International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

STAYNEX: An Explainable, AI-Powered Customer Churn Intelligence and Retention Decision-Support System for E-Commerce

Karan Rautela
Dept. of Computer Science and Engineering
HMR Institute of Technology and Management, Delhi, India

Vardaan Gaur
Dept. of Computer Science and Engineering
HMR Institute of Technology and Management, Delhi, India

Hardik Kaushik
Dept. of Computer Science and Engineering
HMR Institute of Technology and Management, Delhi, India

***Abstract*—Customer churn erodes revenue in e-commerce, yet most retention teams learn about a lost customer only after the last purchase. This paper presents STAYNEX, a web-based churn intelligence system that predicts the churn probability of each customer, assigns a risk tier and a business segment, explains the behavioural drivers behind every score, recommends retention actions, and lets analysts test what-if scenarios before spending marketing budget. The system combines a cleaning and feature engineering pipeline built on recency, frequency and monetary indicators plus composite engagement and return-cancellation measures with a Random Forest classifier, a lightweight explainability engine that relies only on native tree importances, a rule-based recommender, a what-if simulator, a revenue-at-risk estimator and a rule-based analyst, all served through a Flask application with interactive dashboards. On a synthetic dataset of 5,000 customers with a churn rate of 27.0 percent, the model reaches a ROC-AUC of 0.76 and an accuracy of 0.76 on a held-out test set. A logistic regression baseline reaches 0.78 AUC, which shows that the synthetic data are largely linear and that the contribution of this work is the integrated decision-support pipeline rather than a new classifier. A case study shows that a combined intervention lowers the predicted churn probability of a critical-risk customer from 0.87 to 0.58. We discuss limitations and outline validation on real transaction data.**

Keywords - customer churn, e-commerce, explainable AI, random forest, customer retention, what-if simulation, customer segmentation

I. INTRODUCTION

Online retailers operate in a market where switching costs are low and competing offers are one click away. Acquiring a new customer is generally more expensive than keeping an existing one [1], so the ability to recognise customers who are drifting away, and to act while they can still be won back, has direct financial value. Churn in non-contractual settings such as e-commerce is especially hard to manage because customers never formally cancel; they simply stop buying, and the loss is discovered only when the silence has already lasted for months.

Machine learning has been widely applied to churn prediction [2], [3], but a prediction alone rarely changes what a business does. A marketing manager who receives a list of probabilities still has to decide why each customer is at risk, which intervention fits, whether the intervention is worth its cost, and how much revenue is actually exposed. Black-box scores also reduce trust: if the reason for a high score cannot be inspected, analysts hesitate to act on it [4], [5].

This paper presents STAYNEX, an end-to-end system that closes this gap. It treats churn prediction as the first step of a decision-support workflow rather than the final output. The main contributions are:

- an integrated pipeline that cleans raw customer data, engineers recency-frequency-monetary (RFM) and composite behavioural features, and scores every customer with a calibrated risk tier and business segment;
- a dependency-light explainability engine that produces signed, per-customer driver scores from native tree importances, making explanations available on platforms where compiled libraries such as SHAP are difficult to install;
- a rule-based recommender and a what-if simulator that translate risk drivers into concrete actions and let analysts compare churn risk before and after a behavioural change;
- a revenue-at-risk estimator and an interactive web application with role-friendly dashboards and a natural-language analyst widget.

The remainder of the paper is organised as follows. Section II reviews related work, Section III formulates the problem and describes the data, Section IV details the system, Section V reports experiments, Section VI discusses limitations, and Section VII concludes.

II. RELATED WORK

Churn prediction has been studied with logistic regression, decision trees, support vector machines, neural networks and ensembles. Comparative studies report that ensemble and boosted tree methods are competitive on tabular customer data, although the best model depends on the dataset and on how class imbalance is handled [2], [6]. Random forests [7] are popular in practice because they need little tuning, tolerate mixed feature scales and non-linear relationships, and expose feature importances.

RFM analysis summarises customer value through the recency of the last purchase, the frequency of purchases and the monetary value spent, and underpins much of customer-base

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

analysis in marketing [8]. STAYNEX uses RFM indicators as its core features and complements them with engagement and service-quality signals such as logins, product views, wishlist activity, returns, cancellations and support tickets.

Explainability methods such as LIME [5] and SHAP [4] attribute a prediction to input features and are central to the interpretable machine learning literature [9]. Exact SHAP values for tree ensembles rely on compiled extensions, which can be troublesome on newer platforms. STAYNEX therefore adopts a simpler signed attribution built from scikit-learn's native importances, trading exactness for portability and speed.

Finally, most published churn work stops at model evaluation. Systems that connect prediction to explanation, action, simulation and financial impact in one interface are less common, and this integrated workflow is the focus of the present work.

III. PROBLEM FORMULATION AND DATASET

A. Problem Formulation

Let each customer i be described by a feature vector x of behavioural, transactional and service attributes, and let y be a binary label that equals 1 if the customer has churned. The goal is to learn a function that returns the churn probability p for each customer, map p to one of four ordered risk tiers (LOW below 0.25, MEDIUM below 0.50, HIGH below 0.75, CRITICAL otherwise), and, for every customer, produce an explanation, a set of recommended actions and an estimate of the revenue exposed to churn.

B. Data Source

STAYNEX reads customers from a MySQL database, or from a CSV file when no database is configured. Because real transaction logs are confidential, the evaluation in this paper uses a built-in synthetic generator that produces 5,025 customer records with demographics, purchase history, recency, cart abandonment, returns, cancellations, logins, product views, wishlist items, session duration and support tickets. The churn label is drawn from a noisy logistic function of recency, cart abandonment, login activity, order frequency, support tickets, return and cancellation ratio and session length. The generator also injects realistic quality problems: 25 duplicated customer identifiers, 158 missing ages, 44 negative order values, 42 negative login counts and mixed date formats.

C. Cleaning and Preprocessing

The cleaning stage removes duplicated identifiers, converts all dates to a single format, replaces negative magnitudes by their absolute value, imputes remaining missing numeric values with the column median, fills missing categorical values with an explicit unknown label, and enforces valid ranges. After cleaning, 5,000 unique customers remain, of which 27.0 percent are labelled as churned.

TABLE I. FEATURES USED BY STAYNEX

Group	Features
Recency	days since last purchase
Frequency	total orders, monthly order frequency, tenure (days)
Monetary	average order value, total spending
Behaviour	cart abandonments, logins (30 days), product views, wishlist items, session minutes
Service	support tickets, returns, cancellations
Composite	engagement score, return/cancellation ratio
Demographic	age (model); gender, city, payment method (analytics only)

IV. PROPOSED SYSTEM

STAYNEX follows a layered architecture (Fig. 1). A data layer loads and cleans customer records; a machine learning layer engineers features and trains the classifier; a service layer implements segmentation, explanation, recommendation, simulation, revenue estimation and the analyst; and a Flask presentation layer exposes dashboards and JSON endpoints. Models are serialised with Joblib and loaded once at start-up, so every page and API call scores customers from an in-memory cache.

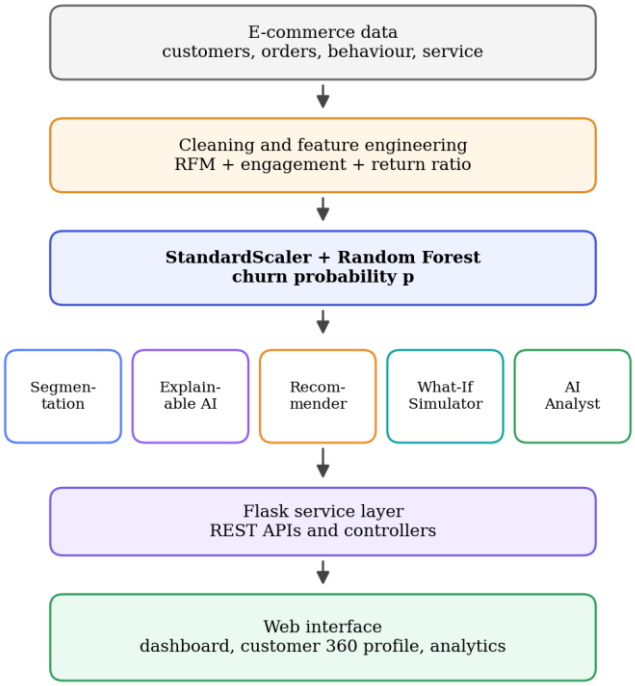


Fig. 1. Layered architecture of STAYNEX.

A. Feature Engineering

Recency is taken directly from the days since the last purchase. Frequency is the number of orders divided by tenure in months, and monetary value is captured by the average order value and total spending. Two composite indicators are added. The engagement score E combines four normalised signals on a scale of 0 to 100:

$$E = \Sigma\, w_k \cdot \min(x_k / c_k, 1) \qquad (1)$$

where the signals x are logins in the last 30 days, product views, wishlist items and average session minutes, the weights w are 35, 25, 15 and 25, and the caps c are 30, 200, 15 and 30. The return-cancellation ratio is the sum of returns and cancellations divided by total orders. In total, 17 numeric features are used by the model.

B. Churn Prediction Model

The features are standardised with a StandardScaler fitted on the training split, and a Random Forest classifier with 300 trees, a maximum depth of 12, a minimum leaf size of 5 and balanced class weights is trained on 80 percent of the customers. The remaining 20 percent form a stratified test set.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

The trained model, scaler and ordered feature list are saved so that the web application and the simulator apply exactly the same transformation. Customers are mapped to the four risk tiers defined in Section III.

C. Explainable Risk Drivers

For each customer, the explainability engine standardises every feature value and computes a signed contribution:

$$c_j = I_j \cdot z_j \cdot d_j \quad (2)$$

$$z_j = (x_j - \mu_j) / \sigma_j \quad (3)$$

where I is the global Random Forest importance of feature j, z is the standardised value (clipped to three standard deviations), and d is a direction of +1 when raising the feature by one standard deviation from the population mean increases the predicted churn probability, and -1 otherwise. The five largest absolute contributions are normalised to percentages and shown as the top risk drivers, each labelled as increasing or reducing risk. This mirrors the intent of SHAP summaries without compiled dependencies, but it is a heuristic attribution and not an exact Shapley decomposition.

D. Segmentation and Recommendations

Customers are assigned to one of five segments. New customers have a short tenure and at most two orders; lost customers have not purchased for more than 150 days; VIP customers belong to the top 15 percent by spending and have a churn probability below 0.5; at-risk customers have a probability of 0.5 or more; the remainder are loyal customers. The recommendation engine maps behavioural anomalies to actions, for example a cart-recovery voucher for four or more abandoned carts, a win-back campaign after 90 days of inactivity, proactive support outreach for three or more tickets, product-fit assistance for a high return-cancellation ratio, engagement boosters for a low engagement score, and concierge outreach for high-spending customers at HIGH or CRITICAL risk. Recommendations are sorted by priority.

E. What-If Simulator

The simulator takes a customer's current features and a set of behavioural changes, such as fewer abandoned carts, more logins or a shorter recency, applies the changes within valid bounds, recomputes the composite features, rescales the vector and scores it again with the same model. It returns the probability before and after, the absolute and relative change, the tier transition and the expected revenue retained.

F. Revenue at Risk

The expected monetary loss for customer i is the product of lifetime spending S and the churn probability p:

$$R_i = S_i \cdot p_i \quad (4)$$

Summing R over customers, tiers or segments gives the revenue exposed to churn for a portfolio, which helps to prioritise campaigns by financial impact rather than by probability alone.

G. Web Application and AI Analyst

The Flask application provides an overview dashboard with key performance indicators, a risk-distribution chart and a top-risk table; a searchable, filterable customer list; a customer profile with a churn gauge, the driver chart, recommendations, recent orders and the interactive simulator; and an analytics page with churn by city and payment method. Simulations and analyst queries are asynchronous requests, so pages do not reload. The AI analyst is a rule-based module that recognises question intents, such as the main reason for churn, the highest-risk customers, revenue at risk or churn by city, and answers from the scored customer table and the global feature importances.

V. EXPERIMENTS AND RESULTS

A. Experimental Setup

The 5000 cleaned customers were split into 4000 training and 1000 test customers with stratification. All models use the same standardised features. Besides the deployed Random Forest, we trained a logistic regression with balanced class weights and a gradient boosting classifier with default settings as baselines. Metrics are computed on the test set at a probability threshold of 0.5; ROC-AUC is threshold-independent. Because churn is the minority class, precision, recall and F1 are reported for the churn class.

B. Model Comparison

TABLE II. TEST-SET PERFORMANCE OF CHURN MODELS

Model	Acc.	Prec.	Rec.	F1	AUC
Logistic Reg.	0.730	0.500	0.667	0.571	0.779
Grad. Boosting	0.779	0.676	0.348	0.460	0.764
Random Forest	0.764	0.573	0.496	0.532	0.762

Table II shows that the three models are close in AUC. The Random Forest obtains an AUC of 0.762 and an accuracy of 0.764, and five-fold cross-validation on the full data gives a mean AUC of 0.759 with a standard deviation of 0.020, so the result is stable. Logistic regression is slightly better in AUC (0.779) and recall (0.667), and gradient boosting has the highest accuracy (0.779) but the lowest recall (0.348). These differences are small. The most likely reason that a linear model performs well is that the synthetic labels were generated from a logistic function, so non-linear ensembles have little additional structure to exploit. We keep the Random Forest as the deployed model because it handles non-linear effects and feature interactions that are expected in real transaction data, and because its importances support the explanation module. Fig. 2 shows the ROC curves.

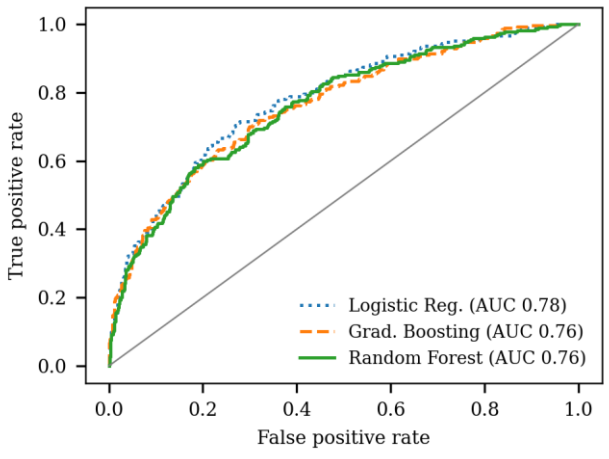


Fig. 2. ROC curves of the three models on the test set.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

C. Global Feature Importance

Fig. 3 shows the eight most important features of the deployed model. Days since last purchase is the strongest predictor (18.4 percent of total importance), followed by Logins (30 days) (12.3 percent) and engagement score (11.3 percent). Recency and engagement therefore dominate, in line with the way the data were generated and with the common observation that inactivity is the earliest warning sign of churn. Because the engagement score is derived from logins, views and sessions, part of its importance overlaps with those raw features.

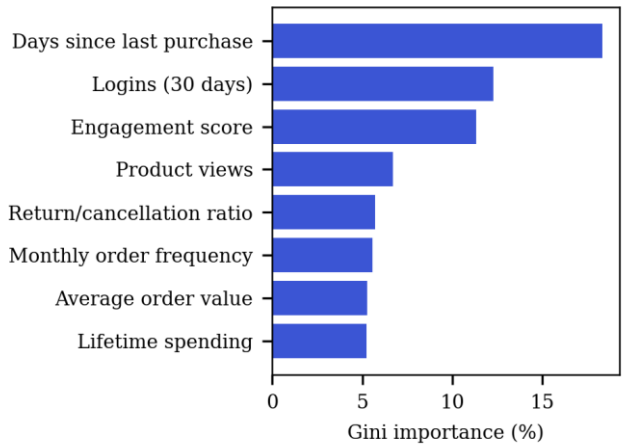


Fig. 3. Top eight features by Random Forest importance.

D. Risk Tiers and Revenue at Risk

TABLE III. RISK TIERS AND REVENUE AT RISK ON THE TEST SET

Tier	Customers	Revenue at risk ($)
LOW	346	48,991
MEDIUM	420	137,826
HIGH	178	91,392
CRITICAL	56	42,623
Total	**1000**	**320,833**

Table III applies the deployed model to the 1000 test customers. 234 customers (23.4 percent) fall in the HIGH or CRITICAL tiers, which are the natural targets for immediate action. The expected revenue at risk is $320,833, or 37.4 percent of the $857,841 lifetime spending of these customers. The MEDIUM tier holds the largest share of exposure because of its size, which illustrates why ranking by revenue at risk can differ from ranking by probability.

E. Case Study: Explanation and What-If Simulation

To illustrate the workflow, consider test customer 12156, who has 5 abandoned carts, 3 logins in the last 30 days, 151 days since the last purchase and 2 support tickets. The model assigns a churn probability of 0.87 (CRITICAL). The top driver is days since last purchase (32.1 percent of the explained impact), followed by logins (30 days) (18.2 percent) and engagement score (17.0 percent); all three increase risk. The recommender therefore proposes a cart-recovery voucher, a win-back campaign and an engagement booster.

TABLE IV. WHAT-IF SCENARIOS FOR THE CASE-STUDY CUSTOMER

Quantity	Now	Scen. A	Scen. B
Cart abandonments	5	3	2
Logins (30 days)	3	8	13
Days since purchase	151	136	121
Support tickets	2	2	1
Churn probability	0.87	0.79	0.58
Risk tier	CRITICAL	CRITICAL	HIGH

Scenario A applies the example change used in the interface (two fewer abandoned carts, five more logins and 15 fewer days since the last purchase). It reduces the probability only to 0.79, a relative drop of 8.5 percent, and the customer stays CRITICAL. Scenario B combines a larger change in all four levers and lowers the probability to 0.58, a relative drop of 33.3 percent, moving the customer to the HIGH tier with an estimated $119.11 of revenue retained. The simulator thus shows analysts how large an intervention must be before the risk level changes, which supports cost-benefit decisions on voucher size or outreach effort.

VI. DISCUSSION AND LIMITATIONS

Synthetic data. All quantitative results come from a synthetic dataset whose labels follow a known noisy logistic process. The numbers demonstrate that the pipeline works end to end, not that the model would achieve the same accuracy on a real shop. Validation on public or industrial e-commerce data with a time-based split and a clearly defined churn window is the most important next step.

Model choice. On this data, logistic regression matched or exceeded the Random Forest, so the value of the Random Forest rests on assumptions about real data rather than on the reported results. Threshold tuning, probability calibration and cost-sensitive learning would also improve the recall of the churn class, which is only about half at the default threshold.

Explanations. The signed attribution is a transparent heuristic that combines global importances with local standardised values. It has not been validated against exact SHAP values or permutation importance, and correlated features, such as the engagement score and logins, can share credit.

Simulation. The simulator reports how the model's prediction changes when features change. This is an association learned from observational data, not a causal estimate of intervention effect, and features such as recency cannot always be changed independently. Uplift modelling or controlled experiments would be needed to measure the true effect of a retention action.

Operational scope. The recommender and analyst are rule-based, so their coverage is limited to the rules written by the developers. The deployed application scores the customer table it was trained on, so dashboard figures should not be read as out-of-sample estimates.

VII. CONCLUSION AND FUTURE WORK

This paper presented STAYNEX, a churn intelligence system that turns customer behaviour data into churn probabilities, risk tiers, segments, explanations, recommended actions, what-if comparisons and revenue-at-risk estimates inside one interactive web application. Experiments on a synthetic dataset gave a ROC-AUC of 0.76 for the Random

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

Forest and showed that the supporting modules provide actionable detail for individual customers. Future work will evaluate the system on real e-commerce data, compare the explanation engine with SHAP and permutation importance, add gradient boosting and calibration, replace the simulator's associative estimates with uplift models, and add role-based access and live data integration.

REFERENCES

[1] F. F. Reichheld and W. E. Sasser, Jr., “Zero defections: Quality comes to services,” Harvard Business Review, vol. 68, no. 5, pp. 105–111, 1990.
[2] A. Vafeiadis, K. Diamantaras, G. Sarigiannidis, and K. Chatzisavvas, “A comparison of machine learning techniques for customer churn prediction,” Simulation Modelling Practice and Theory, vol. 55, pp. 1–9, 2015.
[3] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[4] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 4765–4774.
[5] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should I trust you? Explaining the predictions of any classifier,” in Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.
[6] J. Burez and D. Van den Poel, “Handling class imbalance in customer churn prediction,” Expert Systems with Applications, vol. 36, no. 3, pp. 4626–4636, 2009.
[7] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
[8] P. S. Fader, B. G. S. Hardie, and K. L. Lee, “RFM and CLV: Using iso-value curves for customer base analysis,” Journal of Marketing Research, vol. 42, no. 4, pp. 415–430, 2005.
[9] C. Molnar, Interpretable Machine Learning, 2nd ed., 2022. [Online]. Available: https://christophm.github.io/interpretable-ml-book/

(This work is licensed under a Creative Commons Attribution 4.0 International License.)