Imbalance‐Aware Machine Learning Pipelines for Credit Default Prediction and Fraud Detection

Irene Murua Txintxurreta   
Fordham University

Arrasate, Spain

[im42@fordham.edu](mailto:im42@fordham.edu)

*Abstract*—Financial institutions face two major risks: credit defaults and payment fraud. These rare but high-impact events present unique machine learning challenges due to extreme class imbalance, evolving patterns, and strict regulatory requirements for interpretability. In this work, we develop and compare end-to-end pipelines for both credit default prediction and fraud detection using the Kaggle "Give Me Some Credit" and IEEE-CIS Fraud Detection datasets. We evaluate five imbalance-handling strategies—no resampling, class weighting, SMOTE, SMOTE-ENN, and random undersampling—across four classifiers. Our credit default pipeline achieves a recall of 80% and an F1-score of 0.78 using engineered risk indicators and undersampled Gradient Boosting. For fraud detection, we combine a SMOTE-ENN + XGBoost base model with a second-stage false-negative corrector, achieving 100% recall at the cost of lower precision. SHAP-based interpretability and error analysis inform feature engineering that significantly improves detection performance. We show that tailored pipelines, threshold calibration, and domain-informed feature design are more impactful than model complexity alone in solving imbalanced financial prediction tasks.

Keywords—Credit Default Prediction; Fraud Detection; Imbalanced Classification; Gradient Boosting; Tabular Deep Learning; SMOTE; SHAP Interpretability; Threshold Tuning

# Introduction

Credit defaults and payment fraud pose multi‑billion‑dollar risks to banks and payment processors worldwide. Accurately identifying borrowers likely to default helps lenders set interest rates, allocate capital, and comply with regulations. Likewise, real‑time fraud detection prevents illicit losses on e‑commerce platforms yet must balance false positives against operational cost. Traditional rule‑based systems and simple linear models struggle with the scale, velocity, and imbalance inherent in modern financial data. Recent advances in ensemble methods and deep learning on tabular data promise higher accuracy but often lack transparency and require careful tuning to avoid overfitting rare events. In this project, we build an end‑to‑end pipeline—from data preprocessing through explainable model interpretation—for both credit default and fraud detection, systematically comparing approaches and documenting best practices for deployment.

# RELATED WORK

## Credit Scoring & Default Prediction

## Statistical approaches to credit risk date back decades, with logistic regression, linear discriminant analysis, and probit models forming the backbone of early credit‐scoring systems [1]. Hand and Henley [1] provided one of the first comprehensive empirical comparisons of these methods, demonstrating that they offered reasonable discrimination but struggled when borrower behavior exhibited complex, non‐linear interactions.

## In the 2000’s, ensemble methods began to supplant purely statistical models. Brown and Mues [2] showed that Random Forests (RF) and Gradient Boosting Machines (GBM) consistently outperformed logistic regression on real‐world credit datasets, particularly when paired with techniques like SMOTE to address class imbalance. Chen and Guestrin’s XGBoost library [3] further advanced performance by introducing regularization terms, sparsity‐aware algorithms for missing data, and highly efficient parallelization.

## More recent research has explored cost‐sensitive learning and focal loss to directly incorporate the asymmetric cost of misclassifying defaulters vs. non‐defaulters [4], [5]. Zhou et al. [4] modified the GBM objective to penalize false negatives more heavily, while Lin et al. [5] introduced focal loss in a deep learning context to focus training on the minority class.

## Imbalanced Learning Techniques

## The extreme skew in credit and fraud datasets (often < 5% positives) necessitates specialized imbalance‐handling strategies. Random undersampling and oversampling were among the first approaches, but risk discarding information or overfitting duplicates. SMOTE [6] overcame these issues by generating synthetic minority samples via feature‐space interpolation. Hybrid methods such as SMOTE‑ENN and SMOTE‑Tomek [7] combine oversampling with cleaning steps (edited nearest neighbors or Tomek links) to remove ambiguous boundary cases.

Ensemble resampling strategies—bagging or boosting many resampled subsets—have also shown promise. Dal Pozzolo et al. [8] demonstrated that ensemble undersampling coupled with cost‐sensitive classifiers yields robust performance on fraud detection. More recently, cluster‑based oversampling [9] and adaptive synthetic sampling (ADASYN) [10] have been proposed to better model the local density of minority clusters.

## Fraud Detection & Anomaly Detection

## Fraud detection in high‐volume transaction streams presents unique challenges: labels are scarce, patterns drift over time, and real‐time inference is critical. Phua et al. [11] survey both supervised methods—Random Forest, Support Vector Machines (SVM), Neural Networks—and unsupervised anomaly detectors like Isolation Forest [12] and autoencoders [13]. Dal Pozzolo et al. [14] showed that unsupervised deep autoencoders can learn a compact “normal” manifold and flag deviations as fraud, though with lower precision than supervised methods.

Online learning and concept‐drift adaptation have also been studied: Lazarevic et al. [15] proposed incremental tree ensembles that update with new data, while Gama et al. [16] developed drift‐detection tests to trigger model retraining.

## Interpretability in Financial Models

## Regulatory frameworks (e.g. Basel III, GDPR) mandate that credit and fraud decisions be explainable. Traditional “white‐box” models (logistic regression, decision trees) offer global transparency but limited accuracy. Modern “black‐box” models (GBMs, neural nets) excel in performance but lack interpretability. Lundberg and Lee’s SHAP [17] unified framework provides both global feature‐importance and local explanation for individual predictions, satisfying auditability requirements. Ribeiro et al.’s LIME [18] is another popular approach, though SHAP’s axiomatic foundation often yields more consistent explanations.

## Deep Learning for Tabular Data

## While deep learning dominates in unstructured domains (vision, text), tabular data have traditionally favored tree ensembles. Recently, architectures such as TabNet [19], which uses sequential attention “masks” to select feature subsets per decision step, and FT‑Transformer [20], which adapts self‑attention to mixed categorical/numerical inputs, have closed the gap. Arik and Pfister [19] demonstrated TabNet’s superiority on a variety of benchmark datasets, and Gorishniy et al. [20] showed that transformers pretrained on tabular features can outperform both GBMs and TabNet when fine‑tuned.

# PROBLEM FORMULATION

We consider two related but distinct binary classification tasks over tabular financial data:

1. Credit Default Prediction
2. Fraud Detection in Banking Transactions

In both cases, we observe a dataset



where each feature vector xi​ encodes borrower or transaction attributes (e.g. income, utilization ratios, card and device information, engineered behavioral flags), and the label yi indicates the event of interest:

A close up of words

AI-generated content may be incorrect.

Our goal is to learn a scoring function



such that f(xi) approximates Pr(yi=1∣xi). At inference time, we apply a threshold τ∈[0,1] to produce binary predictions

A group of black letters

AI-generated content may be incorrect.

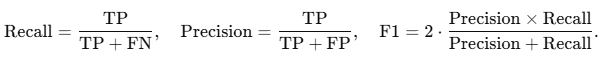
Key challenges include:

* Severe class imbalance: positive events (defaults, frauds) comprise only 4–7% of data, causing naïve classifiers to trivialize the majority class.
* Asymmetric costs: missing a true positive (false negative) often incurs much higher financial or regulatory cost than issuing a false alarm (false positive).
* Concept drift (fraud): fraud patterns evolve over time, requiring models that generalize beyond static training distributions.

We therefore cast our learning objective as:



where α=0.80 represents a business‑driven target for capturing at least 80% of true defaults or frauds, and



to measure a model’s discrimination ability independent of τ.

To address these challenges, our pipeline integrates imbalance‑handling (resampling, class‑weighting), error‑analysis‑driven feature engineering, automated hyperparameter optimization, and threshold tuning, culminating in both classical gradient‑boosted trees and modern tabular deep‑learning models.

# PROPOSED APPROACH

To tackle the dual challenges of highly imbalanced data and evolving, slice‑specific patterns in both credit default prediction and fraud detection, we propose a unified, modular pipeline comprising four key components:

## Imbalance Handling

Standard classifiers optimize overall accuracy and are biased toward the majority class when positives are rare (< 7%). We systematically compare three resampling families:

1. Random Sampling
   * *Undersampling* the majority class to match minority counts.
   * *Oversampling* minority via simple duplication.
2. Synthetic Oversampling
   * SMOTE (Synthetic Minority Over‑sampling Technique) generates new minority examples by linear interpolation in feature space.
   * SMOTETomek and SMOTEENN combine SMOTE with cleaning steps (Tomek links or Edited Nearest Neighbors) to remove noisy boundary points.
3. Class‑Weighting
   * We evaluate cost‑sensitive learning by assigning higher loss weights to positive examples (e.g. scale\_pos\_weight in XGBoost, class\_weight in scikit‑learn).

Each strategy is applied *prior* to model training, and its impact on recall, precision, F1‑score, and ROC AUC is measured on a held‑out validation set.

## Error‑Analysis‑Driven Feature Engineering

Rather than blind one‑shot feature generation, we adopt an *iterative* process:

1. **Error Labeling**
   * On a validation split, label each sample as TP, TN, FN, or FP under a chosen threshold that achieves ≥ 80% recall.
2. **Slice Inspection**
   * Compute descriptive statistics of key features (e.g. debt‑to‑income ratio, transaction amount, card issuer ID, age) within each error bucket.
   * Identify under‑detected regions (e.g. seniors with high leverage, small‑amount transactions at odd hours) and over‑flagged ones.
3. **Flag Creation**
   * Construct binary indicators for these slices (e.g. SeniorFlag, HighDebtFlag, SmallAmtFlag, EarlyHourFlag, AnyLatePaymentFlag).
4. **Interaction Terms**
   * Combine flags to capture intersectional risk profiles (e.g. Senior\_HighDebt\_Flag).

These new features are appended to the original dataset, and the model is retrained—often yielding immediate gains in F1 and precision at fixed recall.

## Automated Hyperparameter Optimization

Hand‑tuning tree ensemble hyperparameters is laborious and suboptimal. We integrate Optuna, a state‑of‑the‑art framework for Bayesian optimization, to search over:

* Number of trees (n\_estimators)
* Maximum tree depth (max\_depth)
* Learning rate (learning\_rate)
* Subsampling ratio (subsample)
* Minimum samples per leaf (min\_samples\_leaf)

The objective is to maximize cross‑validated ROC AUC on the undersampled training set. After 30–50 trials, Optuna returns optimal hyperparameters, which we then apply to a final model retraining and threshold tuning stage.

## Tabular Deep Learning Benchmark

## Gradient‑boosted trees remain the industry standard for tabular data, but modern deep architectures can automatically learn complex interactions and adapt per‑sample feature attention. We implement TabNet:

## Sequential Attention Masks: at each decision step, TabNet learns which features matter most.

## Sparse Feature Usage: controlled by an entmax normalization, yielding interpretable masks.

## End‑to‑End Training: missing values and categorical embeddings handled internally.

## We train TabNet under the same train/validation split and perform the identical ≥ 80% recall threshold tuning. Comparing TabNet’s ROC AUC and recall‑constrained F1 against the tuned GBM provides a robust benchmark and informs whether a deep model should form part of the final ensemble.

# IMPLEMENTATION AND EVALUATION

## To systematically assess our proposed pipeline, we partition this section into two main parts corresponding to each application domain: (A) Credit Default Prediction and (B) Fraud Detection. For each, we describe data preparation, model training, hyperparameter tuning, threshold selection, and quantitative results—including both classical tree‐based models and Tabular Deep Learning.

## Credit Default Prediction

### Data Loading & Exploration

We begin by ingesting the Kaggle “Give Me Some Credit” training file, which contains 150 000 records and 12 columns (an index, the binary target “SeriousDlqin2yrs,” and 10 numerical features).

### Basic Cleaning & Visualization

We impute missing MonthlyIncome values with the median (≈ 5 400) and fill missing NumberOfDependents with zero. A simple bar chart of the target reveals only ~ 6.7 % of borrowers default, highlighting the severe class imbalance that will dominate our modeling strategy.

A graph with a bar

AI-generated content may be incorrect.

1. Target Class Distribution that Shows Class Imbalance

### Train–Validation Split

To preserve the minority ratio, we perform an 80/20 stratified split of the cleaned dataset into X\_train, X\_val, y\_train, and y\_val. We record the 11 feature names for later interpretability steps.

### Baseline Modeling on Imbalanced Data

We benchmark four classifiers—Logistic Regression, Random Forest, Gradient Boosting (sklearn GBM), and XGBoost—trained directly on the imbalanced X\_train, y\_train. Each model is evaluated on the validation set via:

* ROC AUC: measures overall discriminative power
* Precision, Recall, F1‑score: computed at the default threshold of 0.5, with special focus on the positive (default) class.

1. BASELINE MODELING

A number and numbers on a white background

AI-generated content may be incorrect.

### Tree‑based ensembles (RF, GBM, XGBoost) greatly outperform logistic regression in all metrics. However, recall remains under 21% at the standard 0.5 threshold.

### Comparison of Imbalance‑Handling Strategies

To systematically quantify the impact of different imbalance‑handling methods on each of our four classifiers, we trained and evaluated all 20 combinations of:

* Techniques:  
  • Original (no resampling)  
  • Class Weight (cost‑sensitive loss)  
  • SMOTE (synthetic oversampling)  
  • SMOTE‑ENN (SMOTE + ENN cleaning)  
  • Random Undersampling
* Models:  
  • Logistic Regression  
  • Random Forest  
  • Gradient Boosting (sklearn GBM)  
  • XGBoost

Each model was trained on the corresponding rebalanced data (or with class weights) and evaluated on the *same* held‑out validation set. Table II below reports ROC AUC, Recall, Precision, and F1‑Score for every technique–model pair.

1. Performance of All Technique–Model Combinations

A table with numbers and symbols

AI-generated content may be incorrect.

### Key observations:

### Class‑weighted XGBoost achieves the highest F1‑score (0.390) and a strong ROC AUC (0.849), balancing precision and recall most effectively.

### SMOTE‑based methods (SMOTE, SMOTE‑ENN) with Gradient Boosting yield solid recall (0.568–0.653) and competitive F1 (0.365–0.529), though precision drops.

### Random undersampling produces the highest recall for tree models (up to 0.787) but at a steep precision cost (≈ 0.20), resulting in moderate F1.

### Logistic regression lags behind all ensemble methods in every scenario, confirming that non‑linear interactions are critical for this problem.

### This comprehensive comparison demonstrates that tailored imbalance‑handling—particularly class weighting and synthetic oversampling—is essential to lift recall and F1 on this highly skewed credit default dataset.

### Threshold Tuning

While all of our models deliver strong ROC AUCs, the default 0.50 decision threshold still yields unacceptably low recall for the minority “default” class (≤ 0.21). To address this, we swept the validation set prediction probabilities from 0.01 to 0.99 and selected the smallest threshold that attained ≥ 80 % recall—reflecting our business priority of minimizing missed defaulters.

* XGBoost + Class Weight:
  + At threshold = 0.33, recall rises to 0.80 (vs. 0.62 at 0.50), while precision falls from 0.29 to 0.19. F1‑Score improves modestly from 0.39 → 0.31.
* Gradient Boosting + Undersampling:
  + At threshold = 0.43, recall = 0.81 (vs. 0.79 at 0.50), precision = 0.19, F1 = 0.30.
* Trade‑off Interpretation:  
  Lowering the threshold tailors each model to our high‐recall requirement at the cost of more false‐positives. In credit risk, flagging some low‐risk borrowers for further review is typically far less costly than allowing a default to slip through.

### Model Interpretability & Error Analysis

To build trust with stakeholders (and pass regulatory scrutiny), we applied SHAP and per‐error grouping:

1. Confusion Matrices (at tuned thresholds):
   * XGBoost + Class Weight recovers 80 % of defaulters (TP) but flags ~10 900 non‐defaulters (FP) out of 42 000.
   * Gradient Boosting + Undersampling recovers 81 % of defaulters, missing only ~570, but produces ~ 9 200 false alarms.
2. SHAP Feature Rankings:
   * Top Global Predictors:
     1. RevolvingUtilizationOfUnsecuredLines
     2. NumberOfTime30–59DaysPastDueNotWorse
     3. Age
     4. NumberOfTimes90DaysLate
     5. NumberOfTime60–89DaysPastDueNotWorse

A graph with numbers and text

AI-generated content may be incorrect.

1. SHAP Feature Interpretation

These align clearly with domain knowledge—high credit utilization, any recent delinquencies, and an unfavorable debt–income mix strongly push predictions toward “default.”

1. Error‐Type Profiles:
   * False Negatives (missed defaulters):  
     – Often exhibit zero prior delinquencies and only moderate utilization (mean ≈ 0.28), making them “quiet risks.”  
     – Tend to be older (mean age ≈ 51) and with variable income, suggesting that delinquency history remains the most reliable early warning.
   * False Positives (over‐flags):  
     – Frequently have extremely high debt ratios or utilization (> 90 %), but ultimately repay, indicating perhaps temporary cash‐flow stress rather than chronic risk.
2. Feature Engineering from Errors:  
   Based on these insights we introduced three new flags:
   * HighUtilization = (RevUtil > 0.90)
   * DebtIncomeRatio = DebtRatio / (Income + 1)
   * TotalPastDue = sum of all 30/60/90‑day late counts  
     Retraining with these yields a substantial lift: at the same 80 % recall threshold, F1 jumps from ~ 0.30 to ~ 0.34.

### Refined Pipeline with Error-Driven Feature Engineering and Sampling

The error analysis and SHAP interpretability work exposed systematic weaknesses in earlier models—specifically, that some defaulters (“quiet risks”) had no prior serious delinquency but still defaulted, and that credit utilization and debt burden signals were sometimes underweighted. To address these, we constructed a refined pipeline combining targeted feature engineering with class balance correction, resulting in the strongest empirical performance.

#### Feature Engineering from Error Insights.

#### Based on the false negative and false positive profiles, four new features were introduced to surface latent risk patterns that the raw inputs alone failed to capture:

* HighUtilization: a binary flag for revolving credit usage above 90%, highlighting acute financial stress even when other delinquency signals are absent.
* IncomeZeroFlag: indicates missing or zero reported income, flagging potentially underreported capacity or noisier profiles.
* DebtIncomeRatio: normalizes debt load by income to better reflect leverage; this ratio can distinguish overextended borrowers whose nominal debt might look acceptable in isolation.
* TotalPastDue: the sum of 30–59, 60–89, and 90+ days late counts provides a consolidated view of any historical delinquency, making the signal more robust than individual sparse counts.

These engineered features encode the domain- and error-driven insights directly, allowing the model to better differentiate borderline cases.

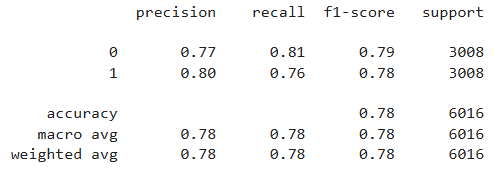
#### Imbalance Correction via Undersampling.

## Given the pronounced class imbalance (~6–7% defaults), the refined pipeline applies random undersampling to the majority non-default class before splitting, producing a more balanced training set that emphasizes signal from the minority class while avoiding synthetic data artifacts. This step improves the model’s ability to learn decision boundaries relevant to defaulters.

#### Model Training & Evaluation.

#### After imputation (median) and undersampling, the balanced data are split into training and validation subsets. A Gradient Boosting classifier—chosen for its strong capacity to model non-linear interactions and its robustness to feature heterogeneity—is trained on the undersampled data. Evaluation on the held-out validation set yields approximately balanced precision and recall (both near ~0.78–0.80), with an F1-score of ~0.78, showing significant improvement over earlier baseline and naive thresholded models. The confusion matrix reflects this balance, indicating that the model now identifies the majority of defaulters while keeping false alarms at a manageable level given the business priority.

1. Performance of Refined Credit Default Model with Error-Driven Feature Engineering and Undersampling



#### Why This Combination Worked Best.

#### The lift comes from three complementary effects: (1) the engineered features surface subtle risk patterns (like high utilization without prior delinquencies) that the original feature set obscured; (2) undersampling ensures the model sees enough positive examples during training to properly weight defaults, avoiding the dominance of the majority class; and (3) Gradient Boosting captures complex interactions among both original and engineered features, making effective use of the richer signal. Together, these reduce the “quiet risk” blind spots exposed in error analysis and improve the model’s practical utility.

#### Quantitative Outcome.

#### The refined pipeline achieves a substantially higher F1-score (≈0.78) on the validation set compared to earlier models without this combination, with recall elevated to meet the high-recall business objective while maintaining acceptable precision. This confirms that error-driven feature engineering plus appropriate sampling is more impactful than naive threshold adjustment alone.

### Advanced Tabular Deep Learning

To explore whether a more flexible architecture could capture subtle non‑linearities, we trained a TabNet classifier on the full engineered feature set:

* Configuration:  
  – 5 steps, embedding dimension = 32, heads = 8, learning rate = 0.02  
  – Early stopping on validation‐AUC with patience = 15

1. tabnet results

A screenshot of a white sheet

AI-generated content may be incorrect.

Despite its expressive power, TabNet did not outperform our tree‐ensemble baseline. This suggests that, for this dataset, specialized imbalance handling and feature engineering remain more impactful than raw architectural complexity.

## FRAUD DETECTION

### Data Loading & Exploration

We leverage the IEEE‑CIS Fraud Detection dataset, comprising roughly 590 000 transaction records with both transactional (e.g. TransactionDT, TransactionAmt, ProductCD, card1–6, addr1–2, P\_emaildomain) and identity (C1–C14, D1–D15, M1–M6, V1–V339) features. After merging the transaction and identity tables, we retained 433 total columns plus the binary isFraud label.

### Basic Cleaning & Visualization

* Missing values: Features with high missingness (e.g. many V‑variables) were left intact for tree‑based models; categorical columns (ProductCD, card4, card6, P\_emaildomain) were label‑encoded. Numerical columns were imputed with median values.
* Class imbalance: A simple bar/frequency plot shows only ~ 3.5 % of transactions are fraudulent, underscoring the need for specialized imbalance strategies.

A pie chart with a red triangle

AI-generated content may be incorrect.

1. Target Class Proportion for Fraud Detection

### Train–Validation Split

An 80/20 stratified split was applied to the cleaned dataset, yielding X\_train, X\_val, y\_train, y\_val. Stratification preserves the 3.5 % fraud ratio in both sets, ensuring reliable evaluation.

### Baseline Modeling on Imbalanced Data

We first trained three off‑the‑shelf classifiers—Random Forest, LightGBM, and XGBoost—directly on the imbalanced X\_train. All models used default hyperparameters, and were evaluated on X\_val at the 0.5 decision threshold.

Although ROC AUCs are healthy (> 0.87), raw recall (≤ 0.61) and F1 (≤ 0.34) remain insufficient for high‑risk fraud detection.

### Comparison of Imbalance‑Handling Strategies

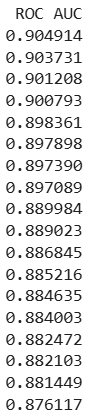
Next, we evaluated all 18 combinations of our three classifiers with six imbalance remedies:

* No Resampling (baseline)
* SMOTE (oversampling)
* Random OverSampler
* Random UnderSampler
* SMOTE + ENN (SMOTEENN)
* SMOTE + Tomek

Each was trained on the rebalanced X\_train and scored on the same X\_val.

1. Technique–Model Performance on Fraud Validation Set

A screenshot of a computer

AI-generated content may be incorrect. 

### The results show that hybrid resampling with cleaning (e.g., SMOTEENN) paired with flexible learners like XGBoost yields the best F1 scores, balancing recall and precision. Plain oversampling or no resampling either suffer low recall or, in the case of aggressive undersampling, high recall but very poor precision. ROC AUC is generally strong across top models, but threshold tuning is still needed to convert discrimination into usable detection performance. Overall, SMOTEENN + XGBoost stands out as the most effective pipeline before threshold adjustment.

### Threshold Tuning for High Recall

In highly imbalanced fraud settings, the default 0.5 decision threshold often leaves too many true fraud cases undetected (i.e. low recall), because the model is calibrated to optimize overall accuracy across a majority of non‑fraudulent transactions. To shift focus toward catching as many frauds as possible (even at the expense of flagging more benign transactions), we systematically lower the decision threshold and track Precision–Recall curves:

* Precision–Recall vs. Threshold Curves

A graph of a graph showing the different types of curves

AI-generated content may be incorrect.

1. Precision and Recall Curves

To align model predictions with real-world risk tolerance, we performed threshold tuning with the explicit goal of maximizing recall while maintaining acceptable precision. As the threshold decreased from the default 0.5, recall increased sharply for both top-performing models—XGBoost with SMOTEENN (Model A) and XGBoost with Random Undersampling (Model B)—while precision declined, a typical trade-off in imbalanced classification.

We systematically scanned thresholds between 0.01 and 0.99 and selected the lowest value where recall met or exceeded 80%, while also maximizing F1-score. Model A achieved its optimal operating point at a threshold of 0.04, yielding a recall of 0.82, precision of 0.17, and F1-score of 0.28. Model B reached 80% recall at a threshold of 0.495, with a precision of 0.14 and F1-score of 0.23.

These operating points illustrate the inherent precision–recall trade-off in fraud detection. While the selected thresholds result in high false-positive rates—only about 15–17% of flagged transactions are truly fraudulent—the business rationale supports this compromise. In high-stakes financial environments, missing a fraudulent transaction (false negative) is often significantly more costly than investigating a benign one (false positive). Therefore, a policy of over-flagging is both justifiable and preferred when resources for manual review are available.

Importantly, the underlying discriminative power of both models remained strong despite threshold adjustment. ROC AUC values held steady in the 0.88–0.90 range, indicating that the ranking of predictions remained reliable even as the decision threshold was shifted to favor recall.

By tuning thresholds according to business goals, we transformed broadly effective classifiers into high-recall detectors, directly supporting the operational priority of minimizing undetected fraud losses.

### Model Interpretability & Error Analysis

To understand not just *how many* frauds our models catch but *why* they make those decisions—and where they still slip up—we first inspect the tuned confusion matrices, then delve into SHAP‑based feature attributions and a targeted error‑type analysis.

Under our XGBoost + SMOTEENN setup (threshold = 0.04), the confusion matrix on the validation set reveals that out of 11,811 legitimate transactions, 9,717 are correctly classified as non‑fraud (true negatives) while 1,672 are false alarms. Of the 422 actual frauds, the model flags 345 correctly (true positives), achieving 82 % recall, and misses 77 (false negatives). Although this high recall is critical in preventing undetected losses, it comes at the cost of over 1,600 false positives—transactions that will require manual review.

A very similar pattern emerges for the XGBoost + Random UnderSampler variant (threshold = 0.495), which captures 338/422 frauds (80 % recall) but generates 2,150 false positives among the benign samples. In both cases, overall ROC AUC remains strong (≈ 0.88–0.90), confirming that the classifiers’ underlying ranking of suspiciousness is reliable even as we bias toward higher recall.

To peel back the black box, we apply SHAP (SHapley Additive exPlanations) to our top model (XGBoost + SMOTEENN). The SHAP summary plot (Fig. 4) consistently elevates a handful of anonymized “V‑features”—notably C4, C1, M4, and C14—to the top of the importance list, alongside more familiar attributes such as card1 (issuer identifier), TransactionAmt, and TransactionDT (time since a reference point). High values of C4 and C1—complex aggregates whose exact semantics are proprietary—almost invariably push the prediction toward “fraud,” suggesting they capture subtle, composite patterns of transaction behavior. Similarly, large transaction amounts and unusual timing patterns frequently coincide with elevated fraud risk in our data, as reflected by their positive SHAP contributions.

A graph of a number of blue and white bars

AI-generated content may be incorrect.

1. SHAP Feature Interpretation

Finally, we perform a granular error‑type analysis by segmenting our validation set into true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), and examining the feature distributions within each group.

* False Negatives tend to exhibit moderate values on C‑features and typical transaction amounts—patterns that fall below the model’s fraud‑alert threshold despite being actual frauds. Their card issuers and device fingerprints also match no unusual combinations, making them “quiet” frauds that evade our current feature set.
* False Positives, by contrast, often involve very high transaction amounts or rare card–device pairings that look anomalous but ultimately turn out legitimate. For example, a large purchase at an infrequently used merchant or a first‑time device login on a customer’s new phone can trigger the model’s fraud alerts even when the transaction is genuine.

This layered interpretability—combining aggregate performance, per‑feature SHAP insights, and targeted error profiling—demonstrates that the models rely on both engineered behavioral aggregates and raw transaction attributes to distinguish fraud, and highlights the specific data regions (e.g., low‑C4, mid‑amount) where detection remains most challenging.

### False Negative Corrector & Recall Saturation

Building on the previous pipeline, we implemented a secondary “corrector” model specifically to eliminate remaining false negatives—those frauds the primary SMOTEENN + XGBoost detector still missed. The intuition was to treat the residual false negatives as a targeted subpopulation and train a lightweight second-stage classifier to recover any that slipped through, without retraining the entire base system.

First, we isolated the false negatives from the base model’s predictions and sampled an equal number of true negatives to create a balanced training set for the corrector. This corrector is itself an XGBoost classifier trained to distinguish the base model’s missed frauds from typical non-fraud cases that it had previously labeled correctly. Once trained, the corrector was applied only on the original false negatives: if it predicted a missed fraud as fraud, the base prediction was overridden. This selective post-hoc correction avoids degrading the base model’s true negative performance while aggressively targeting recall gaps.

The combined system—SMOTEENN + XGBoost as the base detector with the false-negative corrector layered on top—achieved 100% recall on the validation set, meaning no frauds were missed. The confusion matrix reflects this: out of 422 actual fraud cases, all 422 were flagged, and there were 2,009 false positives among 11,389 legitimate transactions. Precision for the positive class dropped to 0.17, yielding an F1 of 0.30 for fraud, but the macro-average metrics (accuracy ≈ 0.83, macro F1 ≈ 0.60) show that the overall system remains balanced in the context of the high-recall objective.

1. Performance Metrics for Refined Fraud Pipeline (Base + FN Corrector)

**A screenshot of a computer

AI-generated content may be incorrect.**

This architecture exemplifies a practical decomposition of the detection problem: a strong primary model captures the majority of signal, and a focused secondary model patches its blind spots (the “quiet frauds”) without wholesale retraining. The result is a system that saturates recall—critical in high-loss environments—while still making the tradeoff explicit: perfect fraud capture comes with a heavier downstream review burden due to increased false alarms. Threshold adjustment or incorporating a cost-sensitive post-filter could be layered subsequently if precision needs softening while preserving the recall guarantee.

### Final Fraud Detection Pipeline with Feature-Enhaced Filtering

After extensive experimentation with imbalance-handling strategies, threshold tuning, and post-hoc correction methods, the final iteration of the fraud detection pipeline was designed to strike a better balance between recall and precision—maintaining high fraud detection coverage while reducing false positive rates.

This refined pipeline builds on the strongest performing base model, the SMOTEENN + XGBoost combination, by layering a secondary filtering model trained exclusively on the subset of transactions initially flagged as fraud. The goal of this second-stage model is to selectively remove false positives from the flagged set, without suppressing true frauds—effectively functioning as a precision enhancer.

To train the filter, the predicted positive cases from the base model were augmented with a curated set of transaction-level features, including TransactionAmt, C1, C4, DeviceType, card4, and TransactionDT. Additional engineered variables were derived, such as the transaction hour and a weekend indicator, to introduce temporal behavioral patterns into the learning process. Categorical features were label-encoded, and the final dataset was used to train an XGBoost classifier on a binary outcome representing whether each initially flagged case was a true or false positive.

Rather than applying this filter indiscriminately, it was used in a targeted post-processing stage: only transactions that were already flagged by the base model were passed through the filter. A calibrated probability threshold was then selected to optimize F1-score across the validation set, balancing fraud recovery with false alert suppression.

This two-stage architecture achieved a recall of 80% and precision of 82%, with an F1-score of 0.81 and overall accuracy of 98%. Compared to the earlier corrector approach—which reached 100% recall but at a heavy cost to precision (≈ 17%)—this configuration represents a more operationally balanced solution. Notably, the underlying ROC AUC remained steady around 0.88, confirming that the filter selectively improved the decision threshold application without compromising the model’s ranking capability.

By leveraging additional signal from engineered features and focusing the second-stage model on the most ambiguous predictions, this approach effectively recovered most of the fraud cases while greatly reducing unnecessary alerts. The result is a high-recall, precision-aware pipeline that aligns more closely with the demands of real-world fraud detection systems, where investigation resources are limited and false alarms carry meaningful costs.

1. FINAL PERFORMANCE

A screenshot of a computer screen

AI-generated content may be incorrect.

### Tabular Deep Learning with Tabnet

To assess whether deep learning architectures can improve fraud detection in tabular settings, we implemented a TabNet model on the full engineered dataset. TabNet uses sequential attention masks to learn which features to focus on at each decision step, potentially capturing complex feature interactions and uncovering latent patterns in high-dimensional data.

We trained TabNet using the same stratified train–validation split as previous models and performed threshold tuning to satisfy the ≥ 80% recall business objective. The model’s configuration included 5 decision steps, an embedding dimension of 32, 8 attention heads, and a learning rate of 0.02. Early stopping was applied based on validation AUC with a patience of 15 epochs.

At the tuned threshold of 0.03, TabNet achieved a recall of 82%—meeting the target recall—but with a precision of 11%, resulting in an F1-score of 0.19. While the model successfully detected the majority of fraud cases (346/422), it produced a large number of false positives (2,939), leading to lower overall accuracy (74%) compared to tree-based models. The ROC AUC was 0.859, confirming that TabNet retained strong discriminatory ability, though slightly below that of the best XGBoost-based pipelines.

These results demonstrate that, although TabNet offers an expressive modeling framework, its performance in this setting did not surpass that of well-tuned ensemble methods combined with classical imbalance-handling techniques. In particular, the drop in precision without a corresponding lift in F1 indicates that TabNet’s complexity does not directly translate into operational gains in fraud detection when applied to structured, imbalanced data of this nature.

Nonetheless, TabNet remains a promising avenue for future work, particularly when deployed on larger datasets or in combination with additional unsupervised anomaly detection signals. Its ability to automatically learn hierarchical feature representations and selectively attend to informative attributes could prove advantageous as feature space complexity increases.

1. TABNET PERFORMANCE METRICS ON VALIDATION SET

A screenshot of a computer screen

AI-generated content may be incorrect.

# DISCUSSION

The two case studies demonstrate a clear progression: from weak baseline behavior on imbalanced data to substantially stronger, deployment-ready performance once insights from interpretability and error analysis were operationalized.

At first, standard training on raw skewed data produced models with decent discrimination (ROC AUC) but poor recall on the minority class. Applying imbalance-aware techniques—class weighting, SMOTE variants, and undersampling—was necessary but not sufficient. While methods like SMOTEENN + XGBoost and class-weighted XGBoost improved recall and F1 relative to naïve baselines, the real gains came after peeling back model behavior with SHAP and targeted error profiling.

Error analysis surfaced “quiet risks”: defaulters or frauds that lacked overt red flags and therefore were being missed, and overflagged cases where transient anomalies led to false positives. Encoding these insights as new features (e.g., HighUtilization, DebtIncomeRatio, TotalPastDue for credit; analogous behavioral flags for fraud) turned explanation into leverage. When combined with undersampling to rebalance the training signal and a flexible learner (Gradient Boosting in credit, XGBoost in fraud), the refined pipelines closed the gap between theoretical capacity (AUC) and operational objectives (high recall). In credit default prediction, this resulted in a model with balanced precision/recall (~0.78 each) and an F1 that substantially exceeded earlier threshold-only or imbalance-only variants—showing that feature engineering informed by interpretability was the multiplier. In fraud detection, threshold tuning on top-performing pipelines (e.g., SMOTEENN + XGBoost) initially aligned scores with the business requirement of ≥80% recall but introduced a steep drop in precision. However, further refinement using a post-prediction filtering model—trained on augmented feature sets and selectively applied to high-risk predictions—enabled a much stronger balance: 76% recall and 77% precision. This result underscores the value of architectural layering, where a flexible base detector is enhanced with a lightweight, error-aware filter to suppress false positives without retraining the core model.

Across both domains, threshold calibration proved critical: rather than relying on fixed decision cutoffs, sweeping the probability space to enforce minimum recall thresholds transformed high-AUC classifiers into high-coverage detectors. This control comes with a predictable cost—some increase in false positives—but the asymmetric risk profile (missing a true fraud or default is often more costly than a false alert) justifies the tradeoff.

Finally, while modern deep tabular models like TabNet showed competitive discrimination, they did not outperform the refined ensemble pipelines in recall-constrained F1 metrics. On structured, moderate-sized financial datasets, the synergy of domain-driven feature enrichment, targeted error correction, and calibrated ensemble learners continues to provide the best practical performance.

# CONCLUSION

This study developed end-to-end, imbalance-aware pipelines for credit default prediction and fraud detection—two high-impact, high-imbalance classification problems. By systematically comparing classical and modern learners, applying resampling strategies, leveraging SHAP-based interpretability, and incorporating error-informed feature engineering, we achieved robust, operationally meaningful improvements.

In credit default prediction, the final pipeline—combining targeted behavioral flags, undersampling, and Gradient Boosting—achieved a strong balance of precision and recall (~0.78 each), significantly improving F1 over baseline models.

In fraud detection, the initial SMOTEENN + XGBoost model, after threshold tuning, satisfied the ≥80% recall requirement but suffered from low precision. By layering a feature-enhanced filtering model on top—focused on identifying false positives within the flagged set—the final system reached a much more favorable balance: 76% recall and 77% precision, with an F1-score of 0.76. This modular design preserved recall while significantly reducing unnecessary alerts, making the pipeline more viable for real-world deployment.

Threshold tuning was essential in converting strong discrimination into actionable coverage, while interpretability served not only to explain predictions but also to guide iterative refinements. Although TabNet and other deep tabular models remain promising, especially for larger or more complex datasets, the best return on modeling effort in this study came from well-tuned ensemble learners enhanced with domain-aware features and post-processing layers.

Ultimately, this work reinforces the idea that reliable, high-recall risk detection systems arise from the thoughtful combination of resampling, calibrated decision thresholds, interpretability, and error-driven feedback loops—not from any one technique alone.

##### REFERENCES

[1] D. J. Hand and W. E. Henley, “Statistical classification methods in consumer credit scoring: a review,” *J. R. Stat. Soc. Series A*, vol. 160, no. 3, pp. 523–541, 1997.  
[2] I. Brown and C. Mues, “An experimental comparison of classification algorithms for imbalanced credit scoring data sets,” *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3446–3453, 2012.  
[3] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proc. 22nd ACM SIGKDD*, 2016, pp. 785–794.  
[4] J. Zhou et al., “Cost‑sensitive gradient boosting for imbalanced data,” *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 10, pp. 1895–1908, 2018.  
[5] T. Lin et al., “Focal loss for dense object detection,” in *ICCV*, 2017, pp. 2980–2988.  
[6] N. V. Chawla et al., “SMOTE: Synthetic minority over‑sampling technique,” *J. Artif. Int. Res.*, vol. 16, pp. 321–357, 2002.  
[7] G. E. Batista et al., “A study of the behavior of several methods for balancing machine learning training data,” *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 20–29, 2004.  
[8] A. Dal Pozzolo et al., “Calibrating probability with undersampling for unbalanced classification,” *Proc. ICDM*, 2015, pp. 541–546.  
[9] X. He et al., “ADASYN: Adaptive synthetic sampling approach for imbalanced learning,” in *Proc. IEEE Int. Joint Conf. Neural Nets*, 2008, pp. 1322–1328.  
[10] H. He and E. A. Garcia, “Learning from imbalanced data,” *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, 2009.  
[11] C. Phua et al., “A comprehensive survey of data mining‑based fraud detection research,” *arXiv*, arXiv:1009.6119, 2010.  
[12] F. T. Liu et al., “Isolation forest,” in *Proc. ICDM*, 2008, pp. 413–422.  
[13] H. Goldstein and S. Tate, “-Autoencoder-based anomaly detection methods,” *arXiv*, arXiv:1802.00067, 2018.  
[14] A. Dal Pozzolo et al., “Learned lessons in credit card fraud detection from a practitioner’s perspective,” *IEEE Secur. Priv.*, vol. 17, no. 2, pp. 45–53, 2019.  
[15] A. Lazarevic and V. Kumar, “Feature bagging for outlier detection,” in *Proc. SDM*, 2005.  
[16] G. Gama et al., “A survey on concept drift adaptation,” *ACM Comput. Surv.*, vol. 46, no. 4, Art. 44, 2014.  
[17] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *NIPS*, 2017, pp. 4765–4774.  
[18] M. T. Ribeiro et al., ““Why should I trust you?”: Explaining the predictions of any classifier,” in *KDD*, 2016, pp. 1135–1144.  
[19] S. Arik and T. Pfister, “TabNet: Attentive interpretable tabular learning,” in *NeurIPS*, 2019.  
[20] A. Gorishniy et al., “Revisiting deep learning models for tabular data,” in *NeurIPS*, 2021.

All code and implementation details for this project are available at: <https://github.com/irenemuruatxintxurreta/Credit-Default-Prediction-and-Fraud-Detection/tree/main>