Bank Customer Churn Analysis: Summary, Key Findings, and Recommendations

Prepared by: Isaiah Edem Essien

i.essien@alustudent.com

June 15, 2025

Link to Presentation Slide: Click Here

Introduction

This analysis set out to build and compare predictive models for customer churn at a retail bank, with the ultimate goal of identifying at-risk clients and informing targeted retention strategies. The project encompasses data exploration, feature preparation, model training (using logistic regression, XGBoost, and a neural network), formal evaluation against business-relevant metrics, and deployment of the best model via an API.

Key Definitions

1. **Precision**: The fraction of instances predicted as positive that are actually positive. It is calculated as True Positives divided by the sum of True Positives and False Positives. In churn modeling, precision tells us how many of the customers flagged as "will churn" truly end up churning.
2. **Accuracy**: The overall proportion of correct predictions, found by dividing the sum of True Positives and True Negatives by the total number of cases. While easy to interpret, accuracy can be misleading when one class (e.g., non-churners) dominates.
3. **Loss**: A numerical measure of prediction error used during model training. For binary classification, binary cross-entropy (also called log loss) is common. It quantifies the distance between the predicted probabilities and the true binary labels; lower loss indicates a better fit.

4. **F_1 Score**: The harmonic mean of precision and recall, defined as

   $F_1 = 2 \times (\text{precision} \times \text{recall}) / (\text{precision} + \text{recall})$.

   It balances the trade-off between false positives and false negatives, yielding a single metric especially useful under class imbalance.

5. **AUC (ROC-AUC)**: The area under the Receiver Operating Characteristic curve, which plots True Positive Rate against False Positive Rate across all classification thresholds. An AUC of 0.5 indicates random guessing, while 1.0 signifies perfect discrimination between classes.

6. **Shapley Value (SHAP)**: A feature-attribution method from cooperative game theory. Each feature's Shapley value represents its average marginal contribution to the model's prediction across all possible combinations of features, providing consistent and locally accurate explanations of individual predictions.

7. **Churn Rate**: The percentage of customers who discontinue service within a given period. In this dataset, the churn rate was 18.2 percent, meaning roughly 18 out of every 100 customers closed their accounts over the study window**.**

Data Overview and Preparation

Link to data

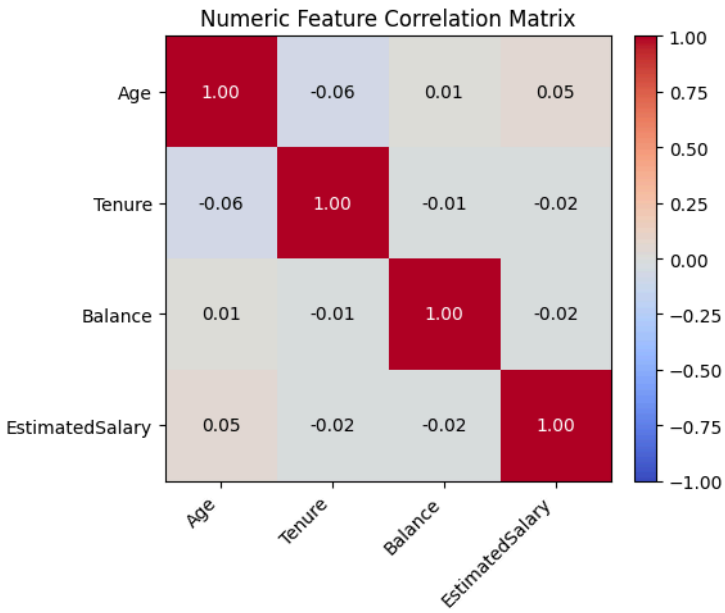
Link to Notebook

Link to Project GitHub Repository

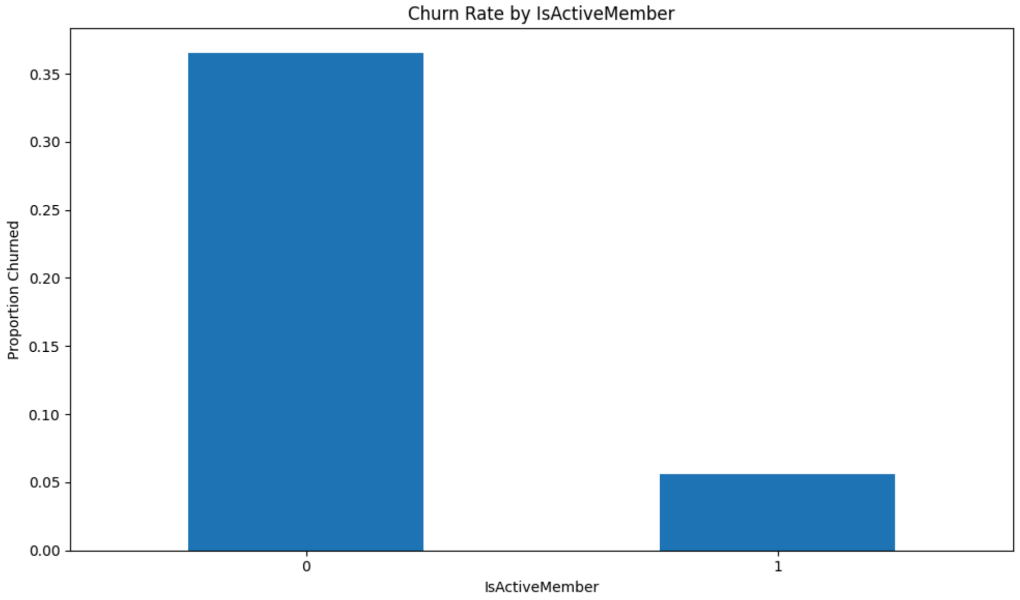
An Excel file containing 1,000 customer records and nine columns—among them CustomerID, demographic and account features, and a binary Churn flag—was loaded into a pandas DataFrame. No missing values were found. I recoded Churn to 0 (stay) and 1 (leave) and dropped the non-predictive CustomerID. An 80/20 train–test split with stratification on the churn label preserved the 18.2 percent churn rate in both subsets.

Exploratory Analysis

Continuous features (Age, Tenure, Balance, EstimatedSalary) were summarized with histograms and descriptive statistics. Binary flags (HasCreditCard, IsActiveMember) and product counts were examined with bar charts. A correlation heatmap showed all pairwise correlations under ±0.1, indicating negligible multicollinearity. The most striking univariate finding was that inactive members churn at approximately 36.4 percent versus 5.7 percent for active members. Age exhibits a U-shaped relationship with churn, high among the youngest (under 30) and oldest (over 60) customers, and lowest in middle age.



Correlation HeatMap of Data



Churn rate by Number of Active users

Data Preprocessing and Imbalance Handling

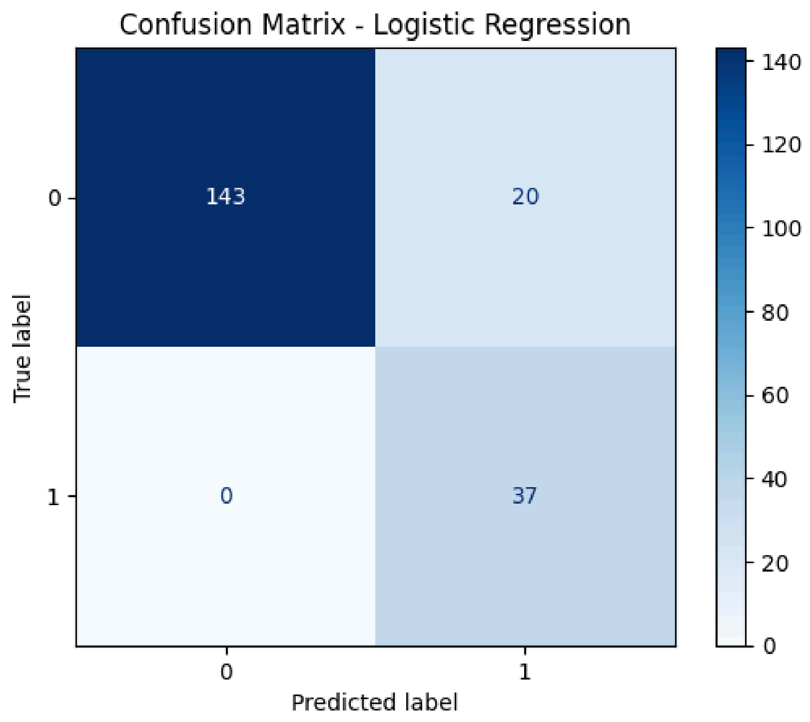
To prepare features for modeling, I standardized the five continuous variables using a pre-fitted StandardScaler. Standardization places all numeric inputs on the same scale, which is essential for gradient-based methods (logistic regression, neural networks) and prevents features with larger magnitudes from dominating distance-based splits in tree ensembles. I left the binary flags (HasCreditCard, IsActiveMember) unscaled, since their 0/1 encoding already conveys categorical presence or absence.

I experimented with SMOTE to synthetically oversample the minority churn class and reduce bias toward the 82 percent majority of non-churners. In practice, the synthetic samples generated by SMOTE clustered too close to class boundaries and introduced ambiguous examples that degraded validation performance. Consequently, I abandoned SMOTE in favor of algorithmic imbalance handling—using class_weight='balanced' in logistic regression and scale_pos_weight in XGBoost—to preserve the integrity of real customer profiles and maintain discrimination power.

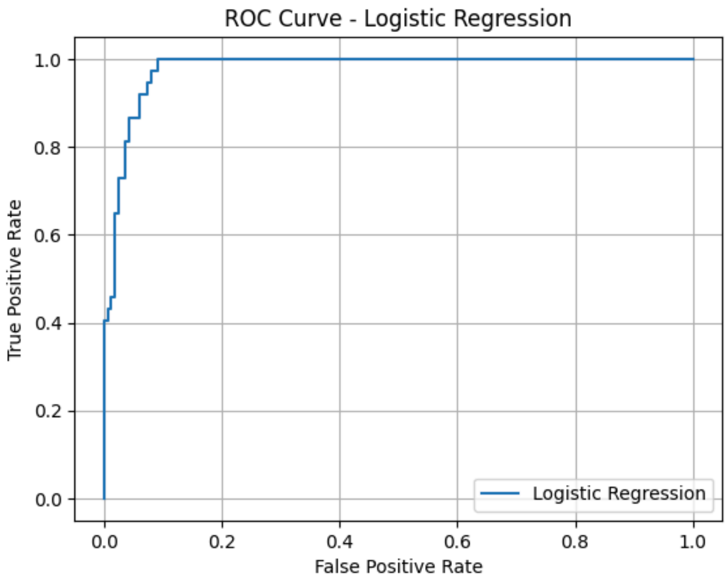
Model Development

Logistic Regression

I trained a baseline logistic regression with class_weight='balanced' to compensate for the 4.5 to 1 imbalance. On the test set, it achieved 90% accuracy, a recall of 1.00 for churners (no false negatives), a precision of 0.65 (35% of predicted churners were false positives), and an AUC of 0.979. This model reliably identifies every at-risk customer but would generate extra outreach.



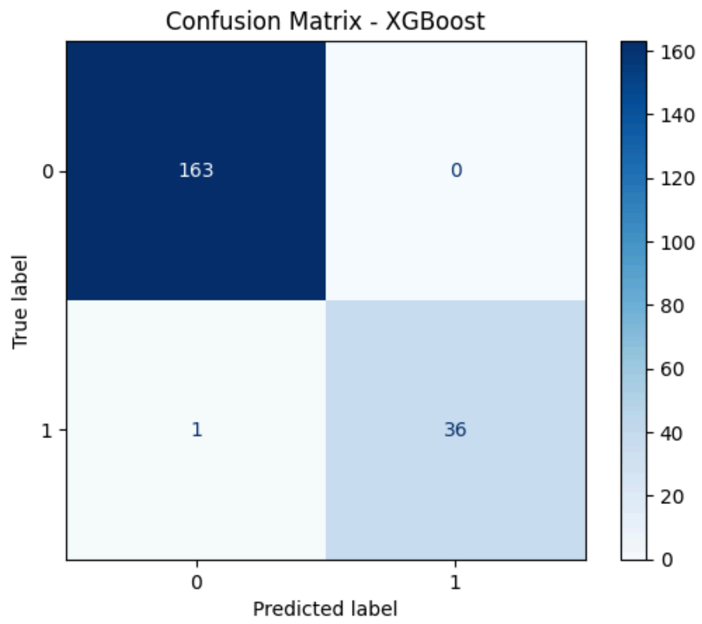
Confusion Matrix of Logistic Regression Model



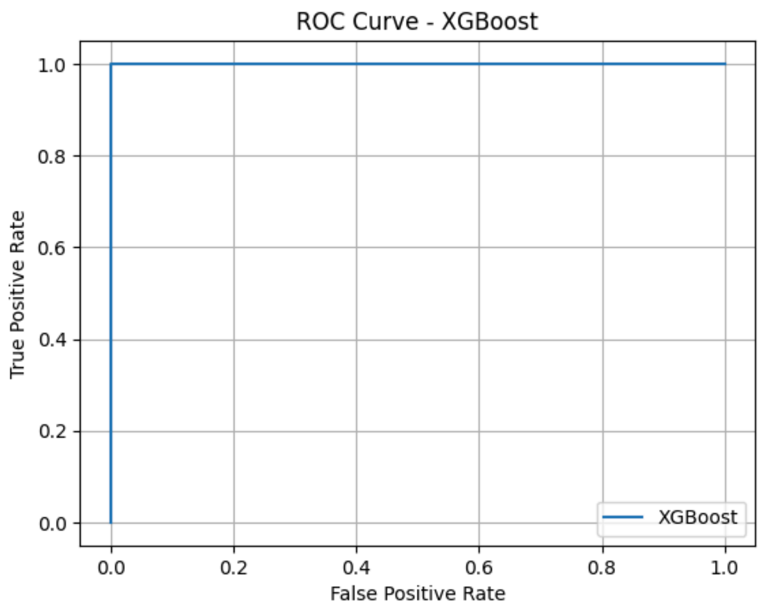
Logistic Regression ROC curve

XGBoost

An XGBoost classifier was configured with scale_pos_weight equal to the ratio of non-churners to churners. It delivered near-perfect performance on the hold-out set: 99.5 percent accuracy, precision of 1.00, recall of 0.97 (one missed churner), and AUC of 1.00. While these metrics signal exceptional discrimination, they also raise overfitting concerns given the modest test sample.



Confusion matrix of the XGB model

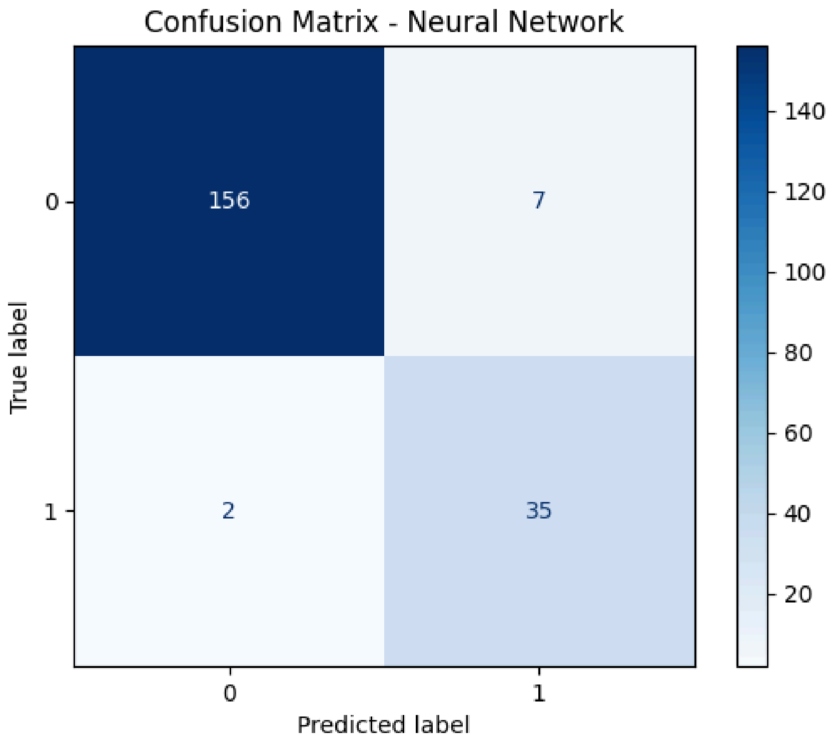


XGB ROC curve

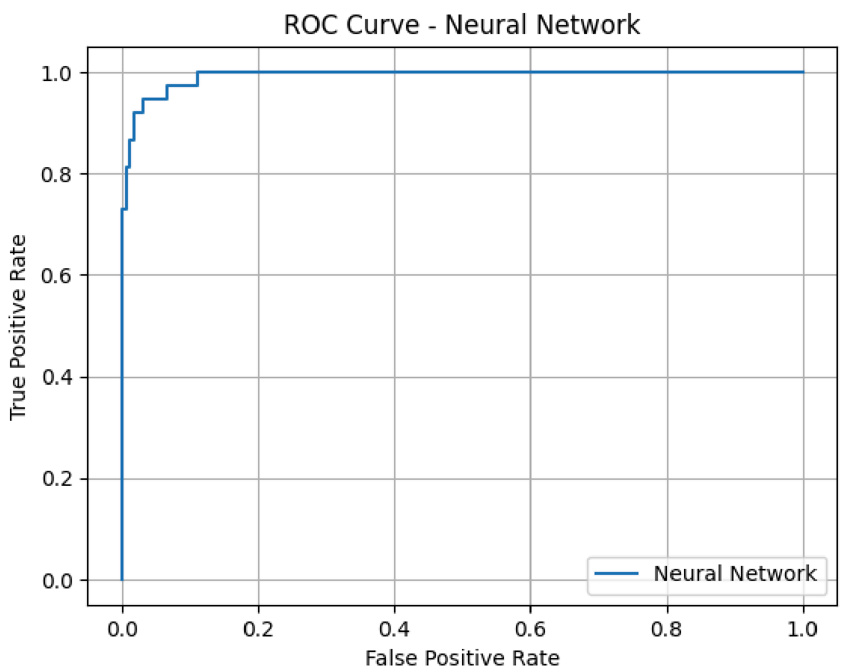
Neural Network

A feed-forward network with two hidden layers (64→32 units), 30 percent dropout, and early stopping on validation AUC was trained with class weights. It reached an AUC of 0.992, a recall of

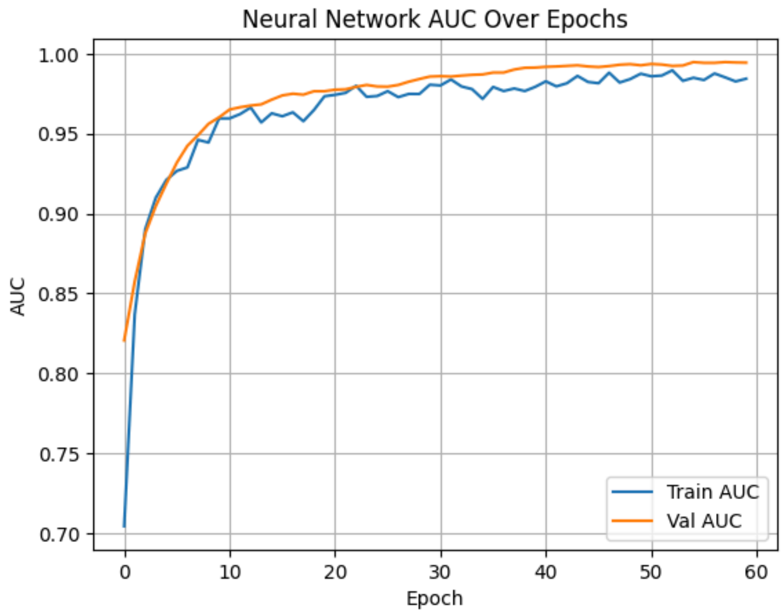
0.95, and a precision of 0.83. Training and validation AUC curves rose together to about 0.99 by epoch 20 and plateaued, indicating good generalization without pronounced overfitting.



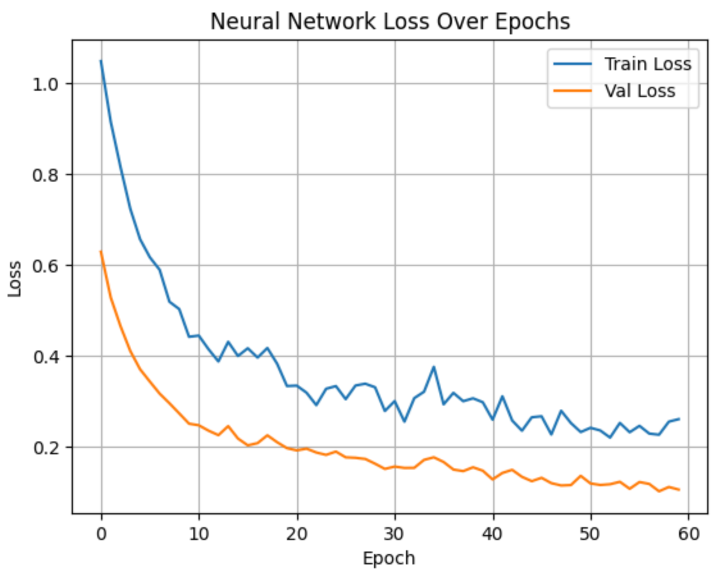
Confusion Matrix of the Neural Network Model



Neural network ROC curve



Neural Network Training and Validation Accuracy



Neural Network Training and Validation Loss

Findings

- **Churn Rate:** 18.2 percent of customers churn, creating a moderate class imbalance.

- **Key Driver:** Membership activity is the strongest churn predictor; inactive customers are six times more likely to churn.

- **Secondary Factors:** Credit-card ownership and the number of products have only a marginal impact.

- **Age Effect:** Younger and senior cohorts carry higher churn risk, suggesting the need for age-based engagement strategies.

- **Model Performance:** XGBoost achieved the highest raw metrics but requires further cross-validation and regularization to ensure stability. The neural network and logistic regression demonstrate that both simple and complex architectures can capture the churn signal, with trade-offs between interpretability and precision.

Recommendations for Customer Retention

1. **Engagement Campaign for Inactive Members:** Proactively target the 36 percent risk group with personalized offers, account reviews, or loyalty incentives.

2. **Age-Segmented Outreach:** Develop distinct messaging and offers for customers under 30 and over 60, addressing their specific financial needs and concerns.

3. **Tiered Incentives Based on Product Usage:** While product count alone is not a strong predictor, bundling additional services (e.g., premium checking with savings) may increase

stickiness.

4. **Monitor High-Balance Accounts:** Seniors with large balances exhibit elevated churn; consider relationship-manager check-ins or fee waivers for these clients.

Deployment and Integration

Link to Deployed Model on Render

To facilitate experimentation and integration, the best-performing model was serialized and exposed via a FastAPI endpoint. A simple JSON API accepts customer attributes, applies the same numeric scaling, and returns both churn probability and a binary prediction. Front-end teams or business analysts can submit sample profiles to "play around" with the model in real time.

For production rollout, I recommend embedding the API behind the bank's secure gateway, configuring a batch scoring job within the CRM system to flag at-risk customers daily, and exposing dashboards—populated by the model's outputs and SHAP-based feature explanations—that empower relationship managers to prioritize outreach.

Next Steps

Before full production, conduct repeated stratified cross-validation and incorporate early-stopping or L1/L2 regularization on the tree-based model to guard against overfitting. Implement SHAP for model-agnostic feature attribution in stakeholder reports and establish quarterly retraining to capture evolving customer behavior patterns.