

# From Classification to Decision Optimization: A Cost-Sensitive Risk Prioritization Framework for Credit Card Fraud Detection

## Abstract

Credit card fraud detection is often approached as a binary classification problem. However, extreme class imbalance and asymmetric misclassification costs render accuracy-based evaluation misleading and operationally ineffective. This paper presents a data-driven framework that reframes fraud detection as a **risk prioritization and decision optimization problem**. Using anonymized European credit card transaction data, we develop a probabilistic risk model, construct a cost-sensitive decision policy, and demonstrate how threshold selection materially impacts financial loss and operational workload. The results highlight the importance of aligning machine learning outputs with business decisions rather than optimizing predictive metrics in isolation.

## 1. Introduction

Fraud detection plays a critical role in financial services, directly impacting revenue protection, customer trust, and regulatory compliance. Traditional machine learning approaches frequently emphasize predictive accuracy or ROC-AUC, metrics that are ill-suited for highly imbalanced problems. In real-world fraud systems, the objective is not to predict fraud perfectly but to **allocate limited investigative resources efficiently** while minimizing financial loss.

This study proposes a framework that integrates machine learning, cost-sensitive evaluation, and decision science to support fraud operations.

## 2. Dataset Description

The dataset consists of 284,807 credit card transactions conducted by European cardholders over two days in September 2013. Only 492 transactions are labeled as fraudulent, resulting in a fraud rate of 0.172%.

All features except **Time** and **Amount** have undergone Principal Component Analysis (PCA) to preserve confidentiality. Consequently, features represent latent behavioral components rather than interpretable transaction attributes.

## 3. Problem Framing

Due to severe class imbalance:

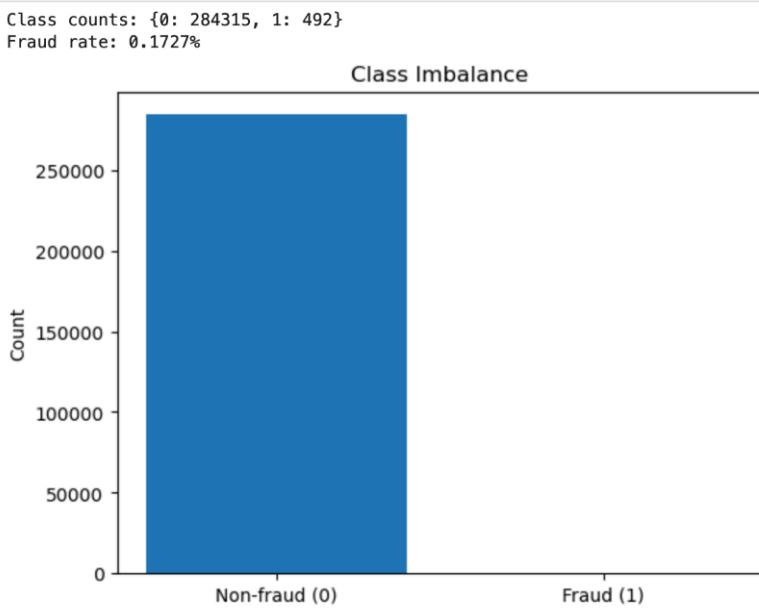
- Accuracy becomes meaningless (predicting all transactions as non-fraud yields >99.8% accuracy).
- False negatives incur direct financial losses.
- False positives generate customer dissatisfaction and operational costs.

Therefore, fraud detection is reframed as:

**A risk scoring and decision optimization problem under cost and capacity constraints.**

## 4. Exploratory Data Analysis and Risk Characteristics

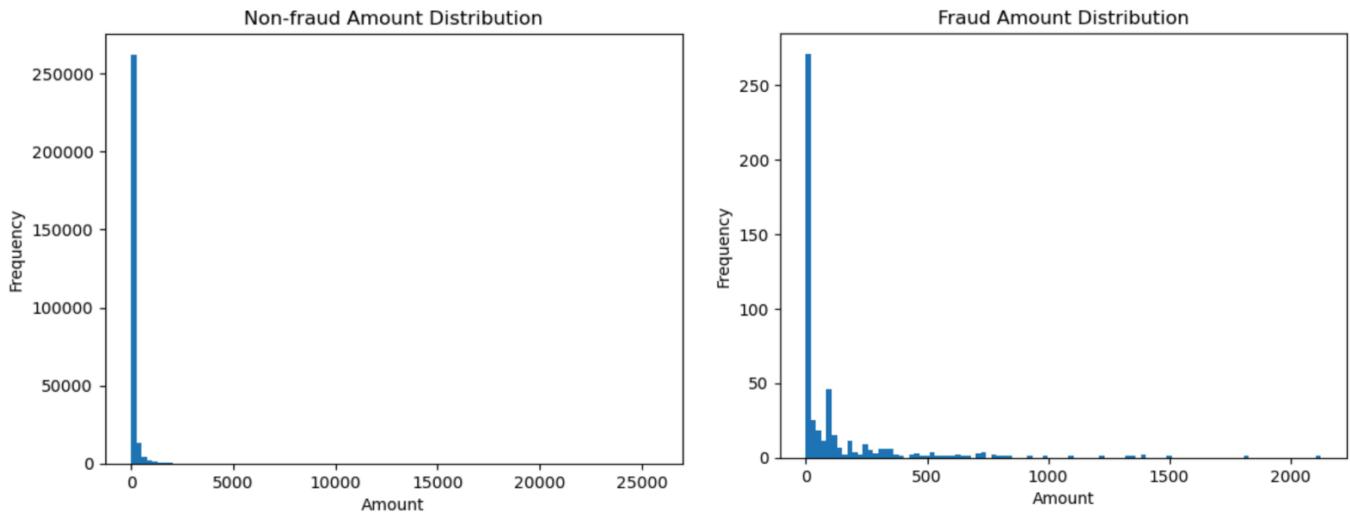
### 4.1 Class Imbalance and Base Rate Risk



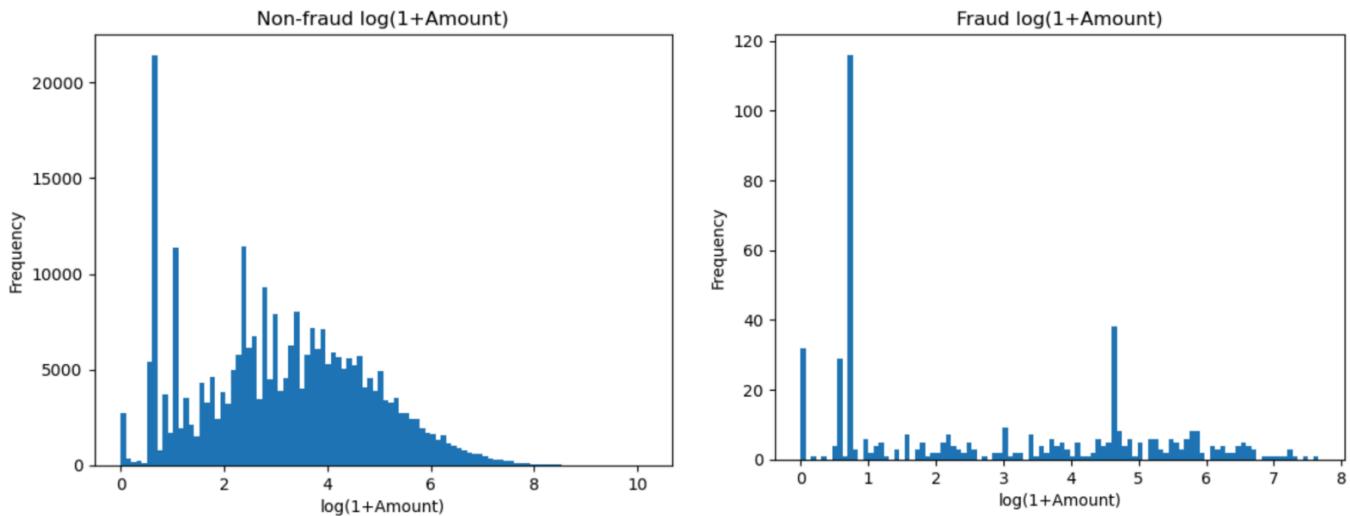
*This figure illustrates the extreme class imbalance in the dataset, with fraudulent transactions accounting for only 0.172% of all observations. The distribution highlights a fundamental challenge in fraud detection: naïve classifiers can achieve deceptively high accuracy by predicting all transactions as non-fraud. Consequently, accuracy-based metrics are inappropriate and potentially harmful for decision-making in this context.*

=> **The low base rate implies that even a small false positive rate can overwhelm operational capacity, while a small false negative rate can lead to disproportionate financial losses. This motivates the use of Precision–Recall based metrics and cost-sensitive evaluation.**

## 4.2 Transaction Amount and Loss Exposure



*Figure: Transaction Amount Distribution - Raw Scale*

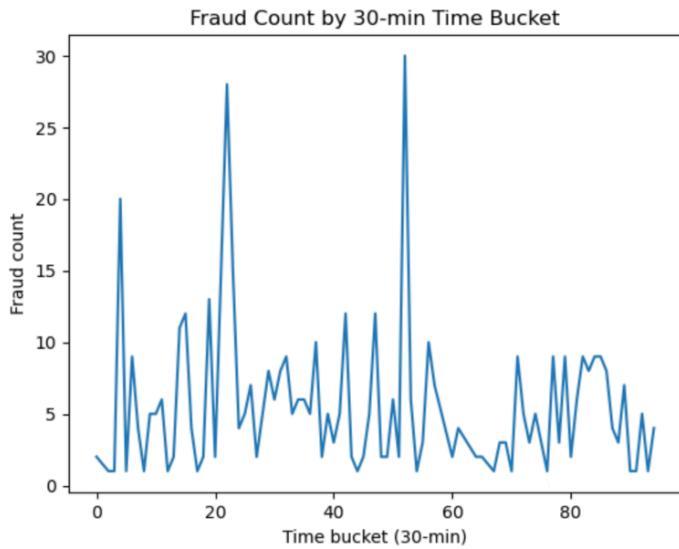


*Figure: Transaction Amount Distribution - Log Scale*

Those figures compare transaction amount distributions for fraudulent and non-fraudulent transactions. While most transactions are low-value, fraudulent transactions exhibit a heavier tail, indicating exposure to large potential losses.

=> This observation justifies using transaction amount as a proxy for financial loss in cost-sensitive modeling.  
Missing a single high-value fraudulent transaction can outweigh the cost of flagging hundreds of legitimate transactions for review.

## 4.3 Temporal Patterns in Fraud Activity



Fraud occurrences show temporal clustering rather than uniform distribution across time. This suggests that fraud risk is not static and may be influenced by time-dependent behavioral patterns.

=> **Although time is not explicitly modeled as a sequential process in this study, the observed clustering motivates future extensions involving temporal backtesting and time-aware modeling.**

## 5. Methodology

### 5.1 Baseline Model

A logistic regression model with class weighting was trained to establish a probability baseline. Performance was evaluated using **Area Under the Precision-Recall Curve (AUPRC)**, consistent with best practices for rare-event detection. The baseline model is not intended for deployment, but serves to establish a probabilistic reference point and validate the appropriateness of Precision–Recall based evaluation under extreme class imbalance.

### 5.2 Tree-Based Risk Model

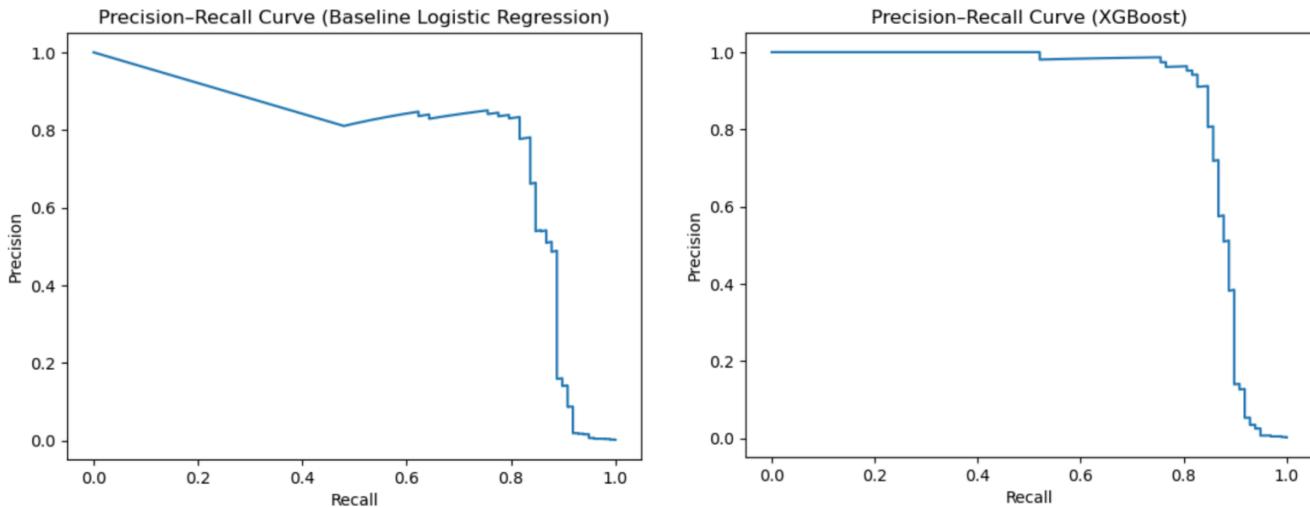
An XGBoost classifier was employed to capture nonlinear interactions among PCA components. Class imbalance was handled using **scale\_pos\_weight** rather than resampling to avoid generating artificial data in latent feature space. The model outputs calibrated probabilities representing transaction-level fraud risk. The purpose of the tree-based model is not to make final fraud decisions, but to produce a reliable **risk ranking** that can be translated into business actions through a downstream decision policy.

## 5.3 Evaluation Metrics

Model performance was evaluated using:

- Precision-Recall curves
- PR-AUC (Average Precision)

However, model metrics alone were deemed insufficient for deployment decisions.



The baseline logistic regression model establishes a probability reference point, while the XGBoost model demonstrates superior performance in the high-recall, low-precision region—precisely where fraud detection systems operate.

=> The improvement in PR-AUC indicates that the tree-based model more effectively ranks fraudulent transactions ahead of legitimate ones. However, this improvement alone does not determine deployment decisions, as operational costs and capacity constraints must still be considered.

## 6. Decision Optimization and Expected Loss

### 6.1 Expected Loss Framework

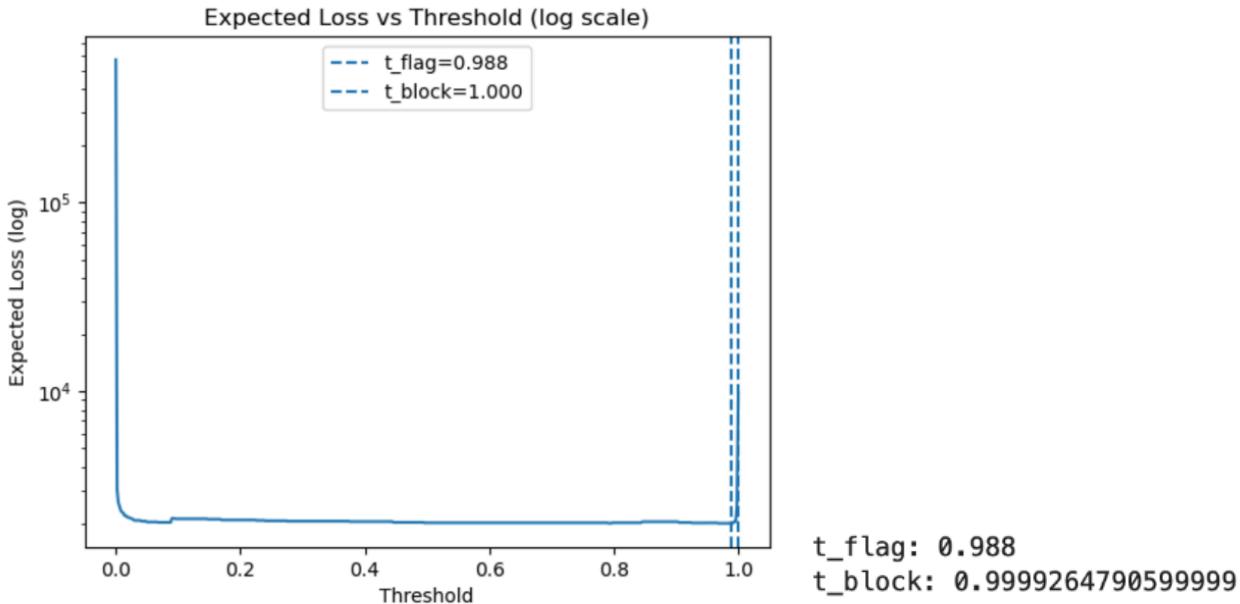
Rather than selecting a probability threshold arbitrarily (e.g., 0.5), expected loss is computed as:

- **False Negative Cost:** Missed fraud transaction amount
- **False Positive Cost:** Fixed operational and customer friction cost

This framework explicitly links model outputs to business outcomes.

## 6.2 Threshold Selection and Loss Tradeoffs

Expected loss was computed across a range of probability thresholds. The optimal threshold minimized total expected loss rather than maximizing predictive metrics.



The figure shows expected loss across probability thresholds on a logarithmic scale. A clear minimum emerges, defining the optimal intervention threshold (`t_flag`).

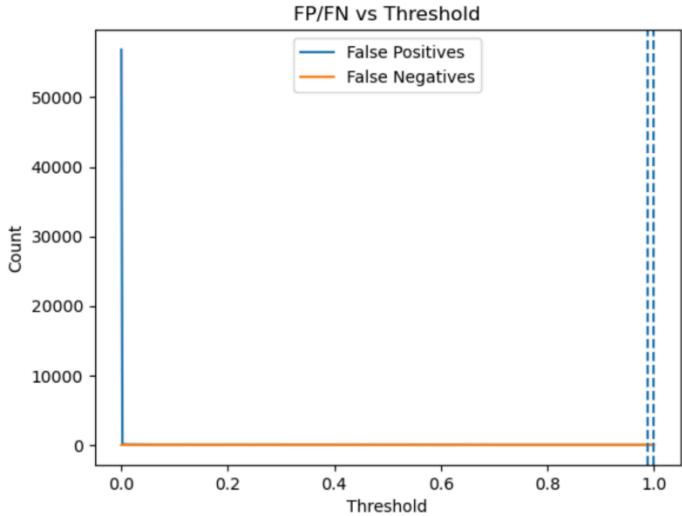
=> Expected loss increases rapidly when the threshold is set too high (missed fraud dominates) or too low (false positives dominate). The optimal threshold represents a balance between financial protection and customer experience.

## 6.3 Tiered Decision Framework

Transactions were assigned to three action tiers:

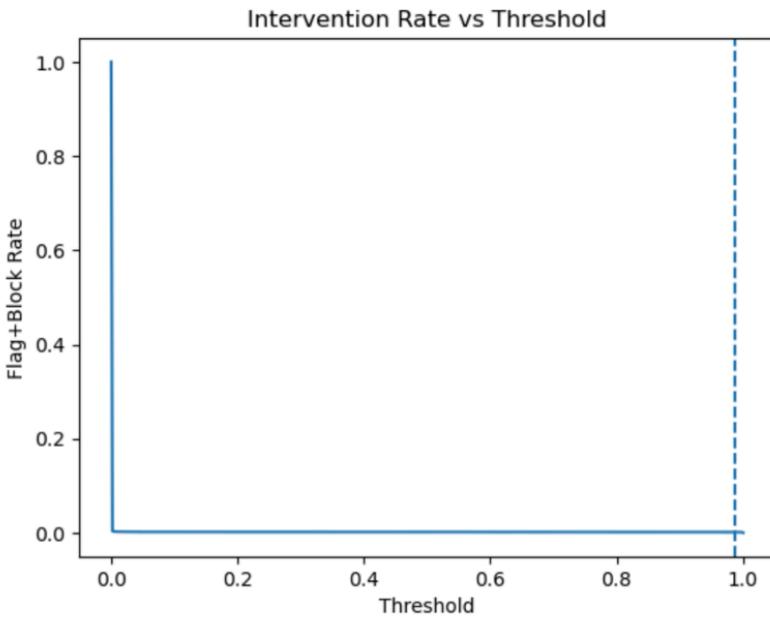
- **Allow:** low-risk transactions
- **Flag:** transactions requiring human review
- **Block:** extreme-risk transactions (top risk tail)

This structure mirrors real-world fraud operations.



The figure illustrates how false positives decrease while false negatives increase as the threshold rises.

=> This tradeoff is central to fraud operations. Business stakeholders can use this visualization to align model behavior with risk appetite and service-level objectives.



This figure plots the proportion of transactions requiring intervention (Flag + Block) across thresholds.

=> The intervention rate directly translates model decisions into operational workload. It enables decision-makers to select thresholds that respect review capacity while maintaining acceptable fraud coverage.

## 7. Results

---

```
TP: 79 FP: 3 FN: 19 TN: 56861
Precision: 0.9634 | Recall: 0.8061
Expected Loss (policy): 2000.00
Block count: 57
Block fraud rate: 0.9824561403508771
```

### 7.1 Risk Prioritization Effectiveness

The optimized decision policy successfully concentrates fraudulent transactions within the highest-risk segments. The **Block** tier exhibits a substantially higher fraud rate than the dataset baseline, confirming that the model effectively prioritizes risk rather than indiscriminately flagging transactions.

#### **Business Impact:**

- Fraud analysts focus attention on a small subset of transactions with materially higher risk.
- Automated blocking is reserved for extreme-risk cases, minimizing unnecessary customer disruption.

### 7.2 Financial Impact and Loss Reduction

By selecting thresholds that minimize expected loss rather than maximizing predictive metrics, the proposed framework achieves a more economically efficient outcome.

#### **Business Impact:**

- High-value fraud cases are prioritized, reducing exposure to catastrophic losses.
- The system avoids excessive false positives that would degrade customer trust and increase support costs.

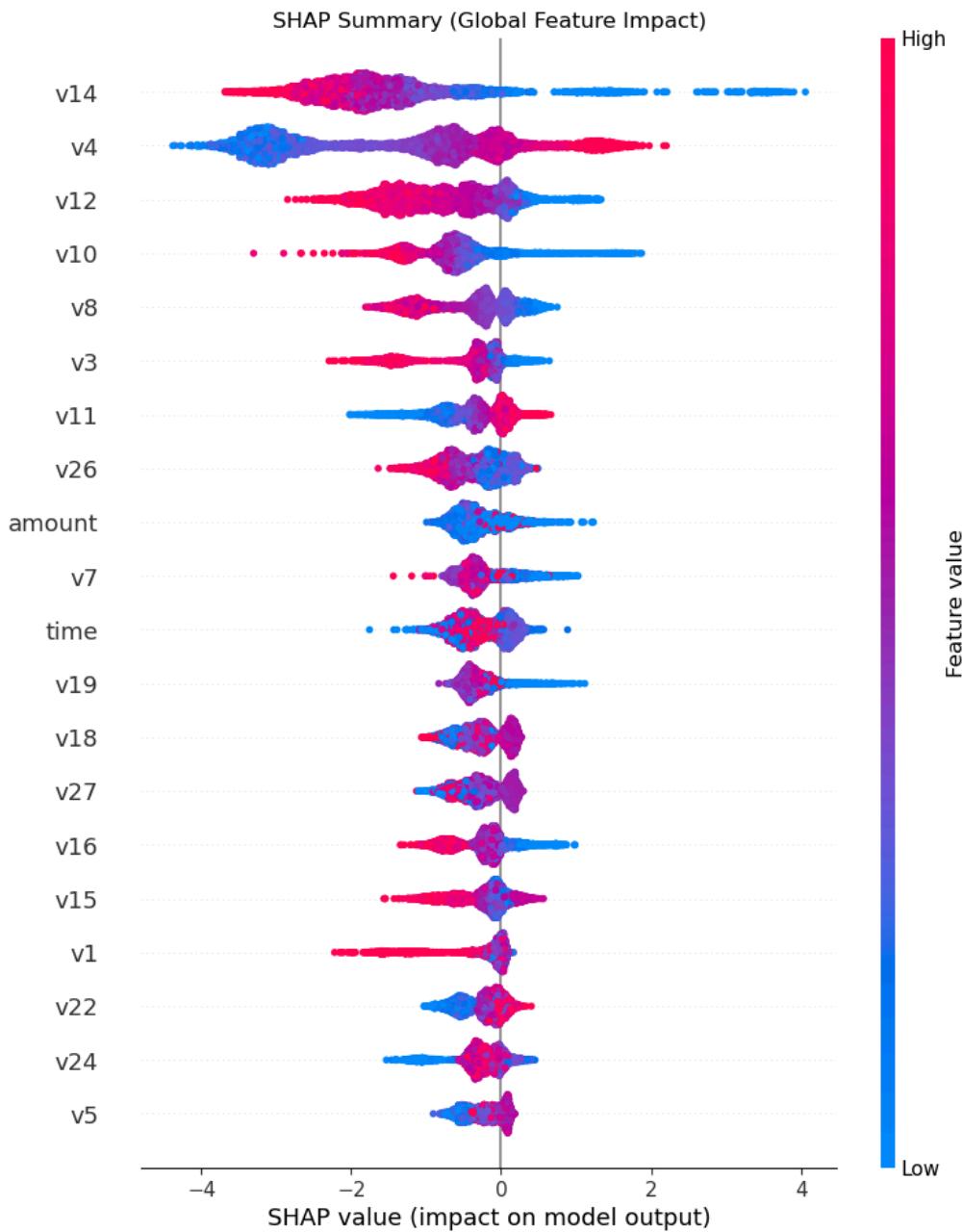
### 7.3 Decision Transparency

Separating risk scoring from decision thresholds improves transparency. Thresholds can be adjusted in response to changing risk appetite or capacity constraints without retraining models.

#### **Business Impact:**

- Model outputs can be audited independently of threshold choices.
- Thresholds can be adjusted without retraining models, enabling rapid response to changing business conditions.

## 8. Explainability and Governance



SHAP analysis identifies latent PCA components that most strongly influence risk scores.

=> Given the PCA-transformed features, SHAP is used for model monitoring and governance, not for customer-facing explanations.

**Governance Implications:**

- Detects model drift through changes in SHAP distributions.
- Support regulatory audits by documenting model behavior at an aggregate level.

## 9. Limitations

- PCA features restrict interpretability.
- Cost assumptions are simplified proxies.
- The dataset covers a limited temporal window.

Despite these constraints, the framework remains valid and extensible for real-world deployment.

## 10. Next Iterations and Extensions

### 10.1 Capacity-Constrained Optimization

Future work should explicitly incorporate a maximum daily review capacity. Instead of selecting thresholds solely by expected loss, optimization can be performed under constraints such as:

- maximum number of flagged transactions per day,
- maximum customer contact volume.

This extension aligns model deployment with staffing and service-level agreements.

### 10.2 Probability Calibration

Calibrating predicted probabilities using Platt scaling or isotonic regression can improve threshold stability over time. Well-calibrated probabilities enable consistent decision-making across changing fraud environments.

### 10.3 Temporal Backtesting

A rolling-window backtesting framework would simulate real deployment conditions. This would allow evaluation of:

- policy robustness under concept drift,
- lag effects between fraud occurrence and label availability.

## 10.4 Feedback Loops and Human-in-the-Loop Learning

In production systems, analyst feedback provides valuable signals. Integrating investigation outcomes can:

- refine cost assumptions,
- update thresholds dynamically,
- support semi-supervised learning.

## 10.5 Multi-Objective Optimization

Beyond financial loss, future models can incorporate additional objectives:

- customer lifetime value,
- regulatory risk,
- fairness and bias constraints.

This transforms fraud detection into a holistic risk management system.

## 11. Conclusion

This project demonstrates that effective fraud detection systems require more than accurate models. By integrating cost-sensitive evaluation and decision science, machine learning can be transformed into a practical tool for financial risk management. The proposed framework provides a foundation for building interpretable, scalable, and business-aligned fraud detection systems.

## 12. Appendix

Full code work: [https://github.com/Playmaker08/credit\\_card\\_fraud\\_detection](https://github.com/Playmaker08/credit_card_fraud_detection)

