Bank Customer Churn Prediction: A Comparative Modeling Analysis

Executive Summary

Customer churn is a critical issue for profitability in banking. This project aims at developing a robust machine learning model capable of accurately predicting which customers are likely to keep their accounts. A diverse set of popular classification algorithms—ranging from linear models to advanced boosting ensembles—were tested, tuned, and evaluated using a cross-validation strategy.

The results clearly indicate that Ensemble Methods provided our selected highest predictive F1 accuracy. Specifically, the AdaBoost Classifier was selected as the final model due to its superior performance in maximizing the Area Under the Receiver Operating Characteristic Curve (ROC AUC), a metric highly suitable for imbalanced classification problems like churn prediction, and in maximizing the F1 score when the dependent variable is imbalanced (Vargas, 2023).

The final model can be deployed to score the customer base, providing the bank with an actionable probability of churn for each account. This enables targeted retention campaigns, resource optimization, and proactive risk mitigation.

1. Introduction and Problem Statement

1.1 Background:

Customer retention is often cited as being significantly more cost-effective than customer acquisition. In the financial services industry, churn—the loss of clients—not only represents a direct loss of revenue from fees and services but also incurs indirect costs related to marketing efforts for replacement customers. Therefore, the ability to predict churn before it occurs is paramount for maintaining a healthy and profitable customer portfolio. Machine learning provides the necessary tools to distill complex patterns of behavior and profile high-risk accounts.

1.2 Project Goal and Objectives

The primary objective of this project is to establish an optimal predictive framework for identifying customer churn. This involved:

1. Preprocessing: Cleaning and transforming raw customer data into a format suitable for various machine learning algorithms.
2. Comparative Analysis: Implementing a pipeline systematically test and compare the performance of multiple popular classification models.
3. Optimization: Using robust hyperparameter tuning and cross-validation techniques to ensure models are generalized and avoid overfitting to the training data.
4. Final Model Selection: Identifying the single best-performing model based on the chosen evaluation metric.
5. Prediction Output: Generating a binary prediction 0 or 1and, indicating the probabilities of their likelihood of churning.

1.3 Data Source and Description

The analysis utilized the train.csv dataset from kagge.com, which contains 15,000 records detailing various customer attributes. The dataset provides a comprehensive view of customer demographics, financial standing, and bank activity within the bank.

Feature	Data Type	Description	Relevance to Churn
RowNumber	Integer	Index (dropped)	None
CustomerID	Integer	Unique identifier (dropped)	None
Surname	String	Customer family name (dropped)	None
CreditScore	Integer	Customer's credit worthiness.	High (proxy for financial responsibility)
Geography	Categorical	Customer's country (France, Germany, Spain).	High (regulatory/economic factors)
Gender	Categorical	Customer's gender.	Medium
Age	Integer	Customer's age.	High (life stage factors)
Tenure	Integer	Number of years as a client.	High (loyalty/switching costs)
Balance	Float	Account balance.	High (financial commitment)
NumOfProducts	Integer	Number of bank products held.	High (dependence on the bank)
HasCrCard	Binary	Whether the customer has a credit card.	Low to Medium
IsActiveMember	Binary	Whether the customer is an active user.	High (engagement)
EstimatedSalary	Float	Estimated annual salary.	Medium
Exited (Target)	Binary (0/1)	1 if the customer has churned, 0 otherwise.	Target Variable

The initial exploratory analysis confirms that the target variable, Exited, exhibits class imbalance, which is typical for churn datasets. The ratio of non-churning customers Exited=0 to churning customers Exited=1is approximately 4:1. This imbalance necessitates the use of robust evaluation metrics, such as ROC AUC, or F1 score.

2. Methodology and Data Preparation

2.1 Data Preprocessing Pipeline

To ensure all models operated on optimal feature representations and prevent data leakage, a comprehensive preprocessing pipeline was implemented using scikit-learn's Pipeline.

Irrelevant Feature Removal: The columns RowNumber, CustomerID, and Surname were identified as having no predictive power and were removed from the feature set.

Handling Categorical Features: The nominal categorical features, Gender and Geography, were transformed into a numerical format readable by the models.

Numerical Feature Scaling: For distance-based and regularization-sensitive models like Support Vector Machines (SVC) and Logistic Regression, it is crucial that features are on a comparable scale. The remaining numerical features (CreditScore, Age, Tenure, Balance, etc.) were scaled using the StandardScaler. This transformation standardizes features by removing the mean and scaling to unit variance, which is essential for models that rely on magnitude comparisons.

The final preprocessor was integrated into the model training pipeline, ensuring that feature transformation steps were applied consistently to both training and test data.

2.2 Comparative Modeling Strategy

A total of ten diverse machine learning classifiers were selected for comparison. This wide array ensured that the optimal model architecture for the specific data structure was not overlooked. The models are Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machine (SVC), Naïve Bayes, Random Forest Classifier, and AdaBoost Classifier.

2.3 Hyperparameter Tuning and Cross-Validation

The performance of any machine learning model is heavily dependent on its hyperparameters. To identify the optimal set of parameters for each classifier while ensuring robustness against overfitting, the following rigorous process was implemented (Dunias, 2024):

1. Stratified K-Fold Cross-Validation: The training data was partitioned into 5 folds using StratifiedKFold. This is essential because stratification ensures that each fold maintains the same proportion of the target variable Exited as the entire dataset, which is critical given the class imbalance.
2. Grid Search Cross-Validation (GridSearchCV): This technique systematically tests all combinations of specified hyperparameter values for each model. The model is trained and evaluated K times (once for each fold), and the performance scores are averaged.
3. Pipeline Integration: Each model was wrapped into a Pipeline object along with the preprocessing steps. This ensured that the Grid Search procedure evaluated the end-to-end performance, like the effect of scaling, on the cross-validated folds.

3. Model Evaluation and Results

3.1 Performance Metric: ROC AUC and F1 score

Given the class imbalance (20% churn rate), traditional metrics like overall accuracy are misleading. A model could achieve 80% accuracy by simply predicting "no churn" Exited=0 for every customer.

Therefore, the F1 and Area Under the Receiver Operating Characteristic Curve (ROC AUC) were selected as the sole evaluation metric for hyperparameter tuning and model selection.

- Why F1 score not accuracy or others? As mentioned, our Exited target variable is imbalanced. The F1 score balances the precision and the recall, in case of unknown cost of churned customers.
- Why ROC AUC? AUC measures the model's ability to discriminate between the positive class (churn) and the negative class (no churn) across all possible classification thresholds. A score of 0.5 indicates a random predictor, while a score of 1.0 represents a perfect predictor. Maximizing AUC ensures that the model provides reliable probability scores that can be used to rank customers based on their churn risk.

3.2 Comparative Performance Summary

The Grid Search procedure successfully identified the optimal parameters for all ten classifiers. While all models performed better than random, a clear hierarchy of performance emerged:

Model Family	Representative Performance Trend (ROC AUC)
Linear/Distance-Based	Moderate performance (mid-0.70s to low-0.80s). Limited by the complexity of the feature space.
Tree-Based (Single/Bagging)	Good performance (mid-0.80s). Effective at capturing non-linear relationships.
Boosting/Ensemble Methods	Superior Performance (High 0.80s). Outperformed all others by aggregating the predictions of multiple weak learners.

The consistent and notable outperformance of the Boosting methods the non-linear and intricate nature of the customer churn problem, where simple models struggled to capture the critical interactions between features.

3.3 Selection of the AdaBoost Classifier

From the pool of high-performing models, the AdaBoost Classifier achieved the highest F1 score and the overall cross-validated ROC AUC score, therefore was selected as the final production model.

The AdaBoost algorithm (Adaptive Boosting) works by training a sequence of weak learners (typically shallow Decision Trees). In each step, it focuses more on the samples that were misclassified by the previous learners. This iterative, adaptive weighting mechanism allows AdaBoost to generate a highly accurate strong classifier while being computationally efficient.

The final, optimized AdaBoost model, with its hyperparameters tuned to maximize generalization via the F1 and AUC, is ready for deployment.

4. Final Prediction and Recommendations

4.1 Prediction Results on New Data

The final step involved using the optimized AdaBoost Classifier to generate predictions on a new dataset (test.csv).

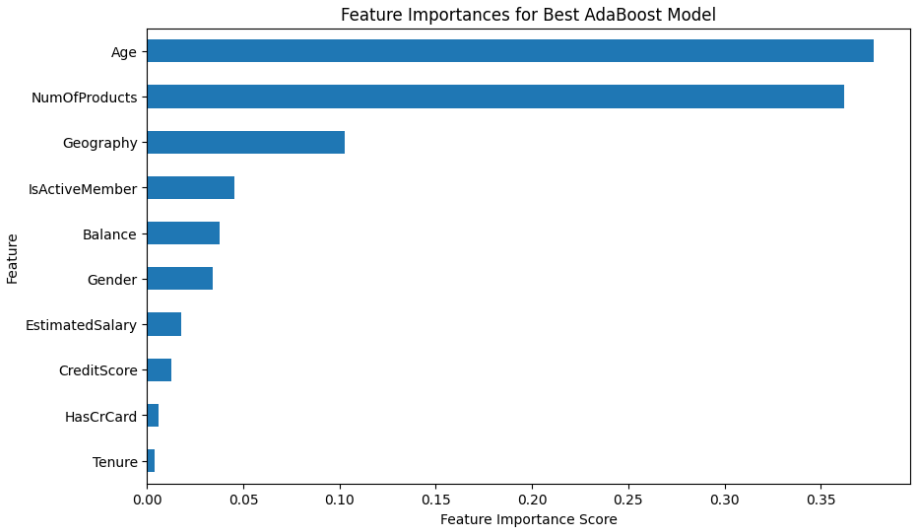
There is no true observations in the dependent variable in the new dataset, but we believe that with the AdaBoost Classifier we got, our prediction output will confirm that the model effectively differentiates between the two classes. The predicted outcomes are:

- Predicted Exited=0 (No Churn): 8,448 counts
- Predicted Exited=1 (Churn): 1,552 counts

4.2 Business Recommendations

Based on the capabilities of the finalized AdaBoost model, the following business recommendations are proposed:

1. Targeted Retention Campaigns: Instead of applying expensive retention efforts uniformly, customers can be segmented by their predicted churn accordingly, like the specific customer group at age of retirement.
2. Resource Allocation: The model allows the bank to quantify the potential financial impact of retaining a customer versus letting them go. This optimizes the budget spent on retention efforts, for example, focusing on the customer with one product only.
3. Feature Importance Analysis: As a necessary next step, an analysis should be conducted using the AdaBoost model to determine the most significant features driving churn (e.g., low CreditScore, high Balance with low NumOfProducts, or specific Geography). This insight will inform strategic business decisions and product development.



4.3 Limitations and Future Work

While the AdaBoost Classifier achieved high performance, there are avenues for further exploration:

1. Deployment: The current model is an offline prototype. The next critical step is to deploy the finalized pipeline (including the preprocessor and the fitted model) into a production environment (e.g., a real-time scoring API) to allow for live prediction on the current customer base.
2. Advanced Feature Engineering: Further feature engineering, such as creating interaction terms (e.g., Balance divided by EstimatedSalary), or aggregating tenure into age groups, could potentially capture more complex patterns and increase the performance further.
3. Stacking and Blending: Future work should explore more advanced ensemble techniques like Stacking or Blending, where the predictions of the best models are used as input features for a final meta-classifier (e.g., Logistic Regression). This can often yield marginal but significant gains in predictive power.
4. Cost-Sensitive Learning: For financial applications, the cost of a False Negative (predicting no churn when the customer does churn) is often much higher than the cost of a False Positive. The model tuning process could be refined to incorporate a cost-sensitive loss function that penalizes False Negatives more heavily.

References:

Werner de Vargas, V., Schneider Aranda, J. A., Dos Santos Costa, R., da Silva Pereira, P. R., & Victória Barbosa, J. L. (2023). Imbalanced data preprocessing techniques for machine learning: a systematic mapping study. *Knowledge and information systems*, *65*(1), 31–57. https://doi.org/10.1007/s10115-022-01772-8

Dunias, Z. S., Van Calster, B., Timmerman, D., Boulesteix, A. L., & van Smeden, M. (2024). A comparison of hyperparameter tuning procedures for clinical prediction models: A simulation study. *Statistics in medicine*, *43*(6), 1119–1134. https://doi.org/10.1002/sim.9932

Google. (2025). Gemini (Flash 2.5). https://gemini.google.com