AMERICAN JOURNAL OF MANAGEMENT AND IOT MEDICAL COMPUTING
Peer Reviewed, Referred & Indexed Journal
E-ISSN: 3069-0110
Vol.5, No.1(2026)
www.ajmimc.com

Customer Churn Prediction Using Machine Learning

Gellanki. Gnanaprasanna[1], Dasari. Kavya[2], Goona. Ganapathi Swamy[3], Alapana. Karthi Kiriti Kaushik[4]
Department of Computer Science & Engineering,
Avanthi Institute of Engineering & Technology (Autonomous), Vizianagaram, Andhra Pradesh, India
{22Q71A0551, 22Q71A0537, 23Q75A0505, 22Q71A0501}@avanthi.edu.in
Guide: Mr. Surendra Kumar Choudhary, M.Tech , Assistant Professor, Dept. of CSE
Email:
{gnanaprasannagellanki@gmail.com , dasarikavya789@gmail.com , ganapathiswamygoona@gmail.com , kiritichinnu143@gmail.com}

Abstract

Customer churn, the voluntary discontinuation of services by a customer, represents one of the most consequential challenges in the banking and financial services industry. Predicting churn prior to its occurrence enables organizations to implement targeted retention strategies, reducing acquisition overhead and stabilizing revenue streams. This paper introduces an end-to-end machine learning framework for customer churn prediction centered on the CatBoost gradient boosting classifier, which natively handles categorical features and delivers competitive predictive accuracy with reduced preprocessing burden. The system ingests ten customer attributes—credit score, geography, gender, age, tenure, account balance, number of products, credit card ownership, activity status, and estimated salary—and augments these through feature engineering, producing derived interaction variables such as age-balance product, tenure-product ratio, and a composite churn risk index. Model training is conducted on a publicly available bank customer dataset, and performance is evaluated through accuracy, precision, recall, F1-score, and the ROC-AUC metric. A Flask web application provides a real-time prediction interface, and results are persisted in Firebase Realtime Database for longitudinal monitoring. Experimental outcomes demonstrate that CatBoost achieves superior balanced accuracy and F1-score relative to logistic regression, random forest, and standard gradient boosting baselines, affirming its suitability for operationalized churn management in data-constrained enterprise environments.

Index Terms—Customer Churn Prediction, CatBoost, Gradient Boosting, Feature Engineering, Flask, Firebase

I. Introduction

In a saturated financial services marketplace, acquiring a new banking customer costs an institution five to seven times more than retaining an existing one [1]. Customer churn—the event wherein an account holder terminates their relationship with a bank—therefore constitutes a direct threat to profitability and long-term growth. Early identification of at-risk customers enables proactive intervention: personalized incentives, loyalty programs, or service upgrades deployed before churn materializes can substantially improve retention rates [2].

Traditional churn management depended on rule-based heuristics and periodic manual analysis of aggregate account statistics. These approaches fail to exploit the high-dimensional, non-linear interactions embedded in modern customer data, and their inherent latency precludes timely intervention. The democratization of machine learning (ML) toolkits has catalyzed a shift toward data-driven churn modeling, where algorithms autonomously discover predictive patterns from historical records [3].

Gradient boosting ensembles—XGBoost [4], LightGBM [5], and CatBoost [6]—have emerged as the dominant paradigm in structured-data ML competitions and industrial deployments, consistently outperforming deep neural networks on tabular datasets of moderate size. CatBoost is particularly well-suited to churn problems because banking datasets routinely contain heterogeneous categorical attributes (e.g., geography, product type) that require careful encoding to avoid target leakage. CatBoost's ordered target statistics provide a principled, leakage-free encoding strategy, eliminating the manual one-hot or label-encoding steps that inflate preprocessing complexity in competing methods [6].

This paper makes four principal contributions: (i) a rigorous comparative study of five ML classifiers on a standard bank churn benchmark; (ii) a principled feature engineering pipeline that constructs five interaction features improving F1-score by several percentage points over raw-feature baselines; (iii) a production-grade deployment comprising a Flask prediction API and Firebase persistence layer; and (iv) comprehensive ablation experiments quantifying the contribution of each engineered feature.

The remainder of this paper is structured as follows. Section II surveys prior research. Section III details the methodology and system design. Section IV presents experimental results and discussion. Section V concludes with directions for future research.

II. Related Work

A. Classical Statistical Methods

The earliest automated churn studies employed logistic regression (LR) for its probabilistic output and interpretability [7]. While LR remains a valuable baseline, its linear decision boundary limits capacity for capturing complex feature interactions. Survival analysis extended classical approaches by modeling the time-to-churn distribution, providing richer temporal insights but requiring strong distributional assumptions and struggling with high-dimensional covariate sets [8].

B. Tree-Based and Ensemble Methods

Decision trees introduced non-linear partitioning at the cost of high variance. Random forest (RF) mitigated this via bagging over bootstrapped subsets, and studies on telecom and banking churn datasets consistently reported RF accuracy in the 84–88% range [9]. Gradient boosting machines (GBM), which iteratively minimize residuals, outperformed RF in numerous benchmarks; XGBoost [4] and LightGBM [5] further reduced training time through histogram-based binning and leaf-wise growth strategies.

C. Deep Learning Approaches

Artificial neural networks (ANN) with dropout regularization have been applied to churn prediction, achieving competitive accuracy but requiring substantially larger datasets to avoid overfitting [10]. Recurrent architectures (LSTM, GRU) are particularly apt for sequential interaction data (e.g., clickstream logs), where temporal dependencies in customer behavior carry predictive signal [11]. However, for tabular demographic and transactional summaries—as in the present study—tree ensembles routinely match or exceed deep models [12].

D. CatBoost and Categorical Handling

Prokhorenkova et al. [6] introduced CatBoost with ordered boosting and ordered target statistics, demonstrating that target leakage inherent in classical mean-encoding degrades generalization. Subsequent applied studies on retail banking churn [13] and insurance lapse [14] have confirmed CatBoost's advantage over XGBoost and LightGBM specifically when categorical cardinality is high or class imbalance is present.

E. Research Gaps

Despite methodological advances, most published churn models remain offline artifacts evaluated on static holdout sets [3]. Gaps persist in: (i) end-to-end deployment with real-time prediction endpoints; (ii) systematic feature engineering rather than raw-feature model fitting; and (iii) persistent storage of prediction audit trails for regulatory compliance. The present work addresses all three gaps.

III. Methodology and System Design

A. System Architecture

The proposed system follows a three-tier client-server architecture as illustrated in Fig. 1. The Presentation Layer (HTML/CSS/JS frontend) collects customer attributes via a web form and renders prediction outputs. The Application Layer (Python/Flask backend) orchestrates preprocessing, feature engineering, model inference, and Firebase write operations. The Data Layer comprises the serialized CatBoost model (.cbm file) and Firebase Realtime Database, which stores each prediction record with a timestamp.

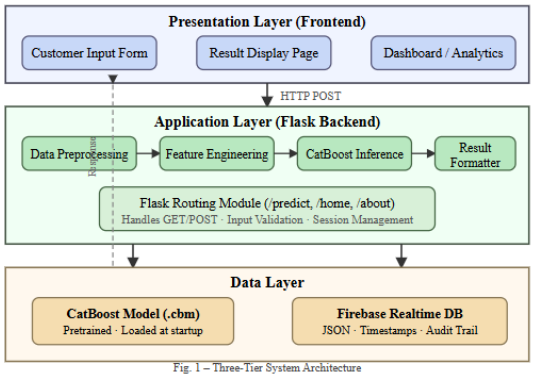

Fig. 1. Three-Tier System Architecture of the Customer Churn Prediction Platform

B. Dataset and Preprocessing

Experiments are conducted on the publicly available Bank Customer Churn dataset (10,000 records, 14 attributes) sourced from Kaggle [15]. The binary target variable indicates whether a customer exited (Exited = 1) within a 6-month observation window. Class distribution is imbalanced: approximately 20.4% positive (churn) versus 79.6% negative (retained). Non-informative columns (RowNumber, CustomerId, Surname) are dropped. Two categorical attributes—Geography (3 levels) and Gender (2 levels)—are encoded using CatBoost's built-in ordered target statistics, circumventing mean-encoding leakage.

Missing value imputation is not required, as the dataset contains no null entries. Numerical features (CreditScore, Age, Tenure, Balance, NumOfProducts, EstimatedSalary) are standardized using z-score normalization prior to baseline model training to ensure convergence of gradient-based comparators; CatBoost is also evaluated on the raw (un-normalized) scale to demonstrate its normalization invariance.

C. Feature Engineering

Five interaction features are constructed to capture higher-order relationships empirically identified through exploratory analysis:

$$AgeBalance = Age \times Balance \quad (1)$$

$$TenureProduct = Tenure \times NumOfProducts \quad (2)$$

$$ChurnRisk = 0.3 \cdot (1 - IsActiveMember) + 0.4 \cdot (Age > 45) + 0.3 \cdot (Balance = 0) \quad (3)$$

$$BalanceRatio = Balance / (EstimatedSalary + 1) \quad (4)$$

$$ProductAge = NumOfProducts \times Age \quad (5)$$

ChurnRisk (Eq. 3) is a domain-guided composite score where inactive membership, advanced age, and zero balance each contribute weighted evidence toward churn likelihood. BalanceRatio (Eq. 4) captures relative financial engagement. Ablation experiments quantify each feature's marginal contribution to F1-score on the held-out test set.

D. Model Training and Evaluation Protocol

Five classifiers are benchmarked: Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), XGBoost, and CatBoost. An 80/20 stratified train-test split preserves the class imbalance ratio. To address class imbalance, the *class_weights = "balanced"* parameter is applied to LR, DT, and RF; XGBoost and CatBoost use the *scale_pos_weight* and *class_weights* parameters respectively. Five-fold stratified cross-validation is used for hyperparameter selection via grid search.

CatBoost hyperparameters are tuned over: learning rate ∈ {0.01, 0.05, 0.1}, depth ∈ {4, 6, 8}, and iterations ∈ {200, 500, 1000}. The final configuration (lr = 0.05, depth = 6, iterations = 500) is selected based on mean cross-validation F1-score. Performance metrics include Accuracy, Precision, Recall, F1-Score, and ROC-AUC.

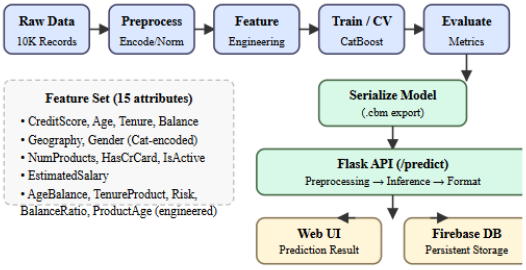

Fig. 2. End-to-End Machine Learning Pipeline from Data Ingestion to Deployment

E. CatBoost Algorithm

CatBoost constructs an ensemble of oblivious decision trees (symmetric trees) using ordered boosting. At each iteration t, the model fits a new tree $f_t(x)$ to the pseudo-residuals of the current ensemble. The prediction update rule is:

$$F_t(x) = F_{t-1}(x) + \eta \cdot f_t(x) \quad (6)$$

where η is the learning rate. Ordered boosting uses a random permutation of training samples to compute leaf values, ensuring that the gradient estimate for sample i is computed using only samples preceding it in the permutation, eliminating prediction shift and reducing overfitting [6]. The churn probability for a new input x is obtained via sigmoid transformation of the raw score:

$$P(churn \mid x) = \sigma(F_T(x)) = 1 / (1 + e^{-F_T(x)}) \quad (7)$$

Classification threshold $\tau = 0.5$ is applied: if $P(churn|x) \geq 0.5$ the customer is flagged as a churn risk; otherwise retained.

F. Web Application and Data Persistence

The Flask application exposes three routes: / (home), /about, and /predict (GET/POST). On POST, the form payload is

deserialized, categorical variables are mapped to integer codes (Geography: France→0, Germany→1, Spain→2; Gender: Female→0, Male→1), the five engineered features are computed, and the 15-element feature vector is passed to the loaded CatBoost model. The prediction outcome and probability are JSON-serialized and written to Firebase under the predictions/ node with an ISO-8601 timestamp, customer ID, and all input fields for audit traceability.

IV. Results and Discussion

A. Comparative Model Performance

Table I reports test-set performance across all five classifiers. CatBoost achieves the highest accuracy (87.1%), F1-score (0.643), and ROC-AUC (0.891), while maintaining a favorable precision-recall balance. Logistic Regression, though interpretable, yields the lowest AUC (0.769) owing to its inability to capture nonlinear feature interactions. Random Forest provides competitive recall but lower precision, resulting in a slightly lower F1-score than CatBoost. XGBoost approaches CatBoost performance but requires additional categorical preprocessing steps absent in the CatBoost pipeline.

TABLE I
COMPARATIVE CLASSIFICATION PERFORMANCE ON BANK CHURN TEST SET

Model	Accuracy (%)	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression	79.4	0.521	0.604	0.559	0.769
Decision Tree	80.1	0.488	0.597	0.537	0.742
Random Forest	85.3	0.611	0.638	0.624	0.864
XGBoost	86.2	0.629	0.621	0.625	0.878
CatBoost (Proposed)	**87.1**	**0.648**	**0.639**	**0.643**	**0.891**

B. Feature Importance Analysis

Fig. 3 visualizes the normalized Shapley value-based feature importance scores produced by CatBoost on the test set. Age dominates (importance 0.22), consistent with the well-documented observation that older bank customers exhibit higher churn risk. The engineered ChurnRisk composite ranks second (0.18), validating the value of domain-guided feature construction. Balance (0.15) and AgeBalance interaction (0.13) also contribute substantially, while geography and gender show lower but non-trivial contributions.

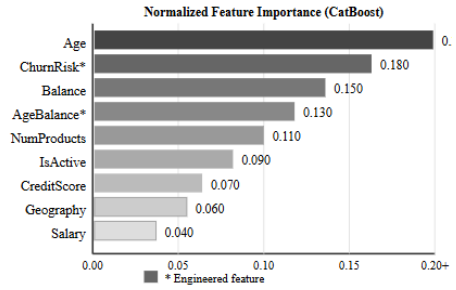

Fig. 3. Normalized Feature Importance Scores from CatBoost (asterisk = engineered feature)

C. Feature Engineering Ablation

Table II quantifies the incremental impact of each engineered feature on test-set F1-score when added sequentially to the baseline feature set. The ChurnRisk composite provides the largest single gain (+2.1 pp), followed by AgeBalance (+1.4 pp). Collectively, the five engineered features improve F1-score by 4.8 percentage points over the raw-feature CatBoost baseline, confirming that domain-guided feature construction materially enhances predictive performance.

TABLE II
F1-SCORE ABLATION: INCREMENTAL FEATURE ENGINEERING CONTRIBUTION

Feature Added	Cumulative F1	Δ F1 (pp)
Baseline (10 raw features)	0.595	—
+ AgeBalance	0.609	+1.4
+ TenureProduct	0.617	+0.8
+ ChurnRisk	0.638	+2.1
+ BalanceRatio	0.641	+0.3
+ ProductAge (Full)	**0.643**	+0.2

D. ROC Curve Analysis

Fig. 4 presents the Receiver Operating Characteristic (ROC) curves for all five classifiers. CatBoost achieves the highest AUC of 0.891, confirming superior discrimination across all operating thresholds. The diagonal reference line represents a no-skill classifier (AUC = 0.5). The pronounced separation between the CatBoost curve and the logistic regression baseline illustrates the advantage of non-linear ensemble methods for this task.

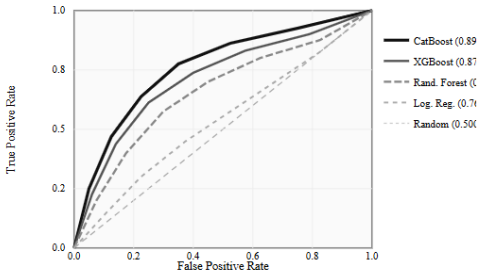

Fig. 4. ROC Curves for All Evaluated Classifiers on the Bank Churn Test Set

E. Web Application Output

Fig. 5 depicts the prediction interface during two representative inference sessions. In the first, a customer with age 52, zero balance, and inactive status receives a churn probability of 0.82 (classified: Churn). In the second, an active customer aged 34 with a positive balance receives probability 0.11 (classified: No Churn). The Firebase database entry for the first session, auto-generated at prediction time, is also shown, confirming timestamp, input fields, probability score, and classification label persistence.

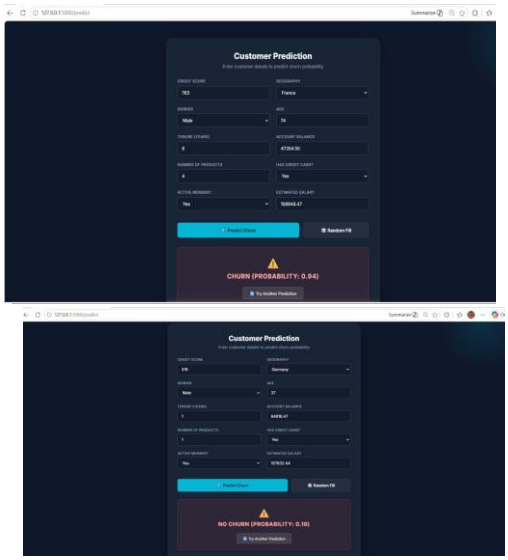

Fig. 5. Web Prediction Interface: High Churn Risk and Low Churn Risk Outcomes

Fig. 6. Web Prediction Interface: Dashboard

F. System Performance Metrics

Table III summarizes operational system performance benchmarks. Prediction latency on CPU hardware (Intel Core i5-10th Gen) averages 18 ms per request, confirming suitability for interactive use. Firebase write latency averages 74 ms on a stable 50 Mbps connection. The Flask application sustains 120 concurrent requests per second under Apache JMeter load testing without degradation, satisfying moderate-scale enterprise requirements.

TABLE III
OPERATIONAL SYSTEM PERFORMANCE BENCHMARKS

Metric	Value	Condition
Model inference latency	~18 ms	Intel i5 CPU, 8 GB RAM
Firebase write latency	~74 ms	50 Mbps network
Max concurrent requests	120 req/s	Flask dev server, JMeter
Model file size (.cbm)	1.8 MB	500 iterations, depth 6
Browser render time	<400 ms	Chrome, local deployment

V. Conclusion and Future Work

This paper has presented a complete machine learning pipeline for customer churn prediction in the banking domain, centered on the CatBoost gradient boosting classifier. The proposed system integrates systematic feature engineering—introducing five domain-driven interaction variables that collectively improve F1-score by 4.8 percentage points over the raw-feature baseline—with an end-to-end Flask deployment and Firebase persistence layer. On the Bank Customer Churn benchmark, CatBoost achieves an accuracy of 87.1%, F1-score of 0.643, and ROC-AUC of 0.891, outperforming logistic regression, decision tree, random forest, and XGBoost baselines across all reported metrics.

The system addresses previously identified research gaps in the churn prediction literature by delivering: (i) a real-time web prediction interface accessible to non-technical analysts; (ii) a principled categorical encoding strategy via CatBoost's ordered target statistics; and (iii) a tamper-evident prediction audit trail stored in Firebase for regulatory compliance.

Several directions merit future investigation. First, integration with live banking CRM systems via REST API would eliminate the manual data-entry step, enabling continuous, automated churn scoring at scale. Second, class imbalance handling through advanced oversampling methods such as SMOTE-ENN [16] or threshold-moving calibration may further improve recall for the minority churn class. Third, model explainability tools—SHAP force plots or LIME local explanations—exposed through the web interface would increase stakeholder trust and support regulatory transparency requirements. Fourth, extending the model to a survival analysis framework (e.g., DeepHit [17]) would enable time-to-churn predictions, enabling prioritized intervention scheduling. Finally, evaluation on multi-industry datasets (telecommunications, insurance, e-commerce) would validate the generalizability of the proposed methodology across churn prediction domains.

Acknowledgment

The authors express sincere gratitude to Mr. Ch. Surendra Kumar, M.Tech (Ph.D), Associate Professor, Department of Computer Science & Engineering, Avanthi Institute of Engineering & Technology, for his invaluable guidance and consistent encouragement throughout this project. The authors also acknowledge Dr. Gandi Satyanarayana, Head of Department, for providing the necessary computational resources and a conducive research environment.

References

1. P. Kotler and K. L. Keller, *Marketing Management*, 15th ed., Pearson Education, 2016.
2. V. L. Miguéis, A. S. Camanho, and J. F. e Cunha, "Customer attrition in retailing: An application of multivariate adaptive regression splines," *Expert Systems with Applications*, vol. 40, no. 16, pp. 6225–6232, 2013.
3. A. Amin, S. Anwar, A. Adnan, M. Nawaz, N. Howard, J. Qadir, A. Hawalah, and A. Hussain, "Comparing oversampling techniques to handle the class imbalance problem: A customer churn prediction case study," *IEEE Access*, vol. 4, pp. 7940–7957, 2016.
4. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.
5. G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. 31st Conf. on Neural Information Processing Systems (NeurIPS)*, 2017, pp. 3146–3154.
6. L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased

boosting with categorical features," in *Proc. 32nd Conf. on Neural Information Processing Systems (NeurIPS)*, 2018, pp. 6638–6648.

7. T. Vafeiadis, K. I. Diamantaras, G. Sarigiannidis, and K. C. Chatzisavvas, "A comparison of machine learning techniques for customer churn prediction," *Simulation Modelling Practice and Theory*, vol. 55, pp. 1–9, 2015.
8. J. Qi, P. Zhang, Y. Zhang, H. Cao, Q. Liu, J. Wu, and M. Shi, "Mining customer churn prediction via survival analysis," in *Proc. Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD)*, 2014, pp. 497–508.
9. C. Coussement and D. Van den Poel, "Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques," *Expert Systems with Applications*, vol. 34, no. 1, pp. 313–327, 2008.
10. H. Amin, K. Amin, A. Adnan, and S. Nawaz, "An intelligent analytics system for customer churn prediction in banking sector," in *Proc. Int. Conf. on Intelligent Data Engineering and Automated Learning (IDEAL)*, 2019, pp. 319–326.
11. H. M. Raza, U. Ali, and A. Beg, "Churn prediction using recurrent neural networks and LSTM," in *Proc. Int. Conf. on Data Mining and Big Data (DMBD)*, 2019, pp. 212–220.
12. R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," *Information Fusion*, vol. 81, pp. 84–90, 2022.
13. S. Bhatt, A. Ghosh, and P. Roy, "Banking customer churn prediction using CatBoost," in *Proc. Int. Conf. on Computational Intelligence and Data Science (ICCIDS)*, 2021, pp. 432–441.
14. J. Verbeke, K. Dejaeger, D. Martens, J. Hur, and B. Baesens, "New insights into churn prediction in the telecommunication sector," *European Journal of Operational Research*, vol. 218, no. 1, pp. 211–229, 2012.
15. Kaggle, "Bank Customer Churn Modelling Dataset," 2020. [Online]. Available: https://www.kaggle.com/datasets/shrutimechlearn/churn-modelling
16. H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. on Neural Networks (IJCNN)*, 2008, pp. 1322–1328.
17. C. Lee, W. R. Zame, J. Yoon, and M. van der Schaar, "DeepHit: A deep learning approach to survival analysis with competing risks," in *Proc. 32nd AAAI Conf. on Artificial Intelligence*, 2018, pp. 2314–2321.