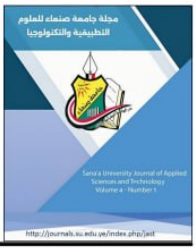
SUN PUBLISHER

مجلة جامعة صنعاء للعلوم التطبيقية والتكنولوجيا

Sana'a University Journal of Applied Sciences and Technology

https://journals.su.edu.ye/index.php/jast/



Machine Learning Algorithms for Customer Churn Prediction in the Banking Sector: A Comparative Study

Ibrahim Ahmed Al-Baltah * **and Sultan Yahya Al-Sultan**

Department of Information Technology, Faculty of Computer and Information Technology, Sana'a University, Sana'a, Yemen.

*Corresponding author: albalta2020@gmail.com

Abstract

Predicting customer churn in retail banking is essential for sustaining profitability. This study compares four supervised machine-learning models—Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), and Artificial Neural Network (ANN)—using the publicly available bank customer churn dataset from Kaggle (10,000 records, 18 attributes; publicly available at Kaggle repository. Data preprocessing included one-hot encoding for categorical variables, label encoding for gender, and feature selection via an ExtraTreesClassifier retaining nine informative predictors (e.g., age, credit score, balance). To address class imbalance (≈80% non-churners vs. 20% churners), models were trained and evaluated with and without the Synthetic Minority Oversampling Technique (SMOTE), which was applied only to the training folds under stratified 5-fold cross-validation. Evaluation metrics comprised accuracy, precision, recall (for churn class), F1-score, ROC–AUC, and PR–AUC. RF achieved the best balance between recall (0.484 imbalanced; 0.619 balanced) and accuracy (0.867 imbalanced), while LR with SMOTE attained the highest recall (0.715) at the cost of reduced accuracy (0.718). Overall, the results highlight RF as the most robust model across both distributions and emphasize the importance of imbalance-aware evaluation in bank churn prediction.

Article Info

Keywords:
Predicting Customer Churn, Machine Learning, Class Imbalance, Random Forest, SMOTE.

Article History:
Received: 30-September-2025,
Revised: 22-November -2025,
Accepted: 31-December-2025,
Available online: 29 January 2026.

1. INTRODUCTION

Customer churn—customers terminating their relationship with a bank—directly erodes revenue and market share in an increasingly competitive retail-banking environment. Even modest reductions in churn rates can translate into substantial gains, because the cost of acquiring new customers is typically much higher than the cost of retaining existing ones. As banks move toward digital channels and self-service platforms, traditional rule-based retention strategies become less effective, making data-driven churn prediction an essential component of customer relationship management [1].

From a modelling perspective, bank churn prediction is a binary classification problem with a strong class imbalance, where the majority of customers stay (non-churners) and a much smaller minority leave (churners). This imbalance complicates learning: naive models can achieve high accuracy by simply predicting the majority class while still failing to identify at-risk customers. Recently, machine learning (ML) and sampling techniques have been utilized to improve churn prediction using algorithms such as logistic regression, decision trees, random forests, gradient boosting, neural networks, and more recent deep or graph-based models [2].

In the banking domain, a growing body of research has applied ML to real customer data. For example, [2] compared several classical classifiers for bank churn and showed that ensemble methods outperformed single models in terms of accuracy and F-score. However, [2] and [3] study churn in a commercial bank in Ethiopia, highlighting the importance of handling imbalances for reliable predictions in emerging markets. Moreover, [4] evaluated ten ML models with multiple resampling techniques on U.S. community bank data and found that RF, XGBoost, AdaBoost, and bagging classifiers dominated in terms of accuracy, F-score, and ROC-AUC. The contributions of the study [5] explore SMOTE-based balancing on bank churn datasets, demonstrating that oversampling helps minority-class recall, but may not consistently improve AUC or precision. A recent study [6] investigated deep learning and temporal or graph-based architectures that incorporate behavioral sequences, yet often at the cost of reduced interpretability and higher data requirements [7].

Despite these advances, there are several gaps in the literature on bank customer churn. First, many studies have evaluated models primarily using overall accuracy, which can be misleading under severe class imbalance; metrics such as ROC-AUC, precision–recall AUC (PR-AUC), and recall for the churn class are not always reported [8]. Second, the experimental design is sometimes under-specified: the order of training/test splitting versus resampling, the use (or not) of stratified folds, and safeguards against data leakage are not always made explicit, making it difficult to reproduce or compare results. Third, even in studies that use the widely adopted Kaggle “Bank Customer Churn” dataset, reported performances vary widely, from moderate to near-perfect accuracy, without a consistent imbalance-aware evaluation framework [9]. Furthermore, some studies provide a multimetric analysis that exposes the trade-off between catching as many churners as possible (high recall) and limiting the number of falsely targeted non-churners (low FPR), which is crucial for designing cost-effective retention campaigns.

To address these gaps, this study develops a transparent and reproducible evaluation framework for bank customer churn prediction using a public retail bank dataset from Kaggle containing 10,000 customers and 18 attributes, including demographic, financial, and behavioral features, such as complaint history, satisfaction score, card type, and loyalty points [9]. After one-hot and label encoding, min–max normalization, and feature importance ranking with the ExtraTreesClassifier, we retained the nine most predictive features for modelling. We then compared four widely used supervised ML models (LR, SVM, ANN, and RF) against two experimental scenarios: (i) the original imbalanced data ($\sim$80% retainers vs. 20% churners), and (ii) a balanced version created by applying SMOTE only to the training data. A stratified 70/30 train–test split and 5-fold stratified cross-validation with fixed random seeds were used to ensure robustness and comparability across the models. The contributions of this study are threefold.

1. Rigorous imbalance-aware learning pipeline. This study establishes a transparent and reproducible machine learning framework for bank churn prediction by strictly separating data splitting, feature engineering, and resampling. SMOTE is applied only within the training folds to prevent data leakage, enabling a fair comparison between imbalanced and balanced learning scenarios.

2. Comprehensive multimetric and statistical performance evaluations. Four widely used supervised algorithms (LR, SVM, ANN, and RF) were evaluated using a rich suite of metrics: accuracy, precision, recall, F1-score, ROC-AUC, PR-AUC, balanced accuracy, and MCC, along with formal significance testing (DeLong and McNemar tests). This yields a more reliable performance interpretation than the accuracy-centric evaluations commonly observed in prior work.

3. Practical decision insights for banking churn management. The results highlight scenarios in which each model excels: RF provides the most stable overall performance, SVM minimizes false positives for cost-sensitive retention programs, and LR with SMOTE maximizes churn detection when recall is prioritized. These findings offer actionable guidance for banks to choose churn prediction strategies aligned with operational goals and cost considerations.

The subsequent sections of this study are organized as follows. A comprehensive review of the current state of the art is presented in Section (2). The study methodology is presented in Section (3). Section (4) presents comparative results of the selected algorithms. Section (5) concludes the study.

2. LITERATURE REVIEW

Customer churn in the banking sector has attracted increasing attention in recent years, especially as institutions face highly competitive markets and imbalanced churn data. A growing body of empirical work has explored which machine-learning algorithms perform best for bank churn prediction, how to handle class imbalance, and how to interpret the resulting models. Most recent studies have converged on three themes: the superiority of tree-based ensembles and advanced models over simple baselines, the importance of explicit imbalance-handling strategies such as SMOTE, and an increasing interest in explainable and deep learning–based approaches.

The work reported in [10] focused on machine learning and interpretability for bank credit products. It developed a churn prediction framework for bank credit card customers that balances data using oversampling techniques and compares several algorithms, ultimately selecting an Extreme Gradient Boosting (XGBoost) model as the core classifier. Their best model achieved very high predictive performance (accuracy and AUC close to 0.97) and was interpreted using SHAP values to identify key drivers of churn, such as credit limit, transaction patterns, and repayment behavior. This study demonstrates that combining powerful ensemble methods with interpretability tools can provide banks with accurate churn predictions and actionable explanations about why customers are leaving.

One of the most comprehensive studies that compared multiple algorithms and resampling strategies on bank churn data was [11]. Their study analyzed a banking churn dataset using a wide portfolio of models—Naïve Bayes, LR, SVM, Decision Tree (DT), RF, Gradient Boosting (GB), XGBoost, and LightGBM—together with several techniques for handling data imbalance. They showed that the hybrid SMOTE-ENN resampling method is the most effective for improving minority-class performance and that LightGBM achieves the best overall results with an accuracy of approximately 0.979 and correspondingly strong ROC-AUC [11].

Peng et al. [12] used a Kaggle bank churn dataset and constructed a GA-XGBoost model combined with multiple oversampling schemes, reporting substantial gains in both the AUC and recall for churners after resampling and feature selection. More broadly, Tam compared several data-level (oversampling, undersampling, hybrid) and algorithm-level approaches for handling imbalances in churn prediction tasks from banking and e-commerce, finding that oversampling methods generally work best on small- and medium-sized datasets and that ensemble models outperform single classifiers, especially when feature selection is guided by SHAP and mRMR [13]. These studies underline that class-imbalance handling is not optional but a central design choice in bank churn modelling.
Another stream of research concentrates on frameworks for bank churn prediction, using many classifiers on real bank data. For example, [5] proposed a machine-learning framework for a large community bank in the southern United States, constructing and comparing ten classification models, including LR, k-NN, SVM, DT, RF, Bagging, AdaBoost, GB, XGBoost, and Extra Trees, under five different sampling strategies. The results show that ensemble tree-based classifiers (RF, XGBoost, AdaBoost, and Bagging) consistently dominate other models in terms of accuracy, F-score, and ROC-AUC on the test observations. In this manner, [14] analyzed an extensive customer-level dataset from a multinational bank and compared RF, LR, DT, and elastic net models, and found that RF provides the best balance between accuracy and discriminatory power, while simultaneously identifying key behavioral and relationship features associated with attrition [14]. In addition, [15] retail-bank churn using GB, DT, and Gaussian Naïve Bayes, reporting very high and closely clustered accuracies, with Naïve Bayes slightly edging out the other models on their particular dataset and emphasized that even quite different learning paradigms can perform similarly when the data are highly informative [15]. Together, these studies reinforce the view that ensemble and tree-based models are particularly strong candidates for operational bank churn systems.

More recently, deep learning has been applied to banking churn with a focus on both performance and data balancing. Thenmozhi et al.. proposed a hyperparameter-tuned deep learning model for churn prediction in the banking sector. Their pipeline consists of data preprocessing, an Improved SMOTE (ISMOTE) procedure to rebalance the classes, and a deep neural network classifier whose hyperparameters are carefully optimized; the resulting system reaches an accuracy of around 97–98% and substantially improves minority-class recognition compared to baseline models [7]. In parallel, [16] introduced an explainable deep-learning-based churn prediction model that combines neural networks with interpretable components, providing both high-accuracy and global/local explanations of churn drivers [16].

At the same time, some studies have relied on classical statistical models as baselines for bank churn. Similarly, [17] applied binary LR to a dataset of 5,000 Indonesian credit card customers and identified a set of significant predictors, including number of dependents, marital status, number of products, months inactive, and transaction counts, achieving an accuracy of approximately 87% on hold-out data. Their work illustrated that LR remains a competitive and interpretable method for churn prediction, especially when carefully specified and supported by significance testing, although it typically underperforms tree-based and deep models on more complex or highly imbalanced datasets.

Beyond individual banks, several cross-domain or methodological studies have also influenced churn modelling in the financial sector. The imbalanced datasets affect the accuracy of machine learning models for churn prediction and show that severe skew towards the majority class leads to biased models that appear accurate but perform poorly on the minority (churn) class. They demonstrated that combining oversampling strategies with ensemble classifiers mitigates this issue and yields more reliable performance [18]. Brito et al. [19], working with a very large retail-bank dataset, compared the impact of resampling techniques versus hyperparameter tuning and concluded that the best results come from integrating both, particularly for gradient-boosting models, where PR-AUC and

ROC-AUC improve significantly once class imbalance is treated explicitly [19]. These more general contributions support the trend towards multimetric evaluation and careful experimental design in churn prediction research.

Furthermore, Tran et al. [20] examined customer churn prediction in the banking sector using a pipeline that combined customer segmentation with several supervised machine-learning models. Using a publicly available bank churn dataset, the authors first applied k-means clustering to partition customers into homogeneous groups and then trained the kNN, LR, DT, RF, and SVM classifiers on both the full sample and the resulting segments. The synthetic minority oversampling technique (SMOTE) was used to alleviate class imbalance. Their results show that RF clearly dominates the other models, achieving an accuracy of approximately 97% on the banking dataset, whereas LR attains the lowest accuracy (approximately 87%). Interestingly, they reported that customer segmentation has only a limited impact on prediction accuracy and that performance is driven mainly by the underlying learning algorithm rather than the segmentation step [20]. This study is therefore important for confirming the strong performance of tree-based ensembles in bank churn prediction, while also illustrating that segmentation alone is not sufficient to solve imbalance or enhance discrimination.

Hambali and Andrew [6] provide a focused empirical analysis of how oversampling affects classification performance in bank churn prediction. Working with an imbalanced retail-bank dataset containing demographic, account, and transaction features, they benchmark multiple classifiers (including logistic regression, k-nearest neighbors, random forest, and a simple neural network) in two scenarios: using raw data and using data rebalanced with SMOTE [6]. Their evaluation was based on accuracy, precision, recall, and F1-score, reported separately for the training and test sets. The authors showed that SMOTE consistently improved minority-class recognition and raises F1-scores across models, with the best configuration achieving test accuracy close to the mid-90% range [6]. However, the analysis remains largely descriptive: performance is compared via point estimates of standard metrics without ROC/PR curve analysis, confidence intervals, or formal statistical tests between models. This makes their study a useful empirical baseline for SMOTE in the banking context, while leaving room for more rigorous multi-metric and inferential comparisons.

The study in [21] investigated customer churn detection in the banking sector using random forest and LightGBM on a Kaggle-based bank churn dataset, and proposed a framework explicitly focused on probability calibration and interpretability. The dataset is first preprocessed to select the variables most directly related to banking churn behavior, and class imbalance is addressed using the SMOTETomek hybrid method, which combines SMOTE oversampling with Tomek-link cleaning to remove borderline and noisy instances. The authors evaluated models with multiple metrics, including accuracy, precision, recall, and F1-score, and importantly, the Brier score, to assess the calibration quality of predicted probabilities [21]. They demonstrated that calibrated random forest and LightGBM models yield more reliable probability estimates than their uncalibrated counterparts, improving the usefulness of churn scores for decision-making. In addition, the study applies SHAP to decompose model predictions into feature contributions, revealing which customer attributes (such as account tenure, balance, and transaction behavior) drive churn risk and providing detailed, model-agnostic explanations to support targeted retention strategies.

One recent work also explores more advanced hybrid strategies for bank churn prediction that combine resampling, ensembles, and temporal modelling. [22] propose a framework that applies multiple oversampling techniques, including SMOTE-Tomek, together with several classifiers such as logistic regression, decision tree, random forest, gradient boosting, XGBoost, and k-nearest neighbors on a heavily imbalanced bank-customer dataset. Their experiments showed that hybrid resampling (SMOTE-Tomek) paired with tree-based ensembles, particularly random forest and gradient boosting, yielded the highest classification performance, with accuracy and F1-scores reported in the high-90% range and notable improvements in minority-class recall compared with simpler baselines. Similarly, [21], summarized in the literature review of other banking studies, used panel-type data from European private banks and adopted dynamic modelling approaches that treat churn as a time-dependent process, applying supervised algorithms such as LR, SVM, RF, and GB to panel data to capture how evolving transaction and loan patterns influence the future likelihood of churn. These two studies collectively highlight that sophisticated resampling and ensemble strategies, sometimes in combination with longitudinal data structures, can substantially enhance churn detection in financial institutions.

In light of the above review, it is clear that prior research has made important progress in applying machine learning to bank customer churn prediction, especially through the use of tree-based ensembles, resampling techniques such as SMOTE, and more recently, deep learning and interpretability tools. However, most existing studies either focus on a single “best” model or on a narrow family of algorithms, and often report results for only one type of data preparation (usually a resampled dataset). In addition, performance is frequently assessed with a limited set of aggregate metrics (typically accuracy, F1-score, and ROC–AUC), with relatively little attention paid to confusion-matrix behavior, precision–recall characteristics, or formal statistical testing of performance differences between models and between imbalance-handling strategies.

To address these gaps, the present study undertakes a systematic and statistically grounded comparison of four widely used supervised learning algorithms—LR, SVM, RF, and artificial neural networks —on a real U.S. bank churn dataset under two clearly defined training regimes: the original imbalanced data and a SMOTE-balanced version of the training set. All models were evaluated within a unified pipeline using stratified train–test splitting and cross-validation, and their performance was analyzed through a multi-metric framework that included accuracy, precision, recall/TPR, FPR, F1-score, balanced accuracy, Youden's J, MCC, ROC–AUC (with DeLong confidence intervals), logistic regression, PR curves, and PR–AUC. Moreover, paired DeLong and McNemar tests were employed to determine whether the observed differences between models and between the imbalanced and balanced scenarios were statistically significant. Thus, the study not only confirms and refines earlier findings regarding the strong performance of tree-based and neural models in banking churn but also provides a transparent, robust baseline against which more complex ensemble or deep-learning approaches can be compared in future work.

3. METHODOLOGY

Recent studies have adopted a similar machine-learning experimental design combining multiple supervised algorithms, feature engineering, stratified cross-validation, and imbalance handling using SMOTE for bank churn prediction [23–27]. These studies demonstrate that pipeline-based workflows and balanced training regimes significantly improve the ROC-AUC and PR-AUC performance, particularly for tree-based and neural network models.

This study adopts a data-driven machine-learning pipeline to model and predict customer churn in the retail banking sector [19]. Various recent studies, such as [23–27] have adopted similar approaches. The main objective was to compare the performance of four widely used supervised learning algorithms (LR, SVM, RF, and ANN) under two distinct training regimes: using the original imbalanced data and using an SMOTE-balanced version of the training data. The methodological workflow comprises three main stages:

1. **Data understanding,** which describes the source dataset and the churn label;

2. **Data Preprocessing** included encoding, normalization, feature selection, imbalance handling, and stratified data splitting.

3. **Model implementation and evaluation,** where the four algorithms were embedded in reproducible scikit-learn/imblearn pipelines, trained with Stratified K-Fold cross-validation, and evaluated on a held-out test set using a rich set of metrics and statistical tests.

3.1. Data Understanding

To investigate the churn problem in the banking domain, we used a publicly available U.S. bank customer dataset from Kaggle. The dataset contains 10,000 customers, each described by 18 attributes or features capturing demographics, accounts, product usage, and satisfaction-related information, together with a binary churn label. The original dataset is listed in Table (1).
The original target feature Exited indicates whether the customer has left the bank (1, churn) or remains with the bank (0, retain). The features include:

- Technical identifiers (RowNumber, CustomerId, Surname),

- Demographic variables (Geography, Gender, Age),

- Relationship characteristics (Tenure, NumOfProducts, HasCrCard, IsActiveMember),

- Financial indicators (CreditScore, Balance, EstimatedSalary), and

- Behavioral/satisfaction features (complaint, satisfaction score, card type, point earned) plus the excited label (later renamed churn during preprocessing).

An initial inspection of the target distribution showed a typical class imbalance: approximately 80% of the customers were non-churners (class 0) and 20% were churners (class 1), giving an imbalance ratio of approximately 4:1. This reflects the reality of churn in banking, but it also motivates the need for explicit handling of class imbalance in the modelling stage.

3.2. Data Preprocessing

All data preparation steps and subsequent modelling were performed in Python using the Jupyter Notebook, relying on the pandas, numpy, scikit-learn, imblearn, matplotlib, and seaborn libraries. The main pre-processing components are described as follows. Figure(1) illus-

Table[1]: Dataset Description

N	Feature	Description
1	Row Number	Quantity of customers
2	Customer ID	Customer identification numbers
3	Surname	The name of the customer (the last name of him or her)
4	Credit Score	Score of credit card usage
5	Geography	Location of the customer
6	Gender	Customer gender
7	Age	Customer's age
8	Tenure	The duration of the account, expressed in months
9	Balance	Customer's main balance
10	NumOfProducts	No of products used by the customer
11	HasCrCard	Does the client have a credit card? card
12	IsActiveMember	Is the customer's account active?
13	Estimated Salary	The customer's estimated salary
14	Complain	Refers to instances where customers or clients file complaints or express dissatisfaction with the bank's products, services
15	Satisfaction Score	A metric that measures the level of satisfaction or happiness expressed by customers
16	Card Type	Refers to the specific type of credit or debit card issued by the bank
17	Point Earned	Point Earned refers to the accumulation of reward points or loyalty points associated with a specific card or account.
18	Exited	Indicates customer left the bank (Churn) or non-churners (Retain)

trates the data-preparation process used in this study.

3.2.1. Handling Missing Values

The raw dataset was inspected for missing values using standard pandas' functions. The U.S. bank churn dataset used in this study does **not** contain missing entries in the variables retained for modelling, so no imputation or row deletion was required. This allows the full sample of 10,000 customers to be exploited and ensures that the observed performance differences are driven by modelling choices rather than by missing data handling.

3.2.2. Encoding the Categorical Features

Because scikit-learn algorithms operate on numerical inputs, the categorical attributes are transformed as follows:

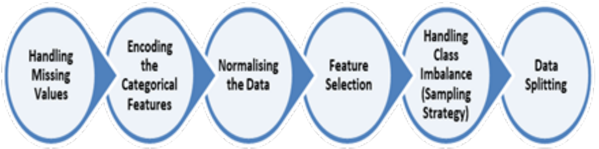

Handling Missing Values
Encoding the Categorical Features
Normalising the Data
Feature Selection
Handling Class Imbalance (Sampling Strategy)
Data Splitting

Figure 1. Data Preprocessing Phases

- Geography and Card Type were converted into dummy (one-hot) variables using pandas.get_dummies, generating indicators such as Geography_France, Geography_Germany, Geography_Spain and Card Type_DIAMOND, Card Type_GOLD, Card Type_PLATINUM, Card Type_SILVER.For example, Table (2) displays the Geography feature after encoding

Table[2]: Encoding the Geography feature by one-hot

Row Number	Geography _France	Geography _Germany	Geography _Spain
1	1	0	0
2	0	0	1
3	1	0	0
4	1	0	0
5	0	0	1

- Gender was encoded using LabelEncoder, mapping the original categories (Female, Male) to integer codes (0, 1). Table(3) displays the gender features after LabelEncoder from raw numbers (5-10) in the dataset.

- Binary attributes (HasCrCard, IsActiveMember) were coded as 0/1 and kept in numeric form.

Table[3]: LabelEncoder of Gender

RowNumber	Gender
5	1
6	1
7	0
8	1
9	1
10	1

A working DataFrame (dforder1) was then created with the target label in the first column, followed by the selected continuous variables, and encoded categorical/binary variables.

3.2.3. Normalizing the Data

To standardize the scale of all predictors and improve the optimization behavior of LR, SVM, and ANN, min–max normalization was applied using MinMaxScaler from sklearn.preprocessing.

The scaler was fitted on the feature matrix and used to transform all predictors into the range [0, 1], yielding a normalized dataset (dforder_copy2). The churn label column was renamed from exit to churn. Tree-based models such as RF are less sensitive to scaling, but applying a uniform normalization simplifies the pipeline and is beneficial for other models.

3.2.4. Feature Selection

To reduce redundancy and focus on the most informative predictors, a feature importance-based selection procedure was used. An ExtraTreesClassifier from sklearn.ensemble was trained on the normalized dataset (dforder_copy3), with all available predictors used to predict churn.

The resulting feature importance scores were extracted and sorted in descending order. The top nine features were selected as the final input set for all the models. These include age, credit score, balance, NumOfProducts, tenure, and a subset of categorical/binary indicators such as Geography, IsActiveMember, HasCrCard, and Gender. This selection simplifies the models and ensures that LR, SVM, RF, and ANN are trained on the same compact and behaviorally meaningful feature subset. Table (4) lists the feature importance ranks in the original dataset.

Table[4]: Features Importance Rank

Features Importance Rank	The Feature	The Value
1	Age	0.178536
2	NumOfProducts	0.136619
3	Balance	0.096037
4	CreditScore	0.089945
5	Point Earned	0.088583
6	EstimatedSalary	0.087466
7	Tenure	0.078719
8	Satisfaction Score	0.064175
9	IsActiveMember	0.035108

As shown in Table (4), age had the largest importance weight ($\sim$0.18), followed by the number of products ($\sim$0.14) and balance ($\sim$0.10), with credit score, reward points, and estimated salary each contributing around 0.08–0.09. Tenure and satisfaction scores also showed meaningful contributions, while active membership, although less influential than others, still added discriminative power. Overall, these results indicate that churn risk in this bank is driven by a combination of demographic (age), portfolio (number of products, card-related rewards), and financial (credit score, balance, income) characteristics, together with engagement indicators (satisfaction and activity status), which aligns with prior findings in banking churn literature.

3.2.5. Handling Class Imbalance (Sampling Strategy)

As the original churn label is heavily skewed ($\approx$ 80% non-churn, 20% churn), this study explicitly examines two training regimes:

1. Scenario A – Imbalanced:
Models were trained on the original imbalanced training data. This baseline reflects the true operating conditions of the banks.

2. Scenario B – SMOTE-balanced:
Models are trained on a synthetically balanced training set obtained via the synthetic minority oversampling technique (SMOTE) implemented by imblearn.over_sampling.SMOTE.
SMOTE generates synthetic minority (churn) instances by interpolating between each churn case and its nearest minority neighbors until the number of churn and non-churn observations in the training set is equal.

In Scenario B, SMOTE is strictly confined to the training portion of the data.

- During cross-validation, SMOTE is placed inside an imblearn pipeline and applied only to the training folds, and never to the validation fold within each split.

- For the final model, SMOTE was fit on the full training set, and 30% of the test set remained untouched and imbalanced.

This design avoids information leakage and allows for a fair and controlled comparison between imbalanced and SMOTE-balanced training for each algorithm.

3.2.6. Data Splitting

After preprocessing and feature selection, the final dataset (dforder_copy3) was split into training and test subsets using train_test_split with a 70/30 ratio, stratified =y to preserve the class proportions, and random_state=42 to ensure reproducibility.

For example, in the logistic regression experiment, the training set contained 5,573 non-churners and 1,427 churners (total 7,000), whereas the test set contained 2,389 non-churners and 611 churners (total 3,000), maintaining the original $\sim$ 4:1 imbalance in both splits. The same stratified split was used for all four algorithms and for both scenarios so that performance differences can be attributed solely to the model and sampling strategy rather than to different random splits.

3.3. Model Implementation and Evaluation

All models were implemented in Python using the Jupyter Notebook, with a modelling pipeline based on scikit-learn and imblearning. A common evaluation function was defined for each algorithm to guarantee a unified experimental procedure across scenarios.

3.3.1. Model Specifications

The four supervised learning algorithms were instantiated as follows:

• LR:
LogisticRegression(solver='liblinear,' random_state=42) embedded in a pipeline with StandardScaler, and in the balanced scenario, SMOTE:

o Imbalanced: StandardScaler → LogisticRegression

o SMOTE-balanced: StandardScaler → SMOTE → LogisticRegression

• SVM:
SVC(kernel='rbf,' C=1.0, gamma='scale,' probability=True, random_state=42) combined with StandardScaler and optional SMOTE

o Imbalanced: StandardScaler → SVC
o SMOTE-balanced: StandardScaler → SMOTE → SVC

• Artificial Neural Network (ANN/MLP):
MLPClassifier(hidden_layer_sizes=(64,32), activation='relu,' solver='adam,' alpha=1e-4, learning_rate='adaptive,' learning_rate_init=1e-3,
max_iter=200,
early_stopping=True,
validation_fraction=0.1,
n_iter_no_change=10,
random_state=42) within:

o Imbalanced: StandardScaler → MLPClassifier
o SMOTE-balanced: StandardScaler → SMOTE → MLPClassifier

• RF:
RandomForestClassifier(n_estimators=100, n_jobs=-1, random_state=42) used with and without SMOTE:

o Imbalanced:
RandomForestClassifier
o SMOTE-balanced: SMOTE → RandomForestClassifier

For algorithms that support probability outputs, a helper function (get_scores) was used to extract either predict_proba (for LR, SVM with probability=True, ANN, and RF) or decision_function, providing continuous churn scores for ROC/PR analysis.

3.3.2. Cross-validation and Scoring

For each algorithm and scenario, the performance on the training set was assessed using Stratified K-Fold cross-validation with 5 folds (StratifiedKFold(n_splits=5, shuffle=True, random_state=42)). The following metrics were computed via cross validation:

- **Accuracy = (TP + TN) / (TP + TN + FP + FN)**

Variable Definitions:
- TP: True Positives — correctly predicted churners
- TN: True Negatives — correctly predicted non-churners
- FP: False Positives — non-churners incorrectly predicted as churners
- FN: False Negatives — churners incorrectly predicted as non-churners

- **Precision (per class)**

Precision_i = TP_i / (TP_i + FP_i)
Variable Definitions:
- TP_i: True Positives for class i
- FP_i: False Positives for class i

- **Precision_macro**

Precision_macro = (1/K) * ∑ Precision_i
Variable Definitions:
- K: Number of classes (K=2 for churn)
- Precision_i: Precision value for each class

- **Recall (per class)**

Recall_i = TP_i / (TP_i + FN_i)
Variable Definitions:
- TP_i: True Positives for class i
- FN_i: False Negatives for class i

- **Recall_macro**,

Recall_macro = (1/K) * ∑ Recall_i
Variable Definitions:
- K: Number of classes
- Recall_i: Recall the values for each class.

- **F1-score (per class)**

F1_i = 2 * (Precision_i * Recall_i) / (Precision_i + Recall_i)
Variable Definitions:
- Precision_i: Precision of class i

- Recall_i: Recall of class i

- **F1_macro,**

F1_macro = (1/K) * ∑ F1_i
Variable Definitions:

- K: Number of classes
- F1_i: F1-score for each class

- **ROC–AUC**

AUC_ROC = ∫ TPR(FPR) d(FPR)
Variable Definitions:

- TPR: TPR = TP / (TP + FN)
- FPR: FPR = FP / (FP + TN)

- **For Logistic Regression, average_precision (PR–AUC)**

AP = ∑ (Recall_n - Recall_(n-1)) * Precision_n
Variable Definitions:

- Recall_n: Recall at threshold n
- Precision_n: Precision at threshold n.

The mean values and standard deviations across folds were reported, providing an estimate of the stability and variability of each model under both imbalanced and SMOTE-balanced training.

3.3.3. Final Test Evaluation and Curve Analysis

After cross-validation, each pipeline (for each algorithm and scenario) was refitted on the full training set and evaluated on the untouched 30% of the test set. The main outputs were:

- **Classification report** (precision, recall, F1-score per class) using classification_report.

- **Confusion matrix** (TP, FP, FN, TN) using confusion_matrix.

- Aggregate metrics such as overall accuracy, TPR (recall), FPR, F1-score, and standard ROC–AUC via roc_auc_score.

For all models, ROC curves were generated using the roc_curve. For Logistic Regression, precision–recall curves and PR–AUC were computed using precision_recall_curve, average_precision_score, and (areunder the curve AUC (). The ROC and PR points were exported in both “wide” and “long” formats for further visualization and analysis.

3.3.4. Statistical Comparison (DeLong and McNemar Tests)

To provide a statistically grounded comparison of the imbalanced versus SMOTE-balanced regimes and between models, this study incorporated two inferential tools:

- A custom implementation of the DeLong method was used to estimate the ROC–AUC, its standard error, and 95% confidence interval, and to perform a paired DeLong test on the same test set. This allows for formal testing of whether the differences in AUC between Scenario A (imbalanced) and Scenario B (SMOTE-balanced) are statistically significant.

- McNemar’s test was applied to paired test predictions using the counts of instances where one model (or scenario) was correct and the other incorrect. This test evaluates whether the difference in the error patterns between the two classifiers is significant beyond random variations.

Finally, the confusion matrix statistics, cross-validation results, DeLong outputs, and McNemar p-values for all four algorithms and both training regimes were consolidated into tables. These tables form the basis for the detailed performance analysis and discussion presented in the results section. Figure (2) summarizes the workflow of the methodology used in this study.

4. RESULTS AND DISCUSSION

4.1. Models Performance

The performance of the four models in this study was evaluated on a stratified 30% hold-out test set using the original 80/20 class distribution and a decision threshold of 0.5. Then, the four models were evaluated when SMOTE oversampling was applied only to the training data to balance the churners and non-churners, while the test set remained imbalanced. In the following subsections, the results of the four models are shown and discussed.

4.1.1. LR:

The LR served as the baseline. On the imbalanced test set, it achieved an accuracy of 0.815 and ROC–AUC of 0.783, but its recall for churners was very low (TPR = 0.218), with a modest F1-score of 0.324, despite a reasonably high precision for churn (PPV = 0.633) and a low false-positive rate (FPR = 0.032). Under SMOTE, LR undergoes a marked change in behavior: churn recall increases dramatically from 0.218 to 0.715, and the F1-score for churn rises to approximately 0.508, but this comes at the cost of lower accuracy (0.718) and a much higher FPR (0.281). For LR, Table (5) shows Cross-Validation (5-fold) – Imbalanced vs. SMOTE, Ta-

Table[5]: Cross-Validation (5-fold) – Imbalanced vs SMOTE

Scenario	Accuracy	Precision (macro)	Recall (macro)	F1 (macro)	ROC-AUC
Imbalanced	0.8070	0.7007	0.5796	0.5901	0.7550
Balanced	0.7039	0.6347	0.6928	0.6370	0.7572

Table[6]: Holdout (30%) – Confusion Matrices and Derived Metrics at Threshold 0.5

Scenario	TP	FP	TN	FN	Holdout Accuracy	Holdout ROC-AUC	TPR (Recall)	FPR
Imbalanced	133	77	2312	478	0.8150	0.783040	0.2177	0.0322
Balanced	437	671	1718	174	0.7183	0.787684	0.7152	0.2809

Table[7]: Derived Metrics at 0.5 (Accuracy, PPV, TPR, TNR, FPR, NPV, F1, Balanced Accuracy, MCC, Youden's J)

Scenario	PPV	TPR	TNR	FPR	NPV	F1	Balanced Acc	MCC	YoudenJ	AUC	Accuracy
Imbalanced	0.6333	0.218	0.968	0.032	0.829	0.324	0.593	0.293	0.185	0.783	0.815
Balanced	0.3944	0.715	0.719	0.281	0.908	0.508	0.717	0.362	0.434	0.788	0.718

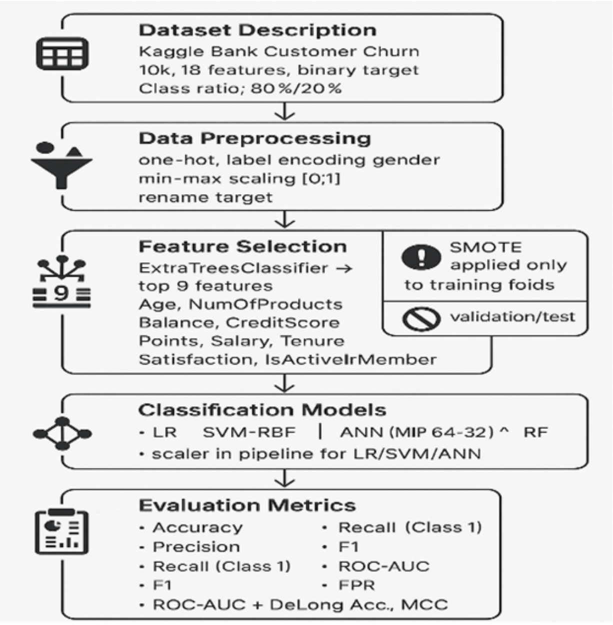


Figure 2. The Methodology Work Flow of this Study

ble (6) shows Holdout (30%) – Confusion Matrices and Derived Metrics at Threshold 0.5, and Table (7) shows the Derived Metrics at 0.5.

The interpretation of tables [(5), (6) and (7)] shows that SMOTE substantially increases recall (from ∼0.218 to ∼0.715), but inflates FPR (from ∼0.032 to ∼0.281) and reduces overall accuracy. Balanced Accuracy and Youden's J improved, indicating better sensitivity, but the precision dropped markedly. LR's linear boundary becomes more permissive under class balancing, which is desirable if capturing churn is paramount, despite more false alarms.

4.1.2. SVM

SVM improved substantially over LR, with a test accuracy of 0.860 and an ROC–AUC of 0.842. Churn recall increased to approximately 0.393, churn precision remained very high (PPV $\approx$ 0.83), and the false-positive rate was extremely low (FPR $\approx$ 0.021). For SVM, SMOTE led to a substantial gain in churn recall (from 0.393 to 0.674) and F1-score (from 0.533 to 0.575), with accuracy remaining relatively high at 0.797 and ROC–AUC around 0.832. For SVM, Table (8) shows Cross-Validation (5-fold) – Imbalanced vs. SMOTE, Table (9) shows Holdout (30%) – Confusion Matrices and Derived Metrics at Threshold 0.5, and Table (10) shows the Derived Metrics at 0.5.

As shown in Tables [(8), (9), and (10)], in the imbalanced setting, SVM maintained a very low FPR (∼0.021) and high precision, but recall was modest. SMOTE shifts the operating point; recall increases (to ∼0.674) with a higher FPR (∼0.172) and lower precision. Youden's J and Balanced Accuracy improved, while overall accuracy declined, consistent with a stricter vs. more permissive decision boundary trade-off.

4.1.3. ANN

The artificial neural network (ANN) model achieved the strongest performance under an imbalance. The ANN attained an accuracy of 0.864, ROC–AUC of 0.858, churn recall of 0.504, and F1-score of 0.602, with an FPR of approximately 0.044 and balanced accuracy of 0.730. However, FPR increased from 0.021 to approximately 0.172, reflecting a shift towards more aggressive churn detection. The ANN showed a similar pattern: recall increased to 0.609 and F1-score to 0.548, but accuracy dropped to 0.795 and AUC to 0.815, with FPR increasing to 0.157, suggesting some overfitting to synthetic minority examples. For ANN, Table (11) shows Cross-Validation (5-fold) – Imbalanced vs. SMOTE, Table (12) shows Holdout (30%) – Confusion Matrices and Derived Metrics at Threshold 0.5, and Table (13) shows the Derived Metrics at 0.5.

It is clear from Tables [(11), (12), and (13)] that the ANN performs strongly under imbalance with a good AUC and balanced metrics. After SMOTE, recall increased, but AUC and accuracy decreased, suggesting mild overfitting to synthetic minority samples and increased false positives. Threshold tuning or class weights may recover

Table[8]: Cross-Validation (5-fold) – Imbalanced vs SMOTE

Scenario	Accuracy	Precision (macro)	Recall (macro)	F1 (macro)	ROC-AUC
Imbalanced	0.8509	0.8359	0.6637	0.6994	0.8088
Balanced	0.7846	0.6860	0.7224	0.6991	0.8012

Table[9]: Holdout (30%) – Confusion Matrices and Derived Metrics at Threshold 0.5

Scenario	TP	FP	TN	FN	Holdout Accuracy	Holdout ROC-AUC	TPR (Recall)	FPR
Imbalanced	240	49	2340	371	0.8600	0.841735	0.3928	0.0205
Balanced	412	410	1979	199	0.7970	0.832136	0.6743	0.1716

Table[10]: Derived Metrics at 0.5 (Accuracy, PPV, TPR, TNR, FPR, NPV, F1, Balanced Accuracy, MCC, Youden's J)

Scenario	PPV	TPR	TNR	FPR	NPV	F1	BalancedAcc	MCC	YoudenJ	AUC	Accuracy
Imbalanced	0.830	0.393	0.10	0.021	0.863	0.533	0.686	0.508	0.372	0.84174	0.860
Balanced	0.501	0.674	0.828	0.172	0.909	0.575	0.751	0.454	0.503	0.83214	0.797

Table[11]: Cross-Validation (5-fold) – Imbalanced vs SMOTE

Scenario	Accuracy	Precision (macro)	Recall (macro)	F1 (macro)	ROC-AUC
Imbalanced	0.8494	0.7974	0.6888	0.7202	0.8296
Balanced	0.7851	0.6825	0.7097	0.6927	0.7885

Table[12]: Holdout (30%) – Confusion Matrices and Derived Metrics at Threshold 0.5

Scenario	TP	FP	TN	FN	Holdout Accuracy	Holdout ROC-AUC	TPR (Recall)	FPR
Imbalanced	308	104	2285	303	0.8643	0.857637	0.5041	0.0435
Balanced	372	375	2014	239	0.7953	0.814716	0.6088	0.1570

Table[13]: Derived Metrics at 0.5 (Accuracy, PPV, TPR, TNR, FPR, NPV, F1, Balanced Accuracy, MCC, Youden's J)

Scenario	PPV	TPR	TNR	FPR	NPV	F1	BalancedAcc	MCC	YoudenJ	AUC	Accuracy
Imbalanced	0.748	0.504	0.957	0.044	0.883	0.602	0.730	0.539	0.461	0.85764	0.8643
Balanced	0.498	0.609	0.843	0.157	0.894	0.548	0.726	0.421	0.452	0.81472	0.795

Table[14]: Cross-Validation (5-fold) – Imbalanced vs SMOTE

Scenario	Accuracy	Precision (macro)	Recall (macro)	F1 (macro)	ROC-AUC
Imbalanced	0.8589	0.8294	0.6953	0.7318	0.8369
Balanced	0.8417	0.7580	0.7325	0.7437	0.8406

the balance without relying on oversampling.

4.1.4. RF

RF provided the most balanced performance, combining high discriminative ability (AUC ≈ 0.86–0.87) with substantially improved churn recall compared to LR and SVM, while still keeping false positives at acceptable levels. The RF remains the most robust model under SMOTE. Its accuracy is 0.856 and ROC–AUC is 0.872, very close to its imbalanced training performance, while churn recall improved from 0.484 to 0.619 and F1-score from 0.598 to 0.636. For RF, Table (14) shows Cross-Validation (5-fold) – Imbalanced vs. SMOTE, Table (15) shows Holdout (30%) – Confusion Matrices and Derived Metrics at Threshold 0.5, and Table (16) shows the Derived Metrics at 0.5.

The results in Tables [(14), (15), and (16)] show the strongest stability according to RF. With SMOTE, recall and F1 improved, whereas accuracy and AUC remained high. FPR increased moderately, but MCC and Balanced Accuracy remained competitive, making RF a robust choice across both scenarios.

Overall, SMOTE consistently boosts the churn recall and F1-scores for all models, but its impact on AUC and accuracy is mixed. For LR and SVM, oversampling is particularly helpful in recovering the minority class, albeit with a substantial increase in false positives. For ANN, the gain in recall was offset by the reduced AUC, hinting at mild overfitting to the synthetic samples. RF, in contrast, benefits the most, with improved recall and F1 while maintaining high AUC and accuracy, making it the best all-round choice in the balanced training scenario as

Table[15]: Holdout (30%) – Confusion Matrices and Derived Metrics at Threshold 0.5

Scenario	TP	FP	TN	FN	Holdout Accuracy	Holdout ROC-AUC	TPR (Recall)	FPR
Imbalanced	296	83	2306	315	0.8673	0.869603	0.4845	0.0347
Balanced	378	199	2190	233	0.8560	0.871556	0.6187	0.0833

Table[16]: Derived Metrics at 0.5 (Accuracy, PPV, TPR, TNR, FPR, NPV, F1, Balanced Accuracy, MCC, Youden's J)

Scenario	PPV	TPR	TNR	FPR	NPV	F1	BalancedAcc	MCC	YoudenJ	AUC	Accuracy
Imbalanced	0.781	0.485	0.965	0.035	0.880	0.598	0.725	0.545	0.450	0.870	0.867
Balanced	0.655	0.619	0.917	0.083	0.904	0.636	0.768	0.547	0.535	0.872	0.856

Table[17]: The Summary Table

Model	Scenario	AUC	Accuracy	TPR	FPR	PPV	F1	BalancedAcc	MCC	YoudenJ
LR	Imbalanced	0.783	0.815	0.218	0.032	0.633	0.324	0.593	0.293	0.185
LR	Balanced	0.788	0.718	0.715	0.281	0.394	0.508	0.717	0.362	0.434
SVM	Imbalanced	0.842	0.860	0.393	0.021	0.830	0.533	0.686	0.508	0.372
SVM	Balanced	0.832	0.797	0.674	0.172	0.501	0.575	0.751	0.454	0.503
ANN	Imbalanced	0.858	0.864	0.505	0.043	0.748	0.602	0.730	0.539	0.461
ANN	Balanced	0.815	0.795	0.609	0.157	0.498	0.548	0.726	0.421	0.452
RF	Imbalanced	0.870	0.867	0.484	0.035	0.781	0.598	0.725	0.545	0.450
RF	Balanced	0.872	0.856	0.619	0.083	0.655	0.636	0.768	0.547	0.535

well. Table (17) provides a comprehensive and summarized comparison of all important results for all models in the case of unbalanced and balanced datasets.

Key takeaways: RF provides the most stable high AUC in both settings; SVM (imbalanced) is optimal when keeping FPR very low; LR (SMOTE) is suitable when maximizing recall is crucial; ANN is strong without SMOTE, but can lose AUC after oversampling.

4.2. Integrated Results and Comparison with Previous Studies

The detailed and summary Table (17) provides a multimetric view of how each model behaves under imbalanced and SMOTE-balanced training. On the original imbalanced test set, RF and ANN achieved the strongest overall performance: RF attained the highest accuracy (approximately 0.87) and ROC–AUC (≈0.87) with a favorable trade-off between churn recall and a low false-positive rate, whereas ANN yielded the highest recall for churners among the four models, with only a modest increase in FPR. SVM performs particularly well when minimizing false alarms is the priority, combining high precision and the lowest FPR in an imbalanced setting, although its churn recall remains moderate. LR forms a useful linear baseline; however, under imbalance, it suffers from very low recall for churners despite achieving reasonable accuracy and AUC, illustrating that accuracy alone can be misleading in this context.

Applying SMOTE to balance the training data systematically increases the churn-class recall and F1-scores across all models, but the extent and side effects differ. LR with SMOTE delivers the highest recall for churners (approximately 0.72) and a clear gain in balanced accuracy, but this is accompanied by a substantial rise in FPR and a decrease in overall accuracy, making it most suitable when the cost of missing a churner is much higher than the cost of targeting a non-churner. SVM and ANN both benefit from oversampling in terms of recall and F1, yet their ROC–AUC and accuracy decline somewhat, suggesting mild overfitting to synthetic minority samples and a shift towards more permissive decision boundaries. RF again proved to be the most robust: under SMOTE, it maintained high accuracy and the best ROC–AUC (≈0.87) while raising churn recall to the 0.62 range of and maintaining FPR at a moderate level. The associated balanced accuracy and MCC remained among the highest, confirming that RF is the most stable choice across both class distributions.

4.3. Cross-scenario comparison and practical implications

The global comparison of all models and scenarios in Table (17) highlights clear trade-offs that are important for banking practices.

- When the accuracy and overall discriminative power are prioritized, RF is the preferred model. It consistently delivered the highest or near-highest accuracy (≈0.867 imbalanced, 0.856 SMOTE) and ROC–AUC (≈0.87 in both cases), along with strong F1 and MCC values, indicating reliable performance in ranking and classifying churners versus non-churners.

- When minimizing false positives is critical (for example, when retention campaigns are expensive), SVM on imbalanced data is attractive: it keeps FPR extremely low (∼0.02) while still detecting a non-trivial subset of churners and maintaining high precision.

- When maximizing churn recall is the main objective, LR with SMOTE achieves the highest recall (∼0.72) and strong balanced accuracy at the expense of many more false positives and lower overall accuracy. This configuration is suitable for early warning or prescreening purposes, where a human or downstream system can further filter flagged customers.

- ANN performs very well under imbalance, with a strong AUC and balanced accuracy, but loses some AUC and accuracy after oversampling. For this dataset, its best use is in the original imbalanced setting, possibly combined with threshold tuning or class weights, instead of heavy oversampling.

From an operational perspective, these results highlight that no single model is universally optimal; instead, the preferred configuration depends on business priorities and cost structures. If a bank wishes to minimize outreach to non-churners (e.g., to control campaign costs or avoid customer irritation), SVM on imbalanced data offers a very conservative option with low FPR and high precision. If the main objective is to capture as many churners as possible, LR or ANN with SMOTE provides a substantially higher recall at the cost of more false positives. For most balanced scenarios in which both accuracy and minority-class recall matter, RF, especially with SMOTE applied in the training folds only, offers the most attractive compromise, as reflected in its combination of high AUC, solid F1, and strong balanced accuracy.

When compared with recent bank-churn studies that employ similar Kaggle-based or proprietary bank datasets and classical machine-learning models—for example, [2], [3], and [5]—the performance of the RF and ANN models in this work is at the upper end of what is typically reported, where test accuracies often fall in the high-70% to mid-80% range and ROC–AUC values are commonly in the mid-0.80s. Consistent with these studies, our results confirm the strong competitiveness of ensemble-based methods such as RF, while also showing that oversampling strategies such as SMOTE primarily boost minority-class recall rather than accuracy or AUC alone. Moreover, many prior works emphasize overall accuracy and, in some cases, ROC–AUC but do not systematically report minority-class (churn) recall, precision–recall AUC, or statistical tests that quantify uncertainty and compare scenarios. In contrast, the present study provides a fully imbalance-aware, multi-metric evaluation that includes DeLong confidence intervals for AUC, PR–AUC, and McNemar tests for paired predictions under imbalanced and SMOTE-balanced training. Together with the feature importance analysis based on ExtraTreesClassifier, these contributions position the proposed framework as a rigorous and practically informative benchmark for future work on customer churn prediction in the retail banking sector. Table (18) summarizes the key findings of some of the latest related studies compared to this study.

5. CONCLUSION

This study addresses the problem of predicting customer churn in retail banking using a publicly available benchmark dataset from Kaggle, which contains 10,000 customers described by 18 demographic, financial, and behavioral features. The proposed framework explicitly focuses on class imbalance, which is intrinsic to churn problems, and compares four widely used supervised machine-learning models—logistic regression (LR), support vector machine (SVM), multilayer perceptron artificial neural network (ANN), and random forest (RF)—under both imbalanced and SMOTE-balanced training scenarios. A transparent experimental design was adopted, combining a stratified 70/30 train–test split, 5-fold stratified cross-validation, and careful separation of resampling and evaluation steps to avoid information leakage.

Using an ExtraTreesClassifier for feature importance analysis, this study found that churn behavior in this bank is primarily driven by a compact set of variables: Age, Number of Products, Balance, Credit Score, Points Earned, Estimated Salary, Tenure, Satisfaction Score, and IsActiveMember. These predictors jointly capture customer lifecycle (age, tenure), engagement, product portfolio (number of products, points earned, active membership, satisfaction), and financial strength (credit score, balance, income), reinforcing the view that churn is a multifactor phenomenon rather than a purely transactional or demographic one.

On the original imbalanced data, the RF and ANN delivered the most balanced performance. RF achieved the highest overall accuracy (approximately 0.867) and ROC–AUC (≈0.87), with a good compromise between churn recall, precision, and false-positive rate. ANN showed similarly strong discriminative ability, with higher churn recall than SVM and LR and acceptable false-positive rates. SVM performed well when the priority was to minimize false positives, achieving a very low FPR and high precision for churners, but at the cost of missing more at-risk customers. Although LR served as a simple baseline, despite reasonable accuracy and AUC, it exhibited very low recall for churners under imbalance, confirming that linear decision boundaries alone are insufficient for reliably detecting minority classes in this setting.

When SMOTE oversampling was applied only to the training data, all four models exhibited higher recall and F1-scores for churners on the held-out test set but with different trade-offs. LR with SMOTE achieved

Table[18]: Key Findings of Related Studies Vs. This Study

Study (year)	Bank data & imbalance	Algorithms	Imbalance handling	Main metrics	Key findings of related studies vs. this study
[28], 2024	Public bank churn dataset; moderately imbalanced	LR, RF	Basic preprocessing, no advanced resampling	Accuracy, precision, recall	RF clearly outperforms LR in accuracy and recall; similar to our finding that tree-based models dominate LR on bank churn.
[2], 2024	Bank churn data (Kaggle-style); imbalanced	LR, DT, GBDT, XGBoost, CatBoost, LightGBM	Class weighting and tuning; no detailed SMOTE pipeline	Accuracy, precision, recall, ROC-AUC	Gradient-boosting methods (especially LightGBM/XGBoost) outperform linear models; reinforces our result that more flexible nonlinear models (RF/ANN/SVM) outperform LR.
[5], 2024	Large real bank dataset; strong imbalance	LR, k-NN, SVM, DT, RF, Bagging, AdaBoost, GBM, XGBoost, Extra Trees	Several resampling strategies (over/under-sampling)	Accuracy, F1, ROC-AUC	Ensemble tree models outperform classical ML; aligns with our finding that RF is the top performer among the four baseline models.
[21], 2024	European bank; imbalanced churn	LR, RF, XGBoost, LightGBM	SMOTE-type resampling + probability calibration	ROC-AUC, Brier score, calibration curves	Show that resampling + calibration improves both AUC and probability quality; our work similarly shows that SMOTE improves minority-class recall and F1, especially for RF and ANN.
[10], 2025	Bank credit-card customers; highly imbalanced	RF, GBDT, Extra Trees, AdaBoost, XGBoost, CatBoost	Random oversampling, SMOTE, Borderline-SMOTE, ADASYN	Accuracy, precision, recall, F1, AUC	XGBoost achieves ∼0.97 on all metrics; combines SMOTE-type methods with interpretability (SHAP, causal inference). Our study is closer to baseline models but similarly confirms the benefit of resampling and multiple metrics.
[29], 2025	Bank churn; imbalanced	RF, XGBoost	SMOTE and SMOTE-ENN	Accuracy, precision, recall, F1	RF and XGBoost around 86% accuracy with more balanced metrics after resampling. Our results likewise show that RF maintains strong performance in both imbalanced and SMOTE-balanced scenarios.
[22], 2025	Bank churn; strongly imbalanced	LR, DT, RF, k-NN, XGBoost, SVM	SMOTE-Tomek	Accuracy, precision, recall, F1	RF and XGBoost top overall; resampling improves minority-class recall. This parallels our comparison of imbalanced vs SMOTE-balanced data, where SMOTE yields better sensitivity for churners.
[30], 2025	Cross-sector (incl. banking)	Many ML and DL models	Various resampling & cost-sensitive methods	Mainly AUC, F1, and accuracy	Confirms that ensembles and hybrid DL models dominate recent literature; our study contributes by providing a rigorous multi-metric baseline comparison for four core algorithms in the banking context.

the highest recall for churners (≈0.72) and improved balanced accuracy; however, its overall accuracy dropped (≈0.718) and the false-positive rate increased sharply, making it suitable for applications where missing a churner is much more costly than incorrectly flagging a non-churner. SVM and ANN both benefited from

SMOTE in terms of recall and F1 but experienced moderate increases in FPR and some degradation in AUC, indicating partial overfitting to the synthetic minority samples. RF again proved to be the most robust: it retained high accuracy and ROC–AUC ($\approx$0.87), while significantly improving churn recall (to $\approx$0.62) and F1, with a comparatively modest rise in FPR. Across both scenarios, RF yielded the strongest values for balanced accuracy and MCC, indicating consistently reliable discrimination between churners and non-churners.

A key contribution of this study is the use of a multimetric imbalance-aware evaluation. Rather than relying on accuracy alone, the study jointly examined churn-class recall, precision, F1-score, ROC–AUC (with DeLong confidence intervals), PR–AUC, false-positive rate, balanced accuracy, MCC, and Youden's J, as well as statistical tests (DeLong and McNemar) to compare scenarios. This richer evaluation revealed that SMOTE does not uniformly improve all metrics; it is particularly effective at boosting minority-class recall, but its effect on the AUC and accuracy depends on the model. For practitioners, the results highlight that model choice and imbalance treatment must be aligned with business priorities: RF for robust, all-round performance, SVM on imbalanced data when false positives are costly, and LR with SMOTE when maximizing churn detection is more important than precision.

From a managerial perspective, the findings suggest that banks can move beyond simplistic, rule-based segmentation, and instead deploy machine-learning models that exploit a small, interpretable set of features, such as age, product holdings, balance, credit quality, loyalty points, satisfaction, and activity status, to identify customers at risk of churn. By selecting an appropriate model–scenario configuration and threshold according to their cost structure and risk tolerance–banks can design more targeted and cost-effective retention campaigns.

This study had some limitations that suggest directions for future research. First, it relies on a single static dataset from one bank. Applying the same framework to multi-bank or longitudinal data with richer behavioral histories would allow validation of the conclusions in broader contexts. Second, although SMOTE was used as a baseline oversampling method, alternative imbalance-handling strategies, such as SMOTE variants, class-weighted losses, cost-sensitive learning, and focal loss, could be explored and compared. Third, although this study focused on classical supervised models, future research could investigate calibrated probability estimates, threshold optimization under explicit cost functions, or hybrid approaches that combine RF or ANN with explainability techniques (e.g., SHAP) to provide more interpretable churn drivers to domain experts.

Despite these limitations, the proposed framework demonstrates that a carefully designed, imbalance-aware, and multimetric evaluation on a widely used public bank-churn dataset can yield robust and actionable insights. In particular, it confirms the strong performance and stability of RF across both imbalanced and balanced settings, clarifies the circumstances under which SMOTE is beneficial, and shows how different model choices translate into concrete trade-offs between capturing churners and avoiding unnecessary intervention.

REFERENCES

[1] I. N. M. Adiputra, P. Wanchai, and P.-C. Lin, "Optimized customer churn prediction using tabular generative adversarial network (GAN)-based hybrid sampling method and cost-sensitive learning," *PeerJ Comput. Sci.*, vol. 11, e2949, 2025.

[2] M. Rahman and V. Kumar, "Machine learning based customer churn prediction in banking," in *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, IEEE, 2020, pp. 1196–1201.

[3] M. H. Seid and M. M. Woldeyohannis, "Customer churn prediction using machine learning: commercial bank of Ethiopia," in *2022 International Conference on Information and Communication Technology for Development for Africa (ICT4DA)*, IEEE, 2022, pp. 1–6.

[4] A. M. Molla, M. A. Yimer, and Y. D. Woldehana, "Customer Churn Prediction Using Machine Learning Techniques: Awash Bank Wolaita Sodo Region," *J. Emerg. Comput. Technol.*, vol. 5, no. 1, pp. 36–46, 2025.

[5] R. Ashraf, "Bank customer churn prediction using machine learning framework," *J Appl Financ Bank*, vol. 14, no. 4, pp. 65–109, 2024.

[6] M. A. Hambali and I. Andrew, "Bank customer churn prediction using SMOTE: A comparative analysis," *Qeios*, 2024.

[7] B. Thenmozhi, C. Jeyabharathi, and S. Vimala, "Customer Churn Prediction in Banking Sectors Using a Hyperparameter-Tuned Deep Learning Model," 2024.

[8] P. P. Singh, F. I. Anik, R. Senapati, A. Sinha, N. Sakib, and E. Hossain, "Investigating customer churn in banking: A machine learning approach and visualization app for data science and management," *Data Sci. Manag.*, vol. 7, no. 1, pp. 7–16, 2024.

[9] J. Nguyen and M. Dupuis, "Closing the Feedback Loop Between UX Design, Software Development, Security Engineering, and Operations," in *Proceedings of the 20th Annual SIG Conference on Information Technology Education*, Tacoma, WA, USA, 2019. DOI: 10.1145/3349266.3351420.

[10] Y. Li and K. Yan, "Prediction of bank credit customers churn based on machine learning and interpretability analysis," *Data Sci. Finance Econ.*, vol. 5, no. 1, pp. 19–34, 2025.

[11] T.-T. Luong, V.-G. Luong, A. H. T. Tran, and T. M. Nguyen, "Application of Machine Learning Techniques for Customer Churn Prediction in the Banking Sector," *Interdiscip. J. Information, Knowledge, Manag.*, vol. 20, p. 009, 2025.

[12] K. Peng, Y. Peng, and W. Li, "Research on customer churn prediction and model interpretability analysis," *Plos one*, vol. 18, no. 12, e0289724, 2023.

[13] L. T. Tam, L. G. Vi, and N. M. Tuan, "Comparison of Methods for Handling Imbalanced Data in Customer Churn Prediction with Feature Selection Using SHAP and mRMR Frameworks," *Cybern. Inf. Technol.*, vol. 25, no. 3, 2025.

[14] A. A.-M. Ragab and E. El Behiry, “Bank Customer Churn Prediction Using Machine Learning,” *J. Eng. Adv. Technol. for Sustain. Appl.*, vol. 1, no. 3, pp. 1–8, 2025.

[15] E. E. Diri, G. O. Diri, K. N. Elliot, N. H. James, R. C. Owhonda, and L. G. Nbaakee, “Behavioural Analysis and Churn Forecasting in Retail Banking Using Machine Learning Models,” 2025.

[16] S. Li and Z. Shen, “Explainable customer churn prediction model based on deep learning,” in *Proceedings of the 2024 3rd Asia Conference on Algorithms, Computing and Machine Learning*, 2024, pp. 282–287.

[17] Enny et al., “Credit Card Customer Churn Prediction With BinaryLogisticRegression,” *J. CendekiaIlmiah*, vol. 4, p. 9, 2025.

[18] R. Suguna, J. Suriya Prakash, H. Aditya Pai, T. Mahesh, V. Vinoth Kumar, and T. E. Yimer, “Mitigating class imbalance in churn prediction with ensemble methods and SMOTE,” *Sci. Reports*, vol. 15, no. 1, p. 16 256, 2025.

[19] J. B. Brito et al., “A framework to improve churn prediction performance in retail banking,” *Financial Innov.*, vol. 10, no. 1, p. 17, 2024.

[20] H. Tran, N. Le, and V.-H. Nguyen, “CUSTOMER CHURN PREDICTION IN THE BANKING SECTOR USING MACHINE LEARNING-BASED CLASSIFICATION MODELS,” *Interdiscip. J. Information, Knowl. Manag.*, vol. 18, 2023.

[21] A.-G. Văduva, S.-V. Oprea, A.-M. Niculae, A. Bâra, and A.-I. Andreescu, “Improving churn detection in the banking sector: a machine learning approach with probability calibration techniques,” *Electronics*, vol. 13, no. 22, p. 4527, 2024.

[22] O. Berrada Chakour, A. Maizate, A. Ettaoufik, and K. Aissaoui, “Enhancing Bank Customer Churn Prediction from an Imbalanced Dataset Using Machine Learning and SMOTE-Tomek Resampling,” in *International Conference on Connected Objects and Artificial Intelligence*, Springer, 2025, pp. 852–860.

[23] A. Zia, S. Khan, and H. Gul, “Machine learning techniques for bank customer churn prediction,” *Appl. Sci.*, 2022.

[24] A. Amin, S. Rahman, and A. Alharbi, “Churn prediction in the banking sector: A comparative machine learning study with resampling,” *SN Appl. Sci.*, 2022.

[25] J. B. G. Brito, G. B. Bucco, R. Heldt, J. L. Becker, C. S. Silveira, and M. J. Anzanello, “A framework to improve churn prediction performance in retail banking,” *Financial Innov.*, 2024.

[26] A. El-Hanafi and M. Mohamed, “Machine learning for bank customer churn prediction with feature engineering,” *J. Financial Serv. Anal.*, 2022.

[27] M. Rahman and M. Islam, “Customer churn prediction using hybrid oversampling and machine learning techniques in banking,” *Expert Syst. with Appl.*, 2023.

[28] S. Du, “Bank Churn Prediction Using Random Forest and Logistic Regression,” in *Proceedings of the 2024 2nd International Conference on Finance, Trade and Business Management (FTBM 2024)*, vol. 304, Springer Nature, 2024, p. 4.

[29] R. Andespa, K. Sadik, C. Suhaeni, and A. M. Soleh, “EVALUATING RANDOM FOREST AND XGBOOST FOR BANK CUSTOMER CHURN PREDICTION ON IMBALANCED DATA USING SMOTE AND SMOTE-ENN,” *MEDIA STATISTIKA*, vol. 18, no. 1, pp. 25–36, 2025.

[30] M. Imani, M. Joudaki, A. Beikmohammadi, and H. R. Arabnia, “Customer churn prediction: A systematic review of recent advances, trends, and challenges in machine learning and deep learning,” *Mach. Learn. Knowl. Extr.*, vol. 7, no. 3, p. 105, 2025.