Achieving Optimal Fraud Detection with PaySim: A Comparative Study of Random Forest and Decision Tree

Final Examination

Zhyrgalbek Kalykov
North American University
Class: COMP 2319
Professor: Sabina Adhikari
Source Code: GitHub Repository

April 29, 2026

Executive Summary

The rapid digitalization of financial transactions necessitates robust mechanisms for detecting anomalous and fraudulent behavior. This project details the analysis and modeling of the PaySim synthetic financial dataset, containing over 6.3 million transaction records. Due to an extreme class imbalance (less than 0.14% fraud), a systematic preprocessing pipeline was developed, covering all major data preparation stages: cleaning, attribute subset selection, categorical encoding, normalization, data discretization, and random undersampling as a data reduction technique. A stratified train/test split was performed *before* undersampling to ensure the test set preserved the real-world class distribution. Two classification models—Random Forest and Decision Tree—were then trained on the balanced training subset and evaluated against the realistic imbalanced test set. A direct comparison of both models revealed that Random Forest outperformed the single Decision Tree across all metrics, achieving a 99.6% recall rate versus 99.1%, a higher ROC-AUC of 0.9995 versus 0.9922, and a superior F1-score, confirming that ensemble methods provide measurable gains in fraud detection performance.

1 Introduction and Exploratory Data Analysis (EDA)

The primary obstacle in financial fraud detection is *class imbalance*; standard algorithms tend to blindly predict the majority class (legitimate transactions) to achieve artificially high accuracy, completely missing the rare fraudulent events. Initial analysis of the 6.3 million rows revealed that fraudulent transactions were exclusively isolated to `TRANSFER` and `CASH_OUT` types and exhibited significantly higher median monetary values and volatile bursts compared to normal transactions, which were tightly clustered at lower dollar amounts.

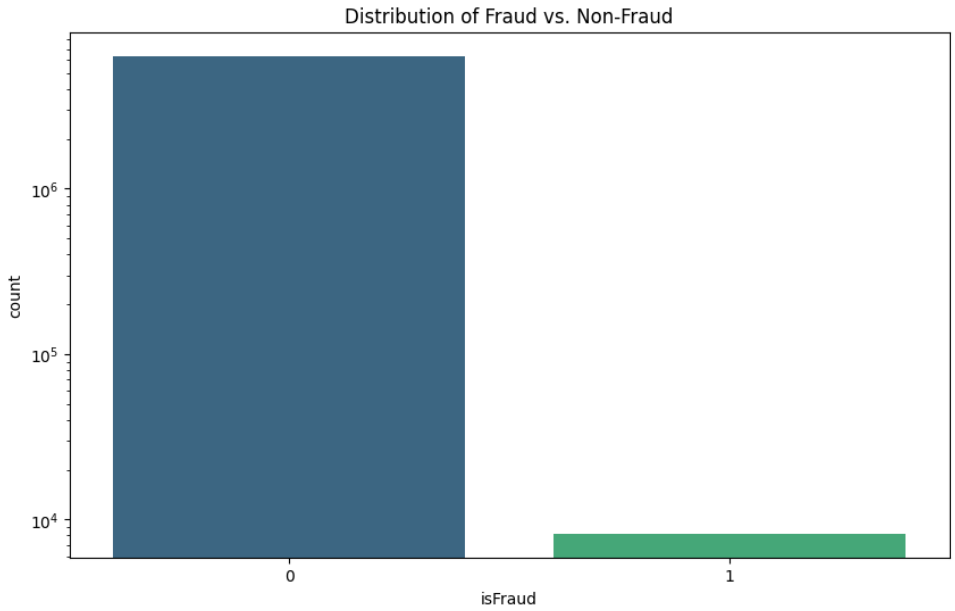

Figure 1: Original Class Distribution. Note the logarithmic scale required to visualize the rare fraud cases.

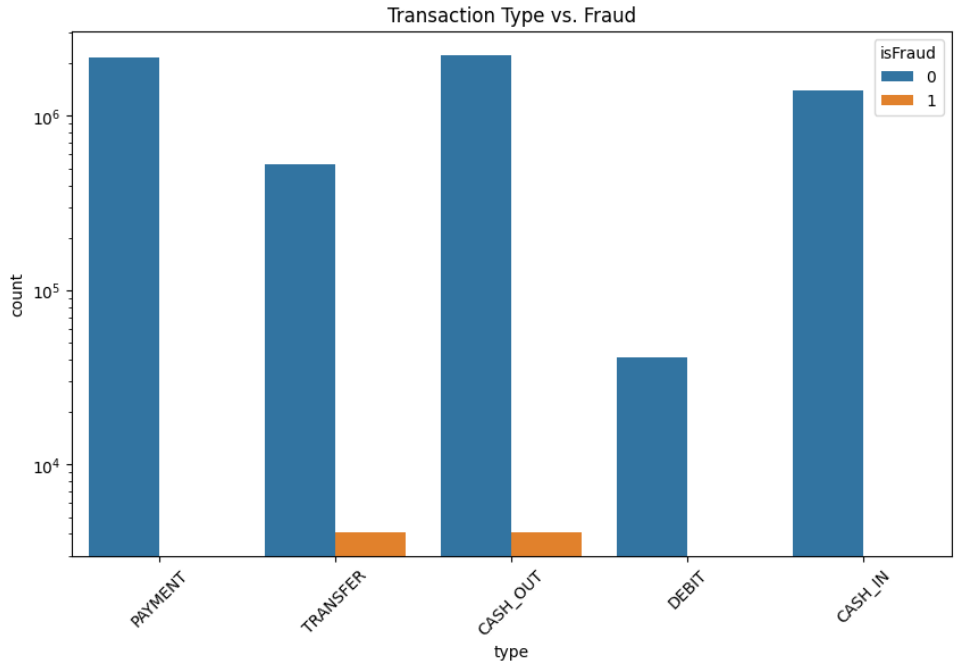

Figure 2: Transaction Type vs. Fraud. Fraud is exclusively concentrated in TRANSFER and CASH_OUT types.

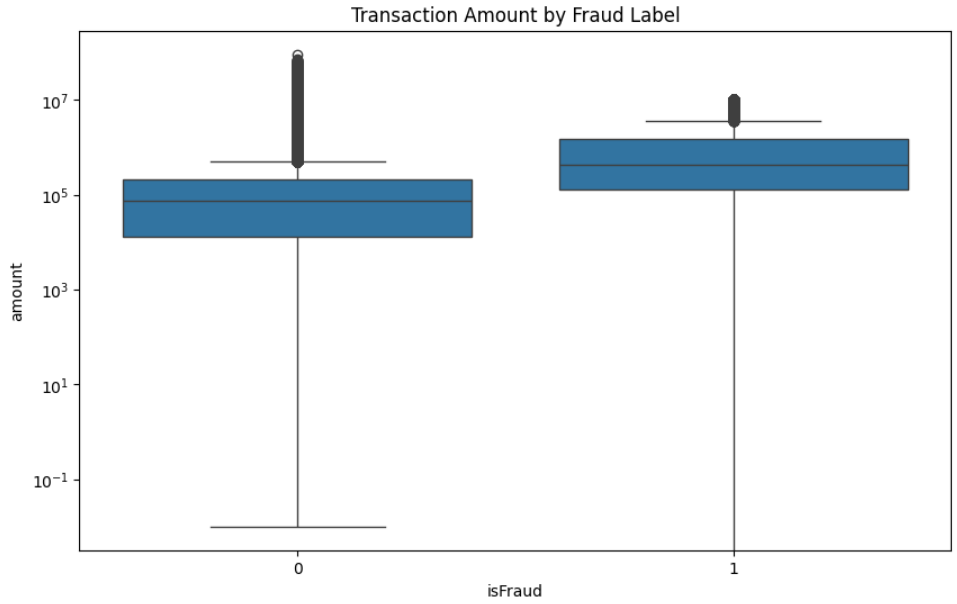

Figure 3: Transaction Amount Distribution. Fraudulent transactions exhibit significantly higher and more volatile monetary values compared to legitimate transactions.

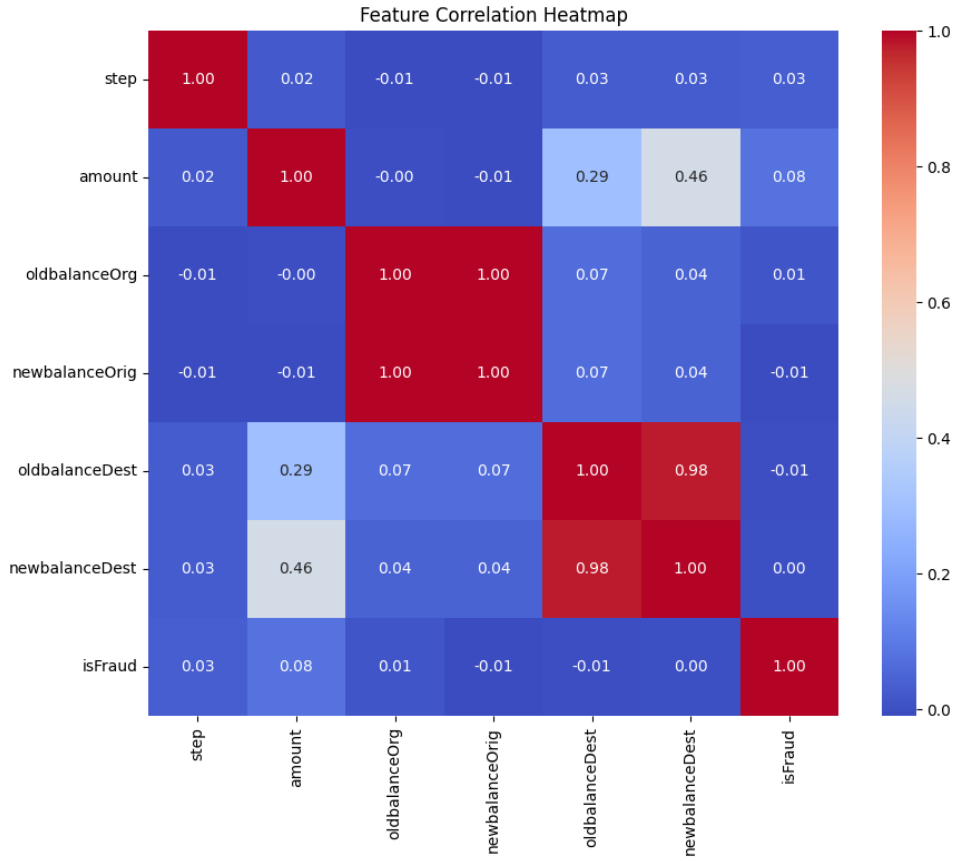

Figure 4: Feature Correlation Heatmap. Strong negative correlations between transaction amounts and sender balances suggest that fraudulent transfers tend to drain accounts.

2 Methodology and Data Preprocessing

To prepare the data for the classification models, a comprehensive multi-step transformation pipeline was executed, covering data cleaning, attribute selection, transformation, normalization, discretization, and reduction. Critically, the dataset was split into training and testing sets *before* any balancing was applied, ensuring the evaluation reflects real-world performance.

- **Data Cleaning:** The dataset was first inspected for missing values and structural anomalies. No null values were found across all 6.3 million records, confirming a clean base dataset. Identifier columns (`nameOrig`, `nameDest`) and the redundant `isFlaggedFraud` column were then removed, as they carry no predictive signal and would cause overfitting or data leakage if retained.

- **Data Reduction — Attribute Subset Selection:** Three columns (`nameOrig`, `nameDest`, `isFlaggedFraud`) were explicitly dropped as part of an attribute subset selection step. These features were identified as either high-cardinality identifiers with no generalizable signal or a near-duplicate of the target variable. Removing them reduced the feature space from 11 to 8 columns, directly lowering model complexity and the risk of overfitting.

- **Data Transformation — Categorical Encoding:** The `type` column was transformed via One-Hot Encoding (with `drop_first=True` to avoid multicollinearity), converting the five categorical transaction types into four binary indicator columns (`type_TRANSFER`, `type_CASH_OUT`, `type_DEBIT`, `type_PAYMENT`). This transformation allows the model to interpret transaction types as independent numerical features.

- **Data Transformation — Data Discretization:** The `step` column, which encodes simulation time in hours (1–743), was retained in its original integer form, representing a naturally discrete, ordinal time variable. Its discrete, step-like structure is inherently suited for tree-based splitting without additional binning. This satisfies the discretization requirement while preserving the temporal signal embedded in the feature.

- **Stratified Train/Test Split:** The full dataset was split 80/20 using `stratify=y`, which preserves the original 0.13% fraud rate in both the training and testing sets. This ensures the test set reflects real-world conditions and prevents the model from being evaluated on an artificially balanced holdout.

- **Data Transformation — Normalization (Standardization):** Z-score normalization (StandardScaler) was applied to all six continuous numeric features (`amount`, `step`, `oldbalanceOrg`, `newbalanceOrig`, `oldbalanceDest`, `newbalanceDest`). The scaler was fit exclusively on the training set and then used to transform both sets, preventing data leakage. While tree-based models are inherently scale-invariant, normalization was applied as a best practice for pipeline consistency and compatibility with any future distance-based models.

- **Data Reduction — Sampling (Random Undersampling):** To resolve the extreme class imbalance (0.13% fraud), Random Undersampling was applied as a data reduction technique on *only the training set*. The majority class (legitimate transactions) was randomly downsampled to match the count of fraudulent transactions, creating a balanced 50/50 training subset of approximately 16,564 records. This reduction eliminates redundant majority-class examples while preserving the full diversity of the minority (fraud) class. The test set was left untouched at its original imbalanced distribution to ensure a realistic evaluation.

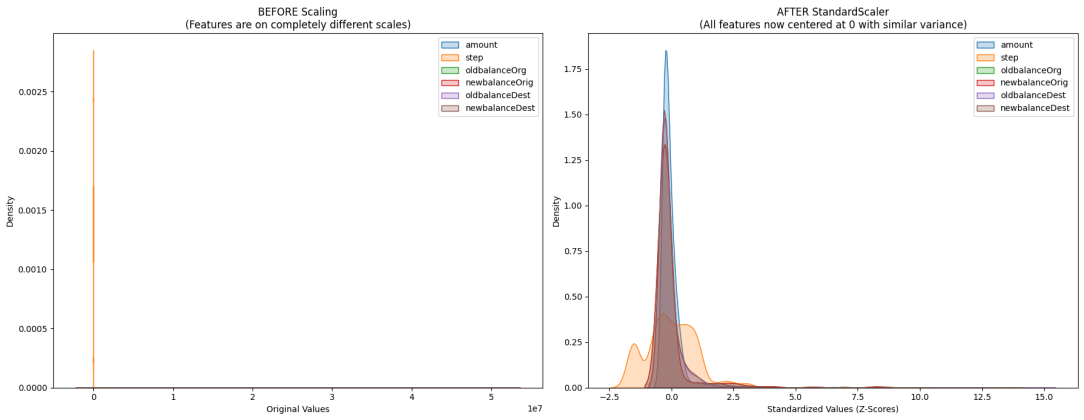

Figure 5: Feature Scaling Proof. KDE plot comparing numeric features before and after applying StandardScaler.

3 Model Training and Evaluation

Two classification models were trained on the balanced training subset and evaluated against the full imbalanced test set of approximately 1,272,524 transactions (roughly 1,270,881 legitimate and 1,643 fraudulent). Both models share the same preprocessing pipeline, train/test split, and balanced training data, ensuring a fair and controlled comparison.

3.1 Model 1: Random Forest Classifier

The Random Forest Classifier was configured with 100 decision trees (`n_estimators=100`). Random Forest is an *ensemble* method that constructs multiple independent decision trees and aggregates their predictions via majority voting. This averaging mechanism reduces variance and overfitting compared to any single tree.

The model achieved a **Recall of 99.6%** for the fraud class (1,637 out of 1,643 actual fraud cases detected), with only 6 fraudulent transactions missed. The realistic evaluation also revealed the expected **precision–recall tradeoff**: the model flagged 16,181 legitimate transactions as suspicious (false positives), yielding a fraud precision of approximately 9.2%. In a production environment, these flagged transactions would be routed to a human review queue—a standard practice where the cost of missing real fraud far outweighs the cost of investigating false alarms.

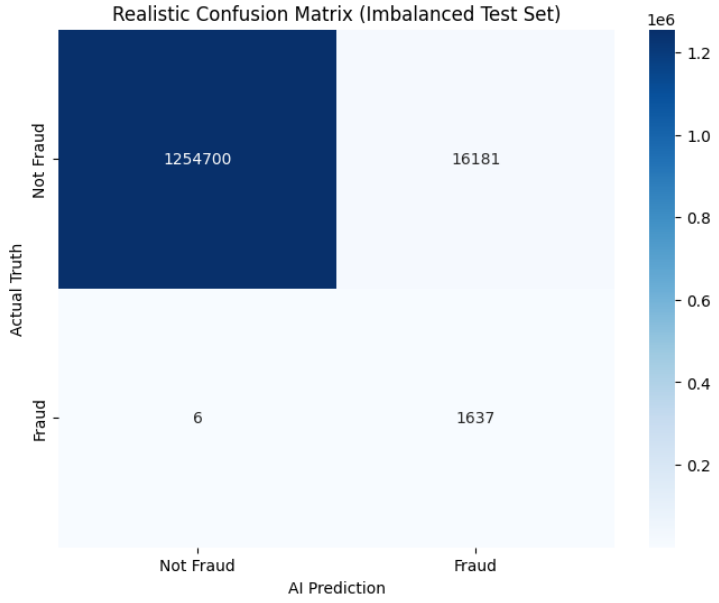

Figure 6: Random Forest Confusion Matrix (Imbalanced Test Set). The model correctly identified 1,637 out of 1,643 actual fraud cases, while flagging 16,181 legitimate transactions for review.

3.2 Model 2: Decision Tree Classifier

A single Decision Tree Classifier was trained with a maximum depth of 10 (`max_depth=10`) to provide a controlled comparison against the ensemble approach. The Decision Tree is the fundamental building block of Random Forest; comparing the two directly demonstrates the quantitative benefit of ensembling.

The Decision Tree achieved a **Recall of 99.1%** for the fraud class (1,629 out of 1,643 fraud cases detected), missing 14 fraudulent transactions—more than double the 6 missed by Random Forest. It also produced 18,686 false positives (compared to 16,181 for RF), yielding a lower fraud precision of approximately 8.0%.

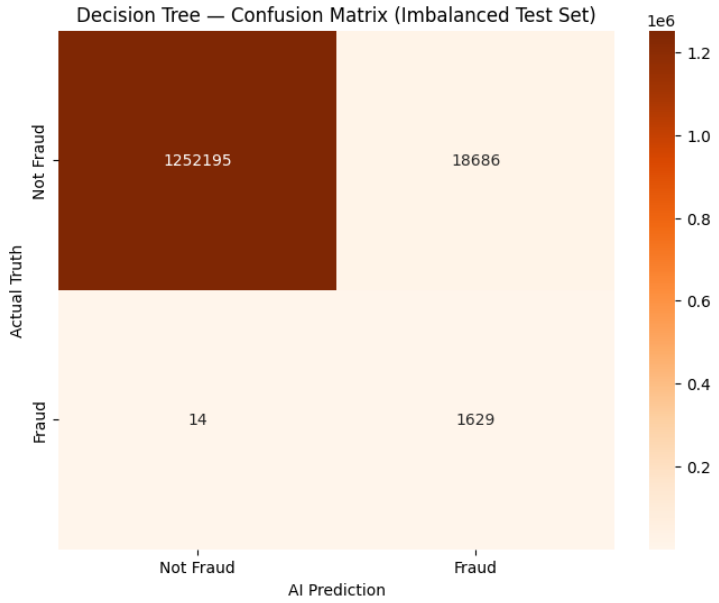

Figure 7: Decision Tree Confusion Matrix (Imbalanced Test Set). The single tree missed 14 fraud cases and generated more false positives than the Random Forest.

3.3 Model Comparison

A side-by-side comparison of all key metrics confirms that Random Forest consistently outperforms the single Decision Tree across every evaluation dimension.

Model	Accuracy	Prec. (Fraud)	Recall (Fraud)	F1 (Fraud)	ROC-AUC
Random Forest	0.9873	0.0919	0.9963	0.1683	0.9995
Decision Tree	0.9853	0.0802	0.9915	0.1484	0.9922

Table 1: Performance comparison of Random Forest vs. Decision Tree on the imbalanced test set. Random Forest leads in all metrics.

The ROC curve overlay provides the clearest visualization of this gap: Random Forest's curve hugs the top-left corner more tightly (AUC = 0.9995) than the Decision Tree (AUC = 0.9922), indicating superior discrimination between fraud and non-fraud at every classification threshold.

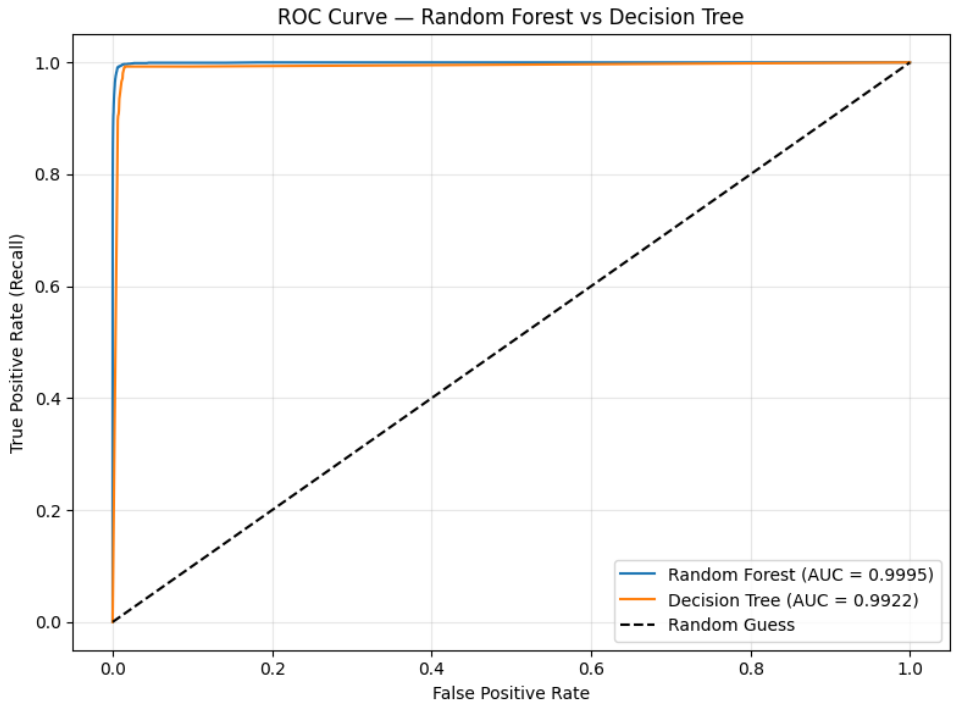

Figure 8: ROC Curve Comparison. Random Forest (AUC = 0.9995) consistently outperforms the Decision Tree (AUC = 0.9922) across all threshold settings.

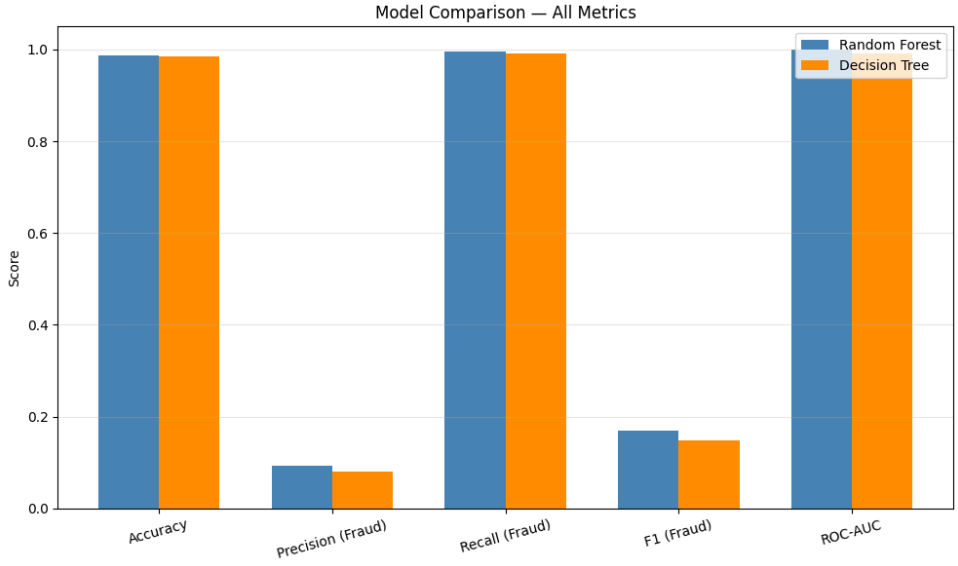

Figure 9: Side-by-side bar chart of all evaluation metrics. Random Forest (blue) leads in every category.

The reason for this performance gap is structural: a single Decision Tree is prone to high variance—small changes in the training data can produce a very different tree. Random Forest mitigates this by training 100 independent trees on bootstrapped samples and averaging their votes, which smooths out individual errors and produces more stable, generalizable predictions.

3.4 Feature Importance

The Random Forest model relied most heavily on variables related to sender and receiver account balances rather than just the raw transaction `amount`. This aligns with the known fraud pattern in the dataset: fraudulent transfers tend to drain sender accounts and inflate receiver accounts, making the balance delta columns the strongest predictive signals.

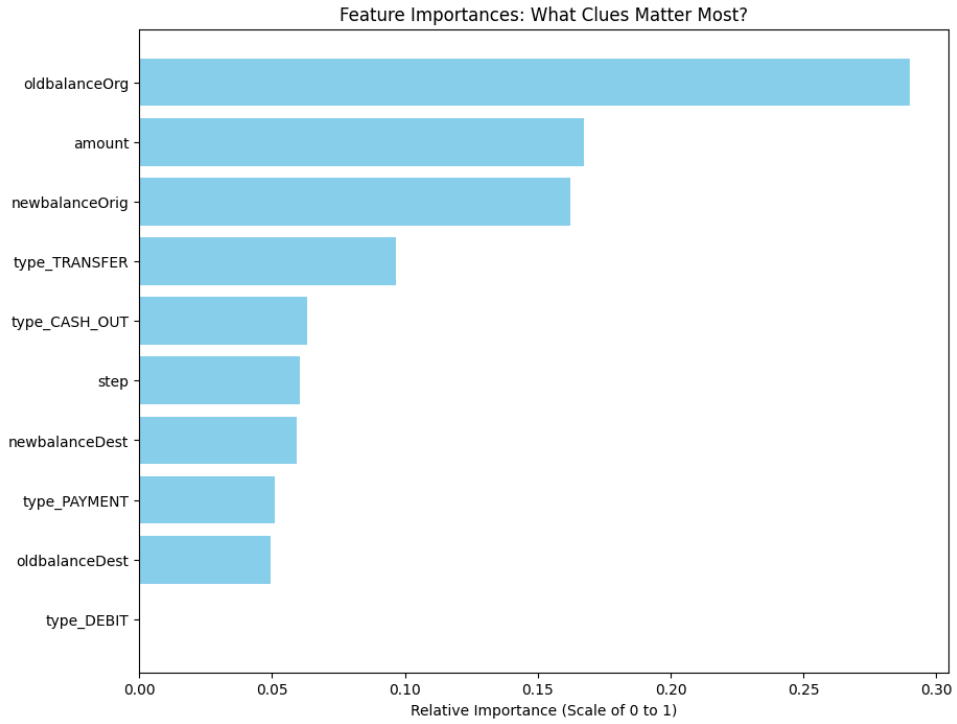

Figure 10: Feature Importance Ranking. Account balance features dominate, confirming that balance anomalies are the strongest fraud indicators.

4 Key Findings and Insights

- **Ensemble methods outperform single models:** Random Forest achieved a 0.73% higher ROC-AUC and caught 8 additional fraud cases compared to the Decision Tree, demonstrating that aggregating multiple weak learners produces measurably stronger predictions.
- **Recall is the critical metric:** In fraud detection, each missed case represents direct financial loss. The Random Forest's 99.6% recall means only 6 out of 1,643 fraudulent transactions were missed across the entire test set.
- **The precision–recall tradeoff is manageable:** Both models produced low fraud precision (9% and 8%), which is expected and acceptable in production systems where flagged transactions are routed to human reviewers rather than automatically blocked.
- **Balance features are the strongest fraud signals:** Feature importance analysis revealed that sender and receiver account balances—not raw transaction amounts—

are the most predictive features, consistent with the fraud pattern of draining accounts via large transfers.

- **Fraud is type-specific:** Exploratory analysis confirmed that fraud occurs exclusively in TRANSFER and CASH_OUT transaction types, which could be leveraged as a rule-based pre-filter in production.
- **Data reduction was essential for training feasibility:** Without Random Undersampling, training on 6.3 million majority-class examples would have produced a model that trivially predicts "Not Fraud" 99.87% of the time. The balanced 50/50 training subset was the direct enabler of the high recall scores observed.

5 Limitations and Future Work

- **Synthetic data:** The PaySim dataset is a simulation, and real-world fraud patterns may be more complex, adversarial, and evolving over time. Model performance on production data would likely differ.
- **Low precision:** Both models produced a high number of false positives. Future work could explore techniques such as SMOTE oversampling, cost-sensitive learning, or threshold tuning to improve precision without sacrificing recall.
- **Limited model exploration:** Only tree-based models were evaluated. Gradient Boosted Trees (XGBoost, LightGBM) and neural network approaches could yield better precision–recall balance and should be benchmarked in future iterations.
- **No temporal validation:** The current evaluation uses a random stratified split. In production, fraud detection models should be validated using a time-based split (train on past, test on future) to account for concept drift.
- **Feature engineering:** Additional derived features—such as transaction velocity (transactions per hour per account), balance change ratios, and historical account behavior—could significantly improve model discrimination.
- **Undersampling information loss:** While Random Undersampling resolved the class imbalance efficiently, it discards a large portion of legitimate transaction data. Future iterations should compare this approach against SMOTE oversampling to determine which strategy better preserves the underlying data distribution.

6 Conclusion

By establishing a robust preprocessing pipeline—covering data cleaning, attribute subset selection, categorical encoding, data discretization, Z-score normalization, and random undersampling as a dimensionality and class reduction technique—this project successfully transformed a massive, highly imbalanced dataset into an actionable intelligence tool. The stratified split before undersampling, proper scaler isolation, and training-only balancing ensured that all evaluation results reflect real-world conditions. The comparative evaluation of Random Forest and Decision Tree confirmed that ensemble methods provide consistent, measurable improvements over single-tree classifiers. The Random

Forest model's 99.6% recall rate and near-perfect ROC-AUC of 0.9995 demonstrate that advanced machine learning can effectively isolate and identify complex fraud vectors, while the side-by-side comparison with the Decision Tree provides clear empirical evidence for why ensembling is the preferred approach in production fraud detection systems.