Banking Churn Project

Customer retention is a key challenge for banks in today's highly competitive financial landscape. Understanding which customers are likely to leave (churn) and why, allows institutions to take timely and effective actions to retain them.

In this project, we aim to develop a predictive model that can identify whether a customer is likely to stay or leave the bank. By analyzing historical customer data—such as demographics, financial behavior, and product usage—we will train a machine learning model capable of predicting churn.

This predictive system will help the bank implement proactive strategies to improve customer satisfaction and reduce attrition, ultimately supporting better business decisions and increasing long-term profitability.

The final objective of this project is not only to predict customer churn, but also to provide actionable insights to the customer service department. By identifying **how many** and **which** customers are at risk of leaving, the bank can prioritize outreach efforts more effectively.

My approach will take into account the **cost-benefit trade-off** of contacting each customer. Instead of targeting all at-risk clients indiscriminately, the model will help the business focus on those customers whose retention is both **likely and valuable**, maximizing the return on investment of retention campaigns.

1. Data Preparation Summary

In this step, we performed a thorough data preparation process to ensure high model performance and reliability:

- **Initial Exploration**:
  Loaded and inspected the dataset, identifying non-numeric columns and renaming the target column (flag_request_closure → target), converting it into binary format (yes → 1, no → 0).
- **Encoding Categorical Variables**:
  - Applied **One-Hot Encoding** to low-cardinality categorical features like gender, income, customer type, and MIFID profile.
  - Used **Target Encoding** for high-cardinality features such as domicile province, residence province, and profession.
- **Data Cleaning**:
  - Removed special characters from column names to prevent parsing issues.
  - Handled missing values by filling them with the **median** of each respective column.
- **Feature Selection and Correlation Analysis**:
  - Analyzed correlation with the target variable to understand which features might be more predictive.
  - Confirmed no strong linear correlation with the target, supporting the choice of using a non-linear model.
- **Initial Feature Importance Assessment**:
  - Trained a baseline **XGBoost** model to get a first insight into feature importance using gain and weight metrics.
- **Class Imbalance Check**:

  - Discovered a highly **imbalanced dataset** (≈ 0.5% churn rate).
  - Decided to work with a representative **sample of 50,000 records** (245 churn vs. 49,755 no-churn) to improve training speed and focus on feature behavior.

NB (note well): In this project, the original dataset contains over 370,000 customer records, with less than 5% of them labeled as churners. This high class imbalance poses a common challenge in churn prediction tasks: most machine learning models tend to favor the majority class (non-churners), leading to high accuracy but poor recall for the minority class.

To address this, I tested different approaches:

- **Resampling the data** to balance the classes, either by oversampling churners or undersampling non-churners
- **Using a reduced dataset** of 50,000 samples, maintaining the original class imbalance

Surprisingly, the **imbalanced subset** outperformed the resampled versions in terms of model precision, recall, and ROI. This is because resampling—especially oversampling—can lead to **overfitting**, as it duplicates the limited positive examples, making the model less generalizable. Undersampling, on the other hand, discards valuable information from the majority class.

By maintaining the natural imbalance in a smaller dataset, the model preserved the real-world distribution and learned more representative patterns, avoiding artificial inflation of performance metrics.

For real-world applications, it is essential to:

- Keep the **data distribution realistic**
- Use **metrics suited for imbalanced classification** (e.g., ROC AUC, Precision-Recall, ROI)
- Consider **cost-sensitive learning** or **adjusted decision thresholds** instead of naive resampling

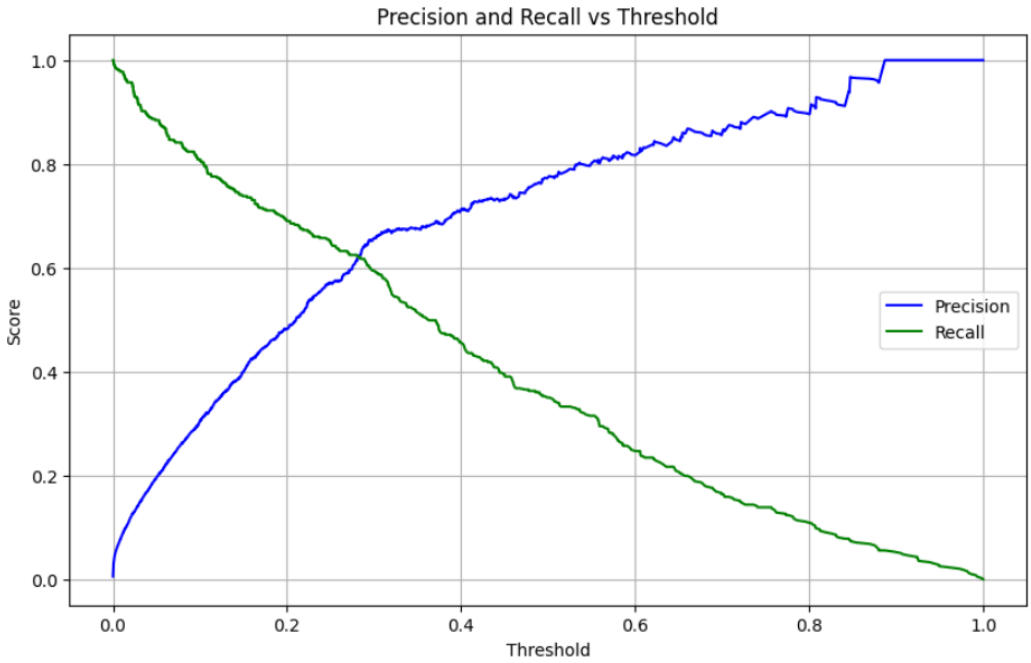
This approach ensures that the predictions remain actionable and aligned with business impact.

2. Modeling

In this phase, we developed and fine-tuned a predictive model to classify which customers are likely to churn.

- **Model Used**:
  Implemented **XGBoost Classifier**, well-suited for handling tabular data and imbalanced classification problems.
- **Key Hyperparameters**:
  - max_depth=3 to prevent overfitting
  - min_child_weight=10 and gamma=0.2 to enforce more conservative splits
  - subsample=0.7 and colsample_bytree=0.7 to increase generalization
  - Regularization terms (reg_alpha, reg_lambda) and scale_pos_weight=1.5 to manage class imbalance

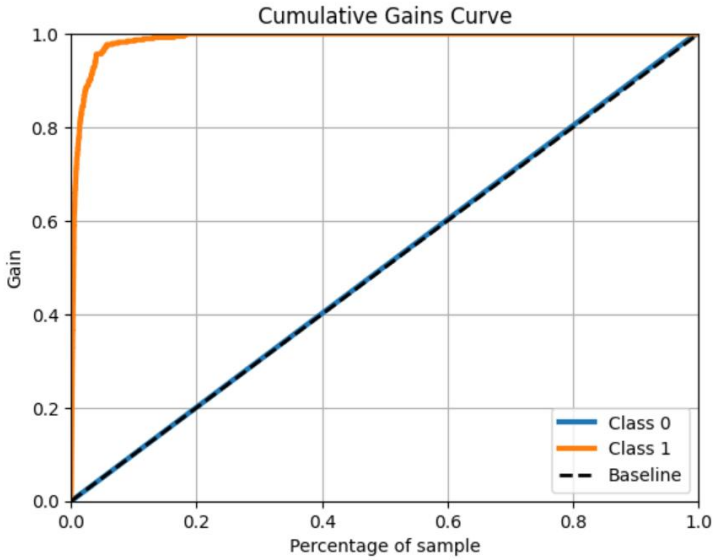
- **Threshold Tuning**:
    - Instead of using the default 0.5 threshold, we **optimized the classification threshold** based on the intersection point of **Precision and Recall**, finding an optimal threshold around **0.28**.
- **Model Evaluation**:
    - Evaluated performance using multiple metrics:
        - **Train Set**: Accuracy ≈ 0.994, F1 ≈ 0.43
        - **Test Set**: Accuracy ≈ 0.996, F1 ≈ 0.62
    - These scores demonstrate a good balance between **generalization** and **predictive power**, with no evident overfitting.
- **Precision-Recall Trade-off**:
    - A **Precision-Recall Curve** was used to visualize and select the most appropriate operating point for the business goal:
      Avoid missing real churners (high recall) while also minimizing false positives (high precision).



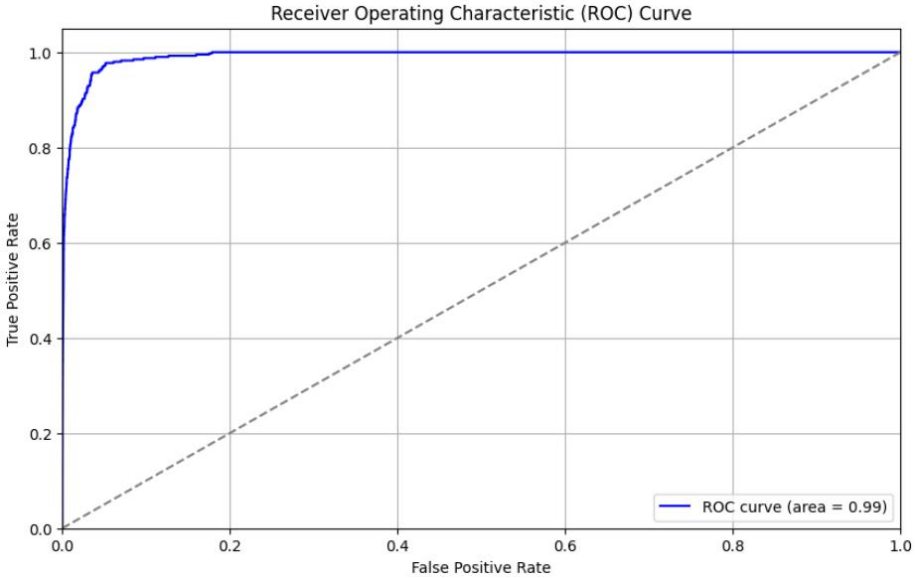
3. Model Evaluation

To assess the real-world effectiveness of our predictive model, we conducted a thorough evaluation using several advanced metrics and visual tools.

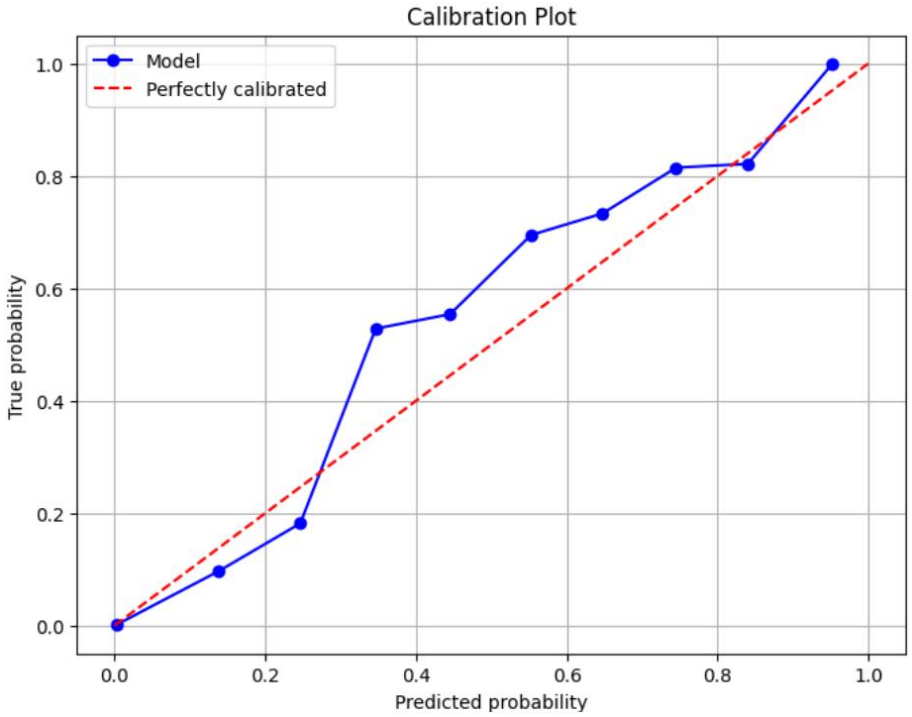
- **Cumulative Gains Curve**:
    - This plot highlights the model's ability to **correctly rank customers at risk**.
    - The curve shows that the model captures a high percentage of churners within the top percentiles of ranked probabilities – a strong indicator of its business value.



- **ROC Curve & AUC**:
    - The **Receiver Operating Characteristic (ROC)** curve provides a view of the trade-off between True Positive Rate and False Positive Rate.
    - Our model achieved an **AUC of 0.99**, confirming that it makes **very few classification errors** and performs almost perfectly at distinguishing churners from non-churners.



- **Calibration Plot**:
    - A well-calibrated model not only predicts who will churn, but also **how likely** they are to churn.
    - The **calibration curve** indicates that predicted probabilities align closely with actual outcomes — showing that **our model is almost perfectly calibrated**.



4. Presenting ROI to Business Users

To support the Marketing team in identifying the optimal classification threshold, I performed an in-depth cumulative ROI analysis across different score bins.
Using the predicted probabilities from the model, I segmented the results into score bins ranging from 0 to 1 (with a 0.01 step) to assess the economic impact at each threshold level.

✅ **For each score bin, I computed:**

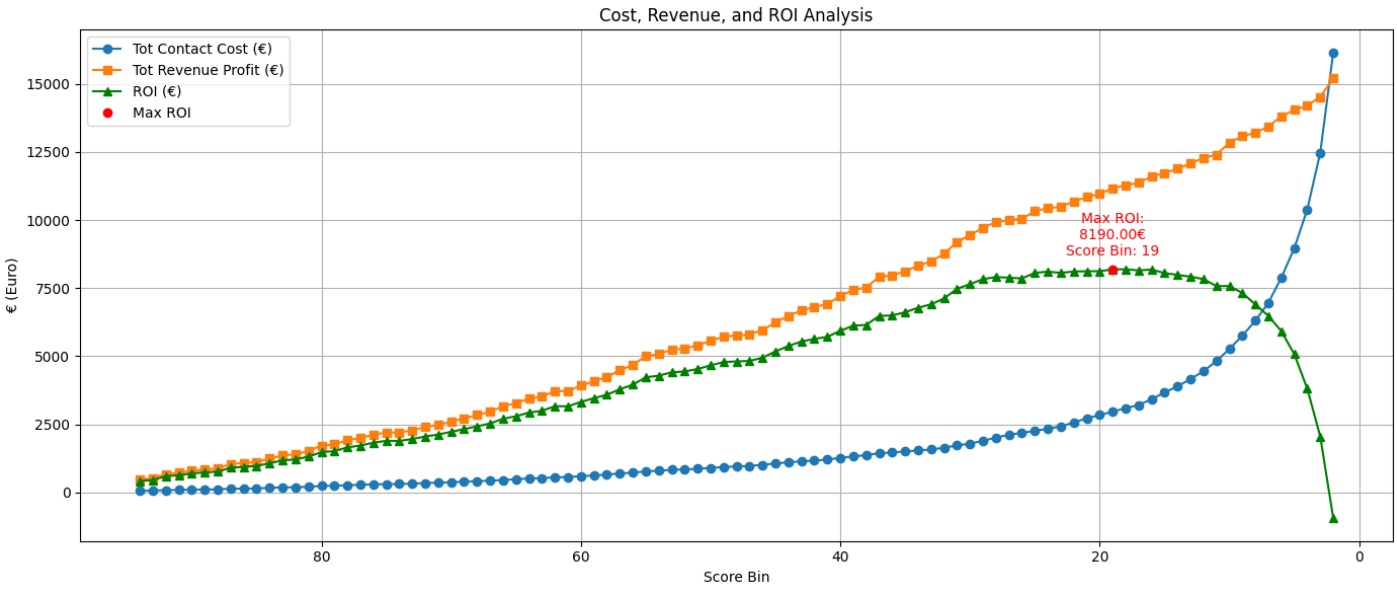
- Cumulative Precision and Recall
- Total Contact Cost (e.g., €5 per lead)
- Expected Revenue Profit (e.g., €40 per converted customer)
- Cumulative ROI = Cumulative Revenue – Cumulative Cost

📈 **Visualization**:

- I created a line plot showing:
    - Total Contact Cost

- Total Revenue Profit
- Cumulative ROI

- The x-axis was reversed (from score bin 99 to 1) to highlight the most confident predictions.
- The bin with the highest ROI was marked to indicate the most cost-effective threshold.



🔍 **Focused insight:**
As a concrete example, I extracted and presented detailed values (precision, recall, ROI, total cost and profit) for score bin 19—an effective way to communicate actionable insights to business stakeholders.

Values for Score Bin 19:

- Cumulative Contacts (Segnalati): 594
- Cumulative Hits (True Positives): 279 over 397
- Precision: 0.47
- Recall: 0.70
- Total Contact Cost: 2,970 €
- Total Revenue Profit: 11,160 €
- ROI (Return on Investment): 8,190 €

CONTACTS

[Linkedin] *(24) Alessio Smerilli | LinkedIn*

[Kaggle] Alessio Smerilli | Contributor | Kaggle

[GitHub] *alessiosmerilli (Alessio Smerilli)*

[Mail] *alessiosmerilli@hotmail.it*