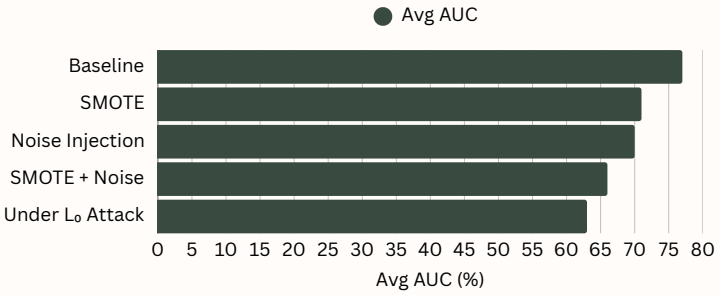
MODEL ROBUSTNESS EVALUATION CREDIT DEFAULT RISK

OBJECTIVE: ENHANCE CREDIT-DEFAULT DETECTION BY EVALUATING AND IMPROVING MODEL ROBUSTNESS.

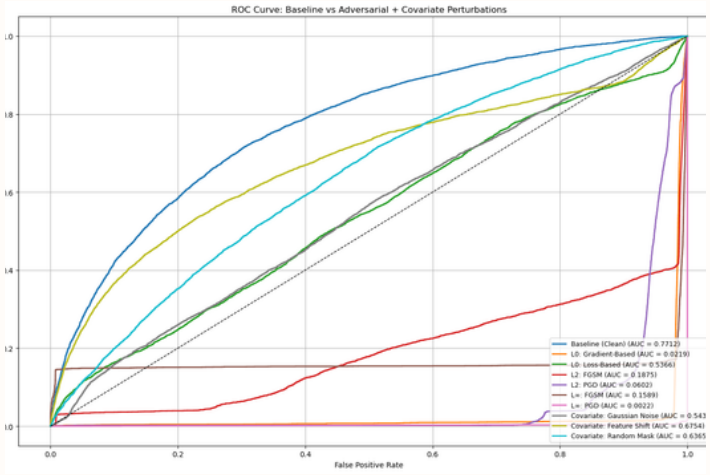
EVALUATION TECHNIQUES IMPLEMENTED



We applied several techniques to evaluate and improve model robustness. These included SMOTE for class balancing, Gaussian noise injection to simulate data variability, and covariate perturbations to test sensitivity. We also introduced adversarial attacks (L_0, L_2, FGSM, loss-based) to stress-test both TabNet and XGBoost under challenging scenarios.

DATASET & PREPROCESSING

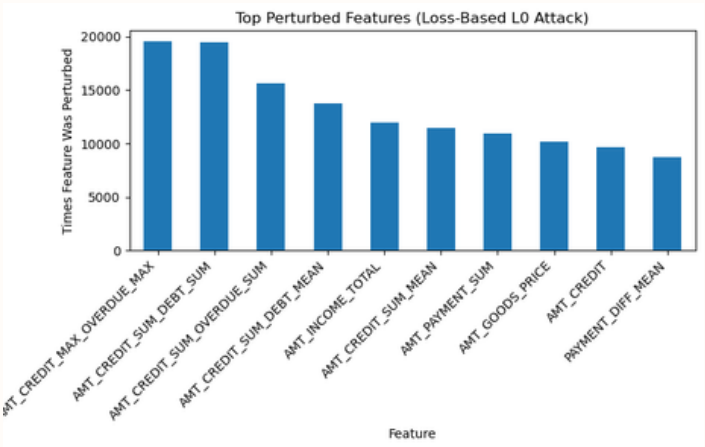
We used the Home Credit Default Risk dataset, which includes over 300,000 client applications and more than 100 features spanning financial behavior, credit history, and demographic data. Given the dataset's real-world complexity, we applied a rigorous preprocessing pipeline. This included dropping identifier and redundant columns, imputing missing values using median or mode depending on the feature type, and applying one-hot encoding to categorical variables. We also performed outlier filtering on continuous features and ensured consistent scaling for deep learning compatibility. These steps were essential to minimize data leakage, reduce noise, and standardize inputs across both the TabNet and XGBoost models for fair evaluation.



PERFORMANCE OF THE TABNET MODEL

TabNet initially achieved strong overall metrics (92% accuracy, AUC 0.77), but only 2% recall for defaulters, revealing a critical weakness. After applying SMOTE and Gaussian noise, recall improved to 66%, and the model showed stronger resilience under perturbations and L_0 attacks compared to XGBoost.

Metric	Before	After
AUC	0.77	0.66
Recall (Class 1)	2%	66%



PERFORMANCE OF THE XGBOOST MODEL

XGBoost delivered strong baseline results on majority class predictions but struggled with minority detection, with Class 1 recall at just 5%. After applying SMOTE and adversarial stress testing, performance gains were limited. The model remained highly sensitive to L_0 and gradient-based attacks, with AUC dropping significantly under perturbations.

Metric	Before	After
AUC	0.75	0.63
Recall (Class 1)	5%	22%

As part of this project, We developed a modular Python toolkit based on our research, enabling easy evaluation of model robustness. It allows teams to test model performance under noise, data shifts, and adversarial attacks, and can be integrated into any ML pipeline.

Prepared by: Yves Assali & Lucas Doan