Machine Learning-Based Analytics for Customer Satisfaction Prediction at Santander Bank

This report presents an end-to-end machine learning pipeline for predicting customer dissatisfaction at Santander Bank, with a focus on feature engineering, model interpretability, and business impact.

Table of Contents

Introduction

1.1 Project Overview

Business Background:

In practical banking operations, customer satisfaction is a key metric for evaluating service quality and user loyalty. However, banks often only have access to highly anonymized, structured customer data, lacking direct business-explanatory variables. Moreover, dissatisfied customers rarely express their dissatisfaction before leaving.

This project is based on anonymized customer data provided by Santander Bank, with the goal of predicting each customer's satisfaction status with the bank’s services. The dataset contains hundreds of anonymized numerical features with no clear business meaning, which presents significant challenges for data understanding and modeling.

Core Tasks of the Project:

Predict the probability that each customer in the test set is a "dissatisfied user (TARGET = 1)." Evaluate models using robust metrics such as ROC-AUC.

Project Objectives:

- Identify customers who are likely to churn or file complaints in advance.
- Explore feature processing methods under conditions where features lack business semantics.
- Validate model performance on unseen data to reduce manual inspection costs and improve management efficiency.

1.2 Data Description

Data File	Description	Dimensions
train.csv	Contains customer features and target variable	76,020rows × 371columns
test.csv	Contains only customer features.	75,822rows × 370columns

Data Characteristics:

1. Large number of features, high dimensionality (370+ features).
2. Presence of duplicate, zero-variance, and highly correlated features.
3. Lack of business interpretability, requiring reliance on statistical features and pattern recognition.
4. Target variable is binary, predicting the probability of dissatisfied users.

Analysis Approach and Methodology

2.1 Overall Analysis Process:

This study aims to predict the probability of customer dissatisfaction and establishes an end-to-end data analysis workflow.

First, exploratory data analysis (EDA) is conducted to identify data quality issues and variable distribution characteristics.

Next, feature engineering is designed based on business intuition and statistical properties.

Finally, multiple models are trained and evaluated to select the one with the best generalization performance.

2.2 Methods and Model Selection Rationale

Baseline Models

- **Decision Tree:** Highly interpretable; used to verify whether the feature engineering is reasonable.
- **Random Forest:** Requires minimal hyperparameter tuning. While a single decision tree has high variance and is prone to overfitting, random forests significantly reduce variance through bagging and feature randomization.

Complex Models

- **XGBoost:** Fits residuals sequentially via boosting, offering better bias control than bagging. Regularization and second-order gradients help prevent overfitting. Automatically handles missing values.
- **LightGBM (LGBM):** Maintains GBDT performance while significantly improving training speed and memory efficiency through histogram-based algorithms and leaf-wise growth strategy. Well-suited for high-dimensional sparse features and practical for real-world large datasets.

Rationale for Evaluation Metrics

The task requirements specify ROC-AUC as the primary metric.

Also ROC-AUC is robust to class imbalance and effectively measures the model's overall discriminative ability.

Rationale for Feature Engineering and Encoding Methods

- **Numerical Features:** Retain original values and construct statistical features.
- **Categorical Features:**
    - **One-Hot Encoding:** Used for low-cardinality categories.
    - **Response Encoding:** Used for high-cardinality categories to mitigate dimensionality explosion.

Combining multiple encoding strategies improves the model's ability to represent different feature types.

2.3 Technical Framework Design

The technical framework adopts a modular design, consisting of six modules:

- Data cleaning,
- Exploratory analysis,
- Feature engineering,
- Model training,
- Model evaluation,
- Model interpretation.

Each module is independent yet logically connected, ensuring the systematic, reproducible, and interpretable nature of the analysis process.

Exploratory Analysis and Feature Engineering

3.1 Data Quality Assessment

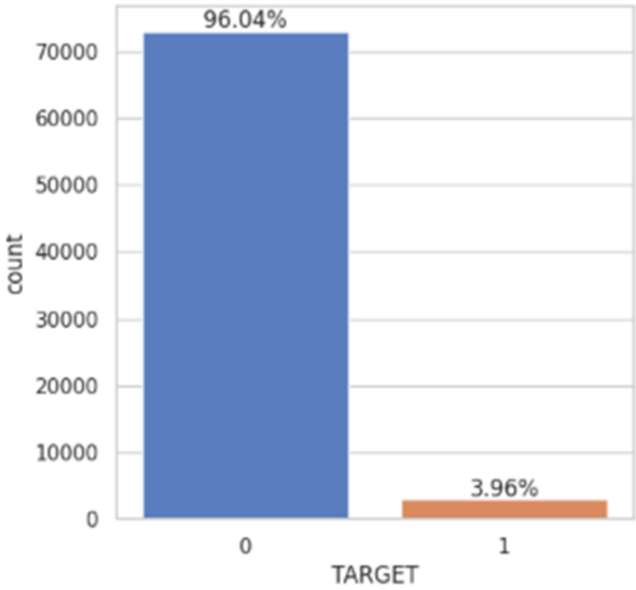
Overview of Data Distribution:

Number of samples in the training set: 76,020

Number of samples in the test set: 75,822

Total number of features: 371

Proportion of unsatisfied customers: 3.96%



Key Challenges Identified:

- Severe class imbalance: Unsatisfied customers account for only 3.96% of the dataset.
- Feature anonymization: Due to the lack of business semantics, modeling relies primarily on statistical patterns rather than domain-driven interpretation.
- High-dimensional feature space: The dataset contains a large number of raw features, many of which are sparse or weakly informative.

3.2 Exploratory Data Analysis (EDA)

var3:

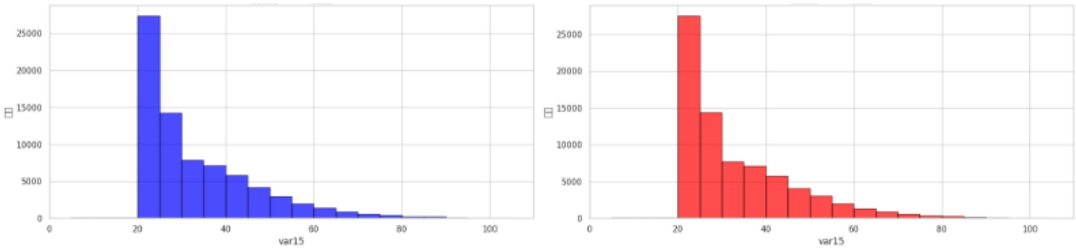
More than 97% of the observations take the value of 2 in both the training and test sets. Although the variable has a relatively high proportion of missing values, replacing missing values with 2 does not alter the distribution of the target variable conditional on this feature.

Accordingly, missing values are imputed with **2** as the chosen preprocessing strategy.

var15:

The variable contains abnormally high values. Customers under 30 account for approximately **56.15%** of the training set and **56.58%** of the test set.

To capture potential nonlinear effects, this variable is subsequently transformed using **binning**.



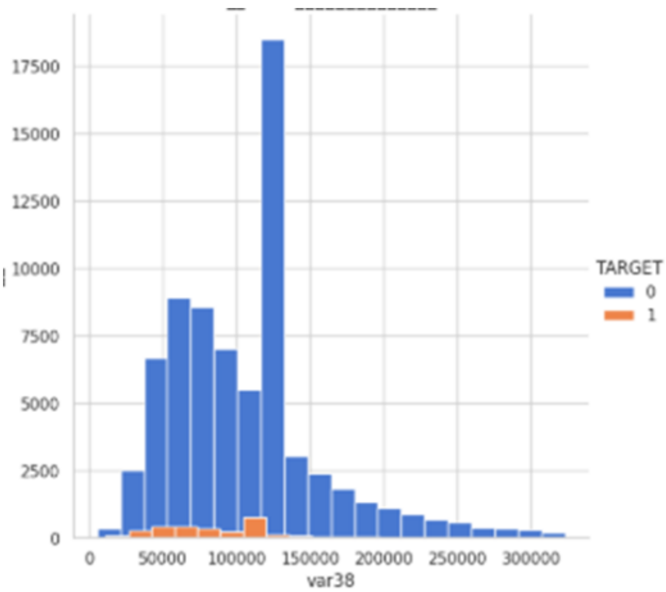
After applying equal-width binning, the results indicate that the majority of unsatisfied customers are concentrated in the 25–45 and 45–65 ranges.

var38:

Exhibits sharp variations in the ranges [0, 10] and [90, 100].

Displays a pronounced right-skewed distribution.

A logarithmic transformation is applied to mitigate the influence of extreme values.



var36 and var21:

These variables contain a small number of unique values and therefore are left **unchanged**.

Features with imp_ and saldo_ prefixes:

- A large proportion of these features contain over 80% zero values
- The non-zero values exhibit a strong right-skewed distribution
- Their distributional characteristics are similar to those of var38

A logarithmic transformation is applied to the non-zero values to increase the model's sensitivity to meaningful numerical variations.

Features with num_ prefix:

- Most features contain only a small number of unique values (e.g., around 5)
- When the value equals 0, the proportion of unsatisfied customers is relatively high

These features are treated as categorical variables, with a threshold on the number of unique values used to guide the encoding strategy.

Features with ind_ prefix:

- Binary features taking values of 0 or 1, with approximately 96% of observations equal to 1
- Individual predictive power is relatively weak

No additional preprocessing is applied; these features are retained in their original form.

3.3 Feature Engineering Strategy

1. Creation of New Features:

- EDA revealed a high prevalence of zero values across features. Therefore, a new feature was created to capture the count of zero or non-zero occurrences for each sample row.
- For highly right-skewed `imp` and `saldo` prefix features, new features were generated by calculating the mean value of these features for each unique value within the interval (50, 210].

2. Feature Selection:

As the feature space expanded, the following selection criteria were applied:

- Remove low-correlation features: Features exhibiting low correlation with the target variable `TARGET` were discarded.
- Remove highly correlated features: Features demonstrating high pairwise correlation with one another were eliminated to reduce redundancy and multicollinearity.

3. Data Transformation:

- Logarithmic Transformation: Applied to `imp` and `saldo` prefix features.

Their distributions were highly right-skewed, which is suboptimal for linear model assumptions and gradient-based optimization processes. The log transform helps mitigate skewness.

4. Feature Encoding Strategy:

A "divide and conquer" encoding strategy was employed to balance dimensionality control with information preservation and leakage risk mitigation.

(1) One-Hot Encoding

Application: Applied to categorical variables with a unique value count in the range (2, 10].

Rationale: Avoids imposing an arbitrary ordinal relationship.

Process: Original features were dropped post-encoding to prevent information duplication.

(2) Response Encoding (Target Encoding)

Application: Applied to high-cardinality categorical features.

Method: Estimates the probability of `TARGET=1` for each category value using Laplace Smoothing.

Control: The smoothing parameter `alpha` is tuned to mitigate the risk of overfitting and target leakage.

5. Standardization (Scaling):

Purpose: To ensure features are on a comparable scale, thereby improving model stability and convergence speed during training.

Model Development and Optimization

4.1 Data Preparation and Evaluation Framework

Following the completion of feature engineering, the processed dataset is utilized for modeling.

- The training set is split into an 85% / 15% ratio for training and validation, respectively.
- Stratified sampling (stratify=y) is employed to ensure the proportion of dissatisfied customers (`TARGET=1`) remains consistent across both the training and validation sets.
- The validation set is used for model selection and hyperparameter tuning to prevent information leakage from the test set.

Evaluation Metric

- ROC-AUC is chosen as the primary evaluation metric.
- Rationale: Given the significant class imbalance in the data, AUC provides a more stable and comprehensive measure of a model's overall discriminative ability.

4.2 Baseline Model Development

To establish a performance benchmark and validate the effectiveness of feature engineering, the following baseline models were initially constructed:

Decision Tree

- Advantages: Intuitive structure and strong interpretability.
- Purpose:

  To quickly verify whether the feature engineering process introduced meaningful predictive signals.

  To analyze the influence of individual features on the prediction outcome.
- Limitations: High variance in a single tree, leading to limited generalization capability.

Random Forest

- Reduces model variance through Bagging and random feature subsampling.
- Offers greater robustness to noise and outliers compared to a single decision tree.
- Serves as a strong baseline model that requires minimal hyperparameter tuning, used to assess the overall effectiveness of the non-linear feature set.

Building upon the baseline models, gradient boosting models were introduced for further performance enhancement.

XGBoost

- Sequentially fits residuals within a Boosting framework, thereby reducing model bias.
- Incorporates second-order gradient information and regularization terms to effectively mitigate overfitting.
- Supports automatic handling of missing values, making it suitable for complex feature engineering scenarios.
- Demonstrates stable performance on structured tabular data and serves as a core model for comparison.

LightGBM

- Employs a histogram-based algorithm, significantly accelerating training speed.
- Utilizes a leaf-wise growth strategy, offering stronger expressive power in high-dimensional feature spaces.
- Well-suited for sparse, high-dimensional data and exhibits good engineering scalability.
- Represents a model type more aligned with deployable solutions in industrial environments.

4.3 Model Selection and Optimization Strategy

- All models are evaluated using a consistent data split method and evaluation metric (AUC).
- Generalization capability is assessed by comparing the AUC performance on the training set versus the validation set.
- The final model is selected based on stable performance and a lower risk of overfitting observed on the validation set.

Results Analysis and Evaluation

5.1 Model Performance Comparison

	model	train_auc	val_auc	n_features	n_samples	best_iteration
0	XGBoost	0.8609	0.8435	185	64617	882.0
1	LightGBM	0.8670	0.8392	185	64617	83.0
2	RandomForest	0.8535	0.8178	185	64617	NaN
3	DecisionTree	0.8775	0.7864	185	64617	NaN

Under consistent data partitioning and the evaluation metric (ROC-AUC), the predictive performance of multiple models was systematically compared. The key findings are summarized below:

Decision Tree

A single tree demonstrated strong fitting capability on the training set, but was highly sensitive to noise, leading to a noticeable decline in validation performance and weaker generalization.

Random Forest

Through bagging and feature randomization, it significantly reduced model variance and exhibited better stability compared to a single decision tree. However, there remains a performance ceiling when modeling complex nonlinear relationships.

LightGBM

This model showed clear advantages in computational efficiency and memory usage. However, under the current parameter configuration, its validation performance was slightly lower than that of XGBoost.

XGBoost

It achieved the best balance between bias and variance, with the smallest gap between training and validation AUC, demonstrating the strongest generalization ability.

Final Model Selection:

- Final Model: XGBoost
- Validation AUC: 0.8435
- Training–Validation AUC Gap: 0.0149

Under identical feature engineering and sample conditions, XGBoost achieved the highest AUC on the validation set while maintaining minimal generalization error. Hence, it was selected as the final model.

5.2 SHAP-Based Model Interpretability Analysis

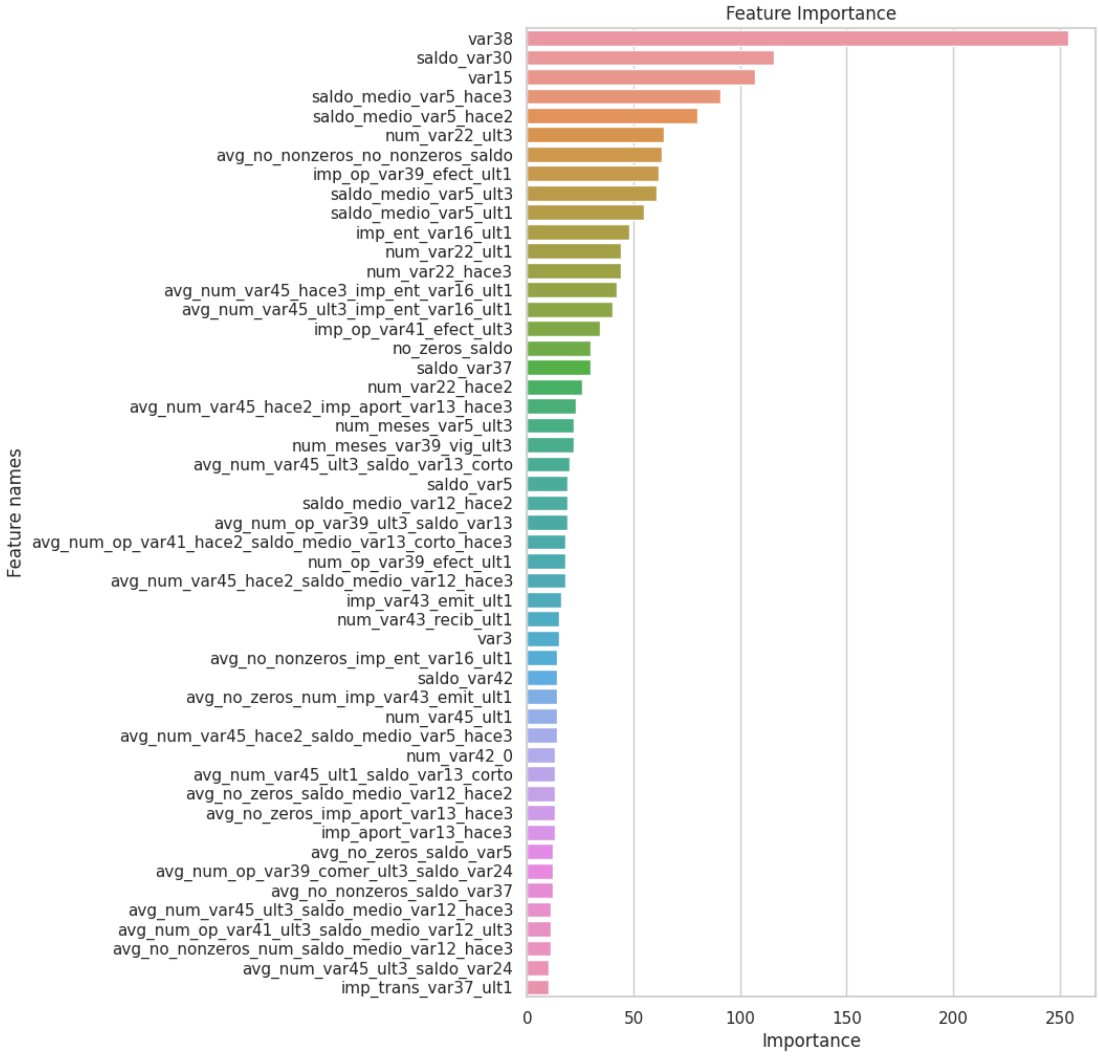


Figure presents the global feature importance derived from SHAP values, reflecting each feature’s average contribution to the model’s predictions.

The results indicate that features such as var_38, saldo_var30, and var15 play dominant roles in driving the model’s predictions, suggesting that customer dissatisfaction is primarily influenced by var_38 related characteristics

5.3 Interpretation of Key Features and Behavioral Patterns

Although all features are anonymized, their distributional characteristics and the ways in which they are utilized by the model allow for interpretation at the behavioral level.

Key Feature Interpretations (Model-based)

var15 (Age-related proxy feature)

This feature can be interpreted as a proxy for the customer's lifecycle stage, as its values are associated with systematic differences in product usage patterns and service expectations observed by the model.

var3 (Special status indicator–type feature)

Despite exhibiting an extremely high missing rate, this feature remains frequently utilized by the model after imputation. This suggests that the presence or absence of this variable may implicitly encode certain latent customer states or account-level flags that are informative for dissatisfaction prediction.

var38 (Proxy for account balance or activity intensity)

This feature displays a pronounced right-skewed distribution and retains strong discriminative power even after logarithmic transformation, indicating that it functions as a proxy for account activity intensity as perceived by the model.

Zero / non-zero usage statistics

These features capture the breadth of customer engagement across multiple products or functionalities. From a modeling perspective, they serve as behavioral proxies for overall account activeness and potential service disengagement.

Summary of Behavioral Patterns

Based on the SHAP-based interpretability analysis, several behavioral patterns can be observed from the model's perspective:

• Customer dissatisfaction does not appear to be an instantaneous or short-term state, but rather the result of accumulated long-term behavioral changes reflected in usage patterns.

• Persistent zero usage across multiple products or features is associated with gradual service disengagement, as identified by the model.

• Importantly, prior to the explicit manifestation of dissatisfaction, the model is able to capture clear early warning signals embedded in customer behavioral data.

5.4 Business Insights and Practical Value

The proposed model achieves an AUC of approximately 0.84 on the validation set, indicating a strong ability to distinguish potentially dissatisfied customers from consistently satisfied ones. This level of predictive performance provides a reliable foundation for downstream decision-making and operational deployment.

From a business perspective, the results suggest that customer dissatisfaction is not a random or isolated outcome, but rather follows identifiable behavioral and account-level patterns that can be systematically detected from historical data.

Importantly, the predicted probability of dissatisfaction can be utilized as a continuous risk score and integrated into real-time or near–real-time monitoring systems, enabling proactive and tiered intervention strategies before dissatisfaction fully materializes.

Recommended Applications

• Segment customers according to their predicted probability of dissatisfaction to support differentiated service strategies.

• Prioritize customer service and retention resources toward high-risk customer segments, thereby improving operational efficiency.

• Under identical resource constraints, focus interventions on customers with the highest predicted risk to achieve a higher return on intervention investment (ROI).

Conclusion and Future Work

6.1 Summary of Findings

This project aims to predict customer dissatisfaction in the banking domain using a high-dimensional dataset composed of anonymized features. A complete analytical pipeline was constructed, encompassing exploratory data analysis, feature engineering, and systematic multi-model comparison.

Through extensive feature engineering and model experimentation, XGBoost was selected as the final model. The model achieved an AUC of 0.8435 on the validation set, while maintaining strong generalization performance across different data splits.

Model interpretability was further enhanced through SHAP-based analysis. The results suggest that customer dissatisfaction does not occur randomly, but is instead associated with stable and identifiable behavioral patterns related to account activity, product usage coverage, and customer lifecycle–related characteristics as perceived by the model.

Overall, the proposed approach demonstrates strong interpretability and practical deployment potential. By transforming predictive outputs into a quantifiable risk score, the model can serve as an actionable early-warning tool to help banks proactively identify customers at elevated risk of dissatisfaction.

6.2 Project Limitations

Despite its strong predictive performance, several limitations of the current study should be acknowledged:

1. **Limited depth of business interpretation due to feature anonymization**
   As original business field definitions are unavailable, feature interpretation relies primarily on statistical properties and model-based behavioral inference rather than domain-specific semantics.
2. **Static modeling without explicit temporal dynamics**
   The current modeling approach is based on cross-sectional features and does not explicitly capture the temporal evolution of customer behavior or long-term interaction dynamics.
3. **Further room for model optimization**
   In this study, model stability and interpretability were prioritized. As a result, large-scale hyperparameter tuning and exhaustive model optimization were not fully explored.

6.3 Future Directions

Future work may extend this project in several directions:

• Incorporate time-series features or behavioral change-rate indicators to better capture customer behavior trajectories over time.

• Combine unsupervised customer segmentation (e.g., clustering) with supervised learning to develop segment-specific prediction models.

• Integrate model outputs into real-world business workflows to empirically evaluate the effectiveness of targeted intervention strategies.

• Subject to computational and resource constraints, further optimize model parameters or explore deep learning–based approaches for potential performance gains.