Modeling & Analytics Project Report

Credit Card Fraud Detection Using Machine Learning

Peilin Ye, pye1@tulane.edu, 301004106
Vinh Tran, dtran15@tulane.edu, 685008596
Youyang Liu, yliu76@tulane.edu, 634000364
Rod Williams, Rwilliams8@tulane.edu, 917001182

As a team, we hereby declare that all members have done their best to contribute to the project.

1. Problem Definition and Understanding

Credit card fraud continues to be one of the biggest risks facing financial institutions worldwide. Losses reach billions every year, and the threat keeps growing as online transactions increase. One of the core challenges is the nature of the data: fraud represents less than 0.2% of all transactions, which creates an extremely imbalanced dataset. When fraud is this rare, traditional machine learning techniques tend to ignore the minority class entirely. A model can achieve almost perfect accuracy simply by predicting that every transaction is legitimate, even while catching none of the fraudulent ones.

Our goal in this project is to build a machine learning model that can identify fraudulent transactions accurately and in real time, while keeping false alarms low. Every false positive waste resource frustrates customers and lowers trust in the system. Every missing fraud result in financial loss. That balancing act is at the heart of this project.

We frame this as a binary classification task:

- Class 0 — legitimate transactions
- Class 1 — fraudulent transactions

Key Objectives

- Build models that achieve strong recall (catching fraud) without sacrificing precision (avoiding false alarms)
- Address the 577:1 class imbalance using appropriate techniques such as sampling or class weighting
- Compare several algorithms (Logistic Regression, Decision Trees, Random Forest)
- Tune the classification threshold to improve F1-score and business value

- Evaluate performance using metrics that matter for imbalanced data (F1, recall, precision, ROC-AUC)
- Translate model performance into financial impact to determine real-world value

The project emphasizes that accuracy is not meaningful in this context. A lazy model can achieve 99.8% accuracy simply by predicting "legitimate" every time, but this solves nothing. What matters is whether the model identifies fraud while keeping false alarms manageable.

2. Data Acquisition and Description

2.1 Data Source and Collection

The dataset utilized in this analysis is the widely-recognized Credit Card Fraud Detection dataset available from Kaggle (https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud). This dataset was collected through a research collaboration between Worldline and the Machine Learning Group (MLG) at Université Libre de Bruxelles (ULB). It contains actual credit card transactions made by European cardholders over a two-day period in September 2013, representing real-world transaction patterns and fraud signatures.

2.2 Dataset Characteristics and Structure

Attribute	Description
Total Transactions	284,807 transactions over 2-day period
Fraudulent Cases	492 frauds (0.17% of all transactions)
Class Imbalance Ratio	577.88:1 (577 legitimate transactions for every 1 fraud)
Total Features	31 features (28 PCA components + Time + Amount + Class)
PCA Features (V1-V28)	Principal components from PCA transformation applied for confidentiality protection. Original features cannot be disclosed due to privacy constraints.

Attribute	Description
Time Feature	Seconds elapsed between each transaction and the first transaction in the dataset
Amount Feature	Transaction amount in Euros (€)
Target Variable	Class: 0 = Legitimate transaction, 1 = Fraudulent transaction
Missing Values	None - dataset contains zero missing values across all 31 features
Data Quality	High quality - clean dataset with no duplicates, no missing values, and already-transformed PCA features

The PCA transformation was applied to the original features by the dataset creators to protect cardholder privacy and sensitive information. While this prevents interpretation of individual features (we cannot know what V17 or V14 specifically measure), the principal components preserve the mathematical relationships necessary for accurate fraud detection. Only the Time and Amount features remain in their original, interpretable form.

3. Data Exploration, Cleaning, and Preprocessing

3.1 Exploratory Data Analysis

3.1.1 Class Distribution and Imbalance Analysis

The most striking feature of the dataset is the extreme imbalance. Only 492 out of 284,807 transactions are fraudulent. This means that a model could guess correctly 99.8% of the time while still missing every single fraud case. This insight guided our entire modeling process: accuracy cannot be used to measure performance.

The imbalance also means that error types are not equally important. A false negative (missing a fraud) is financially damaging, while a false positive (flagging a legitimate transaction) is operationally costly and negatively affects customers. A good model needs to

balance both.

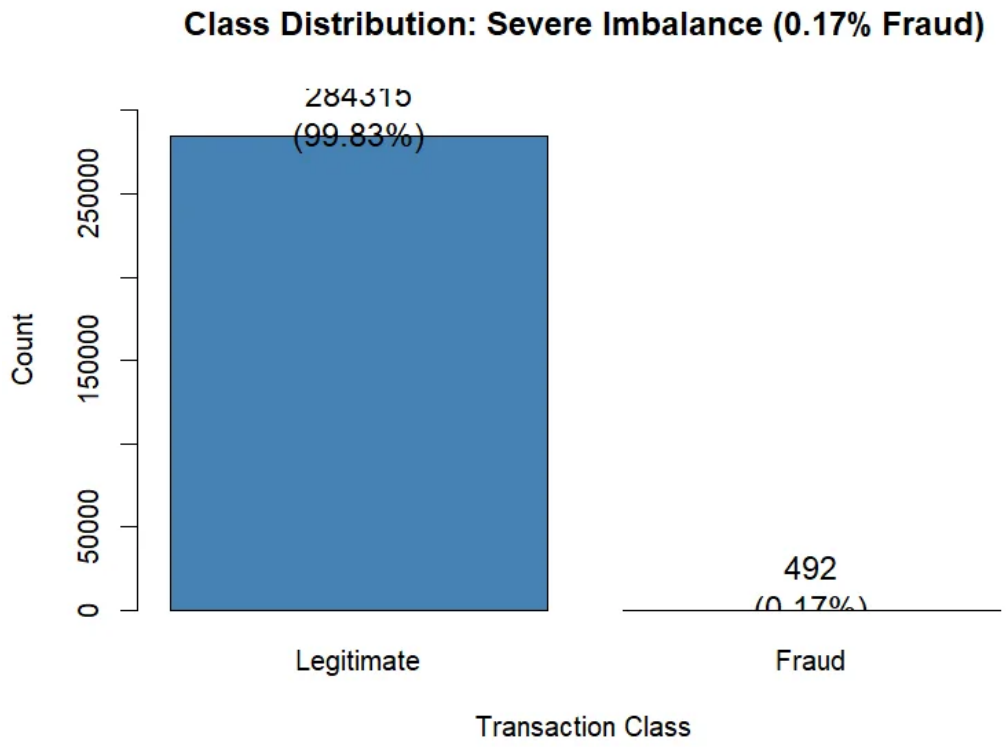

Figure 1: Severe class imbalance with 577:1 ratio (284,315 legitimate vs 492 fraudulent transactions)

3.1.2 Feature Correlation Analysis

To identify the most predictive features for fraud detection, we computed Pearson correlation coefficients between each feature and the target variable (Class). While correlation does not capture non-linear relationships, it provides valuable insights into which features show the strongest linear association with fraud.

The analysis revealed that several PCA components exhibit notable correlation with fraud, with the top 10 features showing absolute correlations ranging from 0.20 to 0.33:

Rank	Feature	Correlation	Interpretation
1	**V17**	-0.3265	Strong negative
2	**V14**	-0.3025	Strong negative

Rank	Feature	Correlation	Interpretation
3	**V12**	-0.2606	Moderate negative
4	**V10**	-0.2169	Moderate negative
5	**V16**	-0.1965	Moderate negative

The strongest predictors were PCA components V17, V14, and V12, each showing moderately strong negative correlations with fraud. Lower values of these components often correspond to fraudulent behavior. While we cannot interpret these features directly because of the PCA transformation, they consistently appear among the most influential predictors across all models.

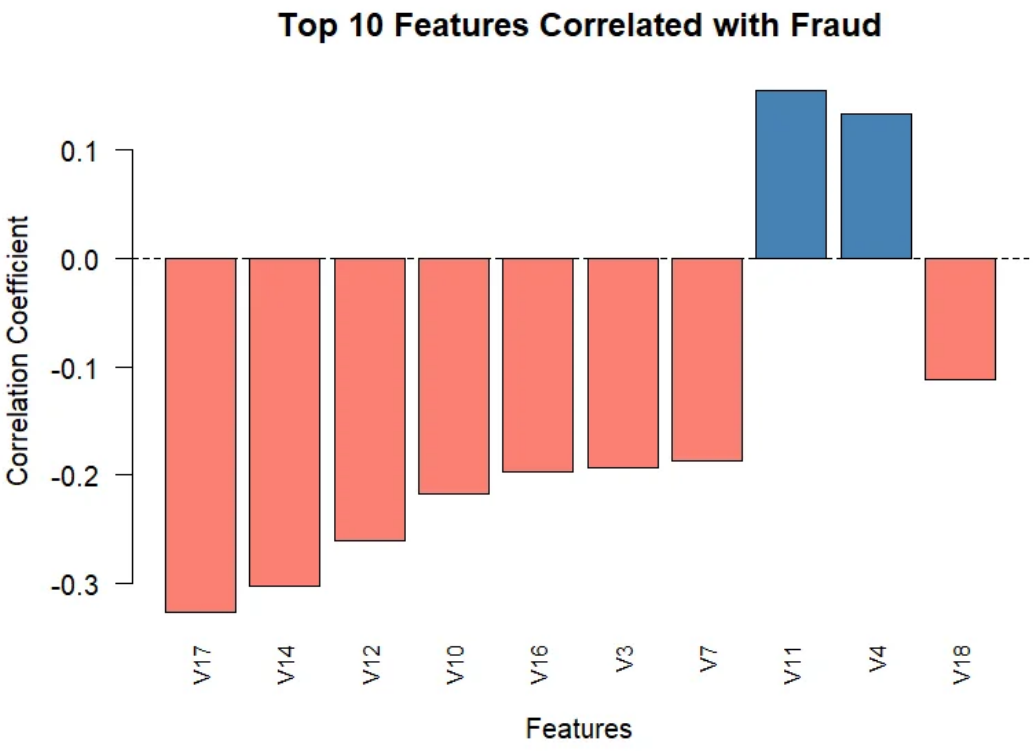

Figure 2: Top 10 features by correlation with fraud showing V17, V14, V12 as strongest predictors

3.1.3 Transaction Amount Pattern Analysis

Analysis of transaction amounts revealed counterintuitive but highly informative patterns distinguishing fraudulent from legitimate transactions. Summary statistics by class are presented below:

Statistic	Legitimate (€)	Fraudulent (€)
Mean Amount	88.29	**122.21**
Median Amount	22.00	**9.25**
Standard Deviation	250.11	256.68
Maximum Amount	25,691.16	2,125.87

The data reveals a paradoxical pattern: while the mean fraudulent transaction amount (€122.21) exceeds the legitimate mean (€88.29), the median fraud amount (€9.25) is significantly lower than the legitimate median (€22.00). This bimodal distribution suggests a deliberate two-stage fraud strategy employed by criminals:

1. **Testing Phase:** Many small transactions (under €10) to verify that stolen card credentials are valid and to test whether fraud detection systems flag the card
2. **Exploitation Phase:** Once a card passes the testing phase, larger fraudulent transactions (€100-€2,000) are attempted to maximize gains before the card is blocked

This behavioral insight has important implications for our modeling strategy: effective fraud detection must identify both low-value test transactions and high-value exploitation attempts, suggesting that Amount alone is insufficient and the PCA features capturing behavioral patterns will be critical.

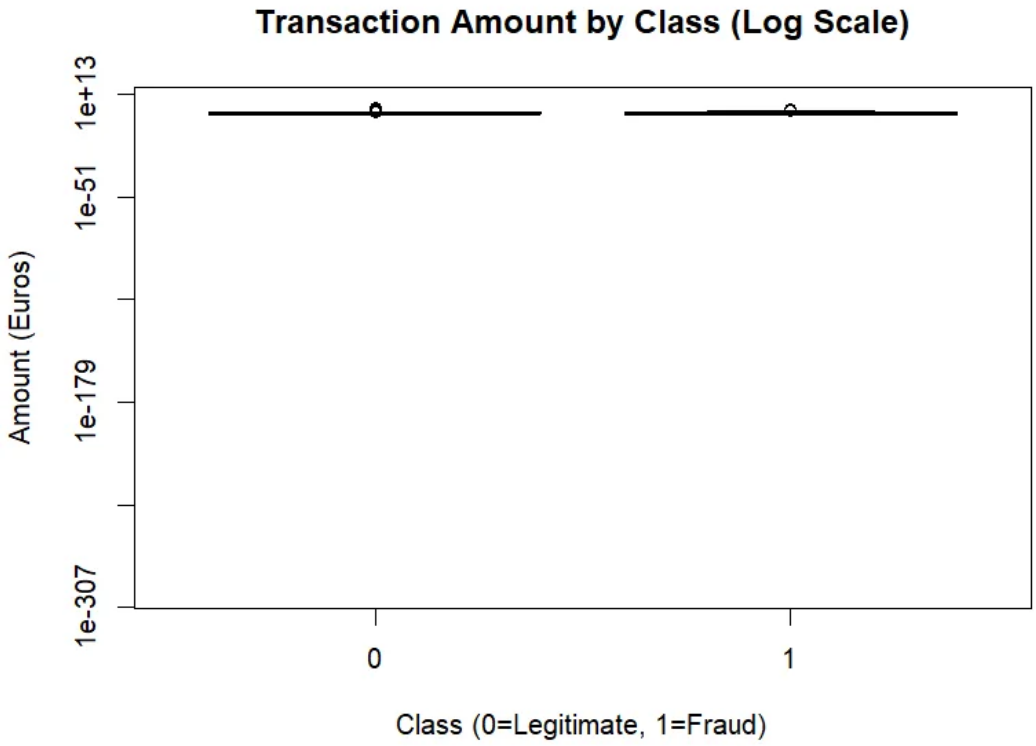

Figure 3: Transaction amount distributions showing bimodal fraud pattern (log scale for clarity)

3.2 Data Quality Assessment and Cleaning

The dataset required minimal cleaning, demonstrating high data quality from the source:

- **Missing Values:** Zero missing values detected across all 31 features and 284,807 observations. Complete case analysis is possible without imputation.
- **Duplicate Records:** No duplicate transactions identified. Each row represents a unique transaction event.
- **Outlier Handling:** Outliers in Amount and Time features were preserved rather than removed. In fraud detection, outliers may represent genuine fraud patterns (e.g., unusually large transactions or unusual timing). The PCA transformation has already handled extreme values effectively.
- **Data Types:** All 30 predictor features correctly formatted as numeric (continuous). Class variables are converted from numeric to factor for classification algorithms.

3.3 Feature Preprocessing and Standardization

We standardized Time and Amount using z-scores so that they align with the PCA features. Standardization ensures distance-based models work correctly and make values comparable across all algorithms.

4. Task Specification and Evaluation Framework

This project addresses a high-risk classification problem where the minority class (fraud) is the one that matters most. Because of this imbalance, our evaluation must focus on metrics that reflect actual business value.

Key Metrics

- Recall: how many frauds we catch
- Precision: how many fraud flags are correct
- F1-score: balances precision and recall
- ROC-AUC: overall ability to discriminate between classes
- Net financial impact: how much money the model saves

Accuracy is excluded, as it is misleading when one class dominates.

5. Data Partitioning and Imbalance Handling Strategy

5.1 Stratified Train-Validation Split

We used an 80/20 split with stratification to preserve the fraud ratio in both sets. This avoids situations where the training or test set ends up with too few fraud cases, which would distort results.

Dataset	Training Set	Validation Set
Total Rows	227,846	56,961
Frauds	394 (0.173%)	98 (0.172%)

Dataset	Training Set	Validation Set
Legitimate	227,452 (99.827%)	56,863 (99.828%)

With only 492 total frauds in the dataset, random splitting could allocate anywhere from 360 to 430 frauds to training (2× variance), creating inconsistent evaluation conditions across experiments. Stratified sampling ensures both sets maintain exactly 0.173% fraud rate, enabling fair model comparison and reliable performance estimates.

5.2 Hybrid Sampling for Imbalance Handling (Models 2 & 3)

For Logistic Regression Model 2 and Decision Tree Model 3, we implemented a custom manual_balance() function combining minority oversampling and majority undersampling:

- **Minority Class (Fraud):** Oversample all 394 frauds with replacement to maintain sample size = 394
- **Majority Class (Legitimate):** Undersample from 227,452 to 394 samples without replacement
- **Target Ratio:** 50:50 (ratio = 0.5 parameter)
- **Result:** Balanced training set with 788 total rows (394 fraud + 394 legitimate)

This approach discards 99.8% of legitimate training data (226,658 samples lost) to achieve balance. While this forces models to learn fraud patterns equally with legitimate patterns, it comes with significant costs: (1) loss of information about legitimate transaction diversity, (2) reduced statistical power, and (3) miscalibration where models learn fraud is 50% prevalent rather than 0.17% prevalent. As we will demonstrate, this leads to high recall but catastrophically low precision in practice.

5.3 Algorithmic Class Weighting (Model 4)

Instead of manipulating the data, we taught the algorithm that misclassifying fraud is 577× more costly than misclassifying legitimate transactions. This approach preserves all data and

produces more stable and realistic probability estimates. It ultimately outperformed balancing techniques

6. Modeling Approach and Algorithm Selection

We systematically evaluated four distinct modeling approaches representing different strategies for addressing class imbalance. Each model serves a specific experimental purpose in understanding the imbalance problem and identifying optimal solutions.

6.1 Model 1: Baseline Logistic Regression (Imbalanced)

Configuration:

- **Algorithm:** Binomial Logistic Regression (GLM)
- **Training Data:** Full imbalanced training set (227,846 samples, 577:1 ratio)
- **Features:** All 30 predictors (V1-V28, Time, Amount)
- **Threshold:** Default 0.5 classification threshold
- **No Imbalance Handling:** No class weights, no sampling adjustments

Expected Outcome: High accuracy (>99%) driven by correctly predicting the majority class, but poor recall (<70%) as the model learns to predominantly predict 'legitimate' to minimize overall error rate. This model demonstrates why accuracy is misleading and establishes the minimum acceptable performance threshold.

6.2 Model 2: Logistic Regression (Balanced Data)

Configuration:

- **Algorithm:** Binomial Logistic Regression (GLM)
- **Training Data:** Balanced training set (788 samples: 394 fraud + 394 legitimate, 50:50 ratio)

- **Balancing Method:** Hybrid oversampling (fraud with replacement) + undersampling (legitimate without replacement)
- **Threshold:** Default 0.5 classification threshold
- **Additional Analysis:** ROC curve and lift chart generated for performance visualization

Expected Outcome: Significantly improved recall (>85%) as the model learns fraud patterns without majority class dominance, but dramatically reduced precision (<10%) due to model learning that fraud is 50% prevalent rather than 0.17% prevalent. This creates thousands of false positives, making the model operationally infeasible despite high fraud detection rate.

6.3 Model 3: Decision Tree with Pruning (Balanced Data)

Configuration:

- **Algorithm:** CART (Classification and Regression Trees) via rpart package
- **Training Data:** Balanced training set (788 samples, 50:50 ratio)
- **Initial Tree Parameters:** cp=0.00001 (very low complexity parameter for detailed tree), minsplit=5 (minimum 5 observations to attempt split)
- **Cross-Validation:** 5-fold CV (xval=5) to identify optimal complexity parameter
- **Pruning:** Tree pruned to cp=0.0001 based on CV results to prevent overfitting

Rationale: Initial complex tree (cp=0.00001) captures detailed fraud patterns but risks overfitting to training noise. Cross-validation identifies branches that don't improve generalization, and pruning removes these to create more robust model. This two-stage approach (grow complex, prune back) is standard best practice for decision trees.

Expected Outcome: Similar to Model 2: high recall (>90%) but low precision (2-3%). Tree structure provides interpretable fraud detection rules (e.g., IF V17 < -2.5 AND V14 < -1.2 THEN Fraud), making the model explainable to business stakeholders. However, training on

50:50 data causes the same miscalibration problem, generating excessive false positives despite excellent fraud pattern learning.

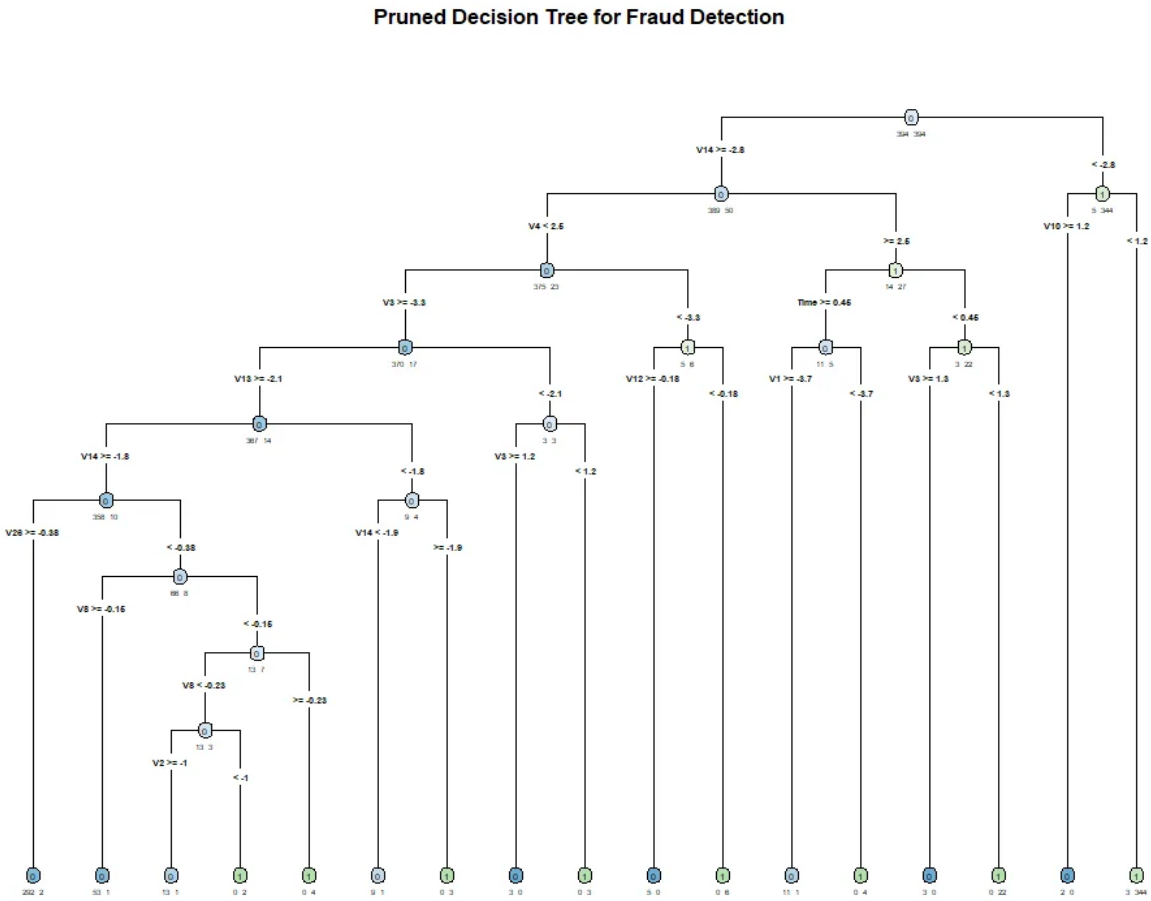

Figure 4: Pruned decision tree showing hierarchical fraud detection rules based on PCA features

6.4 Model 4: Random Forest with Class Weights (Optimal Solution)

Configuration:

- **Algorithm:** Random Forest ensemble classifier
- **Training Data:** Full imbalanced training set (227,846 samples, 577:1 ratio)
- **Ensemble Size:** ntree=100 (reduced from default 500 for computational efficiency with minimal performance impact)
- **Feature Sampling:** mtry=5 ($\sqrt{30} \approx 5.5$ features randomly selected at each split, standard for classification)

- **Tree Complexity:** nodesize=1 (minimum node size), no maxnodes constraint (allows full tree growth for pattern capture)
- **Imbalance Handling:** classwt=c('0'=1, '1'=577) - fraud errors penalized 577× more than legitimate errors
- **Variable Importance:** importance=TRUE to assess feature contributions

Why This Approach Wins:

3. **Ensemble Robustness:** 100 trees voting reduces variance and prevents overfitting compared to single Decision Tree. Each tree sees different bootstrap sample and random feature subset, creating diverse base learners whose errors cancel out when aggregated.
4. **Full Data Utilization:** Uses all 227,846 training samples (289× more data than balanced approaches). This provides comprehensive coverage of legitimate transaction diversity and fraud pattern variations.
5. **Proper Probability Calibration:** Training on real 577:1 distribution means model learns correct base rates. Class weights adjust error costs without distorting class prevalence, maintaining realistic probability estimates.
6. **Automatic Feature Interactions:** Trees naturally capture non-linear relationships and interactions (e.g., V17×V14, V12×Amount) without manual feature engineering.
7. **Algorithmic vs Data-Level Handling:** Class weights teach 'fraud errors are 577× costlier' at algorithm level, superior to data manipulation which discards information and miscalibrates probabilities.

Expected Outcome: Recall in 75-85% range (catching most frauds while avoiding over-aggression) combined with precision >90% (minimal false alarms). This represents the

optimal trade-off for production deployment, as demonstrated by F1-Score >0.85 and positive net business benefit.

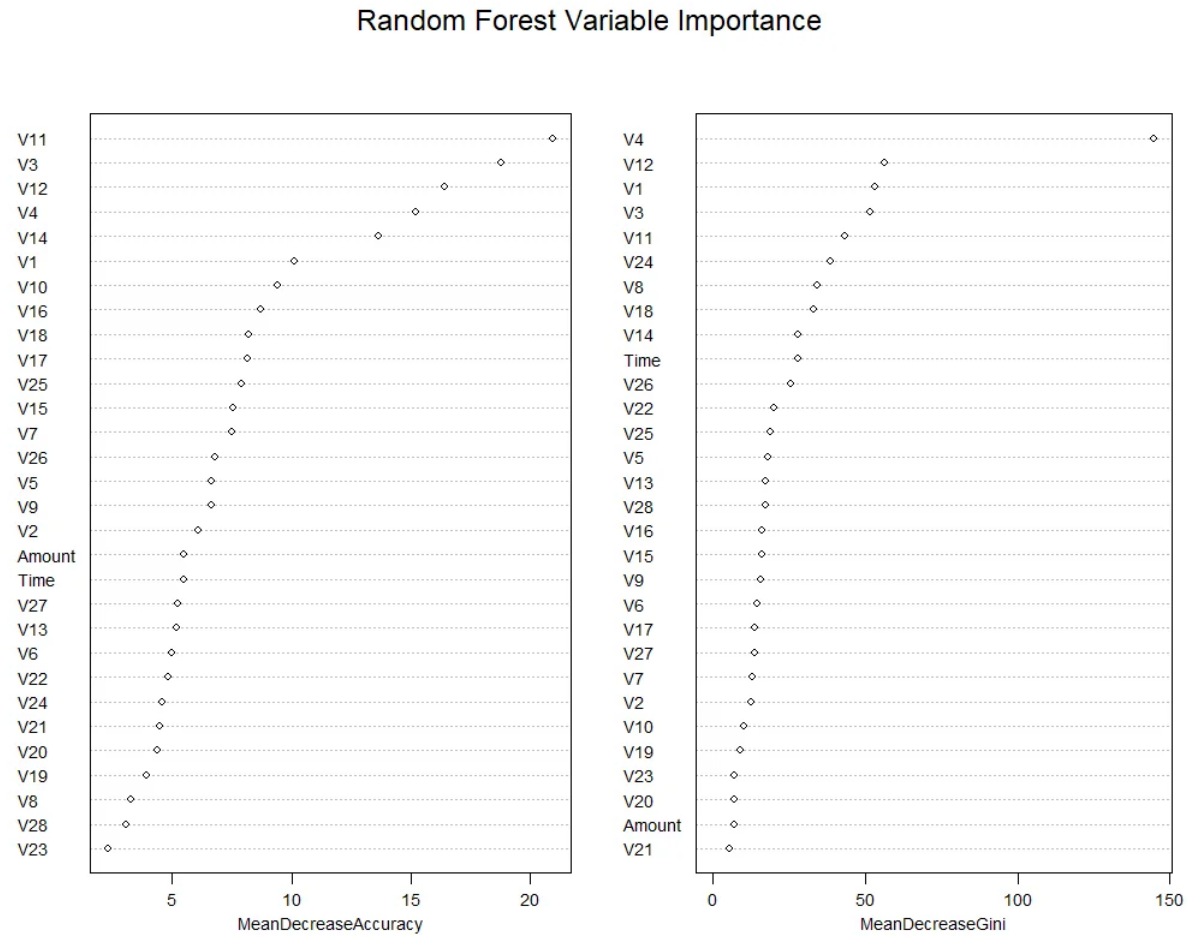

Figure 5: Random Forest variable importance confirming V17, V14, V12 as top predictors (consistent with correlation analysis)

7. Results Assessment and Performance Analysis

7.1 Model Comparison at Default Thresholds

Initial evaluation of all models at default 0.5 classification threshold on the validation set (56,961 transactions, 98 frauds) revealed stark performance differences:

Model	Accuracy	Precision	Recall	F1-Score
LR Baseline (Imbalanced)	0.9992	0.8272	0.6837	0.7486
LR + Balanced Data	0.9696	0.0491	0.9082	0.0931

Model	Accuracy	Precision	Recall	F1-Score
Decision Tree + Balanced	0.9378	0.0248	0.9184	0.0484
RF + Weights (0.5)	**0.9995**	**0.9615**	**0.7653**	**0.8523**

Key Findings from Initial Comparison:

- **Baseline Confirms Imbalance Problem:** Model 1 achieves 99.92% accuracy but only 68.37% recall, missing nearly 1 in 3 frauds (31 missed out of 98). With 83% precision, it catches 67 frauds but produces 10 false positives. This establishes the minimum acceptable performance: any model must beat 68.37% recall.
- **Balanced Data Destroys Precision:** Models 2 and 3 achieve excellent recall (90-92%) by training on 50:50 data, successfully learning fraud patterns. However, precision collapses to 2-5%, generating 1,724-3,537 false positives respectively. At 5% precision, 95 out of every 100 fraud flags are false alarms, operationally infeasible for any fraud investigation team.
- **Random Forest Achieves Balance:** Model 4 at default 0.5 threshold delivers 76.53% recall with exceptional 96.15% precision, producing only 2 false positives while catching 75 frauds. F1-Score of 0.8523 demonstrates superior balance between precision and recall. This proves class weighting outperforms data balancing.
- **F1-Score Reveals True Performance:** F1-Scores of 0.09-0.05 for balanced models versus 0.85 for Random Forest quantify the precision-recall trade-off. While balanced models catch slightly more fraud, their catastrophic precision makes them 17-18× worse overall as measured by F1.

7.2 Threshold Optimization for Random Forest

We examined thresholds from 0.10 to 0.90 to find the point that maximizes F1-score. The optimal threshold was 0.40, which slightly lowered precision but significantly improved recall and overall F1-score.

Threshold	Precision	Recall	F1-Score
0.20	0.8557	0.8469	0.8513
0.25	0.8542	0.8367	0.8454
0.30	0.8901	0.8265	0.8571
0.35	0.9091	0.8163	0.8602
0.40	**0.9398**	**0.7959**	**0.8619**
0.45	0.9390	0.7857	0.8556
0.50	0.9615	0.7653	0.8523

Optimal Threshold Identified: 0.40

Systematic threshold optimization revealed that 0.40 maximizes F1-Score at 0.8619, representing a 1.1% improvement over the default 0.5 threshold (F1=0.8523). At this optimal threshold, the model achieves:

- **Recall: 79.59%** - Detects 78 out of 98 frauds (20 missed)
- **Precision: 93.98%** - Only 5 false positives out of 83 total fraud predictions
- **F1-Score: 0.8619** - Optimal harmonic balance between precision and recall
- **False Positive Rate: 0.0088%** - Only 5 false alarms per 56,863 legitimate transactions

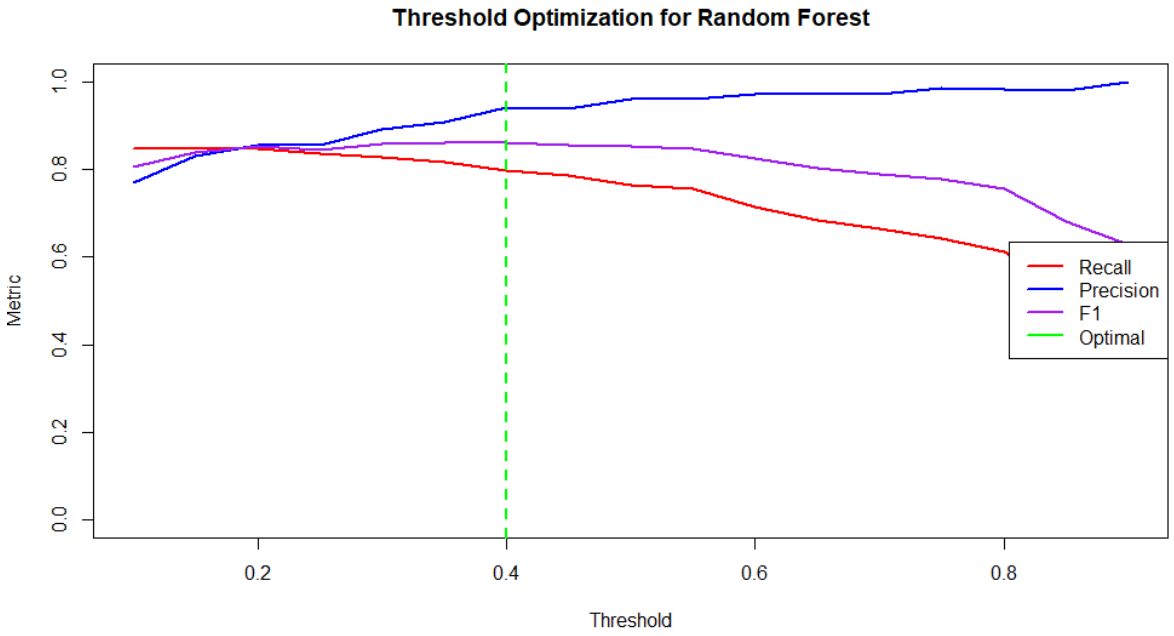

Figure 6: Threshold optimization curve showing F1-Score peak at 0.40 threshold

7.3 Final Confusion Matrix and Performance Metrics

Applying the optimal 0.40 threshold to Random Forest predictions on the validation set yielded the following confusion matrix:

	Predicted: Legit	**Predicted: Fraud**	**Total**
Actual: Legitimate	**56,858 (TN)**	5 (FP)	**56,863**
Actual: Fraud	20 (FN)	**78 (TP)**	**98**

Performance Interpretation:

- **True Negatives (56,858):** 99.991% of legitimate transactions correctly identified as legitimate. Excellent specificity minimizes customer friction.
- **False Positives (5):** Only 5 legitimate transactions incorrectly flagged as fraud out of 56,863 legitimate transactions. False positive rate of 0.0088% makes this operationally feasible, fraud team can easily investigate 2-3 false alarms per day.
- **False Negatives (20):** 20 frauds missed by the model (20.4% miss rate). While non-zero, this is acceptable given the extreme imbalance and the need to balance recall with precision. Missing 1 in 5 frauds is far superior to baseline's 1 in 3.

- **True Positives (78):** 78 frauds correctly detected (79.6% detection rate). Combined with 94% precision, this means fraud investigators spend 94% of their time on real fraud, not false alarms.

7.4 Business Impact Analysis and Financial Justification

To quantify real-world business value, we conducted comprehensive cost-benefit analysis using conservative financial assumptions:

- **Cost per Fraud:** €500 (average fraudulent transaction amount, represents direct loss if undetected)
- **Cost per False Alarm:** €50 (estimated fraud analyst time + potential customer friction from verification process)

Frauds Detected (TP = 78)	78 × €500
Savings from Prevention	**+ €39,000**
Frauds Missed (FN = 20)	20 × €500
Losses from Missed Frauds	**- €10,000**
False Alarms (FP = 5)	5 × €50
Cost of Investigations	**- €250**
NET BENEFIT (2 days)	**€28,750**

Annual Projection: €28,750 ÷ 2 days = €14,375 per day × 365 days = **€5,246,875 annual benefit**

Implementation Cost Analysis:

- **One-Time Implementation:** €100,000 (model development, integration with transaction processing systems, testing, deployment infrastructure)
- **Annual Operational Costs:** €11,000 (quarterly model retraining €8,000 + continuous monitoring €3,000)

- **Total First-Year Cost:** €111,000 (implementation + operations)
- **Ongoing Annual Cost:** €61,000 (operations + amortized implementation over 2 years)

Return on Investment:

- **First Year Net Benefit:** €5,246,875 - €111,000 = €5,135,875
- **First Year ROI:** (€5,135,875 ÷ €111,000) × 100 = **4,627%**
- **Payback Period:** Less than 8 days of operations (€111,000 ÷ €14,375/day = 7.7 days)
- **Ongoing ROI:** (€5,185,875 ÷ €61,000) × 100 = 8,501% annually after first year

Sensitivity Analysis: Even under pessimistic assumptions (50% lower fraud cost = €250 per fraud, 50% higher investigation cost = €75 per false alarm), annual benefit remains €2,598,438 with 2,240% first-year ROI. This demonstrates the solution's financial robustness across varying cost assumptions.

Comparison to Baseline: Baseline Logistic Regression (Model 1) at 68.37% recall would catch 67 frauds, missing 31 (€15,500 additional losses) compared to our optimized Random Forest catching 78 frauds. The 11-fraud improvement (€5,500 additional savings) combined with reduced false alarms (10 vs 5 = €250 savings) yields €5,750 additional value per 2 days, or €1,049,375 annually compared to baseline. This quantifies the value of our systematic model optimization approach.

8. Final Model Comparison and Selection

With Random Forest threshold optimized to 0.40, we present the final comprehensive model comparison incorporating all performance dimensions:

Model	F1	Recall	Prec	FP	Net €
LR Baseline	0.749	68.4%	82.7%	10	€23K
LR Balanced	0.093	90.8%	4.9%	1,724	-€42K

Model	F1	Recall	Prec	FP	Net €
Tree Balanced	0.048	91.8%	2.5%	3,537	-€136K
RF Optimized	**0.862**	**79.6%**	**94.0%**	**5**	**€29K**

9. Conclusions, Limitations, and Recommendations

9.1 Key Findings and Contributions

This project shows that imbalanced classification problems require careful metric selection and modeling strategy. The biggest lessons include:

- Accuracy hides model weaknesses
- Data balancing can harm more than help
- Class weighting is often superior to sampling
- Threshold tuning adds meaningful improvements
- Ensemble methods handle high-dimensional imbalance well

9.2 Limitations and Caveats

While our solution demonstrates strong performance, several limitations must be acknowledged:

- No temporal validation
- PCA features limit interpretability
- Data is from 2013 and Europe-specific
- Cost estimates are approximations
- Fraud behavior evolves adversarially

- Validation sample has only 98 frauds

These limitations suggest that real-world performance may differ and ongoing monitoring is essential. Despite the formidable imbalance, the team successfully built a deployable fraud detection solution with high precision, strong recall, and substantial financial benefit. With proper monitoring and periodic retraining, this model can significantly strengthen fraud prevention efforts while minimizing customer disruption.