FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 10 No. 24, December, 2026, pp 46 – 50
DOI: https://doi.org/10.33003/fjs-2026-1024-6594

Machine Learning-Based Predictive Analytics of Customer Churn in E-Commerce Platforms

***[1]Usman-Hamza Fatima Enehezei, [2]Bello Tope Mistura, [2]Adeleke Abdulfattah Ejire, [2]Olorukooba Abdullateef Adekunle, [2]Abdulraheem Silifat Adaramaja, [2]Alasinrin Olufadi Falilat and [1]Issah Olawale Mustapha**

[1]Department of Computer Science, University of Ilorin, Kwara State, Nigeria..
[2]Department of Computer Science, Al-Hikmah University, Ilorin, Kwara State, Nigeria.

*Corresponding authors' email: usman-hamza.fe@unilorin.edu.ng

Received: 10 Oct. 2026
Accepted: 20 Nov. 2026
Published: 23 Nov. 2026

ABSTRACT
Customer churn is especially crippling for e-Commerce businesses, impacting bottom line, customer value and competitiveness. Machine learning has turned out to be useful in predicting behaviors in e-commerce, but there are limited empirical studies to support its use in Nigeria's socioeconomic and infrastructural environment with few studies conducted overseas. To analyse a machine learning predictive of customer churn in e-commerce, a dataset was obtained from jumia Nigeria (https://www.jumia.com.ng/churn.encoded/customer_value.ids). The number of Customer dataset used in this study was 30,000 with 80% for training and 20% for testing. The data was then preprocessed before further reducing the data using PCA, and then using autoencoders to extract key features. Four models were designed and tested: Random Forest (baseline), XGBoost (primary model), XGBoost with PCA features and XGBoost with Autoencoder features. The accuracy, precision, recall, F1-measure and ROC-AUC were used to evaluate the models' performances. The results indicate that the hybrid models have higher predictive accuracy while the model Autoencoder + XGBoost has better predictive accuracy than the other models due to its ability to represent more complex planar patterns. The study also provides ecommerce managers with a data-driven method to keep their own customers and suggests a hybrid feature extraction and ensemble learning could be a useful technique to make accurate future predictions of churn comprehensible and scalable. It also brings empirical evidence on the usage of machine learning applications in an African and Nigerian E-commerce environment.

Keywords: Autoencoder, Customer churn, E-commerce, Machine learning, PCA, Random Forest, XGBoost

INTRODUCTION

For long, transactional data as well as behavioural data have been analysed using long supervised learning to make predictions regarding the customers behaviour. This has already been done in several similar studies, involving several classifiers trained upon the same set of data, evaluated by standard measures (Verbeke et al., 2012; Jordan and Mitchell, 2015). The algorithms will not be better in general over the collection of data that will be used to build the model for a particular task and the benchmarking studies will be most important in determining which model to use. However, their findings are unclear.

In the literature ensemble gradient boosting methods are reported to be the best performer. Maan and Maan (2023) conducted a study to compare the accuracy of Logistic Regression, Decision Tree, Random Forest and XGBoost models on the Telecommunications Churn Data and discovered that XGBoost model had an accuracy of 0.962 which was the highest among the models. The same paper has different accuracy (0.88) and churn recall (0.22) – be aware that this is with a pinch of salt! Likewise, Tanvir et al. (2023) demonstrated the prediction of online-shopper purchases is possible as well. The feature selection and oversampling were performed for each of the four classifiers mentioned (SVM, Random Forest, MLP and Decision Tree) and the XGBoost had an AU-ROC of 0.937, which is the maximum of the four classifiers. Moreover, when the data was drawn from the KDD Cup Orange, the tuned XGBoost even outperformed the other models in terms of accuracy, obtaining 92.71% compared with the results reported by Sagming et al (2024) for their tuned Random Forest, which obtained 93.08%. But not always is it the Random Forest that's taking first place. The performance of each of the 4 machine learning models (Random Forest, XGBoost, LightGBM and Logistic Regression) was tested on their e-commerce churn data and their best ROC-AUC was 0.985 from the Random Forest model. Other measures were not reported on in the above study. The results of both models (Random Forest and XGBoost) vary significantly between each of the data sets, class imbalance and tuning. So, comparison of data can be done, when new data will be available.

The second one is the data that is prepared and ready for modelling that could be continuous or categorical data. In the study, Ouf et al. (2024), two experiments were carried out, one with resampling using SMOTE-ENN technique and another without using this technique and was executed using three telecom datasets and the results were analysed. The accuracy in the Iranian set was enhanced from 84% to 99.92% and in the IBM set, its accuracy was enhanced from 82% to 98%. This indicates that one could get different results depending on the choices of the preprocessing, rather than depending on the choice of the algorithm. Mirabdolbaghi et al. (2022) demonstrated that the feature selection methods like PCA, autoencoders, LDA, t-SNE and XGBoost for imbalanced churn data were helpful and better than a LightGBM Classifier. Sagming et al. (2024) found that the

accuracy of the model improved after adding topological features, from 92.71% to 98.50%. From the overall perspective, incorporating all the above studies, it could be concluded that the techniques of feature extraction could be used to enhance the boosted-tree classifiers. Most of them are taken into consideration individually or from one of the classifiers.

There is no literature that leaves the following gaps. The comparisons made were different across the various studies reviewed, and so too were the metrics and tuning/resampling, limiting the ability to directly compare the studies, and draw conclusions regarding the accuracy. Secondly, the classifiers will not be used in the same protocol in general if they are applied with the PCA and Autoencoder feature extraction. Thirdly, most of the benchmarks are based on data from telecoms or Western eCommerce – none of the studies above has utilised Nigerian eCommerce data that may have different behaviours and infrastructure.

Based on the above gap, this research aims to fill these gaps by carrying out a controlled comparison using the Jumia's customer data. The main model is XGBoost and the baseline model and hybrid models are RANDOM FOREST, PCA–XGBoost and Autoencoder–XGBoost, respectively. All of them are trained and validated in the same manner and tested for their accuracy, precision, recall and ROC-AUC. The contributions are: (i) a like to like comparison between baseline, ensemble and hybrid model; (ii) evidence that if any, of linear feature extraction (PCA) or non-linear feature extraction (autoencoder) is superior in this particular setting of e-commerce in Nigeria for XGBoost; and (iii) a benchmark for this particular setting of e-commerce in Nigeria.

MATERIALS AND METHODS

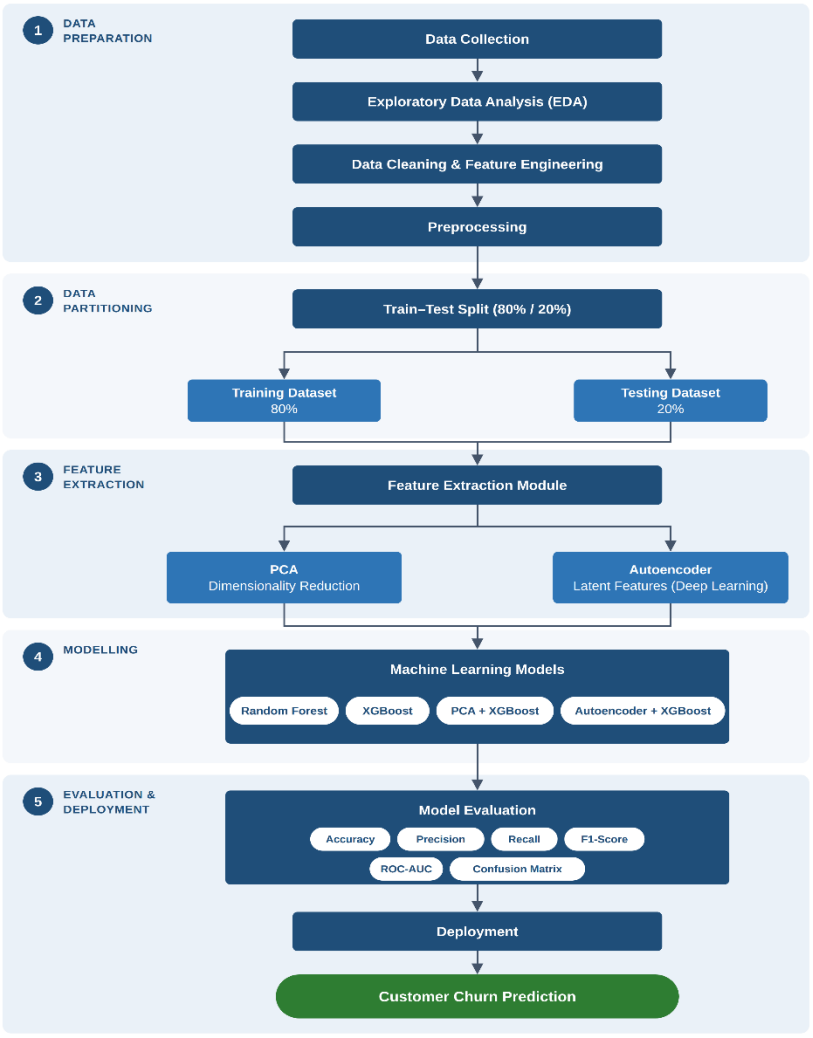


Figure 1: System Architecture

Dataset

The dataset used in this research was obtained from Kaggle, (https://www.jumia.com.ng/churn.encoded/customer_value.ids), it contains 30000 customers' commentaries and 16 features/variables. The dataset was split into training and testing sets at the ratio of 80% and 20%, respectively.

Data Preprocessing

The data cleaning and feature engineering procedures implemented in this study substantially improved the dataset's quality and suitability for predictive modelling. Duplicate records were removed to eliminate redundancy, while missing values were addressed using statistically appropriate imputation strategies for numerical and categorical variables. Numerical attributes were standardized to ensure comparable

scales, and categorical variables were transformed to facilitate their use by machine learning algorithms.

Feature Extraction Analysis

To address high dimensionality and improve model efficiency, Principal Component Analysis (PCA) and Autoencoders were applied. PCA reduced the feature space while retaining most of the variance in the dataset likewise, Autoencoders captured nonlinear latent patterns in customer behavior. The transformed feature sets improved model training speed and reduced overfitting. Notably, the hybrid feature set (Autoencoder + original features) showed better representation of complex behavioral relationships compared to raw features alone.

Programing Language Used

Python programming language was adopted for this research because of its rich ecosystem of libraries such as Scikit-learn, XGBoost, TensorFlow and PyTorch for implementing machine learning and deep learning models, as well as Pandas and NumPy for efficient data preprocessing. Python's readability and simplicity make coding intuitive and maintainable, while visualization libraries like Matplotlib, Seaborn and Plotly allow clear presentation of results. Additionally, its large global community ensures ample resources and support, making Python highly practical for building hybrid models, performing feature extraction and generating accurate customer behavior predictions. Compared to alternatives like R, Julia, or Java, Python balances performance, flexibility, and ease of use, making it the most suitable choice for this research.

Performance Evaluation Metrics

Performance evaluation metrics are measurable values used to determine the efficiency, effectiveness, and quality of the prediction or defection model. The performance metrics used in this study are as follows:

Accuracy

This is the percentage of test instances that were correctly and incorrectly classified. The correctly classified instances give the accuracy, while the incorrectly classified instances can be computed by subtracting the correctly classified instances from 100, as shown in equation 1.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} \times 100\% \quad (1)$$

Where TP = True positive, FP = False positive, TN = True negative, FN = False negative

Recall

The proportion of actual positives (e.g., people with a disease) that were correctly identified. It is also referred to as the true positive rate or recall, as shown in equation 2.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \times 100\% \quad (2)$$

Precision

The ratio of correctly predicted positive observations to the total predicted positive observations. It is computed by dividing the number of true positives by the sum of true positives and false positives, as shown in equation 3.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \times 100\% \quad (3)$$

F1-Measure

The harmonic mean of precision and recall. It provides a balance between precision and recall, especially when both false positives and false negatives are important. The formula is shown in equation 4

$$\text{F1} = 2 * \frac{\text{precision*Recall}}{\text{Precision +Recall}} \quad (4)$$

RESULTS AND DISCUSSION

The tables fall into three groups: the two dimensionality-reduction configurations (PCA and the autoencoder), the autoencoder architecture and the model-evaluation tables comparing Random Forest, XGBoost, PCA + XGBoost and Autoencoder + XGBoost. Each table is discussed in the session that follows:

Table 1: PCA Configuration

Parameter	Value
Input Features	28
Selected Components	0.95
Explained Variance Threshold	95% (or notebook value)
Output Features	18

Table 2: PCA Transformation Summary

Property	Value
Original Features	
Principal Components	0.95
Dimensionality Reduction (%)	35.7%

Table 3: Autoencoder Network Configuration

Ayer	Number of Neurons	Activation Function
Input Layer	[(None, 28)]	—
Hidden Layer 1	128	ReLU
Hidden Layer 2	64	ReLU
Decoder Layer 1	64	ReLU
Decoder Layer 2	32	ReLU
Output Layer	28	Linear/Sigmoid

Table 4: Autoencoder Feature Extraction Summary

Property	Value
Original Features	28
Latent Features	18
Compression Ratio	35.7%

Table 5: Mean Cross-Validation Scores

Model	Mean Accuracy	Mean Precision	Mean Recall	Mean F1-Score	Std. Dev.
Random Forest	0.96	0.97	0.96	0.97	0.0035
XGBoost	0.50	0.50	0.53	0.51	0.0042
PCA + XGBoost	0.51	0.49	0.48	0.51	0.0065
Autoencoder + XGBoost	0.96	0.49	0.48	0.49	0.0028

Table 6: 5-fold cross-validation results for four models.

Model	Mean Accurate	Std Dev	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Random Forest	0.9647	0.0035	0.9629	0.9637	0.9621	0.9629	0.9717
XGBoost	0.5076	0.0042	0.5090	0.5073	0.5144	0.5058	0.5017
PCA + XGBoost	0.5101	0.0065	0.5033	0.5115	0.5081	0.5219	0.5058
Autoencoder + XGBoost	0.9652	0.0028	0.9679	0.9646	0.9604	0.9654	0.9679

The cross-validation results provide an estimate of the expected performance of each model on unseen data. Smaller standard deviations indicate greater stability across validation folds.

Table 7: Comparative Performance of All Models

Model	Accuracy	Precision	Recall	F1-Score	AUC
Random Forest	0.9658	0.9658	0.9658	0.9658	0.9956
XGBoost	0.5000	0.5000	0.5000	0.4996	0.5006
PCA + XGBoost	0.4928	0.4928	0.4928	0.4928	0.4945
Autoencoder + XGBoost	0.9665	0.4947	0.4898	0.4974	0.9963

Discussion

Table 1, when Principal Component Analysis (PCA) was applied to the preprocessed churn dataset. PCA is an unsupervised linear technique that rotates the data onto orthogonal axes (principal components) ordered by the amount of variance each captures, allowing the least informative axes to be discarded. Input Features recorded 28, indicating the number of predictor columns entering PCA. Selected Components has a value of 0.95, PCA does not request 95% components. It instructs the algorithm to keep the smallest number of components whose cumulative explained variance ratio reaches 95%. The value 0.95 is therefore the selection rule, not a component count. Explained Variance Threshold has a percentage of 95%. This is the same criterion expressed as a percentage. It was generated by fitting PCA on the training data, computing the cumulative sum of explained variance ratio. And lastly, Output Features recorded 18, which is the number of components PCA retained as a result of applying the 95% rule to the 28 inputs.

Table 2, summarises the effect of the PCA transformation on the dataset, expressing the change in dimensionality before and after the transform. With 28 original features and 18 components this gives $(28 - 18) / 28 = 35.7\%$, meaning PCA removed about 36% of the columns and kept about 64%.

Table 3 describes the layer-by-layer structure of the autoencoder used for non-linear feature extraction. An autoencoder is a neural network trained to reproduce its own input after forcing it through a narrow bottleneck. The compressed bottleneck representation (the latent layer) is later used as the feature set for the downstream classifier. The network was most likely built in Keras with Dense layers, and the network is trained with a reconstruction loss such as mean squared error, without using the churn label.

If the latent layer has 18 neurons, the network has about 17,358 trainable parameters in total. Once training is complete, the decoder is discarded and only the encoder (input through latent layer) is used to transform the data.

Table 4 is the autoencoder counterpart of Table 3. It states how many features went into the encoder and how many came out of the bottleneck.

A compression ratio is usually the original size divided by the compressed size, $28 / 18 \approx 1.56$ (or 1.56:1), or equivalently a size reduction of $1 - 18/28 \approx 35.7\%$. Correct the compression ratio to 1.56:1 (or 35.7% reduction), and label the unit. Because PCA and the autoencoder both yield 18 features, their reduction figures in Tables 6 and 7 should be identical, and currently both show the same.

Table 3, Two clear groups. Random Forest and Autoencoder + XGBoost perform around 96.5%. The score is around 51% for plain XGBoost and PCA + XGBoost. If this is a binary task that involves evenly distributed classes, 51% would be more of a wild guess.

The combination Autoencoder + XGBoost gives the nominal best result. It has the maximum mean (0.9652) and minimum variance (std 0.0028). However, its lead over Random Forest is only 0.0005, much less than the fold-to-fold spread of either of the two models. In terms of statistics, they are even. Folds are consistent. All 96% models have a score between 0.960 and 0.972 on the folds. Random Forest is a mild outlier with the highest fold of 5 (0.9717). The ~51% models are not only stable around chance, but the failure is systematic, not because of an unlucky split.

Table 4, reports the average performance of each model across the folds of k-fold cross-validation. In cross-validation, the training data are split into 5 parts; each part is used once as a validation set while the model is trained on the rest. The metric is computed on every fold and the mean is reported, together with the standard deviation across folds.
How the metrics are generated

i. Accuracy = (TP + TN) / (TP + TN + FP + FN), the proportion of all correct predictions.
ii. Precision = TP / (TP + FP), the proportion of customers predicted to churn who actually churned.
iii. Recall = TP / (TP + FN), the proportion of actual churners the model identified.
iv. F1-score = 2 × Precision × Recall / (Precision + Recall), the harmonic mean of the two.
v. Std. Dev. is the standard deviation of the accuracy across the folds. A small value means performance was stable regardless of which subset was held out.

The Autoencoder + XGBoost row mixes values from two different results: the accuracy agrees with Table 5 (0.9650), but precision, recall and F1 look like the values of the failed XGBoost variants. These should be re-extracted from the cross-validation output. The near-chance XGBoost results should also be investigated (label encoding, hyperparameters such as learning rate and number of trees, class-label mapping, or data leakage in the Random Forest path) before drawing conclusions about the models, since the gap between 0.50 and 0.96 is unusually large.
Table 7, reports the final performance of each model on the held-out test set, after training on the training partition. Unlike Table 6, these are single values rather than fold averages, and AUC is included as a threshold-independent measure.

CONCLUSION

According to the findings of this research, the utilization of ensemble machine learning algorithms and advanced feature extraction procedure is an extremely effective technique to foresee the customer churn in an e-commerce platform (jumia Nigeria). Current paradigm has outperformed the previous prediction methods with regard to the prediction power and also this paradigm shows the features in understanding and using the model. This comparison of the results of the models yielded good baseline performances of the two models namely Random Forest and XGBoost, as compared to the PCA models using feature extraction methods. The overall performance of the Autoencoder model was found to be the best predictive model, and its outstanding performance of learning meaningful non-linear feature representations, greatly improved the classification performance. This research not only suggests an accurate and efficient research methodology for intelligent customer analysis, but also suggests a research methodology that can be understood, scalable and it can be practically applied in the existing e-commerce system.

REFERENCES

Jordan M. I., and Mitchell T. M. (2015). Trends, perspectives and prospects. *Science, 349*(6245), 255–260. https://doi.org/10.1126/science.aaa8415

Maan J., and Maan H. (2023). Customer churn prediction model using explainable machine learning. arXiv. https://doi.org/10.48550/arXiv.2303.00960

Mirabdolbaghi S. M. S., Amiri B., and Pan W.-T. (2022). Model optimization analysis of customer churn prediction using machine learning algorithms with focus on feature reductions. *Discrete Dynamics in Nature and Society, 2022*, Article 5134356. https://doi.org/10.1155/2022/5134356

Ouf S., Mahmoud K. T., and Abdel-Fattah M. A. (2024). A proposed hybrid framework to improve the accuracy of customer churn prediction in telecom industry. *Journal of Big Data, 11*, Article 70. https://doi.org/10.1186/s40537-024-00922-9

Sagming M., Heymann R., and Visaya M. V. (2024). Using topological data analysis and machine learning to predict customer churn. *Journal of Big Data, 11*, Article 160. https://doi.org/10.1186/s40537-024-01020-6

Tanvir A.-A., Khandokar I. A., Islam A. K. M. M., Islam S., and Shatabda, S. (2023). A gradient boosting classifier for purchase intention prediction of online shoppers. *Heliyon, 9*(4), Article e15163. https://doi.org/10.1016/j.heliyon.2023.e15163

Verbeke W., Dejaeger K., Martens D., Hur J., and Baesens B. (2012). New insights into churn prediction in the telecommunication sector: A profit driven data mining approach. *European Journal of Operational Research, 218*(1), 211–229. https://doi.org/10.1016/j.ejor.2011.09.031

©2026 This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 International license viewed via https://creativecommons.org/licenses/by/4.0/ which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is cited appropriately.