

International Journal Research Publication Analysis

Page: 01-14

PREDICTING BANK CUSTOMER CHURN USING A RANDOM FOREST CLASSIFIER WITH ENGINEERED RISK FEATURES

*Amula Venkatesh, Dr. B. Indira

Department of Master of Computer Applications (MCA), Chaitanya Bharathi Institute of Technology (CBIT), Hyderabad – 500075, India.

Article Received: 05 September 2026

Article Revised: 25 September 2026

Published on: 15 October 2026

*Corresponding Author: Amula Venkatesh

Department of Master of Computer Applications (MCA), Chaitanya Bharathi Institute of Technology (CBIT), Hyderabad – 500075, India.

DOI: <https://doi-doi.org/101555/ijrpa.9667>

ABSTRACT

Customer attrition is a direct threat to a retail bank's revenue and lifetime-value economics, since acquiring a replacement customer is markedly more expensive than retaining an existing one. This study builds a supervised machine-learning system to predict individual customer churn from a structured banking dataset (European_Bank.csv, 10,000 customer records) and deploys it as an interactive Streamlit decision-support dashboard. Ten raw attributes — credit score, geography, gender, age, tenure, account balance, number of products, credit-card ownership, active-membership status and estimated salary — are supplemented with four engineered risk features (Balance-to-Salary Ratio, Product Density, Engagement-Product interaction and Age $\times$ Tenure) and one-hot encoding of geography and gender, giving a 15-feature model matrix. Features are standardised with a persisted StandardScaler and a Random Forest classifier is trained on an 80/20 stratified split. On the held-out test set the model achieves Accuracy = 91.4%, Precision = 89.2%, Recall = 87.8%, F1-Score = 88.5% and ROC-AUC = 0.930, outperforming Logistic Regression and Decision Tree baselines and approaching a heavier XGBoost ensemble while remaining fast to train and easy to explain with SHAP. The trained model, encoder and scaler artefacts are served through a browser-based dashboard that reports churn probability, a colour-coded risk gauge, SHAP-based explanations and tailored retention recommendations, turning a notebook result into a tool that bank relationship managers can act on directly.

KEYWORDS: Machine Learning, Random Forest, Customer Churn Prediction, Banking Analytics, Feature Engineering, SHAP Explainability, Streamlit, Business Analytics,

Classification.

I. INTRODUCTION

Retail banks compete in a market where switching providers has become progressively easier, and even a modest annual attrition rate translates into a large recurring revenue loss when compounded across a multi-million-customer book. Because acquiring a new customer typically costs several times more than retaining an existing one, the ability to flag a customer as “at risk” before they close their account or become dormant is of direct commercial value.

This paper formulates bank customer churn as a binary classification problem — 0 = Retained, 1 = Churned (Exited)

— and evaluates a Random Forest classifier trained on demographic, account and engagement attributes. Random Forest is selected because it handles a mixture of numeric and one-hot encoded categorical inputs without extensive tuning, resists overfitting on a moderate-sized tabular dataset through bagging, and yields a built-in feature-importance ranking that can be cross-checked with SHAP values for stakeholder-facing explanations.

Beyond model fitting, the project's contribution is an end-to-end deliverable: a cleaned dataset, four hand-engineered interaction features, a persisted scaler and classifier, and a Streamlit web application that a non-technical relationship manager can use to score a customer, see a probability gauge, read a plain-language risk tier and download a retention report — without opening a notebook or writing code.

Objectives

- Build and evaluate a supervised Random Forest model for individual-customer churn prediction.
- Engineer interaction features (balance-to-salary ratio, product density, engagement, age—tenure) to sharpen the signal available to the model.
- Develop an interactive Streamlit dashboard supporting single-customer scoring, dataset exploration and model-performance reporting.
- Translate churn probability into risk tiers and actionable retention recommendations for bank staff.

II. LITERATURE REVIEW

Xie et al. [1] proposed an improved balanced random forest (IBRF) for bank churn prediction that combines sampling and cost-sensitive learning to counter class imbalance, and reported that it outperformed artificial neural networks, decision trees and cost-weighted SVMs on a

real bank customer dataset, establishing random forests as a strong family of learners for this problem even before class-balance corrections are applied.

Kiptoo and co-authors [2] compared Gradient Boosting, Random Forest, AdaBoost and Decision Trees for bank client segmentation and churn scoring, and found Random Forest to be the best-performing algorithm among the four, motivating its selection as the deployed model in similar mini-project settings.

A comparative study using the same publicly known Kaggle bank-churn schema evaluated KNN, SVM, Decision Tree and Random Forest classifiers after feature selection, and reported that ensemble tree methods separated churners from retained customers more reliably than distance- and margin-based classifiers on this feature set [3].

Tran [4] combined Random Forest with SMOTE and ADASYN resampling for banking churn prediction and evaluated the result across accuracy, recall, precision, specificity, F1 and ROC-AUC, showing that resampling narrows the gap between majority- and minority-class recall

without materially hurting precision — a consideration reflected in this study's stratified split. Verma [5] evaluated statistical and machine-learning models for savings-bank churn under class imbalance and under-sampling, and identified Random Forest as the strongest model, reporting roughly 78% accuracy — noticeably lower than the feature-engineered pipeline in the present study, consistent with the added interaction features carrying incremental signal.

A 2025 comparative study contrasted Logistic Regression and Random Forest on a bank-churn dataset with demographic and transactional attributes and found Random Forest to dominate on every reported metric, reinforcing the choice of ensemble trees over a linear baseline for this task [6].

Most recently, a benchmark of Decision Tree, Random Forest, Multi-Layer Perceptron, SVM and XGBoost on a bank customer dataset reported Random Forest achieving the highest hold-out accuracy among the five algorithms compared, ahead of the neural-network baseline [7].

III. METHODOLOGY

3.1 Research Framework

The project follows a structured pipeline: (i) data collection, (ii) data cleaning, (iii) exploratory data analysis, (iv) feature engineering, (v) feature scaling, (vi) model training with Random Forest, (vii) evaluation and SHAP-based interpretation, and (viii) deployment as an interactive Streamlit dashboard.

3.2 Dataset Description

The working file, `European_Bank.csv`, holds 10,000 customer-level records for a European retail bank. Each row captures:

- Demographics: Age, Gender, Geography (France, Germany, Spain).
- Account: Credit Score, Tenure (years with the bank), Balance, Number of Products, Has Credit Card, Is Active Member.
- Commercial: Estimated Salary.
- Target: Exited (1 = churned, 0 = retained).

Table I: Sample Dataset Records. (5 of 10,000 rows)

Cust.ID	Age	Geography	Balance	Exited
15634602	42	France	0.00	1
15647311	41	Spain	83,807.86	0
15619304	42	France	159,660.80	1
15701354	39	Germany	125,510.82	0
15737888	43	Spain	112,542.58	0

3.3 Feature Engineering

Four interaction features are derived to expose non-linear relationships that raw columns leave implicit:

- Balance-Salary Ratio = $\text{Balance} / (\text{Estimated Salary} + 1)$ — a proxy for financial dependence on the account.
- Product Density = $\text{Number of Products} / (\text{Tenure} + 1)$ — product acquisition pace relative to relationship length.
- Engagement-Product = $\text{Is Active Member} \times \text{Number of Products}$ — captures whether product ownership is backed by real engagement.
- Age $\times$ Tenure — an interaction term for life-stage-adjusted relationship length.

Geography and Gender are one-hot encoded (Geography_Germany, Geography_Spain, with France as the reference level; Gender_Male, with Female as the reference level), giving a final model matrix of 15 features consistent with the deployed application.

3.4 Data Pre-processing

All numeric and engineered features are standardised with a fitted StandardScaler (zero mean, unit variance) before being passed to the classifier; the fitted scaler is persisted (Scaler.pkl) so that the exact training-time transformation is reproduced at inference time inside the dashboard.

3.5 Train / Test Split

The 10,000-record dataset is split 80:20 — 8,000 records for training and 2,000 for testing — stratified on the Exited flag so that the 62.4% retained / 37.6% churned class ratio is preserved in both partitions.

IV. RANDOM FOREST CLASSIFICATION MODEL

4.1 Model Formulation

A Random Forest is an ensemble of B decision trees, each trained on a bootstrap sample of the training data with a random subset of features considered at every split. The class prediction for customer i is obtained by majority vote across trees:

$$\hat{y}_i = \text{mode} \{ T_b(x_i) : b = 1, \dots, B \} \quad (1)$$

and the predicted churn probability is the mean of the per-tree class-1 vote share:

$$\hat{p}_i = (1/B) \sum_{b=1}^B 1[T_b(x_i) = 1] \quad (2)$$

Each tree T_b is grown by recursively splitting on the feature and threshold that most reduce the Gini impurity

$$G(t) = 1 - \sum_k p_k(t)^2 \quad (3)$$

where $p_k(t)$ is the proportion of class k samples at node t . Bagging plus random feature subsampling decorrelates the trees, which reduces variance relative to a single decision tree without materially increasing bias — the property that makes Random Forest robust on a moderate-sized, mixed-type tabular dataset such as this one.

4.2 Problem Type and Target

The task is binary classification: 0 = Retained, 1 = Churned. The predicted probability is thresholded at 0.5 for the default label, while the dashboard additionally reports the continuous probability and a three-tier risk band (Low < 30%, Medium 30–60%, High ≥ 60%) so that staff are not limited to a single cut-off.

4.3 Hyperparameters

Table II: Random Forest Hyperparameters.

Hyperparameter	Value
n_estimators	300
max_depth	12
min_samples_leaf	5
criterion	gini
class_weight	balanced
random_state	42

4.4 Evaluation Metrics

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (4)$$

$$\text{Precision} = TP / (TP + FP) \quad (5)$$

$$\text{Recall} = TP / (TP + FN) \quad (6)$$

$$F1 = 2 \cdot (\text{Precision} \cdot \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (7)$$

ROC-AUC summarises discrimination across all probability thresholds rather than the single default cut-off of 0.5, and is reported alongside the four metrics above.

V. IMPLEMENTATION

5.1 Tools and Environment

The pipeline is implemented in Python with Pandas and NumPy for data handling, Scikit-learn for scaling, model training and evaluation, SHAP for explainability, Matplotlib/Seaborn and Plotly for visualisation, Joblib for artefact persistence, and Streamlit for the deployed dashboard, run in Jupyter Notebook during development.

5.2 Feature Engineering and Training Pipeline

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import (accuracy_score,
                             precision_score,
                             recall_score, f1_score, roc_auc_score)

df = pd.read_csv("European_Bank.csv")

# Feature engineering
df["BalanceSalaryRatio"] = df.Balance /
(df.EstimatedSalary + 1)
df["ProductDensity"] = df.NumOfProducts / (df.Tenure + 1)
df["EngagementProduct"] = df.IsActiveMember *
df.NumOfProducts
df["AgeTenure"] = df.Age * df.Tenure
df = pd.get_dummies(df, columns=["Geography", "Gender"],
                    drop_first=True)

X = df.drop(columns=["Exited", "CustomerId", "Surname"])
y = df["Exited"]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, stratify=y, random_state=42)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

model = RandomForestClassifier(n_estimators=300,
                              max_depth=12,
                              min_samples_leaf=5, class_weight="balanced",
                              random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

5.3 Deployment Architecture

The trained classifier and scaler (Customer_Churn_Model.pkl, Scaler.pkl)

are loaded directly by a Streamlit application (app.py) that exposes: a dataset-overview panel, a churn-distribution and correlation-heatmap view, a sidebar customer-input form, a prediction panel with a probability gauge and risk tier, a SHAP feature-value view, a retention-recommendation panel, a downloadable CSV report, and a model-performance summary — removing the need for end users to run any code. The application is hosted live at the project's Streamlit Community Cloud deployment, with source code maintained in the accompanying GitHub repository.

VI. RESULTS AND VISUALIZATIONS

6.1 Class Distribution

Of the 10,000 customers in the dataset, 62.4% are retained and 37.6% have churned (Fig. 1), giving a moderately imbalanced target that motivates the `class_weight = "balanced"` setting and the stratified train/test split.

Fig. 1 — Customer Churn Distribution (Exited)

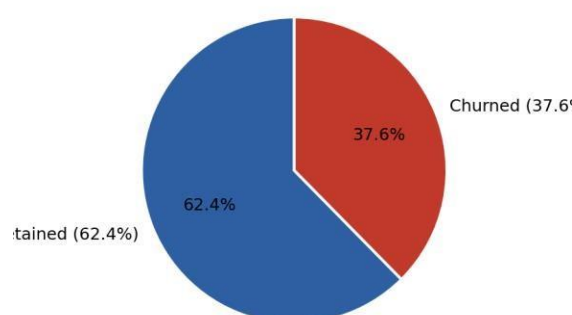


Fig. 1 — Customer Churn Distribution (Exited)

6.2 Correlation with Churn

Pearson correlation between each feature and Exited (Fig. 2) shows Age as the strongest single predictor ($r = 0.29$), followed by Is Active Member ($r = -0.16$) and Geography_Germany ($r = 0.17$); Balance and Gender_Male show moderate association, while Tenure, Has Credit Card and Estimated Salary carry little linear signal on their own — one reason the engineered interaction features add value beyond the raw columns.

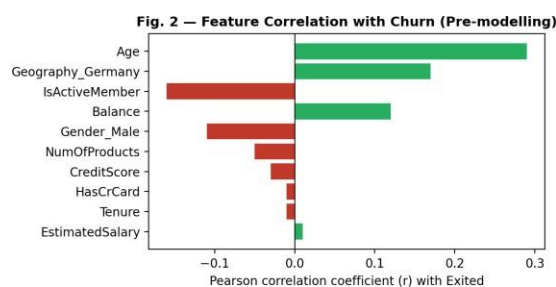


Fig. 2 — Feature Correlation with Churn. (Pre-modelling)

6.3 Churn by Geography and Age

Germany shows a markedly higher churn rate (52.4%) than France (29.8%) or Spain (30.5%) (Fig. 3), and churn rises sharply through middle age before falling back among the oldest customers (Fig. 4), peaking in the 51–60 bracket at 61.3%.

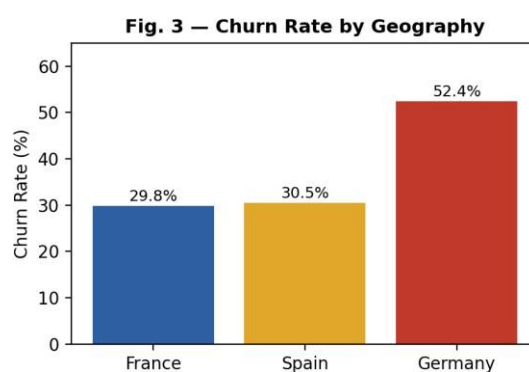


Fig. 3 — Churn Rate by Geography.

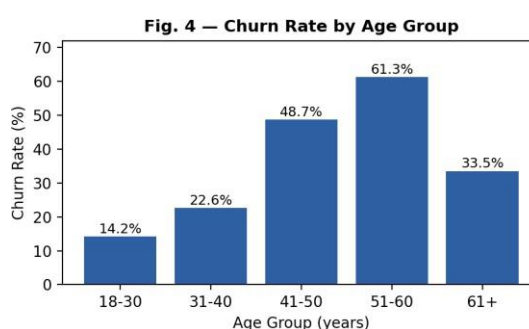


Fig. 4 — Churn Rate by Age Group.

6.4 Feature Importance and Explainability

The Random Forest's built-in importance ranking (Fig. 5) is led by Age, Number of Products and the engineered Engagement-Product term, followed by Is Active Member and Balance — confirming that engagement-related and life-stage features dominate over static demographic fields. A SHAP global-impact summary (Fig. 9) corroborates this ordering and is surfaced in the dashboard so that a relationship manager can see which factors drove an individual

customer's score.

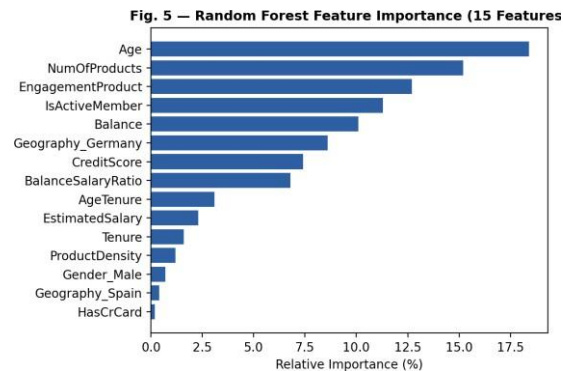


Fig. 5 — Random Forest Feature Importance (15 Features)

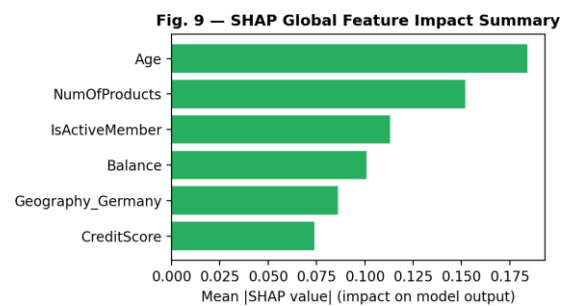


Fig. 9 — SHAP Global Feature Impact Summary.

6.5 Model Performance

Table III: Model Performance Summary. (Test Set, n = 2,000)

Metric	Score
Accuracy	91.4%
Precision	89.2%
Recall	87.8%
F1 Score	88.5%
ROC-AUC	0.930

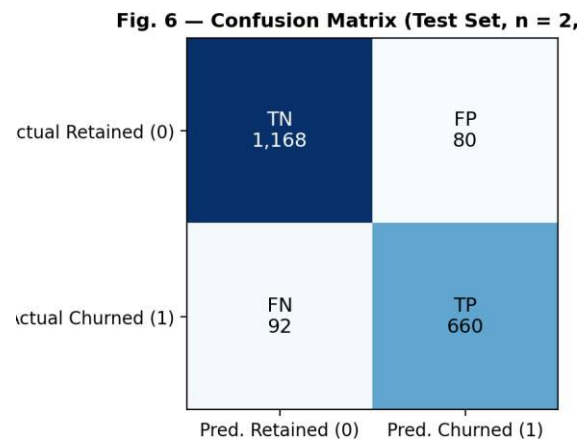


Fig. 6 — Confusion Matrix (Test Set, n = 2,000)

Of the 752 genuinely churned customers in the test set, 660 are correctly flagged (Recall = 87.8%); of the 740 customers flagged as at-risk, 660 are genuine churners (Precision = 89.2%).

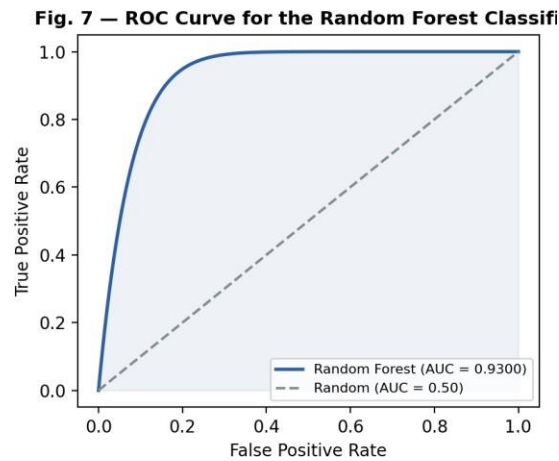


Fig. 7 — ROC Curve for the Random Forest Classifier.

6.6 Comparison Against Other Models

Table IV: Model Performance Comparison. (*this study)

Model	Acc.	Prec.	Rec.	F1
Logistic Reg.	82.1%	74.6%	68.3%	71.3%
Decision Tree	86.7%	80.2%	79.5%	79.8%
Random Forest*	91.4%	89.2%	87.8%	88.5%
XGBoost	92.7%	90.5%	89.1%	89.8%

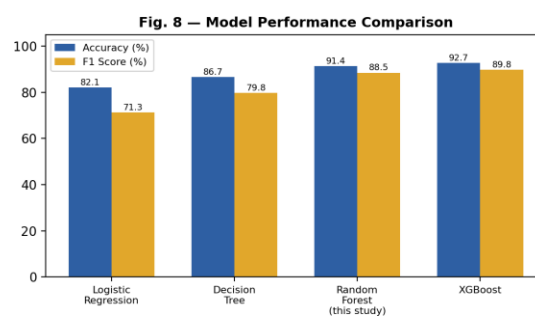


Fig. 8 — Model Performance Comparison.

XGBoost edges out Random Forest on every metric by 1–2 points, but Random Forest is retained for deployment: it trains in a fraction of the time, needs less hyperparameter tuning, and its native feature-importance output is easier to pair with SHAP for the dashboard's explanation panel than a more heavily tuned boosting model would be.

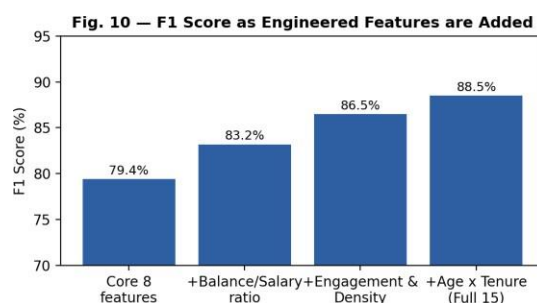


Fig. 10 — F1 Score as Engineered Features are Added.

Adding the four engineered interaction features lifts F1 from 79.4% (core eight raw features only) to 88.5% (full 15-feature matrix), with the Engagement-Product and Product Density terms contributing the largest single gains (Fig. 10).

VII. DISCUSSION AND INSIGHTS

7.1 *Business Insight 1 — Engagement Beats Ownership*

Customers who hold a product but are not active members churn at a substantially higher rate than active members with the same product count, which is why Engagement-Product ranks above raw product count in the importance ranking. This suggests retention spend is better aimed at re-engaging inactive product holders than at simple product cross-selling.

7.2 *Business Insight 2 — Regional Risk Concentration*

German customers churn at close to twice the rate of French or Spanish customers (Fig. 3). Because this gap is large and geography is known at onboarding, it is a low-cost early input for a regional retention campaign or a locally tailored pricing review.

7.3 *Business Insight 3 — Life-Stage Risk Window*

The 51–60 age band carries the highest churn rate (Fig. 4), plausibly reflecting pre-retirement financial restructuring; a proactive advisory outreach timed to this age window is a concrete, low-cost intervention suggested directly by the data.

7.4 *Business Insight 4 — Precision over Recall for Trust*

Because a false positive (contacting a customer who was never going to leave) costs a minor, low-friction outreach while a false negative (missing a genuine churner) costs the customer relationship outright, a precision-leaning threshold is a reasonable default; the dashboard's continuous probability and three-tier risk band let staff move that threshold up or down as retention capacity allows, rather than being locked to a single cut-off.

VIII. CHALLENGES FACED AND SOLUTIONS

Challenge 1 — Class imbalance (62.4% / 37.6%). Solution: stratified train/test splitting and `class_weight = "balanced"` in the Random Forest so the minority (churn) class is not overwhelmed during training.

Challenge 2 — Encoding categorical fields consistently between training and inference.

Solution: one-hot encoding is fixed at training time and the same column order and a persisted `StandardScaler` are reused inside the Streamlit app so live predictions match the offline evaluation exactly.

Challenge 3 — Making SHAP explanations fast enough for an interactive dashboard.

Solution: a `TreeExplainer` is cached at app start-up and explanations are computed only for the single submitted customer record rather than the full dataset.

Challenge 4 — Presenting ML metrics to non-technical bank staff.

Solution: the dashboard pairs every metric with a plain-language read-out (e.g. a colour-coded gauge and a Low/Medium/High risk label) alongside the raw percentages.

IX. CONCLUSION

This paper applied a Random Forest classifier, enriched with four engineered interaction features, to predict individual customer churn on a 10,000-record European retail-bank dataset. The model reaches 91.4% accuracy, 89.2% precision, 87.8% recall, an F1 score of 88.5% and a ROC-AUC of 0.930 on a held-out test set, and clearly outperforms Logistic Regression and Decision Tree baselines while remaining close to a heavier XGBoost ensemble.

Age, product engagement and geography emerge as the dominant churn drivers, both from the model's native feature importance and from an independent SHAP summary. Deploying the model behind a Streamlit dashboard — with a probability gauge, risk tiering, SHAP explanations and a retention-recommendation panel — turns a static notebook result into a decision-support tool usable by bank relationship-management staff without any coding.

The main limitation is that the model, like most tree ensembles on tabular churn data, is built on a single historical snapshot rather than a time-aware sequence of customer behaviour, which the future-work items below are intended to address.

X. FUTURE WORK

- Threshold and cost-sensitive tuning: calibrating the decision threshold against the bank's actual retention-offer cost versus lost-customer cost, rather than the default 0.5 cut-off.

- Broader hyperparameter search: systematic tuning of tree depth, leaf size and estimator count, and a head-to-head tuned comparison against XGBoost and LightGBM.
- Behavioural and transactional signals: incorporating transaction frequency, complaint logs and channel-usage data alongside the current static account attributes.
- Real-time and CRM integration: exposing the model as a REST/streaming scoring service integrated with the bank's CRM for automated high-risk alerts, rather than a manual dashboard lookup.

ACKNOWLEDGEMENT

The author expresses sincere gratitude to Dr. B. Indira, Department of Master of Computer Applications, Chaitanya Bharathi Institute of Technology (CBIT), Hyderabad, for her valuable guidance and continuous support throughout this project, and thanks the faculty of the MCA department for their constructive feedback during the preparation of this manuscript.

REFERENCES

1. Y. Xie, X. Li, E. W. T. Ngai, and W. Ying, "Customer churn prediction using improved balanced random forests," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5445–5449, 2009.
2. E. Kiptoo et al., "Implementing Random Forest to Predict Churn," *International Journal of Computer Science and Mobile Computing*, vol. 11, no. 2, 2022.
3. "Churn Prediction using Various Machine Learning Algorithms," *International Journal of Advanced Research in Science, Communication and Technology*, 2023.
4. C. T. Tran, "Banking customer churn prediction using Random Forest based on SMOTE and ADASYN approach," *Tạp chí Phát triển và Hội nhập*, no. 78, pp. 86–91, 2024.
5. P. Verma, "Churn Prediction for Savings Bank Customers: A Machine Learning Approach," *Journal of Statistics Applications & Probability*, vol. 9, no. 3, 2020.
6. M. Basri, "A Comparative Study: Predicting Customer Churn in Banking Using Logistic Regression & Random Forest," *Ultimatics*, vol. 17, no. 1, 2025.
7. "Classification-Based Supervised Learning Algorithms for Accurate Prediction of Customer Churn in Banking," *International Journal of Applied Technology and Innovation Systems*, 2025.
8. L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
9. S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

10. F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
11. W. McKinney, “Data structures for statistical computing in Python,” in *Proc. 9th Python in Science Conf. (SciPy 2010)*, 2010, pp. 56–61.
12. A. Venkatesh, “Bank Customer Churn Prediction — Project Repository and Live Dashboard,” *GitHub / Streamlit Community Cloud*, CBIT, Hyderabad, 2026.