optimized hyperparameters over five training and testing folds. Multiple evaluation metrics are used to assess algorithm performance, ensuring realistic treatment of the dataset without favoring specific feature classes. This approach minimizes disturbances in the training process, improves result stability, and provides a more representative reflection of the population. The overall structure of the proposed approach is illustrated in Figure 1.

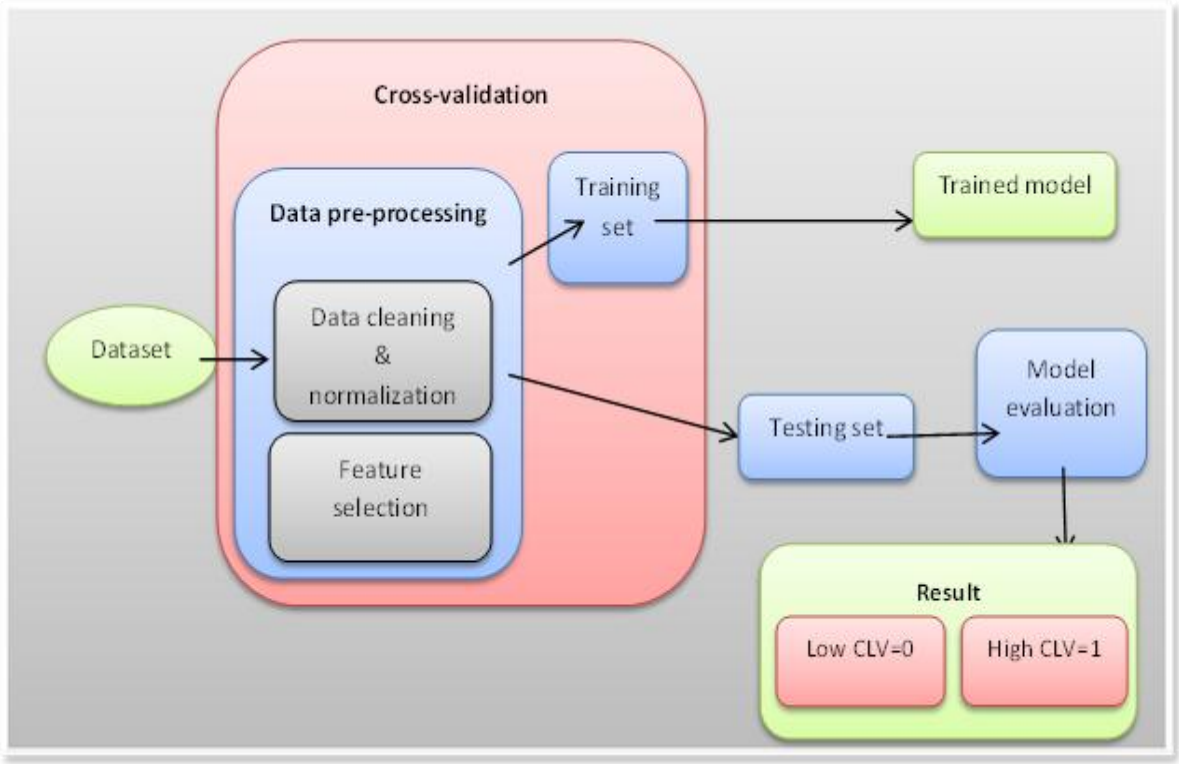


Figure 1. Theoretical Structure for Customer CLV Prediction

CLV prediction with ML algorithms:

Customer Lifetime Value (CLV) represents the present value of all future cash flows generated by a customer over time. Machine learning enables automatic learning from data patterns and supports decision-making without continuous human intervention. It enhances an organization's value by helping develop effective strategies for customer retention and attracting new customers, thereby maintaining market stability (Blattberg et al., 2009). To support CLV prediction, several key elements are considered, including purchase frequency, customer lifetime, average order value, customer loyalty, customer satisfaction, customer churn rate, and RFM values (Ali & Shabn, 2024). Machine learning analyzes data patterns to generate insights that assist in decision-making based on population samples without the need for continuous human involvement. By examining individual customer data, machine learning

enables the segmentation of customers into smaller groups or categories and identifies suitable services and promotional strategies that align with customer behavior (Gil, 2020).

Therefore, this study employs a cross-validation technique as an evaluation method to select optimal hyperparameters and apply them across k-folds for different machine learning algorithms, ensuring stable, reliable, and generalizable model performance (Lumumba et al., 2024).

Methodology

Data Description:

The data for this research were obtained from Kaggle (kaggle.com/datasets), as detailed in Appendix 1, and consist of 148,989 observations. The dataset was divided into training (60%) and testing (40%) sets for binary classification. The classification was connected to customer segmentation using RFM analysis to reduce the potential skewness of CLV values. A median CLV threshold of 66,396.0 was used to define the two classes, where CLV ≥ 66,396.0 was coded as 1 (high CLV) and CLV < 66,396.0 was coded as 0 (low CLV). The median was used to divide the dataset into two relatively balanced groups and reduce the influence of extreme values on the classification process (Tanga & Li, 2023). A higher CLV indicates a stronger customer relationship and greater potential for customer retention. The dataset consists of the following variables:

1. **Numerical data**, such as marital status (0 = single, 1 = married), Vintage-No. (minimum = 1, maximum = 8), and claim amount (minimum = 0, maximum = 20,102).

2. **Categorical data**, such as gender (male, female), area (rural, urban), qualification (high school, bachelor's degree, others), type of policy (silver, platinum, gold), and policy (A, B, C).

3. **Mixed data**, such as income (2L–5L, 5L–10L, more than 10L) and number of policies (1, more than 1).

And the formulas for the algorithms start with logistic regression formulas such as:

$$\mathcal{y}\iota = 1\ (\frac{1}{1} + e^{-(\beta o + \sum_{j=1}^{p} \beta j x i j)} \geq \tau)$$

$$\mathcal{J}(\beta) = -\frac{1}{n}\sum_{i=1}^{n}[\mathcal{y}\iota \log(pi) + (1 - \mathcal{y}\iota)\log(1 - pi)]$$

Then with random forest such as:

$$\hat{y}_i = \arg\max_{k\in\{1,\dots,k\}} P(y = k|xi \in R_m)$$

$$I_{Gini}(R_m) = 1 - \sum_{K=1}^{K} p^2{}_{mk}$$

And then the gradient boosting formulas are as follows:

$$\gamma_m h_m(x) \sum_{m=1}^{M} = F_m(x)$$

$$\log(1 + \exp(-yF(x))) = L(y, F(x))$$

Then the support vector machine formulas are as follows:

$$f(x) = w\ x + b \qquad \hat{y} = \begin{cases} +1,\ if\ w^T\ xi + b \geq 0 \\ -1,\ if\ w^T\ xi + b < 0 \end{cases}$$

$$\mathcal{L}_{hinge} = \sum_{i=1}^{n} \max(0, 1 - yi(\ w^T xi + b))$$

And the neural network formulas are as follows:

$$z = \sum_{i=1}^{n} wixi + b \qquad : a = f(z)$$

$$L = -\frac{1}{m}\sum_{i=1}^{m} [yi \log(ai) + (1 - yi) \log(1 - ai)]$$

b = bias w = weight

xi = observation i's feature

y = true value

$\hat{y}$ = predicted value

K = number of classified segments

τ = classification starting poite of 0.5

βo = beta coeffecient

$_m\mathcal{R}$ = class node

$p^2{}_{mk}$ = part of a node in a leaf

$j\beta$ = feature j's beta coeffecient

n = total number of observations

ai = predicted output

L = loss(error)

z = linear combination

x = input data

Data cleaning and feature engineering:

After data cleaning, potential errors were identified and addressed, particularly because some of the data were collected using a checklist format. The cleaned dataset was then divided into training (60%) and testing (40%) sets using the NumPy and Pandas libraries. Feature scaling was subsequently performed using Min–Max scaling and z-score standardization to transform the data to an appropriate scale. The preprocessing steps were performed within each fold of the five-fold cross-validation to prevent data leakage between the training and validation sets. Feature engineering was then conducted as presented in Table 1.

Table 1. Preprocessing Methods Used for Each Data Type

Data type	Variables	Method used
Numerical	marital_status, vintage, claim-amount	scaling, transformations
Categorical	Gender, area, qualification, policy, type-of-policy	mapping, encoding
mixed	Income, num-policies	splitting, encoding

Validation strategy:

The validation strategy used in this study consists of the following steps:

1. train_size=0.6, holdout-split=0.2, test_size=0.4, stratified.
2. CV for tuning: Stratified K Fold (n_splits=5, shuffle=True, seed=42)
3. Tuning: Randomized Search CV (n_iter=10, scoring='roc-auc')

The randomizedsearch CV method used to optimize the needed hyperparameters through sklearn model-selection library, were the StratifiedKFold got chosen with 5 splits (5-fold), and the data to prevent bias in evaluation, with 10 iterations for the five algorithms, controlling randomness using seed=42 for reproducible results.

Tuning hyperparameters:

To prepare the data for model training, cross-validation was used as the evaluation strategy. Specifically, five-fold stratified cross-validation was applied, and the hyperparameters used for each algorithm are presented in Table 2.

Table 2. Hyperparameters Used for Each Algorithm

Model	Hyperparameters
Logistic regression	{'model__solver', 'model__penalty', 'model__class_weight', 'model__C'}
Random Forest	{'model__n_estimators', 'model__min_samples_split', 'model__max_depth'}
Gradient boosting	{'model__min_samples_leaf', 'model__max_leaf_nodes', 'model__max_iter', 'model__max_depth', 'model__learning_rate', 'model__l2_regularization'}
Linear SVM	{'model__loss', 'model__class_weight', 'model__C'}
Neural Network	{'model__learning_rate_init', 'model__hidden_layer_sizes', 'model__alpha', 'model__activation'}

Evaluation metrics:

All measures used to assess model performance compare the predicted values with the actual values to evaluate the accuracy of the predictions (Obi, 2023).

The confusion matrix used to represent the symbolic values is presented below (Vujović, 2021).

Table 3. Confusion Matrix

		Actual values	
		P	N
Predicted values	P	TP	FP
	N	FN	TN

This refers to the following symbolism (Vujović, 2021):

TP: number of true positive predictions.

TN: number of true negative predictions.

FP: number of false positive predictions.

FN: number of false negative predictions.

Used through the following metrics:

1- Accuracy: It's a classification measure that provides a percentage of outcomes correctly classified into their categories (Miller, 2024).

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

2- Precision: It is a measure that represents the proportion of correct positive predictions among all positive predictions (Platos, 2025).

$$Precision = \frac{TP}{TP + FP}$$

3- Recall: It represents the proportion of actual positive cases that are correctly predicted as positive (Steurer, 2020).

$$Recall = \frac{TP}{TP+FN}$$

4- F1-score: It balances precision and recall by considering both false positives and false negatives, providing a balanced measure of model performance (Platos, 2025).

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

5- ROC-AUC: This metric evaluates the true-positive and false-positive rates. When combined with the area under the curve (AUC), it measures the ability of the algorithms to correctly distinguish between the classified categories using the following formula (Carrington et al., 2022):

$$AUC = \int_0^1 TPR(FPR)\, d(FPR)$$

$TPR(FPR)$

: True positive rate describes the ROC curve

$d(FPR)$

: The change in the false positive area

$0 - 1$

: The false positive rate ranges from 0-1

Statistical comparison ROC-AUC table:

To ensure the selection of the best algorithm for predicting CLV among the five algorithms, a statistical comparison of ROC-AUC values was conducted. The table contains the mean ROC-AUC, paired t test, and Wilcoxon signed-rank test to assess the significance of differences between the algorithms' performance. If $p < .05$, the difference is considered statistically significant; if $p \geq .05$, the difference is considered statistically insignificant. A value of .05 represents the significance level, which is the maximum acceptable probability of rejecting the null hypothesis when it is true. The following formulas were used:

1- paired t-test:

$$t = \frac{\bar{d}}{\frac{sd}{\sqrt{n}}}$$

sd = standared deviation of differences

d = difference

n = number of folds

2- Wilcoxon signed-rank test:

$$w = \sum Ri$$

w = Wilcoxon statistic

Ri = rank of differences

Reproducibility strategy:

For the reproducibility of the machine learning algorithms, a random seed of 42 was used, with the Python random and NumPy random seeds fixed. The random_state parameter was also fixed for algorithms such as SGDClassifier to control stochastic processes and ensure reproducible results. Fixing the seed helps maintain consistent data splitting and random processes across repeated runs, thereby producing comparable model outputs.

Results and Discussion

Cross-validation results and best hyperparameters:

Table 4 shows the algorithms' performance. The LR algorithm can distinguish between customers 0.7711 of the time, which indicates good classification performance, with a low level of regularization (C = 12.915) that might require more complex patterns to show the model's full performance, and a standard level of regularization (L2) to reduce overfitting, with the (liblinear) parameter being more suitable for small datasets. Capturing simple patterns might cause some bias.

The RF algorithm's prediction performance reaches 0.7741; therefore, it is slightly better at classifying customers than LR, with 200 trees to improve robustness through complex learning and a depth of 6 (depth = 6) as the growth level for not very complex behavioral

patterns, with 2 samples required for a split (min_samples_split = 2) for full growth within the depth limits, and 1 sample per leaf (min_samples_leaf = 1), which allows for precise distribution of customers across classes. The use of log2 as the number of features per split reduces similarities between trees, showing the ability to capture nonlinear patterns.

In the case of the GB algorithm, its performance is the best at discriminating between customers so far, with 0.7749. A low learning rate of 0.05 indicates careful learning to avoid overfitting, which provides stable and accurate predictions, combined with 100 trees and 150 samples per leaf for complex classification, and no depth limit or additional regularization parameters.

The Linear SVM algorithm gives a similar classification performance to LR, with strong regularization, where the model avoids noise that the data may contain through a low C parameter (C = 0.0278) for cleaner classification between customers. This is more evident with the squared-hinge loss function and by avoiding any adjustment for class imbalance through the use of none in class weight, which might lead to some bias in classification. Where the data contain some nonlinear patterns, this model might miss them.

Finally, the NN algorithm indicates a higher classification ability than the linear algorithms, with 0.7722, but does not outperform the more complex algorithms (GB and RF). With a hidden layer of 64 neurons, it can capture nonlinear patterns between customer features with a good level of complexity, with the help of the activation = ReLU parameter, and a stable learning rate of 0.001 for complex patterns, with L2 regularization through the parameter alpha = 0.01 to prevent overfitting.

Table 4. Hyperparameters and Cross-Validation Scores for Each Model

Model		Best_CV_ROC_AUC	Best-Params
Logistic regression	0.771120123	{'model__solver': 'liblinear', 'model__penalty': 'l2', 'model__class_weight': None, 'model__C': np.float64(12.915496650148826)}	
Random Forest	0.774122952	{'model__n_estimators': 200, 'model__min_samples_split': 2, 'model__min_samples_leaf': 1, 'model__max_features': 'log2', 'model__max_depth': 6}	
Gradient boosting	0.774914366	{'model__min_samples_leaf': 20, 'model__max_leaf_nodes': 15, 'model__max_iter': 100, 'model__max_depth': None, 'model__learning_rate': 0.05, 'model__l2_regularization': 0.0}	
Linear SVM	0.771098	{'model__loss': 'squared_hinge', 'model__class_weight': None, 'model__C': np.float64(0.027825594022071243)}	
Neural Network	0.772183929	{'model__learning_rate_init': 0.001, 'model__hidden_layer_sizes': (64,), 'model__alpha': 0.01, 'model__activation': 'relu'}	

Evaluation metrics comparison:

Table 5 reveals the algorithms' holdout performance results while predicting to ensure stability, as the results for each of the five-fold CV are shown in Appendix 2. In terms of accuracy, the five algorithms all achieved similar accuracy levels, ranging from 0.744561 for LR to 0.744673 for NN, where the algorithm shows better classification of nonlinear patterns, followed by the rest of the algorithms with a minor difference that is statistically insignificant, as shown in Figure 2.

In terms of precision, the five algorithms all achieved similar precision levels, with LR (0.680976) and NN (0.681299) outperforming the rest of the algorithms, which achieved 0.680913. This shows that only 68% of the positive predictions are high-CLV customers, while 32% are false low-CLV customers, which is a sign of a bias toward high-CLV behavior, as shown in Figure 3.

In terms of recall, the five algorithms all achieved a high recall score, ranging from 0.919826 for NN as the lowest recall to 0.921056 for the rest, aside from LR, which had the second-lowest recall of 0.920608. This shows how the algorithms are biased toward customers with high customer lifetime value, as shown in Figure 4.

In terms of F1, all the algorithms showed similar scores, ranging from 0.782795 to 0.782985, where the difference is statistically insignificant. This shows how balanced the algorithms are but still influenced by recall, indicating how the dataset is affecting the performance of the algorithms, even with complex algorithms that deal with nonlinear patterns in the data, as shown in Figure 5.

In terms of ROC-AUC, the algorithms achieved very close scores with a small difference, but still larger than the previous metrics, ranging from 0.774262 to 0.776018, with Linear SVM as the lowest and NN as the highest. This reveals a moderate classification ability, and even if the results are close to a certain degree for the other metrics, according to the ROC-AUC metric, NN is better, as shown in Figure 6. This contradicts the results in Table 4, which indicate that GB is the best algorithm based on the ROC-AUC metric.

Table 5. Holdout Results for the Five Algorithms' Evaluation Metrics

Model	accuracy	Precision	Recall	F1-score	ROC-AUC
Linear SVM	0.744617	0.680913	0.921056	0.782985	0.774262
Logistic Regression	0.744561	0.680976	0.920608	0.782865	0.774326
Random Forest	0.744617	0.680913	0.921056	0.782985	0.775073
Gradient Boosting	0.744617	0.680913	0.921056	0.782985	0.775615
Neural Network	0.744673	0.681299	0.919826	0.782795	0.776018

The results above are shown in the following figures:

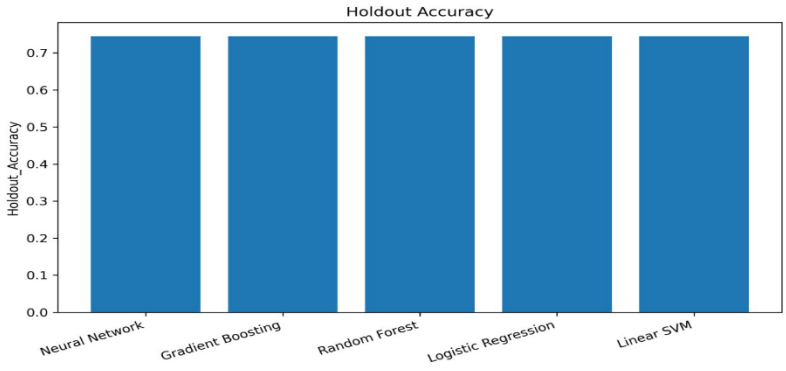


Figure 2. Evaluating algorithms' holdout performance with the accuracy metric

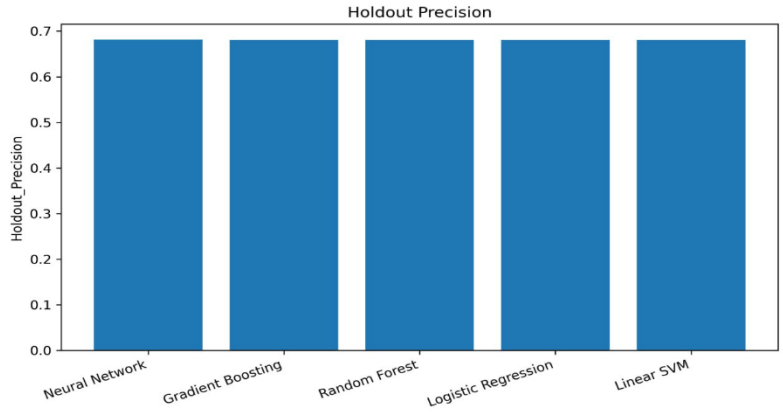


Figure 3. Evaluating algorithms' holdout performance with the precision metric

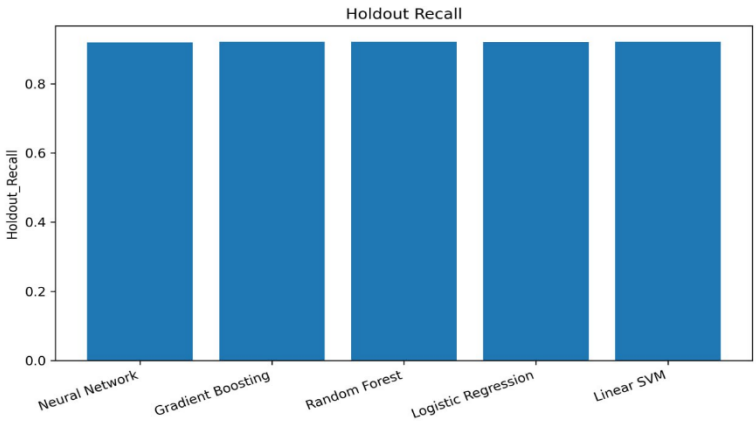


Figure 4. Evaluating algorithms' holdout performance with the recall metric

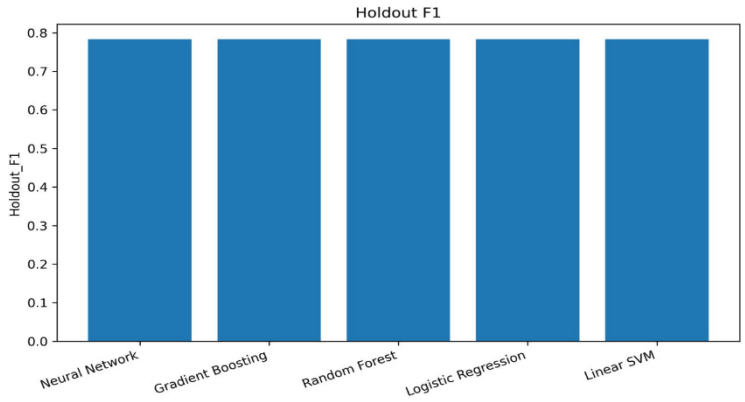


Figure 5. Evaluating algorithms' holdout performance with the F1 metric

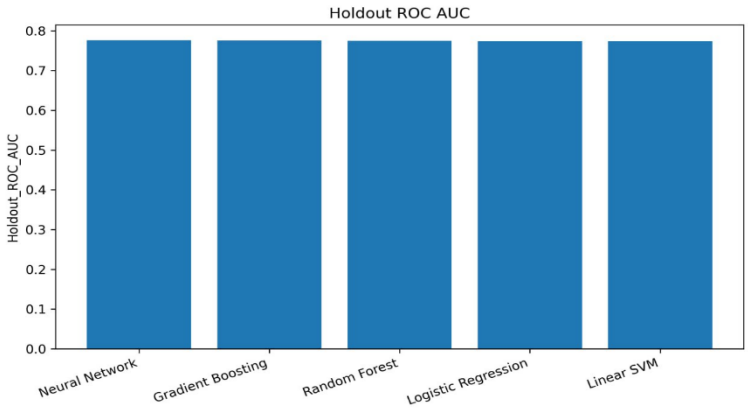


Figure 6. Evaluating algorithms' holdout performance with the ROC-AUC metric

Figure 7 shows the algorithm performance represented in heat maps, which reflect the evaluation metrics' holdout results through the use of the confusion matrix, as shown in Table 3. The results reveal how the algorithms are recall-centered, with a bias toward predicting high-CLV customers in cases where valuable observations may be missed. Figure 8 presents the ROC-AUC results, showing that the algorithms' classification performance is at a high level but is still affected by the patterns in the dataset.

https://jitm.ut.ac.ir/

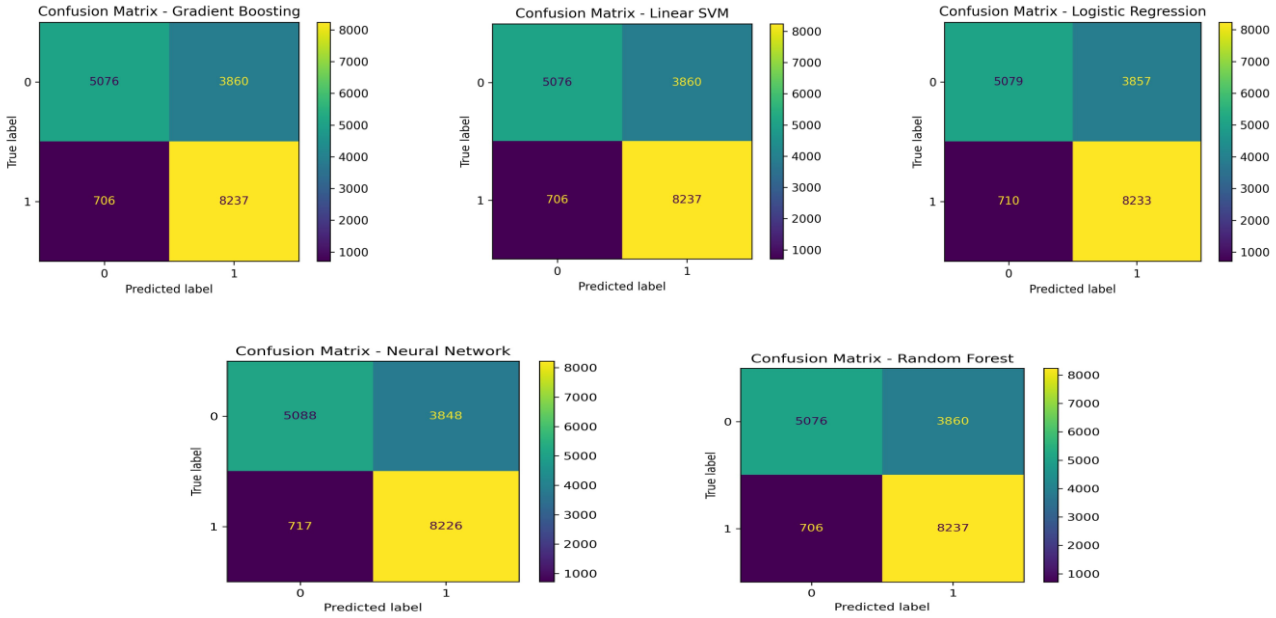


Figure 7. Algorithm holdout performance represented by heat maps

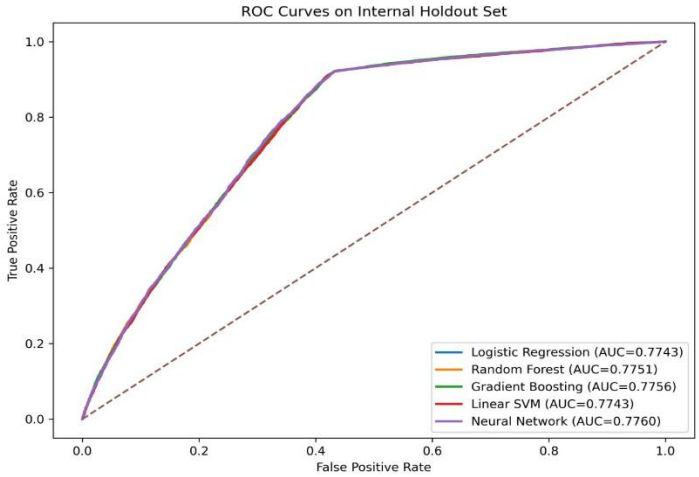


Figure 8. ROC-AUC results in evaluating the algorithms' performance

Paired t-test and Wilcoxon signed-rank test comparison Table:

According to Table 6, these statistical tests were used to identify whether the differences between the algorithms were statistically significant and whether they were randomly distributed. These tests served as a deciding factor in selecting the best algorithm for predicting CLV. The following explanations are provided:

- **Row 1 (LR vs. RF):** According to the *t-test* value (0.00104) and Wilcoxon test value (0.00585), the difference is statistically significant, making RF the better algorithm.
- **Row 2 (LR vs. GB):** According to the *t-test* value (0.000011) and Wilcoxon test value (0.0019), the difference is statistically significant, making GB the better algorithm.
- **Row 3 (LR vs. Linear SVM):** According to the *t-test* value (0.239) and Wilcoxon test value (0.1054), the difference is statistically insignificant.
- **Row 4 (LR vs. NN):** According to the *t-test* value (0.14928) and Wilcoxon test value (0.16015), the difference is statistically insignificant.
- **Row 5 (RF vs. GB):** According to the *t-test* value (0.0179) and Wilcoxon test value (0.0371), the difference is statistically significant, making GB the better algorithm.
- **Row 6 (RF vs. Linear SVM):** According to the *t-test* value (0.0010) and Wilcoxon test value (0.0058), the difference is statistically significant, making RF the better algorithm.
- **Row 7 (RF vs. NN):** According to the *t-test* value (0.0107) and Wilcoxon test value (0.0136), the difference is statistically significant, making RF the better algorithm.
- **Row 8 (GB vs. Linear SVM):** According to the *t-test* value (0.000011) and Wilcoxon test value (0.00195), the difference is statistically significant, making GB the better algorithm.
- **Row 9 (GB vs. NN):** According to the *t* test value (0.00046) and Wilcoxon test value (0.00195), the difference is statistically significant, making GB the better algorithm.
- **Row 10 (Linear SVM vs. NN):** According to the *t-test* value (0.1417) and Wilcoxon test value (0.1601), the difference is statistically insignificant.

Table 6. Paired t-test and Wilcoxon Signed-Rank Test Comparison

Model_A	Model_B	Mean_ROC_AUC_A	Mean_ROC_AUC_B	Paired_t_stat	Paired_t_pvalue	Wilcoxon_stat	Wilcoxon_pvalue
Logistic Regression	Random Forest	0.771120123	0.774122952	-4.750565849	0.001043364	2	0.005859375
Logistic Regression	Gradient Boosting	0.771120123	0.774914366	-8.709959574	1.12E-05	0	0.001953125
Logistic Regression	Linear SVM	0.771120123	0.771097543	1.258476535	0.239884472	11	0.10546875

Logistic Regression	Neural Network	0.771120123	0.772183929	-1.576868975	0.149280398	13	0.16015625
Random Forest	Gradient Boosting	0.774122952	0.774914366	-2.887517811	0.017956727	7	0.037109375
Random Forest	Linear SVM	0.774122952	0.771097543	4.745106542	0.001051377	2	0.005859375
Random Forest	Neural Network	0.774122952	0.772183929	3.205755294	0.010731744	4	0.013671875
Gradient Boosting	Linear SVM	0.774914366	0.771097543	8.689709981	1.14E-05	0	0.001953125
Gradient Boosting	Neural Network	0.774914366	0.772183929	5.339890151	0.000468541	0	0.001953125

Thus; according to the table, the results showed the gradient boosting algorithm which had achieved statistically the highest mean ROC-AUC among the evaluated algorithms ending the contradiction between GB and NN in choosing the mean ROC-AUC as the final factor, while NN achieved the highest holdout ROC-AUC, GB had consistently outperformed NN performance based on the statistical comparison for the CV fold.

Customer CLV predictions:

Predicting Customer Lifetime Value Using the Best Model: According to the previous results in Tables 4 and 6, the Gradient Boosting algorithm is the best-performing model. Based on the results presented in Figure 9, a high level of bias toward high-CLV behavior can be observed because of the nonlinear and complex patterns in the dataset. This high recall level might be the result of consistent and stable predictions across the cross-validation folds.

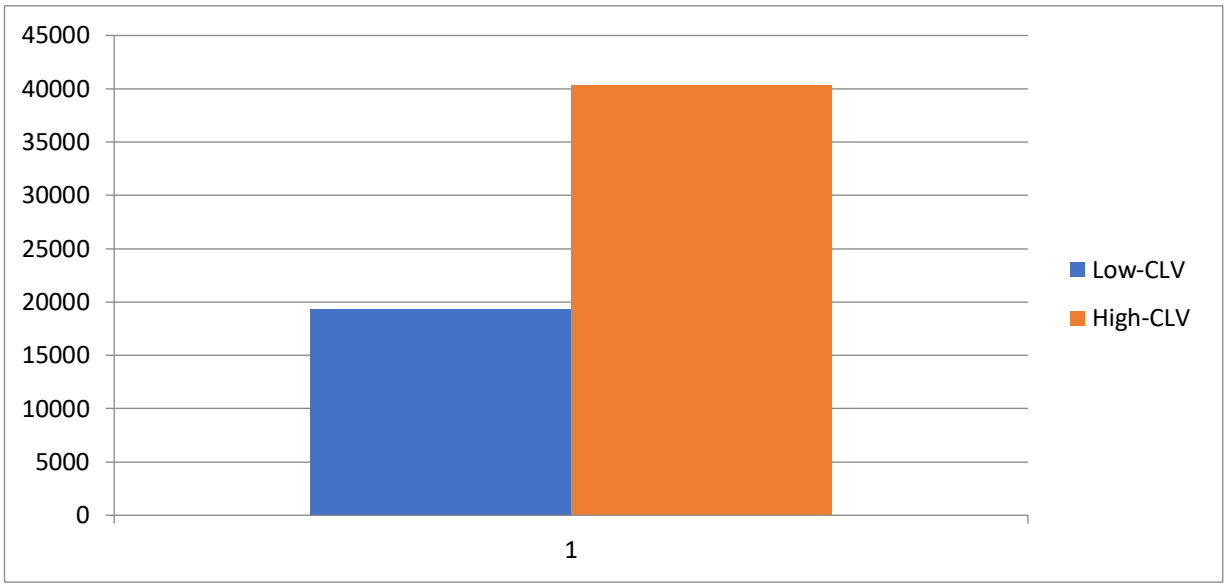


Figure 9. CLV prediction result using the Gradient Boosting algorithm

Conclusion

This study shows that when using ML algorithms with Cross-validation method to increase the algorithms regularization and generalization when dealing with new unseen data, the algorithms used proved to over-predict positives and this recall-dominant behavior is to not miss on some important characteristics that may leave some high CLV customers undetected by the algorithm, and having a high level of error proves that some of the predicted outcomes don't necessarily represent actual cases, and with the algorithms achieving stable predictions on a consisting pace over the folds with repeated sampling conditions, which means that the company is actually filled with high CLV cases, with the algorithms prioritizing sensitivity towards customers with high CLV as shown with the high recall level.

Then, when comparing the algorithms against each other; a pattern is noticed of a high level of recall causing bias towards high CLV customers, and this recall-dominating behavior sets the opportunity for future works involving threshold improvement, and relying on multiple comparison methods such as the evaluation metrics and the paired T test and Wilcoxon signed rank test, we got the final result that shows the gradient boosting algorithm being the one with the most accurate discriminated classes, means that the institutions can support their retention campaign with the use of Gradient Boosting algorithm to help prioritize high CLV customers and optimize marketing strategies.

Conflicts of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

The author's received no financial support for the research, authorship, and/or publication of this article.

References

Abidar, L., Zaidouni, D., El Asri, I., & Ennouaary, A. (2023). Predicting customer segment changes to enhance customer retention: A case study for online retail using machine learning. *International Journal of Advanced Computer Science and Applications (IJACSA), 14*(7).

Akter, J., Roy, A., Rahman, S., Mohona, S., & Ara, J. (2025). Artificial intelligence-driven customer lifetime value (CLV) forecasting: Integrating RFM analysis with machine learning for strategic customer retention. *Journal of Computer Science and Technology Studies, 7*(1).

Ali, A. A. (2026). *Mastering Python from fundamentals to advanced applications* (1st ed.). Al-Nahrain University.

Ali, A., & Mashwani, W. K. (2023). Supervised machine learning algorithms: Applications, challenges, and recommendations. *Proceedings of the Pakistan Academy of Sciences: Physical and Computational Sciences*, *60*(4).

Ali, N., & Shabn, O. S. (2024). Customer lifetime value (CLV) insights for strategic marketing success and its impact on organizational financial performance. *Cogent Business & Management, 11*(1), 2361321. https://doi.org/10.1080/23311975.2024.2361321

Alnuaimi, A. F. A. H., & Albaldawi, T. H. K. (2024). An overview of machine learning classification techniques. *BIO Web of Conferences, 97*, 00133. https://doi.org/10.1051/bioconf/20249700133

Blattberg, R. C., Malthouse, E. C., & Neslin, S. A. (2009). Customer lifetime value: Empirical generalizations and some conceptual questions. *Journal of Interactive Marketing, 23*, 157–168.

Carrington, A. M., Manuel, D. G., Fieguth, P., et al. (2022). Deep ROC analysis and AUC as balanced average accuracy, for improved classifier selection, audit and explanation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 45*(1).

Gil, C. (2020). *The end of marketing: Humanizing your brand in the age of social media and AI*. Kogan Page.

Guolin, K., Meng, Q., Finley, T., Wang, T., Chen, W., et al. (2017). *LightGBM: A highly efficient gradient boosting decision tree*. Proceedings of the 31st International Conference on Neural Information Processing Systems.

Haddadi, A. M., & Hamidi, H. (2025). A hybrid model for improving customer lifetime value prediction using stacking ensemble learning algorithm. *Computers in Human Behavior Reports, 18*, 100616.

Ibrahim, A. A., & Kaml, B. S. (2024). Customer behavior prediction using machine learning with fuzzy multiple linear regression. *Journal of Logistics, Informatics and Service Science, 11*(10).

Insurance Authority. (2016). *Motor vehicle insurance policy against loss and damage*. Central Bank of the United Arab Emirates.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., et al. (2017). *LightGBM: A highly efficient gradient boosting decision tree*. Proceedings of the 31st International Conference on Neural Information Processing Systems.

Lumumba, V. W., Kiprotich, D., Mpaine, M. L., Makena, N. G., & Kavita, M. D. (2024). Comparative analysis of cross-validation techniques: LOOCV, K-fold cross-validation, and repeated K-fold

cross-validation in machine learning models. *American Journal of Theoretical and Applied Statistics, 13*(5).

Mabel, E., Carter, E., & Hayes, M. (2025). *Using RFM analysis to predict high-value customers*. ResearchGate. Retrieved July 22, 2025, from https://www.researchgate.net/publication/392017535

Miller, C., Portlock, T., Nyaga, D. M., & O'Sullivan, J. M. (2024). A review of model evaluation metrics for machine learning in genetics and genomics. *Frontiers in Bioinformatics, 4*, 1457619. https://doi.org/10.3389/fbinf.2024.1457619

Natekin, A., & Knoll, A. (2013). Gradient boosting machines: A tutorial. *Frontiers in Neurorobotics, 7*, 21. https://doi.org/10.3389/fnbot.2013.00021

Obi, J. C. (2023). A comparative study of several classification metrics and their performances on data. *World Journal of Advanced Engineering Technology and Sciences, 8*(1).

Osisanwo, F. Y., Akinsola, J. E. T., Awodele, O., Hinmikaiye, J. O., Olakanmi, O., & Akinjobi, J. (2017). Supervised machine learning algorithms: Classification and comparison. *International Journal of Computer Trends and Technology, 48*(3), 128–138.

Platos, J. (2025). *Fundamentals of machine learning evaluation metrics for machine learning*. Department of Computer Science, Faculty of Electrical Engineering and Computer Science, VSB–Technical University of Ostrava.

Shanbharkar, A., Rakhade, V. M., & Yadav, L. N. (2024). Neural network. *International Journal of Futuristic Innovation in Arts, Humanities and Management, 3*.

Steurer, M., Hill, R. J., & Pfeifer, N. (2020). Metrics for evaluating the performance of machine learning-based automated valuation models. Paper presented at the 36th IARIW Virtual General Conference, University of Graz.

Tang, Y.-C., & Li, K.-H. (2023). A machine-learning approach to setting optimal thresholds and its application in rolling bearing fault diagnosis. *Machine Learning: Science and Technology*. https://doi.org/10.1088/2632-2153/ad0ab3

Vujović, Ž. Đ. (2021). Classification model evaluation metrics. *International Journal of Advanced Computer Science and Applications (IJACSA), 12*(6).

Appendix (1)

Data details:

1- Numerical data such as

- marital_status-Marital Status of the customer {0:Single, 1: Married}
- Vintage-No. of years since the first policy date
- claim_amount-Total Amount Claimed by the customer (in rupees)

2- Categorical data such as:

- gender-Gender of the customer
- area-Area of the customer
- qualification-Highest Qualification of the customer
- type_of_policy-Type of active policy
- policy-Active policy of the customer

3- Mixed data such as:

- income-Income earned in a year (in rupees)
- num_policies-Total no. of policies issued by the customer

Appendix 2

1- Gradient boosting fold-scores:

fold_accur	fold_preci	fold_recall	fold_f1	fold_roc_auc
0.732853	0.671999	0.91012	0.77314	0.76643
0.742781	0.681837	0.910832	0.779872	0.777863
0.738796	0.676001	0.91768	0.778516	0.776031
0.741015	0.6798	0.911658	0.778839	0.77626
0.744721	0.682886	0.914174	0.781782	0.778703
0.742641	0.681055	0.913056	0.780173	0.778451
0.742432	0.68065	0.913906	0.780217	0.777224
0.741802	0.678564	0.919357	0.780818	0.777485
0.738428	0.677409	0.910819	0.776963	0.773824
0.734443	0.67432	0.907325	0.773659	0.766873

2- Random forest fold-scores:

fold_accur	fold_preci	fold_recall	fold_f1	fold_roc_auc
0.733063	0.671959	0.911099	0.773466	0.765912
0.742781	0.681837	0.910832	0.779872	0.776342
0.738796	0.675784	0.918519	0.778673	0.774007
0.741015	0.679725	0.911937	0.778892	0.775309
0.744861	0.682838	0.914873	0.782006	0.779128
0.742641	0.680829	0.913894	0.780331	0.776752
0.742921	0.680807	0.915164	0.780779	0.777087
0.741802	0.678564	0.919357	0.780818	0.777503
0.738428	0.677409	0.910819	0.776963	0.772289
0.734722	0.674419	0.908023	0.773978	0.766899

3- Logistic regression fold-scores:

fold_accur	fold_preci	fold_recall	fold_f1	fold_roc_auc
0.733133	0.672206	0.9104	0.773378	0.762511
0.742641	0.681885	0.910133	0.779647	0.771917
0.738586	0.67572	0.917959	0.77843	0.774272
0.741225	0.679975	0.911798	0.779005	0.772537
0.744861	0.683029	0.914174	0.781876	0.774692
0.742641	0.68098	0.913335	0.780226	0.776729
0.742991	0.680991	0.914745	0.780747	0.772342
0.741872	0.678671	0.919217	0.780838	0.772672
0.738218	0.677382	0.91012	0.776691	0.771143
0.734932	0.674847	0.907185	0.773955	0.762385

4- Neural network fold-scores:

fold_accur	fold_preci	fold_recall	fold_f1	fold_roc_auc
0.733133	0.672669	0.908583	0.773027	0.761451
0.741802	0.683602	0.900769	0.777302	0.77665
0.737398	0.677382	0.907058	0.775574	0.773396
0.740456	0.681962	0.901594	0.776547	0.77066
0.744441	0.68352	0.910819	0.780967	0.776218
0.741173	0.682762	0.901314	0.776961	0.777887
0.742292	0.680734	0.913068	0.779967	0.774021
0.741243	0.679187	0.914885	0.779611	0.775235
0.738288	0.677416	0.91026	0.776764	0.770603
0.734932	0.676647	0.900335	0.772626	0.765719

5- Linear SVM fold-scores:

fold_accur	fold_preci	fold_recall	fold_f1	fold_roc_auc
0.733063	0.671959	0.911099	0.773466	0.762496
0.742781	0.681837	0.910832	0.779872	0.77193
0.738796	0.675784	0.918519	0.778673	0.774258
0.741015	0.679725	0.911937	0.778892	0.772452
0.744861	0.682838	0.914873	0.782006	0.774662
0.742641	0.680829	0.913894	0.780331	0.776709
0.742921	0.680807	0.915164	0.780779	0.77234
0.741802	0.678564	0.919357	0.780818	0.772531
0.738428	0.677409	0.910819	0.776963	0.771213
0.734722	0.674419	0.908023	0.773978	0.762384

Bibliographic information of this paper for citing:

Al-Hashimy, Zahra Muhammed; Ibrahim, Ali & Musharraf Eidan, Feryal (2026). Predicting Customer Lifetime Value Using Machine Learning Algorithms with Cross-Validation. *Journal of Information Technology Management*, 18 (Special Issue), 1-22.
https://doi.org/10.22059/jitm.2026.109971

Copyright © 2026, Zahra Muhammed Al-Hashimy, Ali Ibrahim and Feryal Musharraf Eidan