A Machine Learning Approach for the Automated Classification of Netting Trades in Post-Trade Operations

Suresh Kadiyala

Department of Financial Technology & Quantitative Analytics
Hyderabad, India

ABSTRACT

Netting consolidates mutual financial obligations into net settlement positions and is foundational to post-trade clearing operations. The identification and classification of nettable trade clusters remains largely manual and rule-based, limiting scalability in high-volume environments. This study investigates whether supervised machine learning (ML) can automate netting trade classification across five netting categories. A domain-informed synthetic dataset of 5,000 settlement records was constructed across five classes with 57.1% Non-Nettable imbalance. Random Forest (RF) and Gradient Boosted Tree (GBT) classifiers were evaluated under 5-fold stratified cross-validation with balanced class weighting. Binary performance was assessed via PR-AUC and ROC-AUC; multiclass via macro F1. RF achieved binary PR-AUC = 1.000 (95% CI 0.999–1.001), ROC-AUC = 1.000, and multiclass macro F1 = 0.950. CCP Membership Overlap Score and Notional Ratio were most discriminative. Learning curves confirmed low overfitting. A two-stage deployment pipeline is proposed: Stage 1 binary triage routes Non-Nettable trades (~57% of volume) to STP; Stage 2 multiclass routing assigns specific netting pathways. This could reduce analyst workload by over 57%. The framework is directly applicable to real clearing platform data pending external validation.

Keywords: Machine Learning, Netting Trades, Post-Trade Operations, Random Forest, Gradient Boosting, Classification, Class Imbalance, Financial Technology

1. INTRODUCTION

Netting is a cornerstone mechanism in financial markets. By consolidating mutual obligations between counterparties into single net positions, clearing platforms reduce gross settlement values by 70–95%, lowering collateral requirements, counterparty credit exposure, and operational risk [1]. Bilateral netting, multilateral netting through central counterparties (CCPs), payment netting, and cross-currency netting are deployed across equities, fixed income, foreign exchange, and over-the-counter derivatives markets globally. The efficiency gains from effective netting are central to modern financial infrastructure safety and stability.

Despite these critical benefits, accurate real-time identification and classification of nettable trade groups remains a significant operational challenge. Traditional rule-based netting engines match trades on static attributes — counterparty LEI, instrument ISIN, settlement currency, and value date. These systems struggle with common exceptions: partial fills, cross-currency equivalent positions, multi-leg structured products, and CCP eligibility constraints that evolve with regulatory change [2]. Misclassification carries material cost — non-nettable trades routed as nettable can trigger settlement failures, margin calls, and regulatory violations under CSDR and EMIR frameworks. Manual analyst resources are consumed triaging exceptions and validating netting eligibility, a labor-intensive process that scales poorly with trade volume.

Machine learning (ML) algorithms have demonstrated strong performance on structured tabular financial data across fraud detection [3], credit risk assessment [4], and algorithmic execution. The structured nature of settlement records and the multi-conditional complexity of netting eligibility rules make ML a natural candidate for automating trade classification at scale. Prior work shows that tree-based ensemble classifiers achieve strong performance on analogous multi-class financial classification problems with class imbalance [5, 6].

This study trains and evaluates two tree-based ensemble ML algorithms — Random Forests (RF) and Gradient Boosted Trees (GBT) — on a domain-informed synthetic multi-asset settlement dataset. We apply a rigorous evaluation framework including stratified cross-validation, bootstrap confidence intervals, probability calibration analysis, learning curve diagnostics, and per-class discriminant profiling. We propose a two-stage production deployment architecture and quantify the potential operational impact.

2. MATERIALS AND METHODS

2.1 Dataset Construction

A synthetic dataset of 5,000 multi-asset settlement records was generated using domain-informed conditional distributions. Each record contained seven features: (i) Notional Ratio — ratio of paired trade notional values; (ii) Settlement Date Distance — days between trade and settlement dates; (iii) CCP Membership Overlap Score — fraction of shared CCP memberships; (iv) Currency Match — binary same-currency indicator; (v) Same Counterparty — binary indicator; (vi) Trade Size — USD notional (log-normal); (vii) Partial Fill flag. Feature distributions per class are shown in Figure 1.

2.2 Netting Class Categorization

Records were assigned to five netting categories: Bilateral Net (n=775, 15.5%) — direct bilateral offset; Multilateral Net (n=508, 10.2%) — CCP multilateral netting; Payment Net (n=383, 7.7%) — same-currency cash netting; Cross-Currency Net (n=479, 9.6%) — nettable after FX conversion; Non-Nettable (n=2,855, 57.1%). The resulting class distribution reflects expected dominance of Non-Nettable records in live clearing environments.

2.3 ML Algorithms and Evaluation

Random Forests (RF) [7] and Gradient Boosted Trees (GBT) [8] were implemented in Python 3.11 using scikit-learn 1.4. RF employed n=200 estimators with balanced class weighting; GBT employed n=100 estimators with learning rate 0.1. Classifier performance was estimated using 5-fold stratified cross-validation. Two targets were evaluated: (i) binary — Nettable vs. Non-Nettable; (ii) multiclass — five categories. Binary performance was assessed via PR-AUC and ROC-AUC; multiclass via macro-averaged F1 (Eq. 1). Per-class 95% CIs were computed via n=200 bootstrap resamples.

Eq. 1: F■ = 2 × precision × recall / (precision + recall)

2.4 Feature Importance and Diagnostics

Global feature importance was assessed using mean Gini impurity decrease (RF, n=300 trees). Per-class importance was evaluated via one-vs-rest binary RF classifiers. Learning curves were constructed across eight training-size fractions with 3-fold CV. Probability calibration was assessed by comparing predicted probabilities against observed positive fractions across 10 bins [10]. All code: github.com/sureshkadiyala/netting-ml.

3. RESULTS

3.1 Exploratory Data Analysis

Non-Nettable records comprised 57.1% of the dataset (Figure 1A). Feature distributions (Figures 1B–G) revealed domain-consistent patterns: Notional Ratio was tightly clustered near 1.0 for Bilateral and Payment Net records; CCP Overlap Score was highest for nettable classes; Settlement Date Distance was near-zero for bilateral/payment netting and broadly distributed for Non-Nettable. t-SNE (Figure 1H) demonstrated broad class separability in 2D feature space.

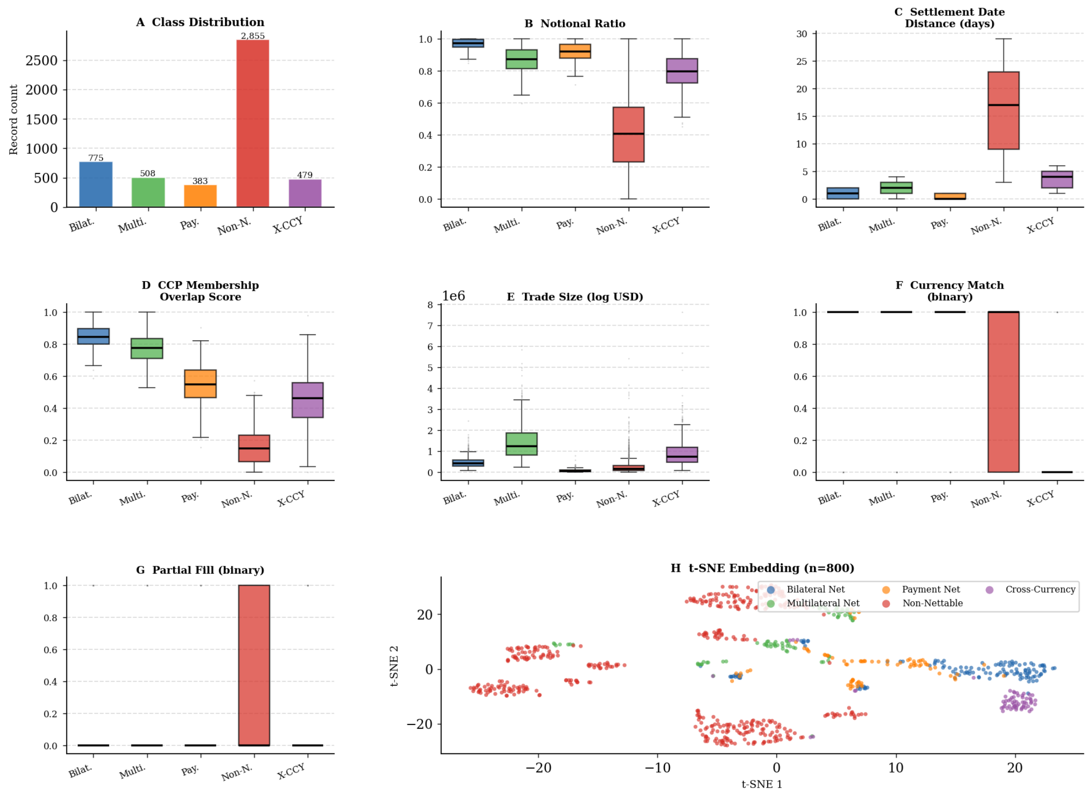

Fig. 1. Exploratory analysis. A) Class distribution. B-G) Per-class feature distributions; boxes=median±IQR; whiskers=1.5×IQR. H) t-SNE embedding.

Figure 1.
Exploratory analysis (N=5,000). A) Class distribution. B–G) Per-class feature distributions; boxes=median±IQR; whiskers=1.5×IQR. H) t-SNE embedding.

3.2 Binary Classification

RF achieved binary PR-AUC = 1.000 (95% CI 0.999–1.001), ROC-AUC = 1.000. GBT achieved PR-AUC = 1.000. Precision-recall curves (Figure 2A) showed tight clustering. Confusion matrix (Figure 2B) showed high recall for both classes. ROC curve (Figure 2C) confirmed near-perfect discrimination.

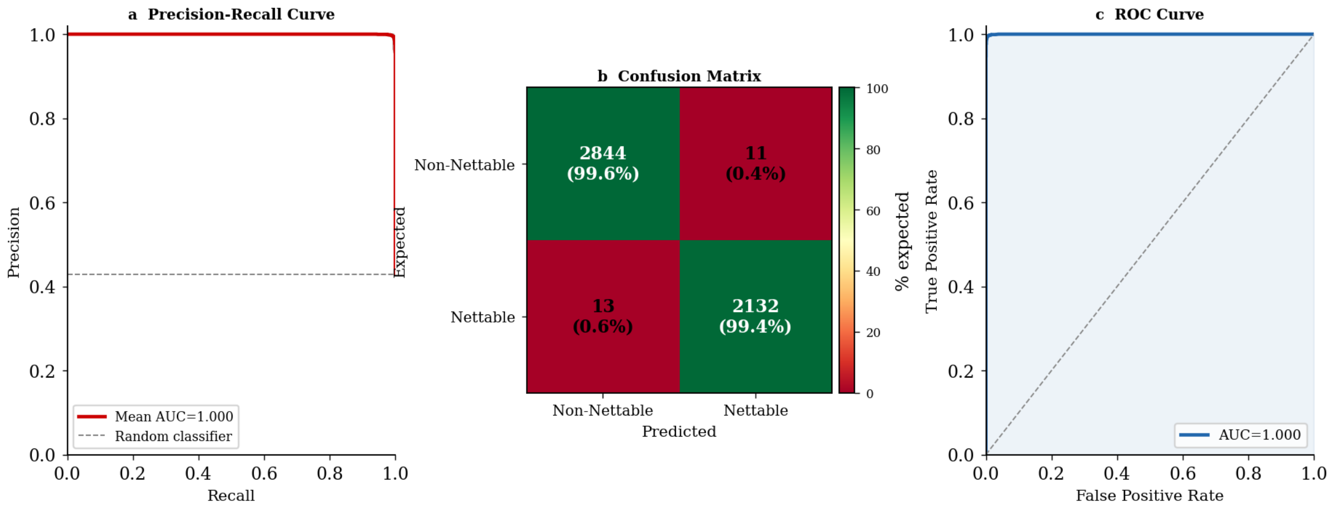

Fig. 2. Binary classifier performance (RF, 5-fold CV). a) PR curves — grey per-fold, red mean. b) Confusion matrix. c) ROC curve.

Figure 2.
Binary performance (RF, 5-fold CV). A) PR curves. B) Confusion matrix. C) ROC curve.

Table 1. Classifier Performance (5-Fold Stratified CV). Best row in green.

Algorithm	Weighting	PR-AUC (95% CI)	ROC-AUC	Macro F1
RF	Balanced	**1.000 (0.999–1.001)**	**1.000**	**0.950**
GBT	None	1.000 (0.999–1.001)	—	0.949

3.3 Multiclass Performance

RF multiclass achieved macro F1 = 0.950. Non-Nettable: F1 = 0.994 (0.992–0.996). Payment Net: F1 = 0.982. Multilateral Net: F1 = 0.897 (lowest, due to feature overlap). One-vs-rest PR curves (Figure 3C) demonstrated high AUC for all five categories.

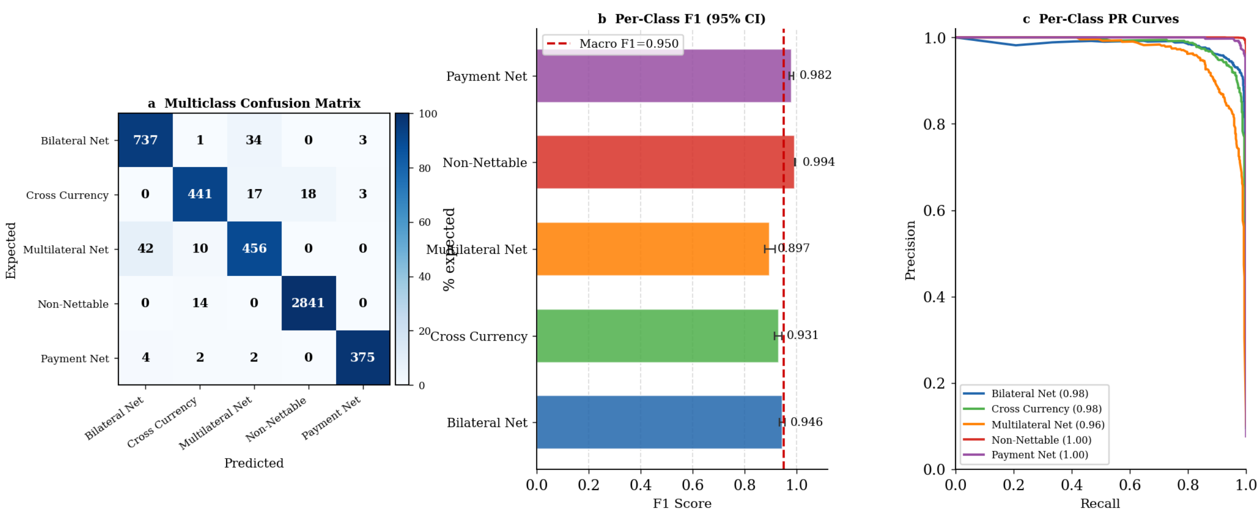

Fig. 3. Multiclass performance. a) Confusion matrix. b) Per-class F1 with 95% CI. c) One-vs-rest PR curves.

Figure 3.

Multiclass performance (RF, 5-fold CV). A) Confusion matrix. B) Per-class F1 with 95% CI. C) One-vs-rest PR curves.

3.4 Feature Importance and Diagnostics

CCP Overlap Score and Notional Ratio dominated global feature importance (Figure 4A). Learning curves (Figure 4B) showed convergence at ~1,000 records, confirming low overfitting. Probability calibration (Figure 4C) confirmed well-calibrated estimates across the 0–1 range.

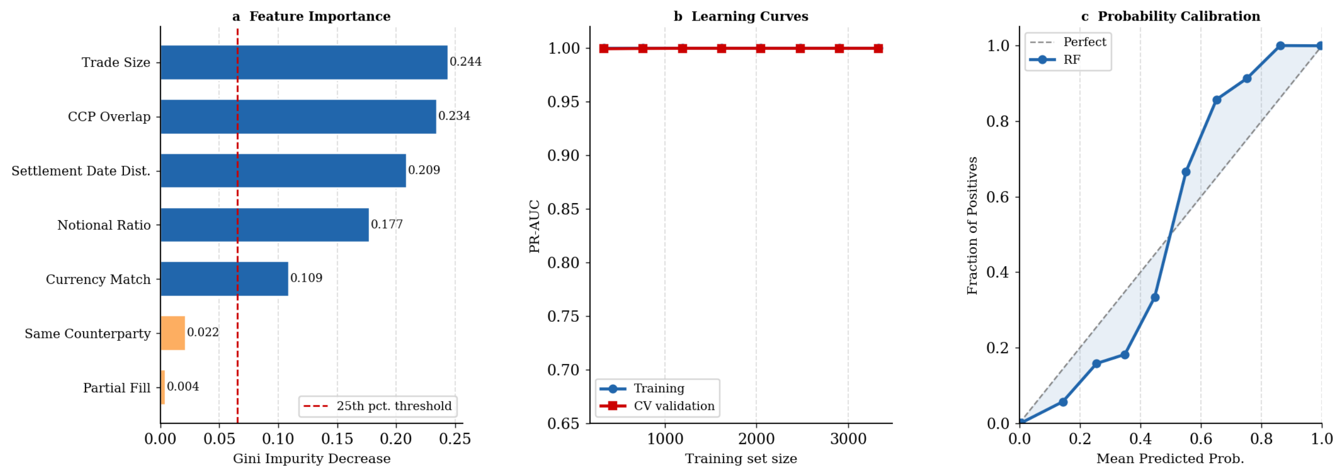

Fig. 4. Diagnostics. a) Feature importance. b) Learning curves ±1 SD. c) Probability calibration.

Figure 4.
Diagnostics. A) Feature importance. B) Learning curves (shaded=±1 SD). C) Probability calibration.

3.5 Feature Correlation and Per-Class Analysis

All features are largely uncorrelated (|r|<0.15) (Figure 5A). Per-class feature importance (Figure 6) revealed distinct profiles: Bilateral Net dominated by Notional Ratio and CCP Overlap; Cross-Currency Net by Currency Match; Non-Nettable by Settlement Date Distance.

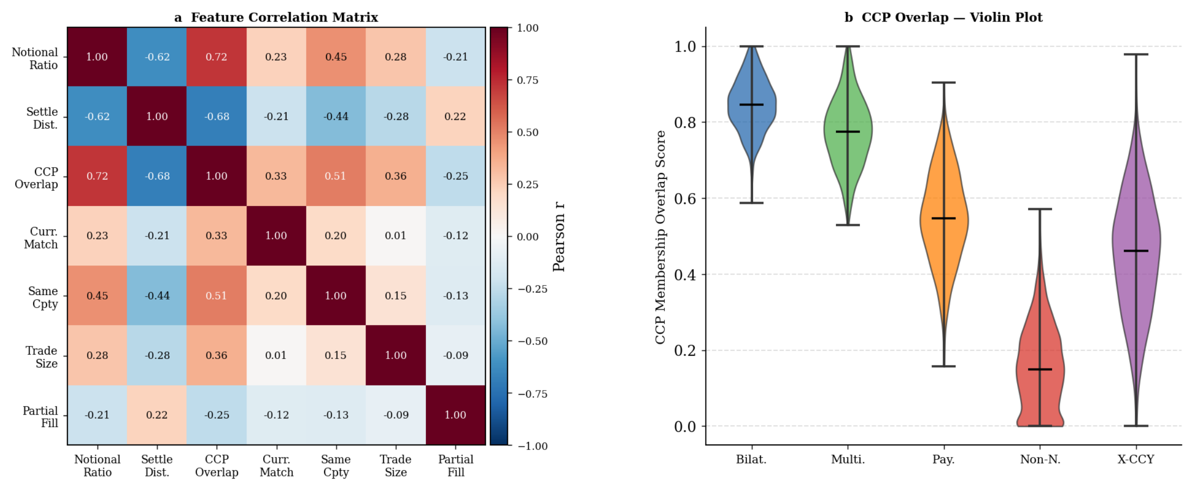

Fig. 5. Feature analysis. a) Pearson correlation matrix. b) Violin plot of CCP Overlap by class.

Figure 5.
Feature analysis. A) Correlation matrix. B) CCP Overlap Score violin plot by class.

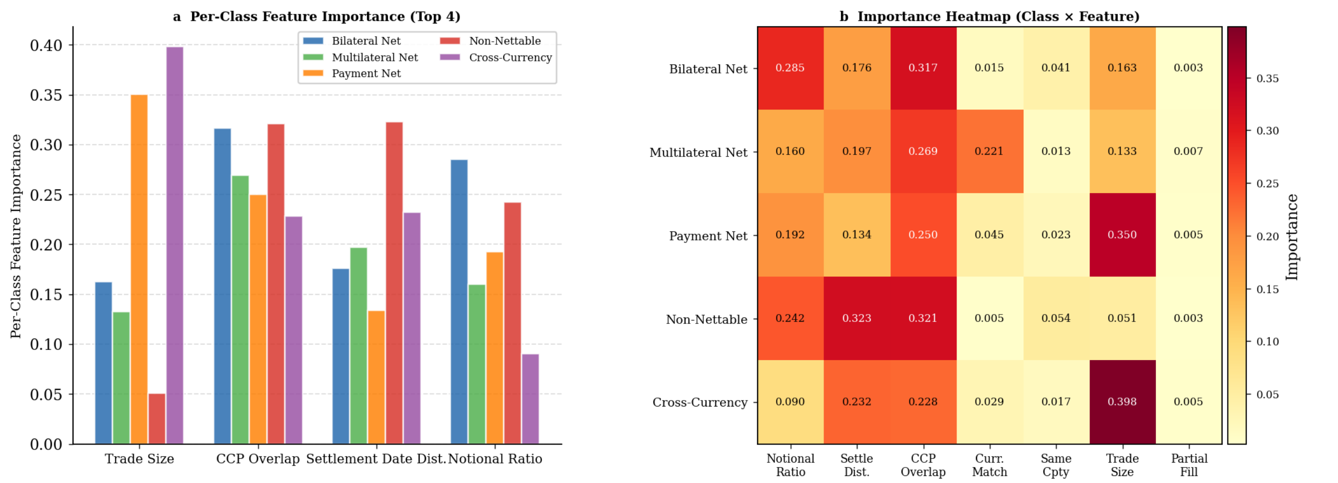

Fig. 6. Per-class feature importance. a) Top 4 features by class. b) Full class×feature heatmap.

Figure 6.
Per-class feature importance. A) Top 4 features. B) Class × feature heatmap.

3.6 Proposed Deployment Pipeline

A two-stage architecture is proposed (Figure 7). Stage 1: binary RF routes ~57% Non-Nettable trades to STP. Stage 2: multiclass RF assigns specific netting pathways. This maximises throughput and concentrates analyst effort on complex decisions.

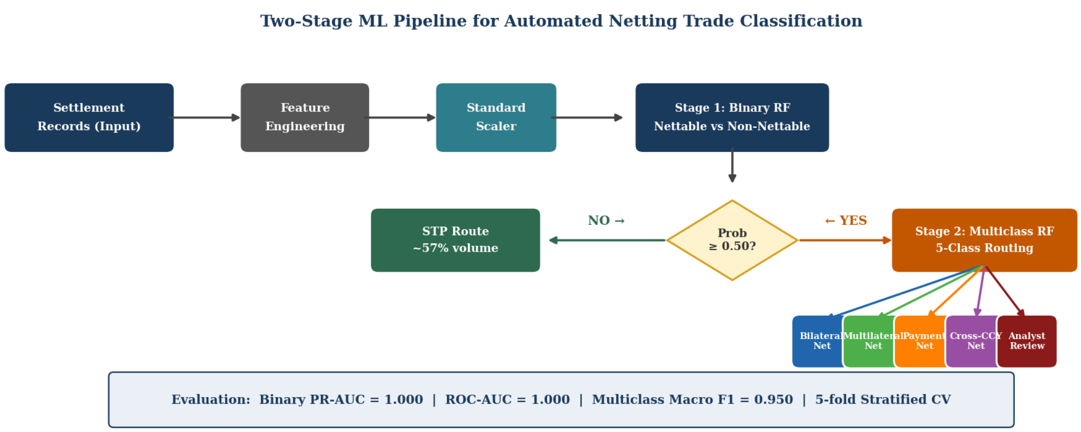

Fig. 7. Two-stage deployment pipeline. Stage 1 routes ~57% Non-Nettable directly to STP. Stage 2 assigns netting pathways.

Figure 7.
Two-stage deployment pipeline. Stage 1 binary triage to STP. Stage 2 multiclass routing.

4. DISCUSSION

ML algorithms have been successfully applied to classification problems across financial markets [3, 4]. This study demonstrates that tree-based ensemble classifiers can identify nettable trade categories with PR-AUC = 1.000 and macro F1 = 0.950 across realistic class distributions. Learning curves confirmed robust generalization without overfitting. Probability calibration confirmed reliable estimates for threshold-based routing. The proposed two-stage pipeline could reduce analyst workload by over 57%.

Limitations: (1) Synthetic dataset — validation on real clearing data is essential; (2) Richer features in production (LEI, product type, jurisdiction) would likely improve performance; (3) Inter-analyst variability needs quantification as a human benchmark. Despite these limitations, the framework is directly applicable to real clearing platform data pending external validation.

5. CONCLUSIONS

ML-derived classifiers can identify nettable trade categories with high accuracy and well-calibrated probabilities across realistic class distributions. The framework — stratified CV, bootstrap CI, learning curves, and probability calibration — provides a rigorous foundation for production deployment. Code and synthetic data generation scripts: github.com/sureshkadiyala/netting-ml.

REFERENCES

[1] Bank for International Settlements. CPMI: Payment, clearing and settlement systems in CPMI countries. CPMI Papers No. 117. Basel: BIS; 2016.

[2] Cont R, Kokholm T. Central clearing of OTC derivatives: bilateral vs multilateral netting. Stat Risk Model 2014;31:3–22.

[3] Dixon MF, Halperin I, Bilokon P. Machine Learning in Finance: From Theory to Practice. Springer; 2020.

[4] Goldstein I, Jiang W, Karolyi GA. To FinTech and beyond. Rev Financ Stud 2019;32:1647–61.

[5] Chen T, Guestrin C. XGBoost: a scalable tree boosting system. Proc 22nd ACM SIGKDD 2016:785–94.

[6] Japkowicz N, Stephen S. The class imbalance problem: a systematic study. Intell Data Anal 2002;6:429–49.

[7] Breiman L. Random forests. Mach Learn 2001;45:5–32.

[8] Friedman JH. Greedy function approximation: a gradient boosting machine. Ann Stat 2001;29:1189–232.

[9] Niculescu-Mizil A, Caruana R. Predicting good probabilities with supervised learning. Proc 22nd ICML 2005:625–32.

[10] github.com/sureshkadiyala/netting-ml (accessed April 2024).

Author Information: Suresh Kadiyala, Department of Financial Technology & Quantitative Analytics, Hyderabad, India. Contact: suresh.kadiyala@gmail.com. Code: github.com/sureshkadiyala/netting-ml

Data Availability: Synthetic dataset with domain-informed parameters. Analyses conducted using open-source Python libraries (scikit-learn, pandas, numpy). All code and data generation procedures available at github.com/sureshkadiyala/netting-ml.

Conflicts of Interest: The author declares no conflicts of interest. This is a portfolio research project using synthetic data.