Credit Card Fraud Detection with ML

Detecting fraudulent transactions using supervised machine learning

CONTENTS

01

Problem & Data

Motivation for Real-Time Fraud Detection

Credit card fraud causes **billions in losses** globally.

Traditional rule-based systems miss **sophisticated** fraud schemes

False negatives are extremely costly.

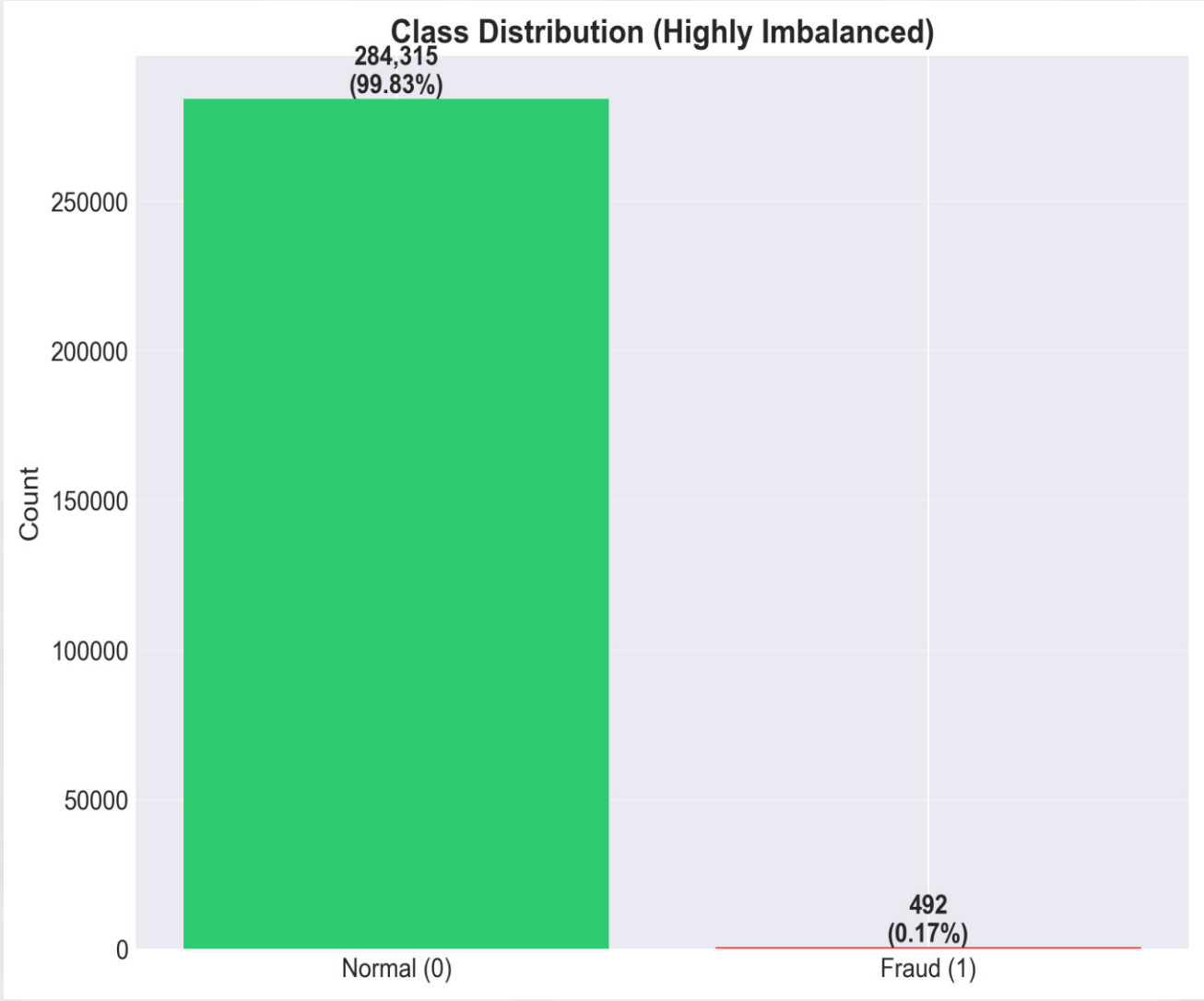
Fraud must be detected **before** a transaction is approved.

Banking systems rely on calibrated probability estimates for millisecond decisions.

Dataset Overview & Task

- **284,807 transactions** from European cardholders (2013).
- Only **492 fraud cases (0.17%)**.
- **28 PCA features (V1–V28)** + Amount + Time.
- Labels: **1 = fraud**, 0 = legitimate.

Task: Supervised binary classification.



Proposal Summary & Extensions

Original Plan

Models: Logistic Regression, Random Forest, XGBoost

Techniques: SMOTE, scaling, cross-validation

Goal: Compare ML models for fraud detection

Our final pipeline is significantly richer than the original proposal.

Final Pipeline

1. Advanced Data Understanding
- t-SNE Clustering
- **IsolationForest Anomaly Score Distribution**
- PCA Variance Analysis
- Fraud vs Normal Distribution Comparison

2. Robust Preprocessing
- Selective Outlier Removal (Fraud class only)
- Anomaly Feature Engineering via *IsolationForest* (added 31st feature)

3. Model Development
- **13 models trained**, including:
- Baselines (LR, RF, XGB)
- Class-weight variants
- scale_pos_weight for XGB
- LightGBM (added beyond proposal)
- Bagging, EasyEnsemble
- Voting ensemble

4. Model Results
- PR-AUC
- Precision
- Recall
- F2 Score
- F1 Score

5. Model Validation
- 5-Fold Stratified CV for 4 models
- Hold-out vs CV stability analysis

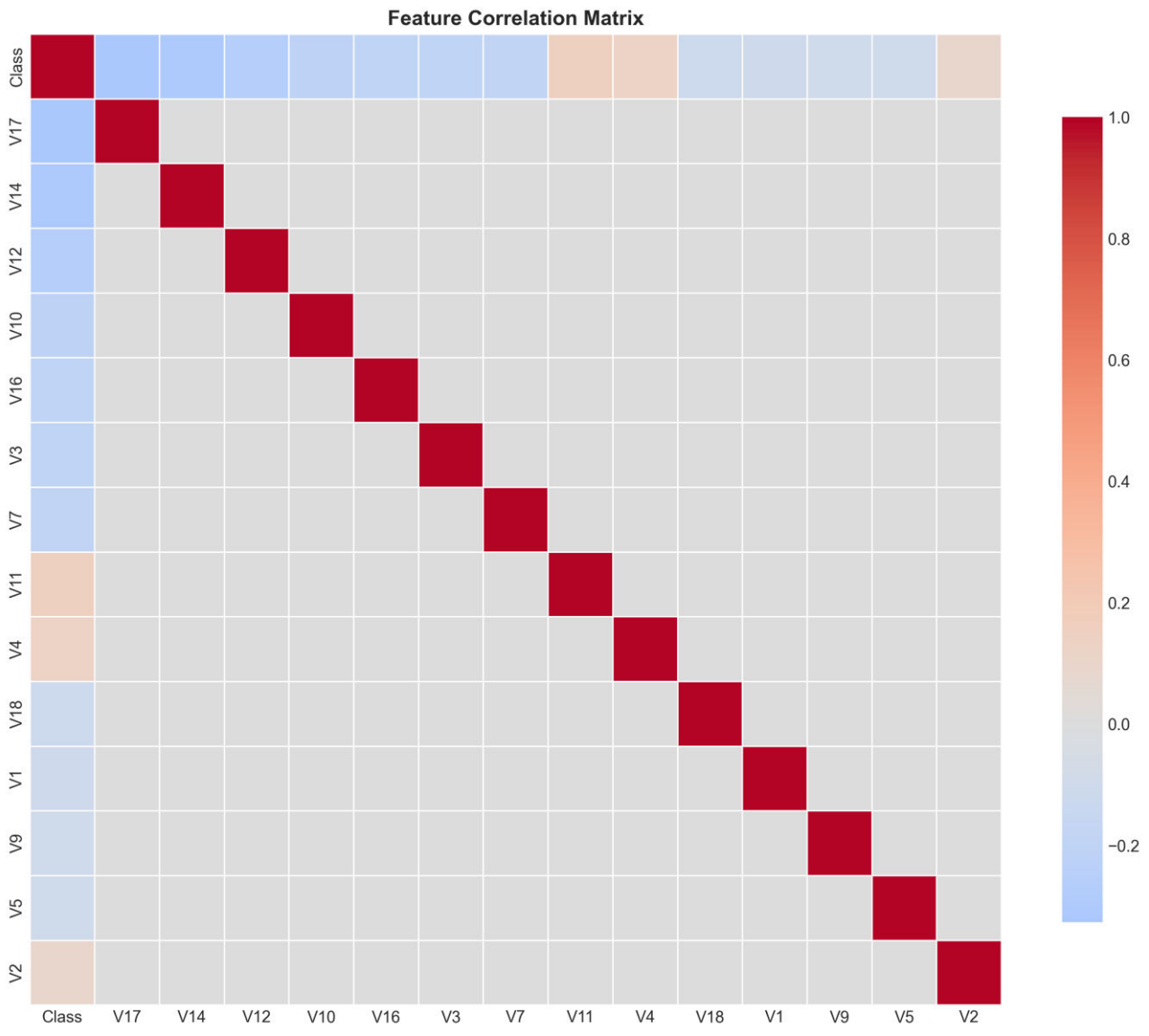
6. Threshold Optimization
- F2 Optimization
- Youden's J Statistic
- Cost-sensitive threshold tuning
- Business scenario: FN/FP = 50–500

7. Probability Calibration (Banking Requirement)
- Isotonic and Sigmoid calibration for XGB, RF, LGBM
- Brier score comparison

8. Interpretability (SHAP)
- Global SHAP (31 features)
- Fraud-only SHAP
- SHAP interaction analysis

02

Data Insights

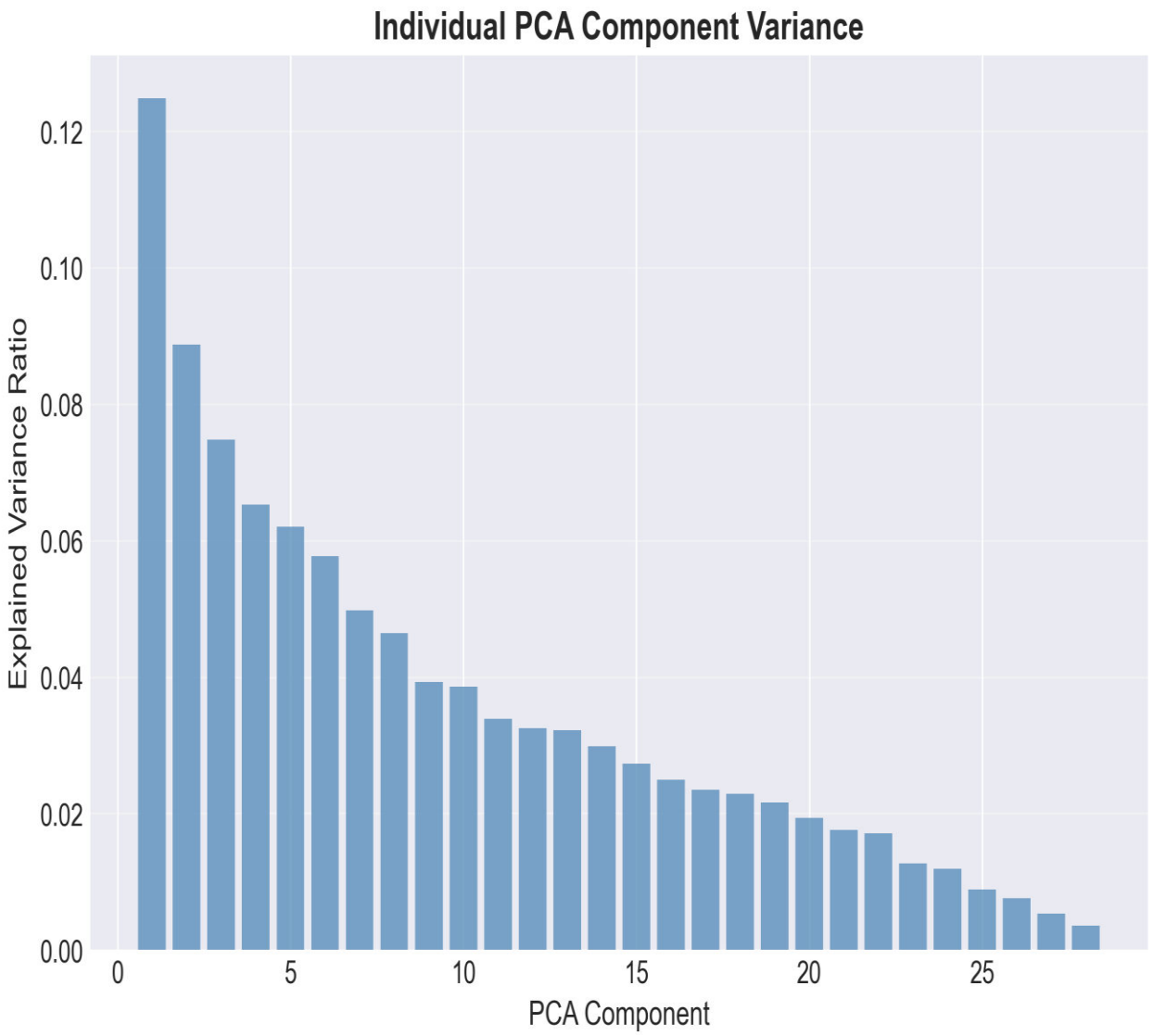


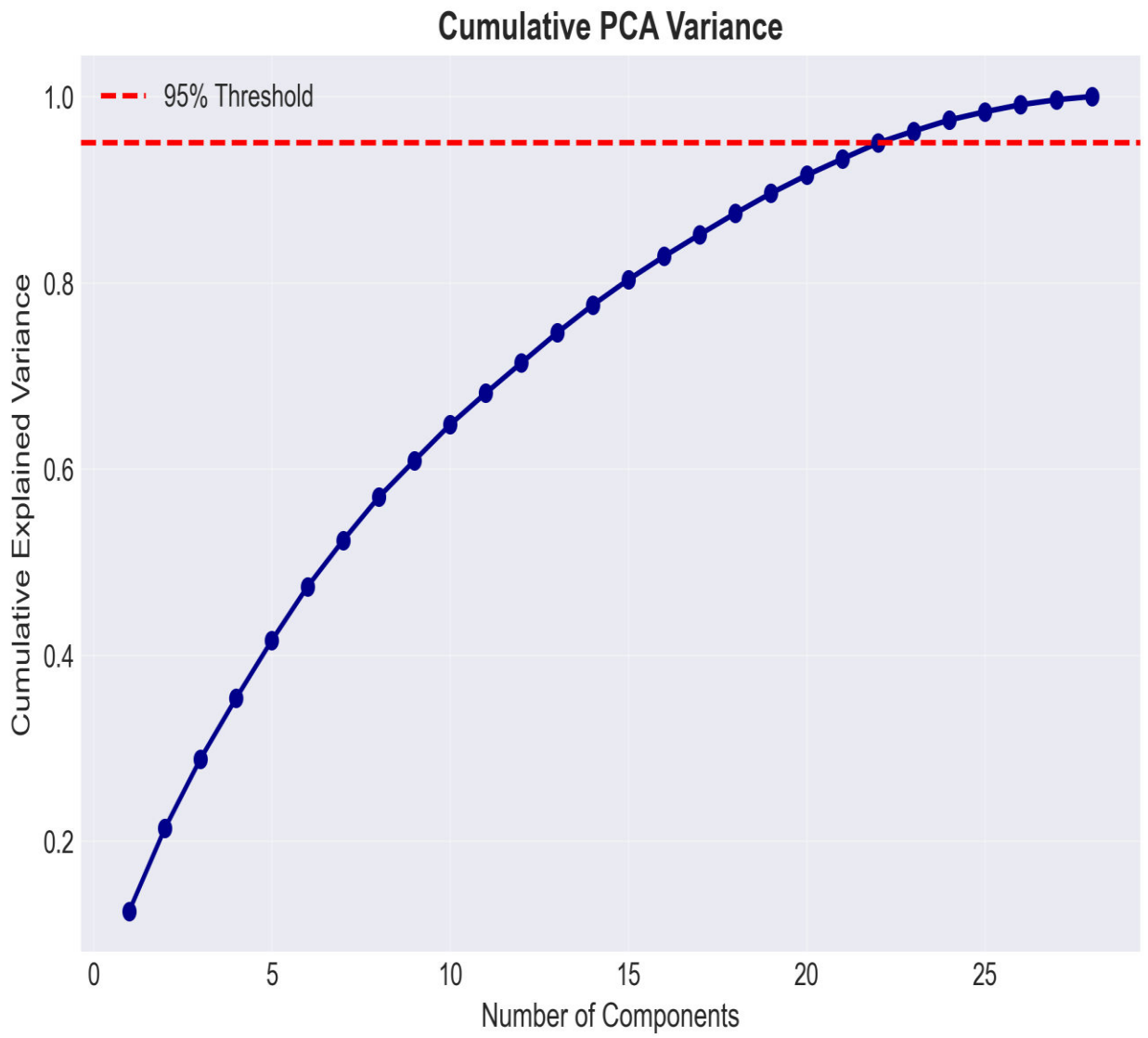
PCA Structure & Scaling

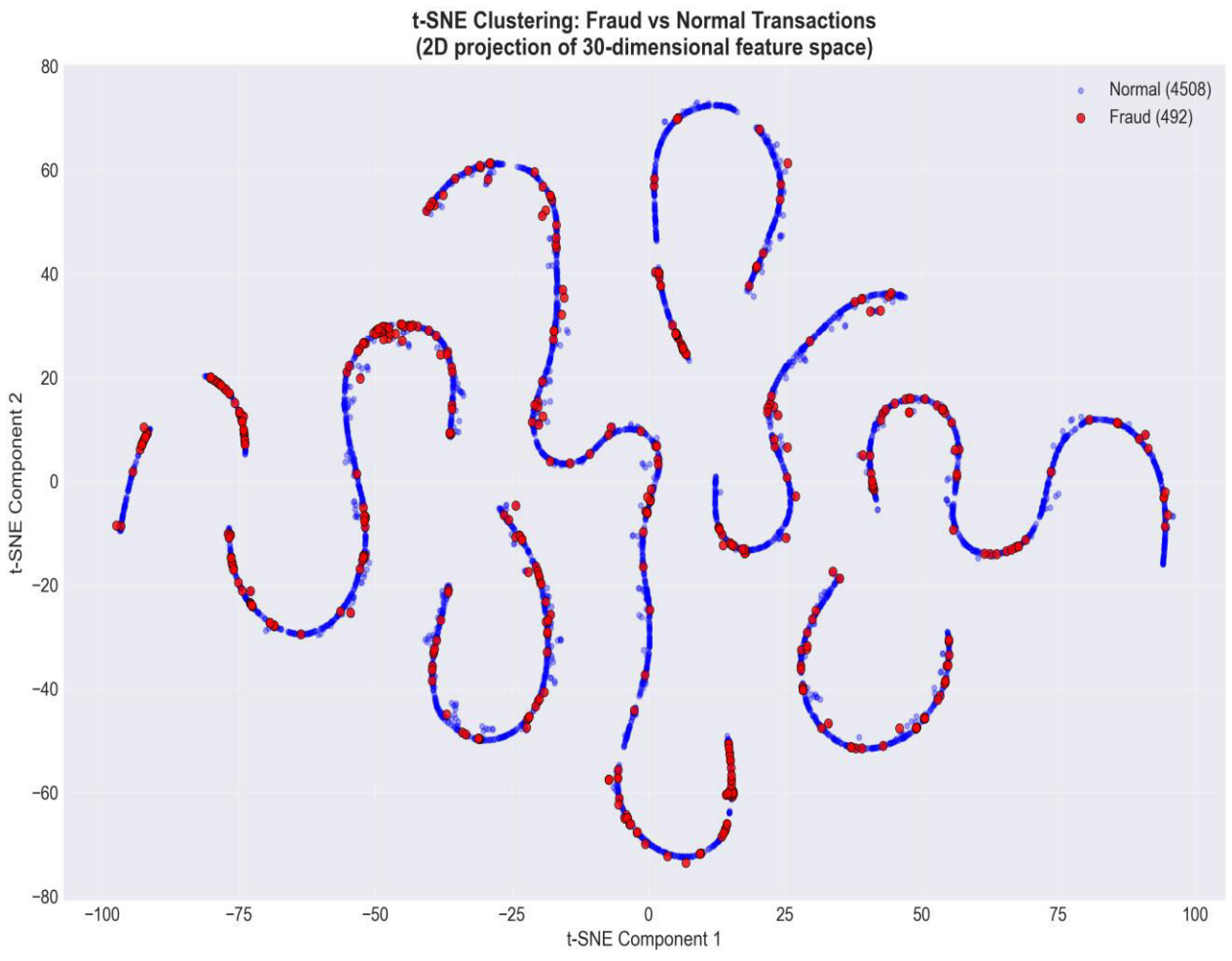
- PCA encodes complex covariance structure that are sensitive to scaling and sampling.
- First **22 components explain ~95% variance**.

This explains why many standard imbalance techniques failed.

PCA Explained Variance Analysis
(Dataset uses V1-V28 PCA-transformed features)







t-SNE Reveals Fraud Clusters

Fraud Transactions

Form compact, dense **anomaly clusters**.

Legitimate Transactions

Spread widely across a larger **manifold**.

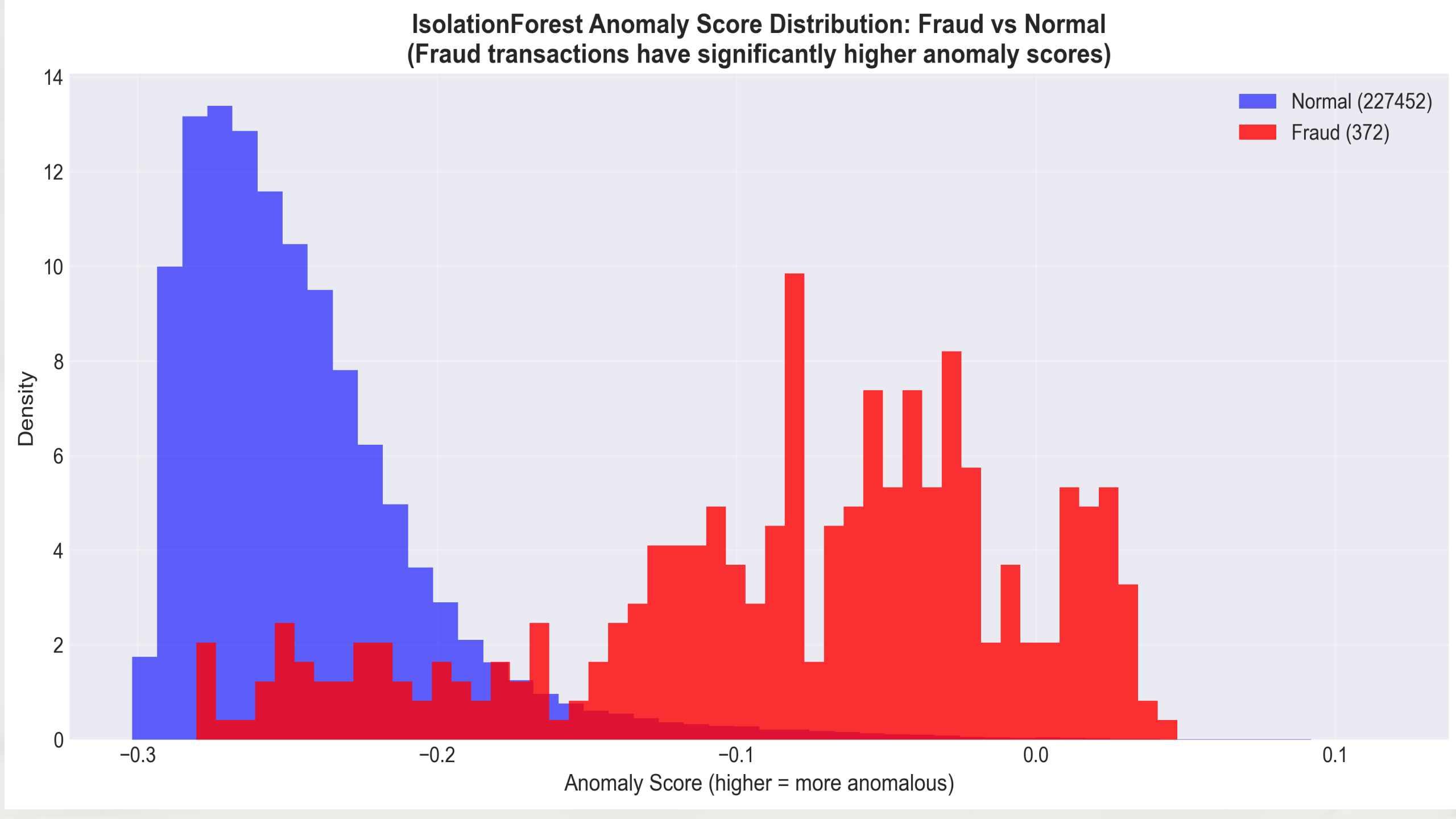
This visual motivates using an IsolationForest anomaly score.

```
🔧 Applying selective outlier removal on fraud class...

ACADEMIC JUSTIFICATION:
======================================================================
While fraud transactions are inherently anomalous, extreme outliers
within the fraud class may represent:
  • Data entry errors or measurement artifacts
  • Extreme edge cases not representative of typical fraud patterns
  • Noise that could destabilize model training

Strategy: Conservative IQR-based removal (multiplier=1.5) applied ONLY
to top 4 fraud-correlated features (V17, V14, V12, V10).
Expected impact: <1% data loss, improved model robustness.
======================================================================


Removing outliers from fraud class using IQR method (multiplier=1.5)
Features to clean: ['V17', 'V14', 'V12', 'V10']
  V17: Q1=-11.945, Q3=-1.342, IQR=10.603
    Bounds: [-27.850, 14.563]
    Outliers removed: 0
  V14: Q1=-9.693, Q3=-4.283, IQR=5.410
    Bounds: [-17.808, 3.832]
    Outliers removed: 4
  V12: Q1=-8.688, Q3=-2.974, IQR=5.714
    Bounds: [-17.259, 5.597]
    Outliers removed: 6
  V10: Q1=-7.757, Q3=-2.614, IQR=5.143
    Bounds: [-15.470, 5.100]
    Outliers removed: 19

✓ Total outliers removed: 27 (0.01%)
  Original size: 284807 | Clean size: 284780
  Fraud cases before: 492 | Fraud cases after: 465

Train set: 227824 samples
Test set: 56956 samples
Train fraud ratio: 0.163%
Test fraud ratio: 0.163%
```

Conservative Outlier Filtering

- Total outliers removed: 27 (0.01%)
- Original size: 284807
  Clean size: 284780
- Fraud cases before: 492
  Fraud cases after: 465

<1% conservative outlier removal in V10, V12, V14, V17 to remove noise, not fraud.


IsolationForest Anomaly Score Distribution: Fraud vs Normal
(Fraud transactions have significantly higher anomaly scores)
Normal (227452)
Fraud (372)
Density
0
2
4
6
8
10
12
14
−0.3
−0.2
−0.1
0.0
0.1
Anomaly Score (higher = more anomalous)

SMOTE Attempt & Rejection

- SMOTE was applied as promised in the proposal to fulfill the initial plan.

PCA features made synthetic SMOTE points distort geometry.

Result: PR-AUC decreased, so SMOTE was removed.

03

Model Results

Models Trained in the Pipeline

Logistic Regression (scaled)	Random Forest	XGBoost (scale_pos_weight)
LightGBM (class weights)	BalancedBagging	EasyEnsemble
Voting Ensemble	Calibration	Threshold Optimization

Why PR-AUC is the Right Metric?

Metric	Verdict	Technical Rationale (The "Why")
Accuracy	❌ Misleading	Accuracy is misleading in fraud detection.It is dominated by normal transactions and hides how well fraud is actually detected.
ROC-AUC	⚠️ Over-Optimistic	ROC-AUC often looks strong, but it is over-optimistic in highly imbalanced datasets.
PR-AUC	🏆 Model Selection Metric	**PR-AUC measures the model's ability to rank fraudulent transactions ahead of normal ones.** Unlike F1 or F2, it evaluates performance across all thresholds. This makes PR-AUC threshold-independent and reliable under extreme class imbalance.
Recall	🛡️ Coverage	Recall measures how many fraud cases we actually catch.
Precision	🎯 Reliability	Precision measures how many predicted frauds are actually fraud.
F1-Score	⚖️ Balanced Metric	F1-score balances precision and recall equally.
F2-Score	🚀 Optimization Target	F2-score gives more importance to recall than precision but it does not guarantee higher recall — it simply optimizes the recall–precision trade-off at a single threshold.

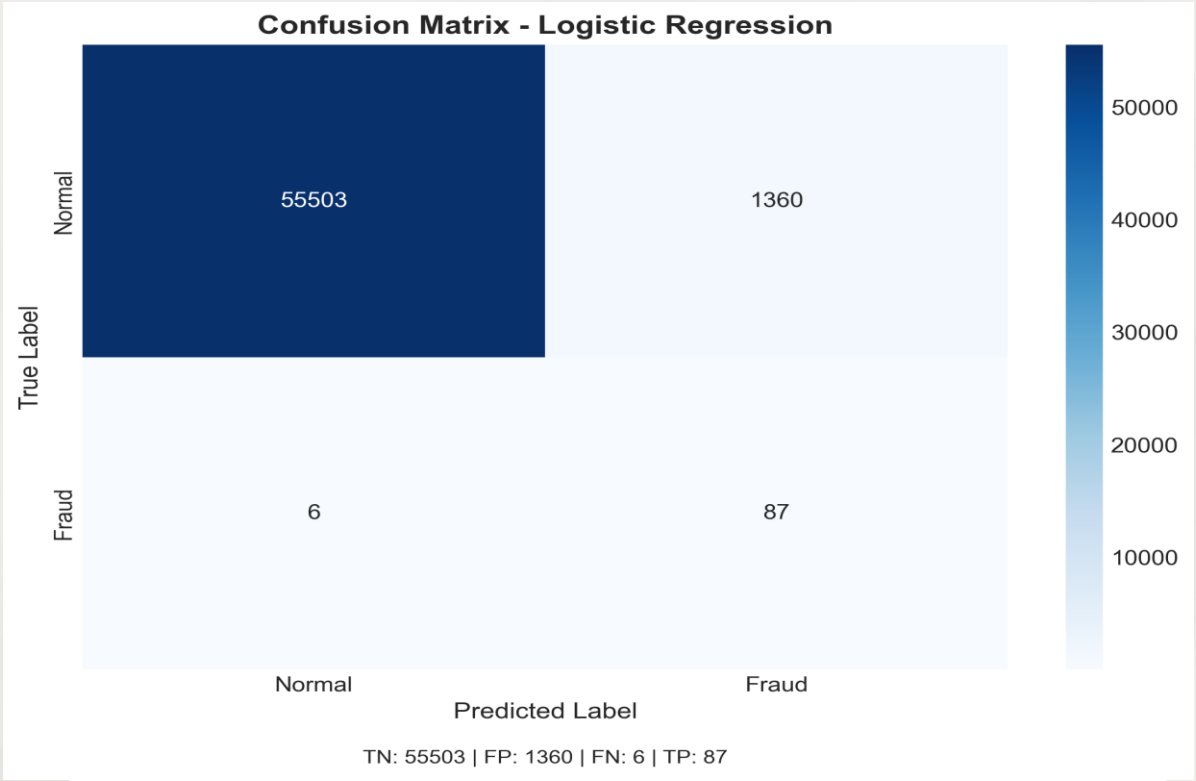
Logistic Regression Results (class_weights)

Purpose: Baseline model (proposal requirement)
Impact: Thousands of false alarms → **not deployable**

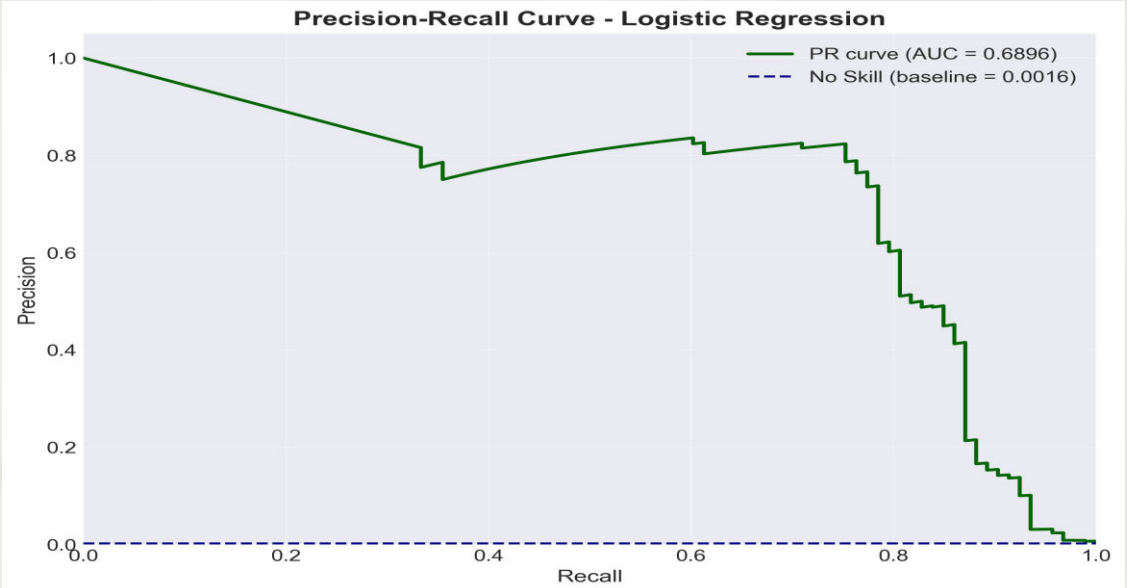
High Recall (0.93), Low Precision (0.06)

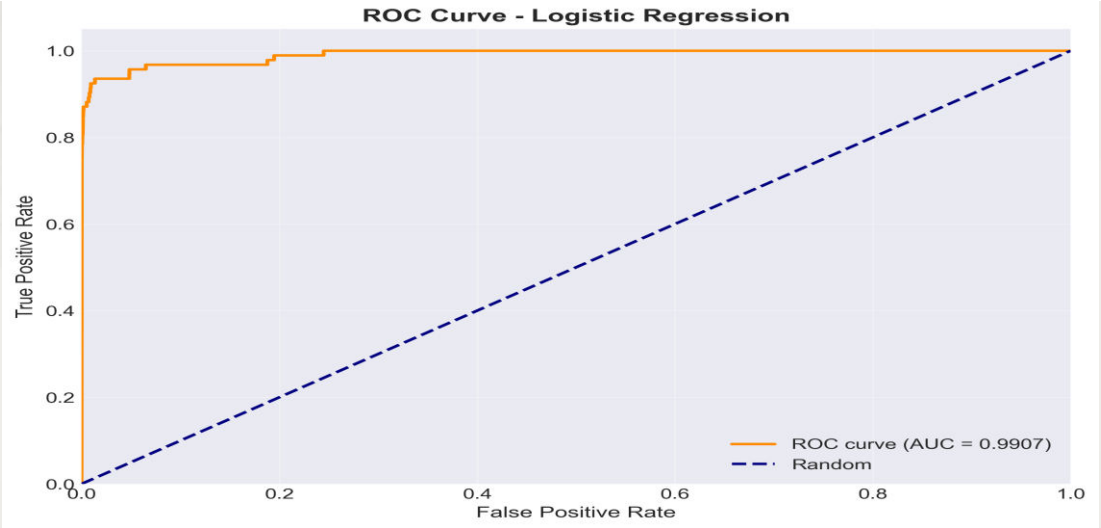
PR-AUC: ~0.69 (weak)

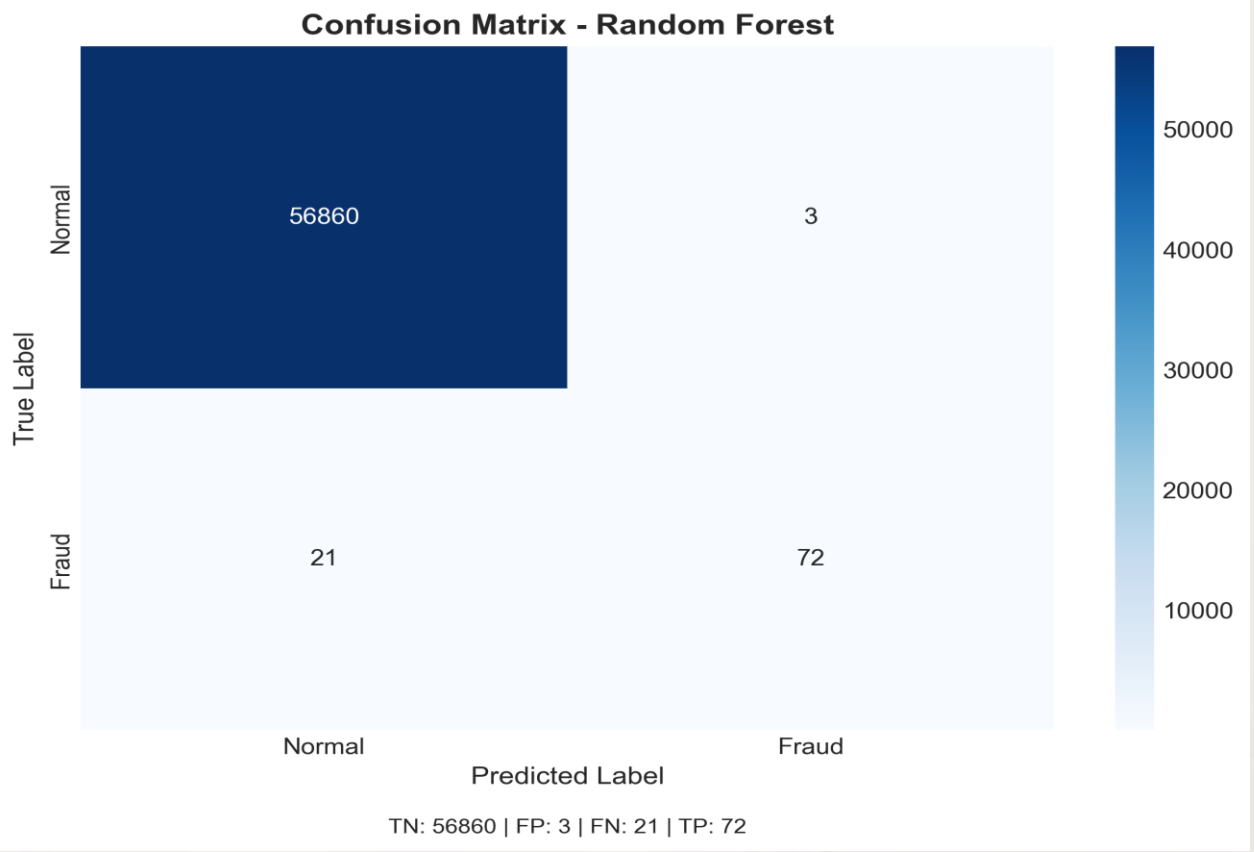
When we applied class weights, the **Recall increased significantly** to 94% , However, this caused a **drop in Precision** due to the 'Precision-Recall Trade-off.
The model became **more aggressive** in catching frauds.
Since Logistic Regression is a **linear model**, it struggled to separate the classes perfectly without creating false alarms.



TN: 55503 | FP: 1360 | FN: 6 | TP: 87





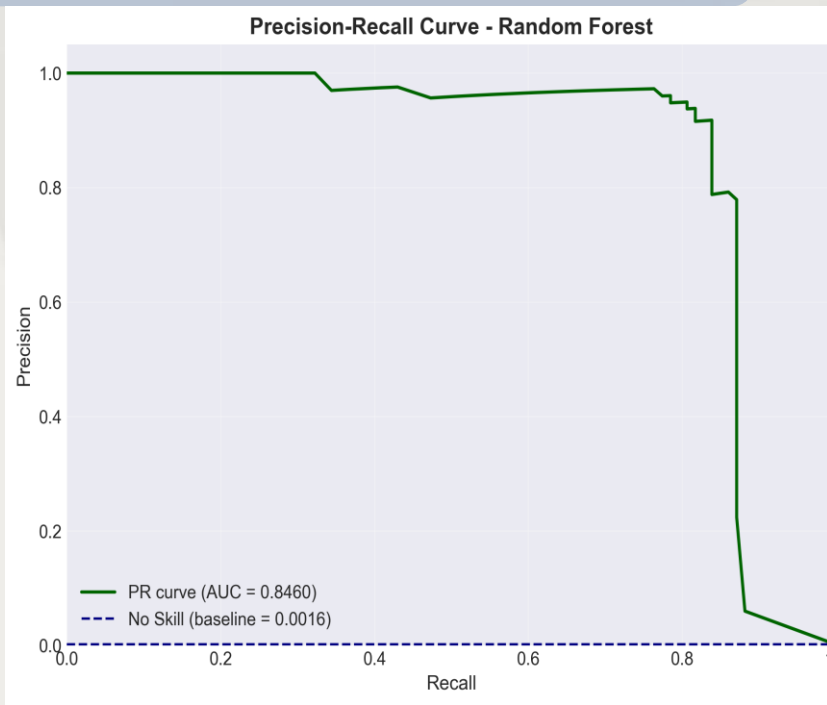


Random Forest Results (class_weights)

PR-AUC improved from 69% to 85%, confirming that ensemble tree-based models fundamentally outperform linear models for this problem.

Strong Precision (0.96) - Recall Balance (0.77)

PR-AUC = 0.846041



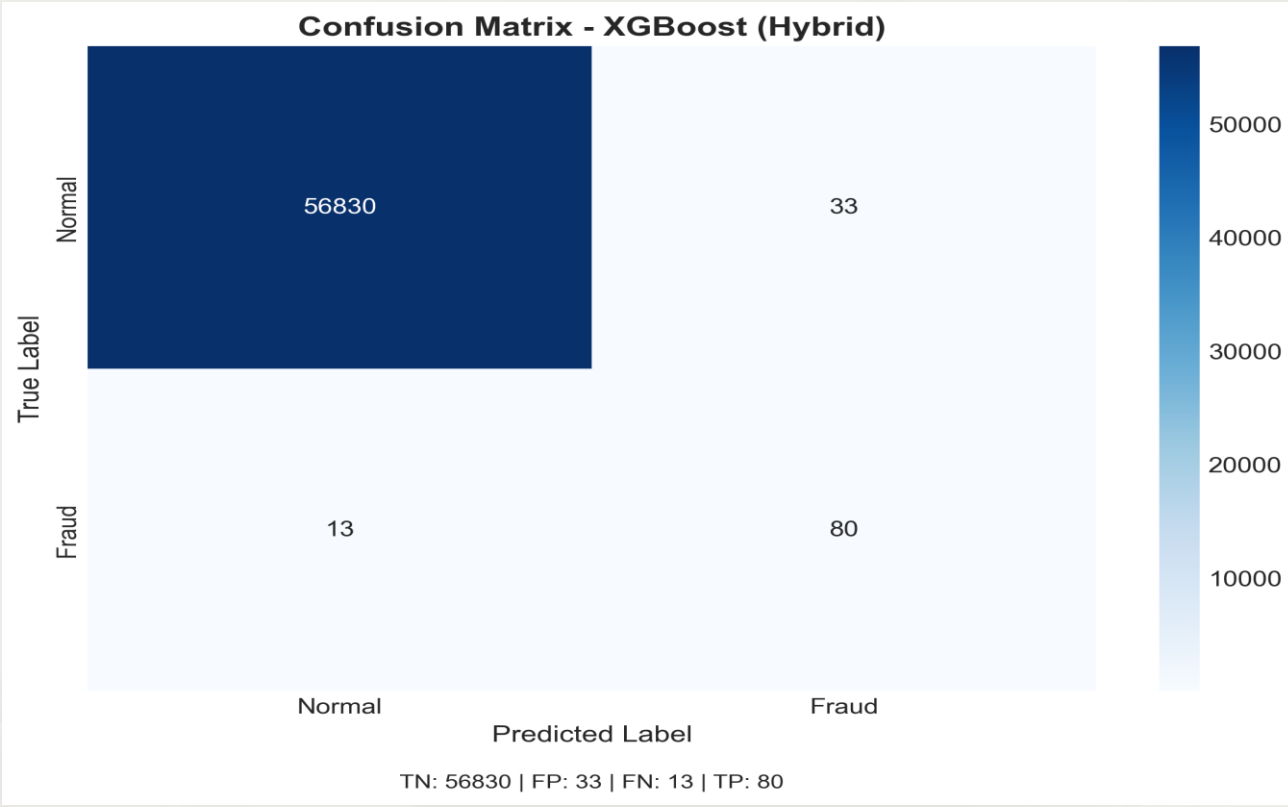
Also ,Precision jumped from 6%(lr's value) to 96%. False positives dropped from just 1360 to 3. This is now operationally viable.

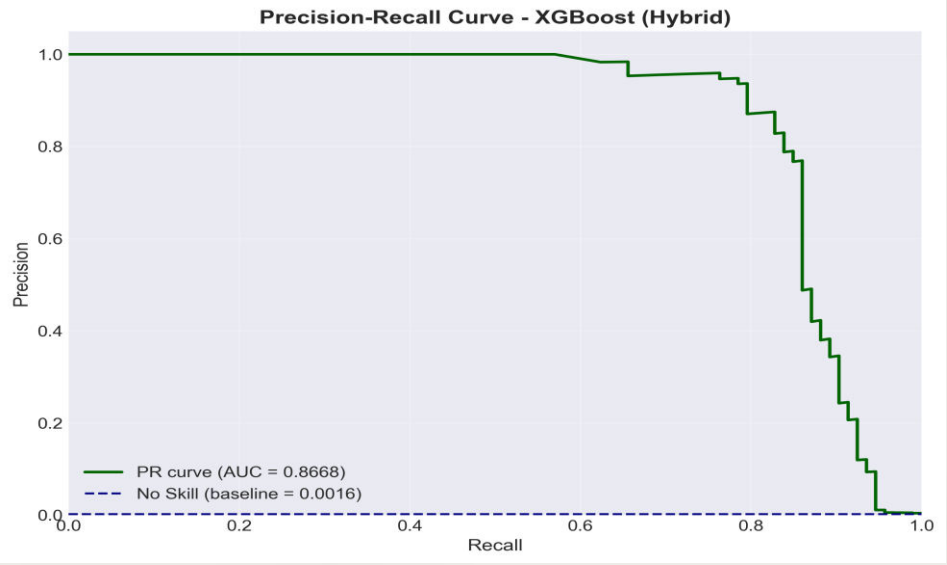
This result validated our direction — but we continued exploring whether gradient boosting methods could push performance even further.

XGBoost Results (scale_pos_weight)

In XGBoost, we used scale_pos_weight to directly address class imbalance and prioritize fraud detection.
This increased recall by penalizing missed fraud cases more strongly.
We then added the anomaly_score feature, which further improved overall detection quality by capturing abnormal transaction behavior.

PR-AUC ≈ 0.867 **Precision = 0.70** **Recall = 0.86**





Metric	XGBoost Baseline	XGBoost Hybrid (scale_pos_weight + anomaly_score)	Impact
Precision	**92.5%**	**70.8%**	↓ Trade-off due to aggressive fraud prioritization
Recall	79.6%	**86.0%**	↑ +6.4pp more frauds detected
PR-AUC	0.851	**0.867**	↑ +1.6% overall fraud detection quality

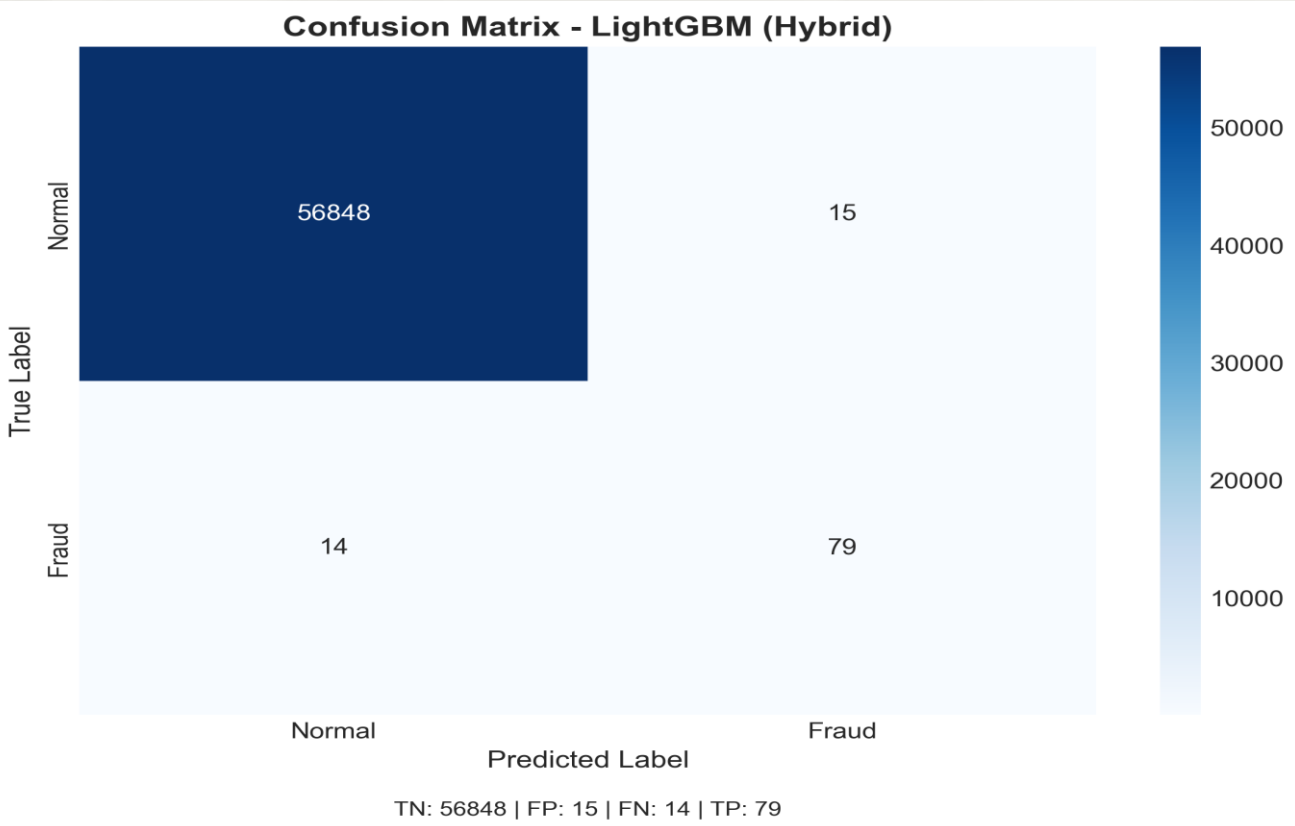
LightGBM Results : Best Baseline (class_weights + anomaly_score)

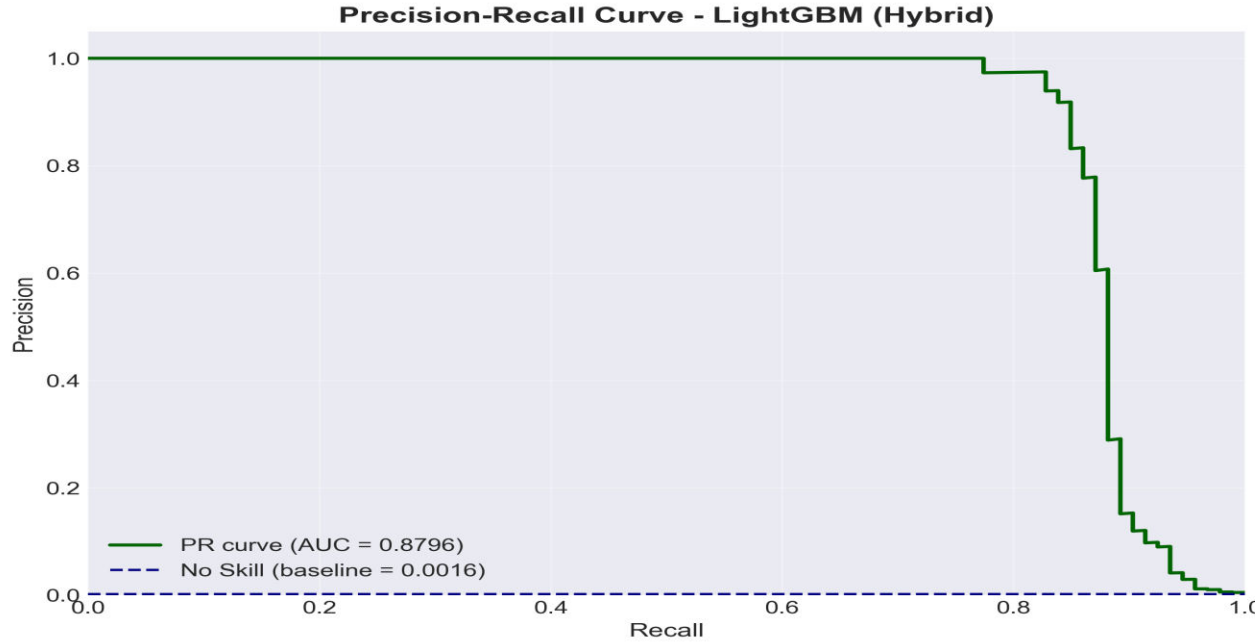
PR-AUC (baseline): 0.8796

Fastest, most stable model

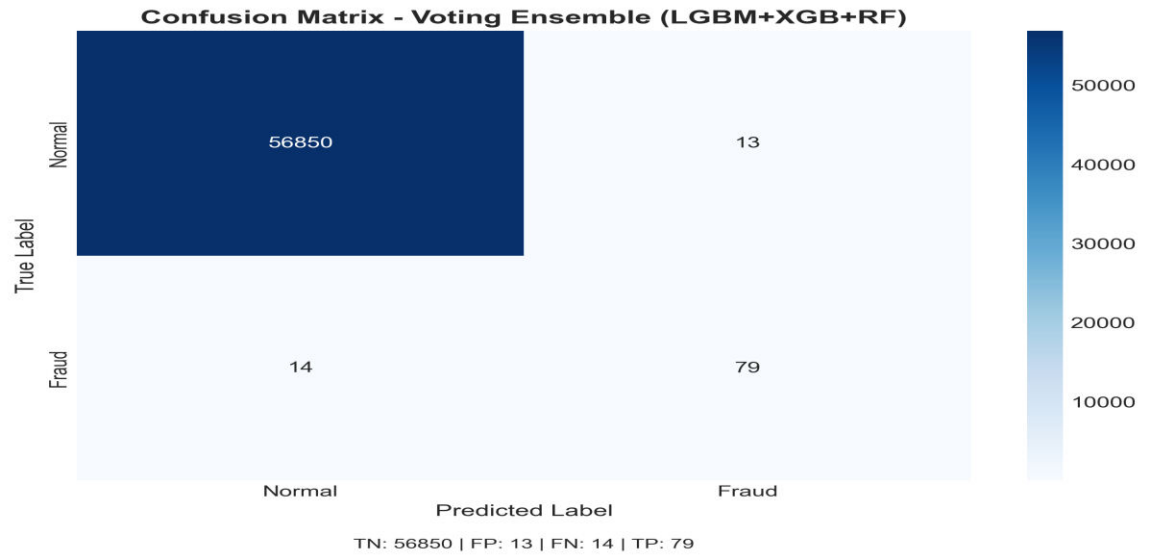
Precision (0.8404) – Recall (0.8494)

Metric	Logistic Regression (Class Weights)	Random Forest (Class Weights)	XGBoost (ScalePos Weight)	LightGBM (Class Weights) 🏆
Precision	0.0601	0.96	0.708	0.8404
Recall	0.9355	0.7742	0.8602	0.8495
F1-Score	0.1130	0.8571	0.7767	0.8449
PR-AUC	0.6896	0.8460	0.8668	0.8796





Voting_Ensemble_LGBM_XGB_RF Results (LGBM=0.5 , XGB=0.3, RF = 0.2)



Feature	Voting Ensemble (RF+XGB+LGBM)	LGBM (Base Model) 🏆
Complexity	🔴 **High** (3 Models)	🟢 **Low** (1 Model)
PR-AUC (Potential)	0.8724	**0.8796** ↗
Precision (Default)	**85.87%**	84.04%
Maintenance	Hard to Maintain	**Easy to Deploy**

Decision

We eliminated the Voting Ensemble because it added high complexity while giving only a small precision gain and a slight drop in PR-AUC compared to the best single model.

Further analysis showed that **LightGBM** achieved the highest PR-AUC and the best precision–recall balance, making it the most robust standalone model.

XGBoost showed strong recall-focused behavior, especially after adding the anomaly feature, but did not outperform LightGBM overall. Therefore, we selected **LightGBM** and kept **XGBoost** only for calibration and threshold optimization, instead of using a more complex ensemble.

Key Rationale

Diminishing Returns:

The Ensemble improved default Precision by only ~1.8% while increasing **operational complexity.**This represents a small performance gain for a large complexity cost.

PR-AUC Signal:

The single LightGBM model achieved a higher PR-AUC (0.88) than the Ensemble (0.87). This indicates that the standalone model has stronger **learning capacity**, leaving more room for improvement through **calibration and threshold optimization**.

Undersampling Methods Fail

- **This dataset is already PCA-transformed.**
- Undersampling removes many normal samples and changes the PCA feature space.
- This damages the data distribution and leads to lower PR-AUC.
- **Class weighting keeps all samples and gives much better performance.**

Undersampling Methods

BalancedBagging, EasyEnsemble

0.58 - 0.71

PR-AUC

Class Weighting

LightGBM, XGBoost

~0.88

PR-AUC

04

Model Validation (Cross-Validation)

Cross-Validation Overview & Correction

Model	Precision (mean ± std)	Recall (mean ± std)	F1 (mean ± std)	PR-AUC (mean ± std)	ROC-AUC (mean ± std)
Logistic Regression (CW)	0.0568 ± 0.0061	0.8979 ± 0.0277	0.1067 ± 0.0108	0.7170 ± 0.0209	0.9715 ± 0.0171
Random Forest (Baseline)	0.9305 ± 0.0367	0.7554 ± 0.0487	0.8319 ± 0.0215	0.8257 ± 0.0165	0.9446 ± 0.0191
XGBoost (SPW + Anomaly Score)	0.8875 ± 0.0360	0.7985 ± 0.0546	0.8383 ± 0.0157	0.8257 ± 0.0134	0.9758 ± 0.0134
LightGBM (CW + Anomaly Score)	0.8854 ± 0.0256	0.7958 ± 0.0505	0.8367 ± 0.0210	0.8341 ± 0.0107	0.9713 ± 0.0116

CV Strategy

5-fold stratified CV reporting Precision, Recall, F1, PR-AUC, ROC-AUC.

Key Correction

5-fold stratified cross-validation is applied to all four model families using the cleaned dataset with the anomaly score feature, while intentionally keeping default probability thresholds and excluding calibration or cost-sensitive tuning to ensure a fair comparison of base model performance.

⚠ The reason of getting less PR-AUC value rather than the final value is because of the amount of data (%64 train, %16 test) we used for cross-validation which is less than the original one (%80 train, %20 test).

XGBOOST
HOLD-OUT vs CV COMPARSION TABLE

XGBoost (scale_pos_weight + anomaly_score)

Metric	Hold-out	CV Mean	CV Std
Precision	0.7080	0.8875	0.0360
Recall	0.8602	0.7985	0.0546
PR-AUC	0.8668	0.8257	0.0134
ROC-AUC	0.9837	0.9758	0.0134

LIGHTGBM
HOLD-OUT vs CV COMPARISON TABLE

LightGBM (Class Weights + Anomaly Score)

Metric	Hold-out	CV Mean	CV Std
Precision	0.8404	0.8854	0.0256
Recall	0.8495	0.7958	0.0505
PR-AUC	0.8796	0.8341	0.0107
ROC-AUC	0.9889	0.9713	0.0116

05

Enhancements

Probability Calibration Matters

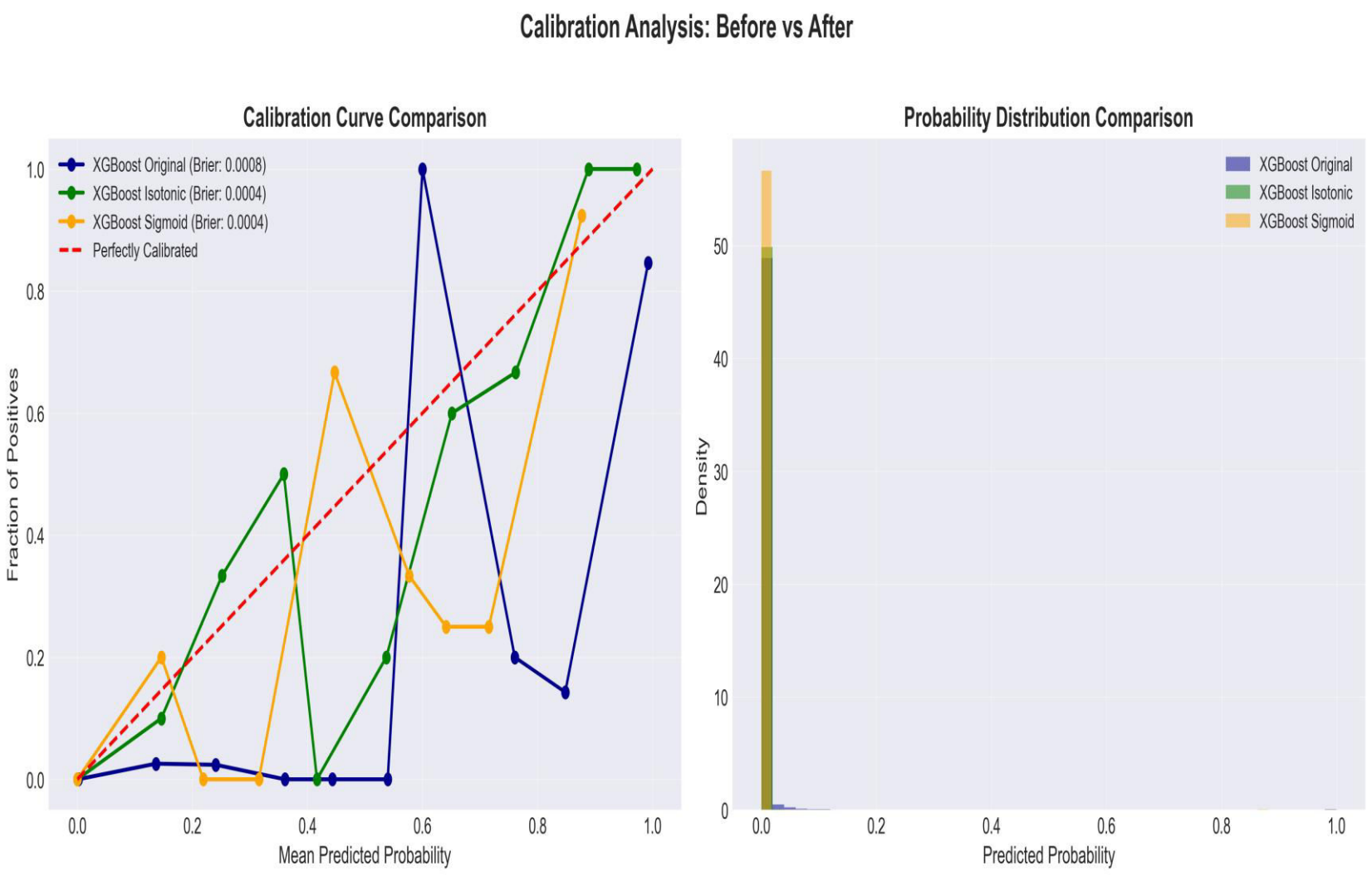
Calibration improves the reliability of probability estimates, which is essential for risk-based decision systems.

- Tree-based models often produce **poorly calibrated probabilities**
- **Isotonic Regression** improves probability reliability
  (better than Sigmoid for complex models)

Brier Score evaluates probability accuracy

- Lower score → better calibration

XGBoost Calibration Comparison



XGB_Calibrated_Isotonic

Precision: 0.9024

Recall: 0.7956

PR-AUC: 0.8787

XGB_Calibrated_Sigmoid

Precision: 0.8191

Recall: 0.8279

PR-AUC: 0.8790

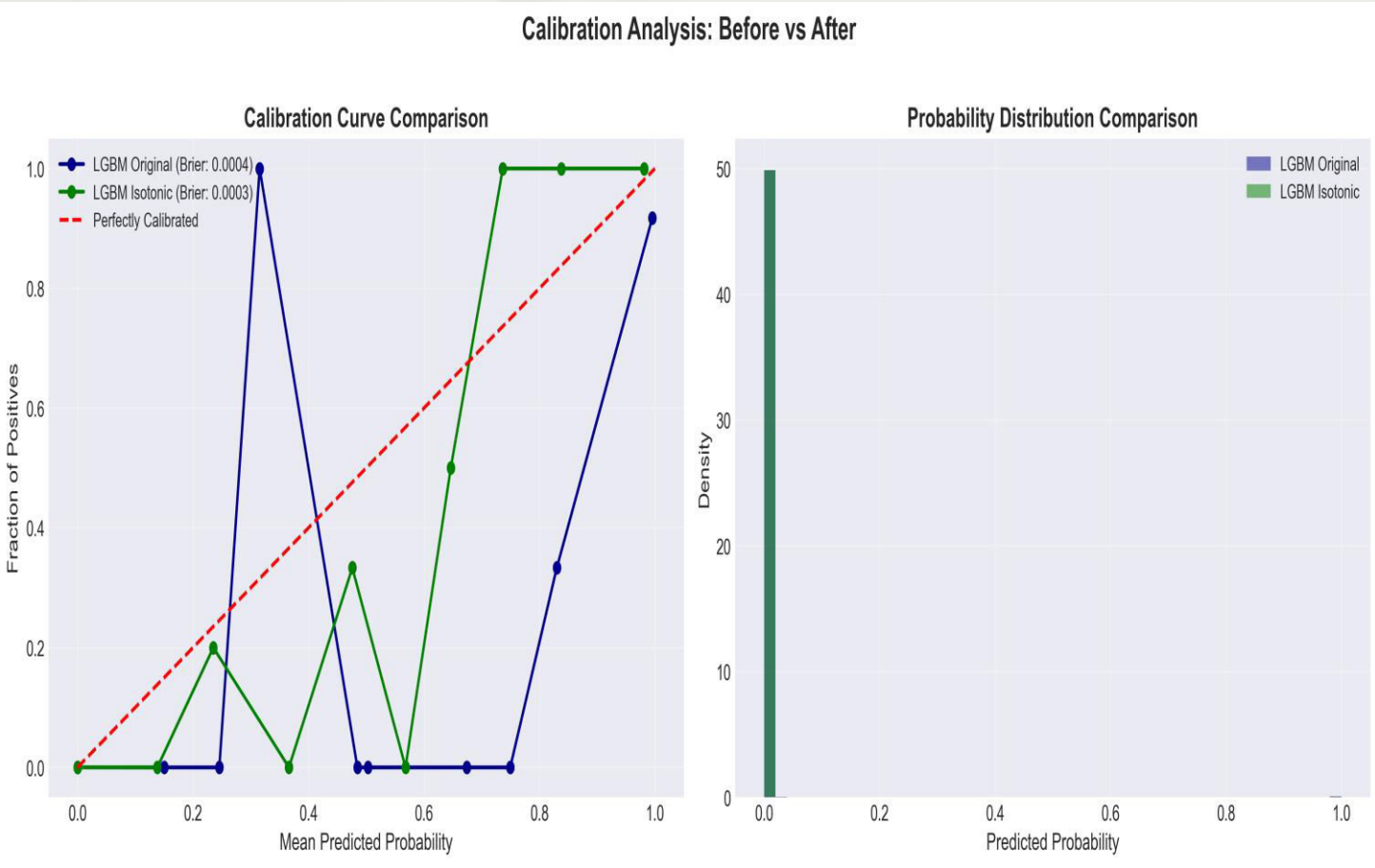
Brier Score

Original: 0.0008

Isotonic: 0.0004

Sigmoid: 0.0004

LightGBM Calibration Comparison



Metric	LGBM (Base / Class Weights)	LGBM (Calibrated – Isotonic)
Brier Score (Lower is Better)	0.0004	**0.0003** 📉
Precision	84.04%	**90.70%** 🚀
Recall	84.95%	83.87%
PR-AUC	87.96%	**88.07%** ↗

06

Threshold Optimization

Why Threshold Optimization Matters

- Fraud datasets are extremely imbalanced → default threshold = 0.50 is suboptimal.
- Banks prioritize catching more fraud (FN cost ≫ FP cost).
- Therefore threshold must be tuned using:
  - **F2-score optimization** (recall-focused) -> SELECTED ✓
  - **Youden's J** (balanced statistical metric)
  - **Cost-sensitive thresholds** (business impact)

Cost-Sensitive Threshold Optimization (LightGBM)

Scenario (FN:FP)	Optimal Threshold	Min Cost	Default Cost	Cost Reduction (%)	Precision	Recall
50 : 1	0.23	664	758	12.40%	0.8511	0.8602
100 : 1	0.23	1314	1508	12.86%	0.8511	0.8602
200 : 1	0.23	2614	3008	13.10%	0.8511	0.8602
500 : 1	0.23	6514	7508	13.24%	0.8511	0.8602

Threshold Optimization (F2 & J) for LightGBM

Threshold Method	Threshold	F2-Score	Recall	Precision	FP (Test)
Default (sklearn)	0.50	0.8515	83.9%	90.7%	8
F2-Optimized ✓	0.60	0.8609	83.9%	96.3%	3
Youden's J ✗	0.0029	0.3494	93.5%	10.0%	786
Cost-Sensitive ✗	0.23	0.8584	86.0%	85.1%	14

Business Insight (Final Takeaway)

- **Selected deployment threshold: 0.60 (F2-optimized)** — achieves the **highest F2-score (0.8609)** with **96.3% precision** while keeping recall unchanged (**83.9%**).

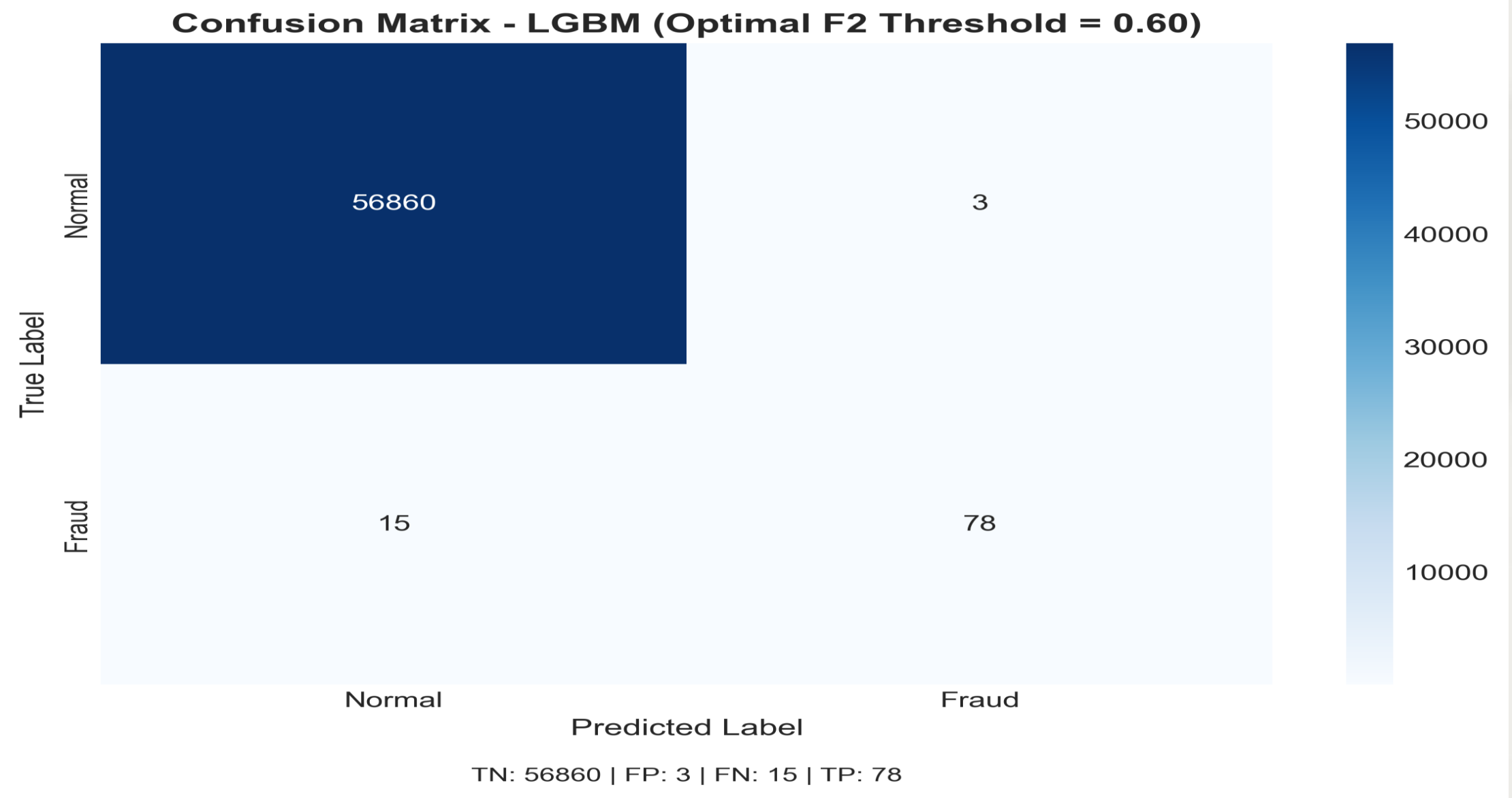
- **Why F2 instead of cost-sensitive tuning?** True **FN/FP cost ratios are uncertain and organization-dependent**; F2 provides a **data-driven, recall-prioritizing** objective without assuming a specific business cost.

- **Operational impact (hold-out test):** only **3 false positives**, resulting in a **manageable alert volume** and reduced analyst workload.

- **Alternative tested:** cost-sensitive threshold (**0.23**) can reduce estimated cost (e.g., **~12–13% under FN:FP = 50–500:1 scenarios**), but increases false positives to **14** (**~4.6× more alerts** than F2-optimized).

- **Practical alignment:** F2 is **widely used** in highly imbalanced detection tasks when **missing positives is more costly** than false alarms, balancing recall (fraud capture) with precision (trust/alert quality).

Optimized Confusion Matrix (LightGBM)



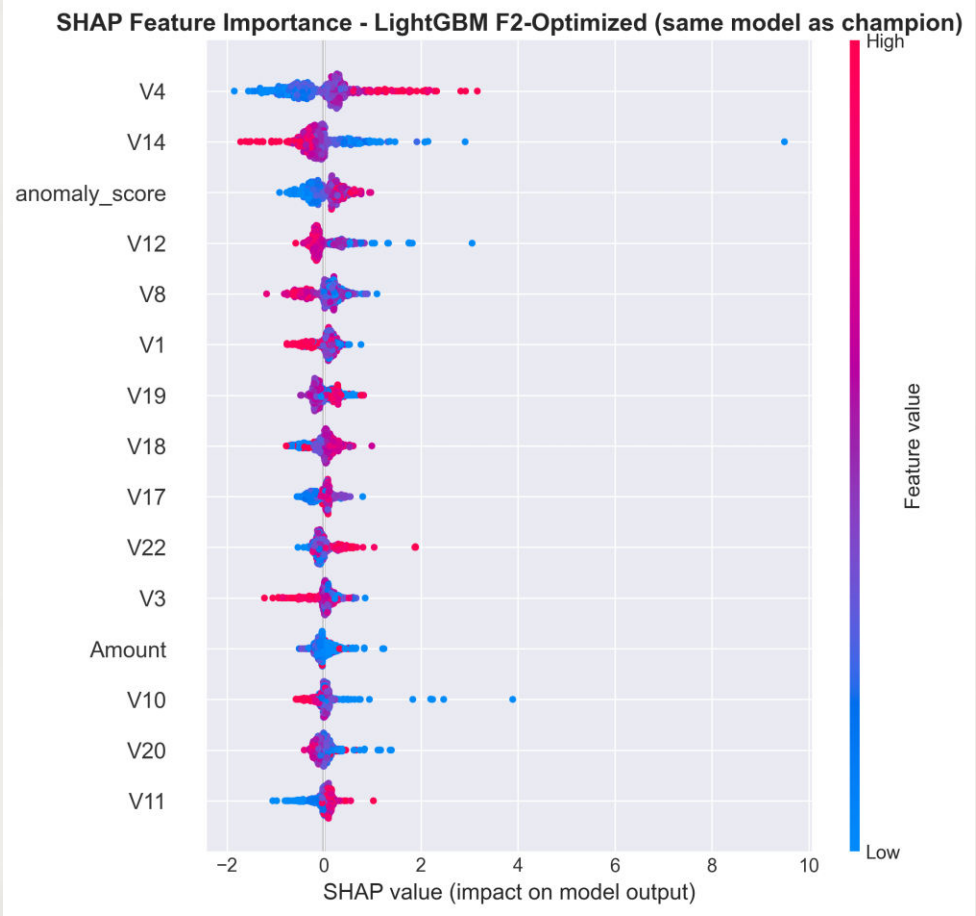
07

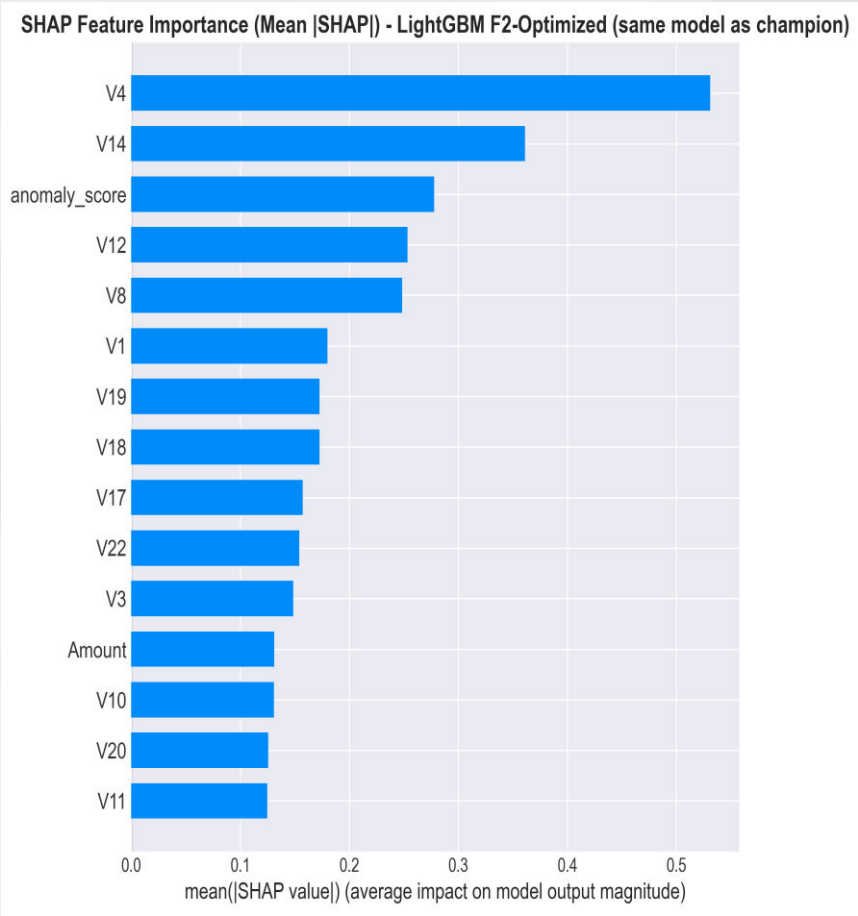
Interpretability

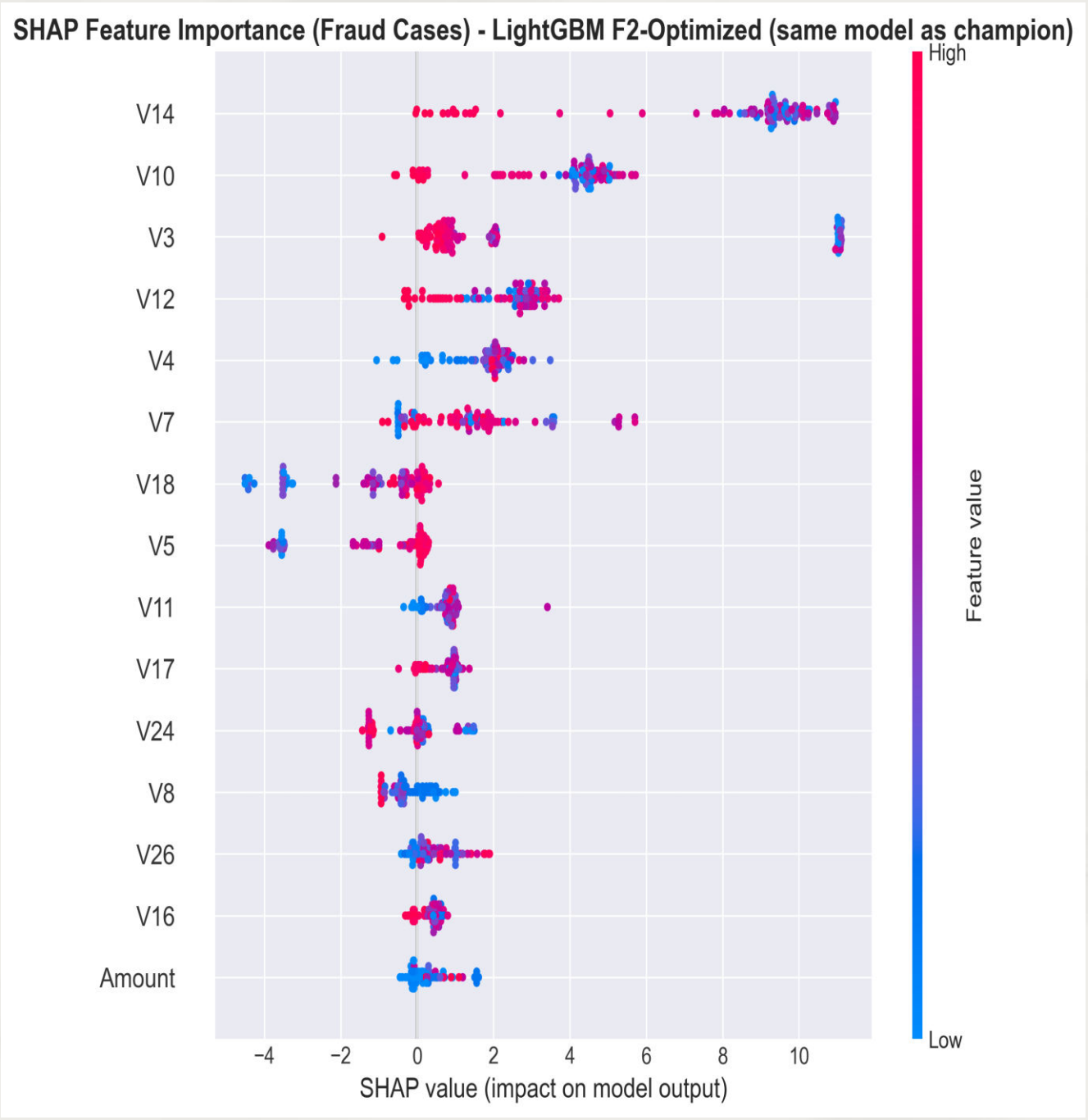
SHAP Global Interpretation

Across all transactions, SHAP reveals the top drivers of the model's predictions.

1. V4
2. V14
3. anomaly_score
4. V12
5. V8



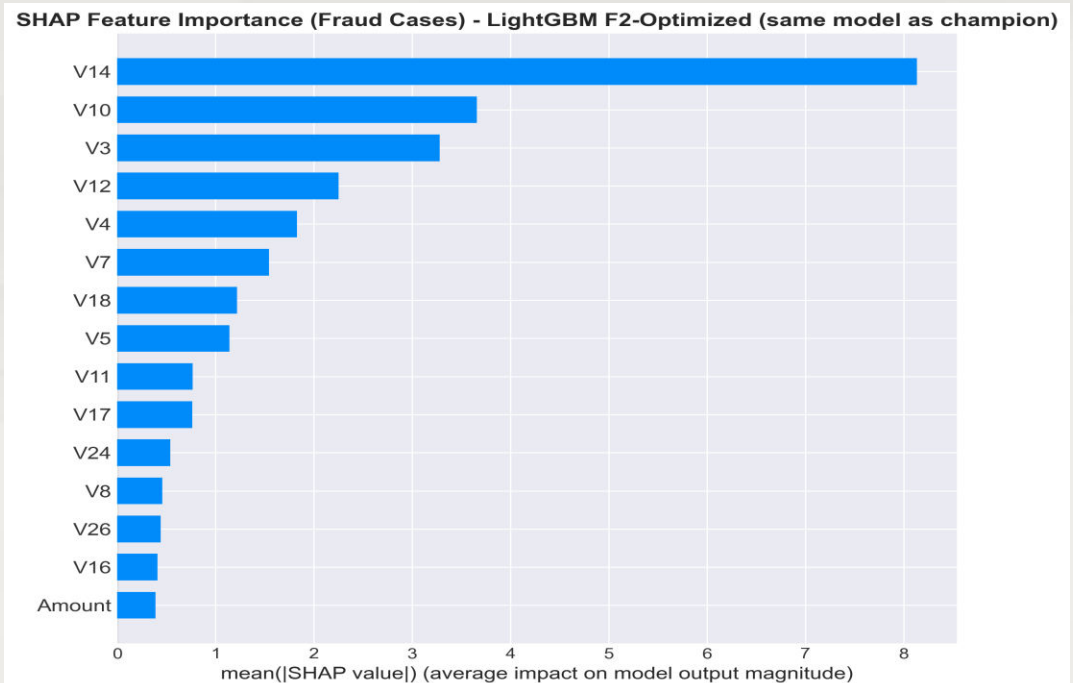




SHAP Fraud-Only Insights

When explaining only fraudulent cases, the patterns that distinguish fraud become clearer.

- V14 dominates fraud-specific detections
- Reveals distinct sub-patterns vs. global

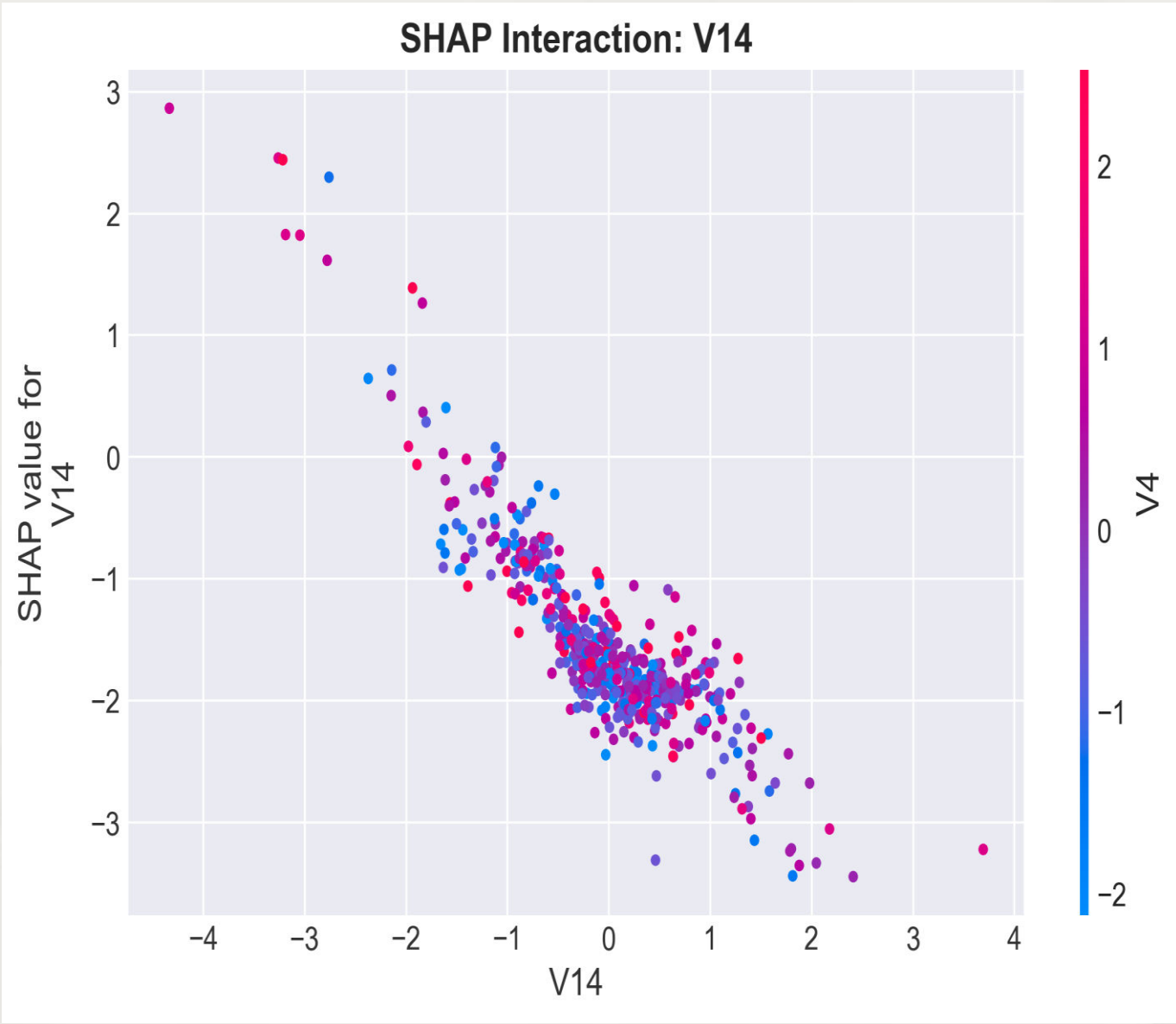


SHAP Interaction Effects

SHAP interaction values uncover how features work together to influence predictions.

- Strong V14 × V4 interaction
- Captures non-linear fraud relationships

This confirms tree ensembles outperform linear models.



08

Conclusion

Final Model Comparison

Model	PR-AUC	Recall	Precision
LGBM_Optimized_F2 🏆	88.07%	83.87%	96.30%
LGBM_Calibrated_Isotonic	88.07%	83.87%	90.69%
LGBM_ClassWeights	87.96%	84.94%	84.04%

⚠ These results show that LightGBM with F2-optimized threshold delivers the best overall trade-off between PR-AUC, recall, and precision, making it our final model.

Final Recommendation

F2-Optimized LightGBM

- PR-AUC: 0.8807
- Precision: 0.9630
- Recall: 0.8387

This model achieves the best trade-off between maximizing fraud recall and minimizing false positives, making it ideal for deployment.

Project Limitations

Limited Interpretability

This dataset is PCA-anonymized for privacy. The 28 features are principal components, not real transaction attributes.

Potential Concept Drift

The dataset comes from 2013. Fraud patterns change over time, and many modern fraud types did not exist then.This limitation comes from the dataset, not from our modeling approach.

No Temporal Sequence Modeling

We used the time feature as a static input and treated each transaction independently. Because of this, we cannot detect behaviors like rapid repeated transactions or unrealistic location changes

SHAP Computational Cost

SHAP is expensive and may need approximation for real-time use.

Conclusion & Deliverables

We successfully built a complete, robust, and interpretable fraud detection system.

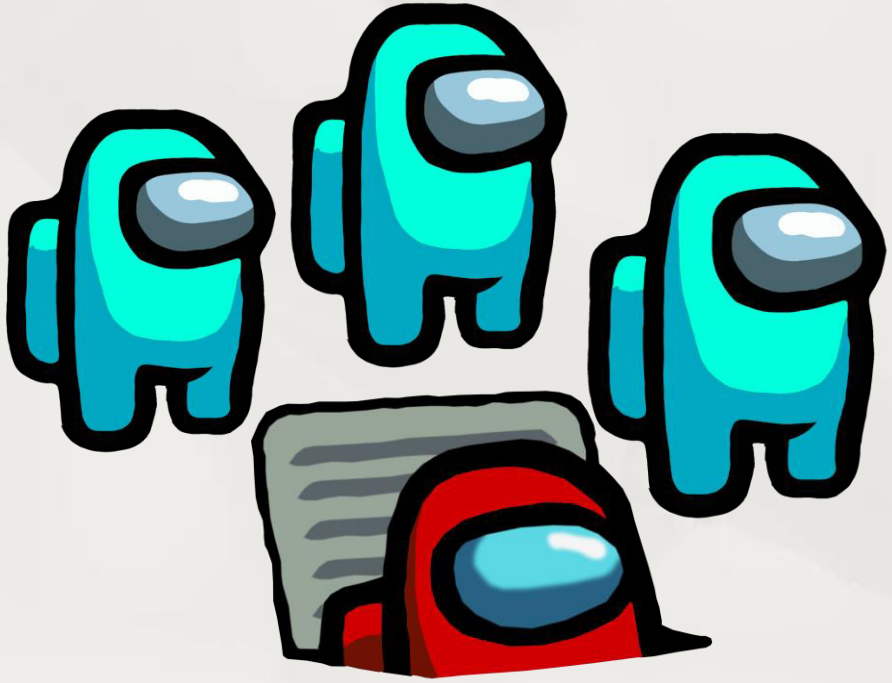
Evaluated 17+ models in 14 final comparisons.

Added anomaly detection, calibration, tuning, and SHAP.

Final model is robust, interpretable, and deployable.

Model	Precision	Recall	F1-Score	F2-Score	MCC	Cohen's Kappa	Balanced Accuracy	ROC-AUC	PR-AUC
LR_Baseline	0.8169	0.6237	0.7073	0.6546	0.7134	0.7069	0.8117	0.9848	0.7213
LR_ClassWeights	0.0601	0.9355	0.1130	0.2391	0.2339	0.1103	0.9558	0.9907	0.6896
RF_Baseline	0.9367	0.7957	0.8605	0.8204	0.8631	0.8603	0.8978	0.9339	0.8488
RF_ClassWeights	0.9600	0.7742	0.8571	0.8054	0.8619	0.8569	0.8871	0.9393	0.8460
XGB_Baseline	0.9250	0.7957	0.8555	0.8186	0.8577	0.8553	0.8978	0.9801	0.8506
XGB_ScalePosWeight	0.7080	0.8602	0.7767	0.8247	0.7800	0.7763	0.9298	0.9837	0.8668
LGBM_ClassWeights	0.8404	0.8495	0.8449	0.8476	0.8447	0.8447	0.9246	0.9889	0.8796
BalancedBagging_Hybrid	0.0820	0.8925	0.1502	0.2999	0.2678	0.1477	0.9381	0.9759	0.5846
EasyEnsemble	0.0389	0.9462	0.0747	0.1669	0.1876	0.0718	0.9540	0.9921	0.7115
Voting_LGBM_XGB_RF	0.8587	0.8495	0.8541	0.8513	0.8538	0.8538	0.9246	0.9896	0.8724
XGB_Calibrated_Isotonic	0.9024	0.7957	0.8457	0.8150	0.8472	0.8455	0.8978	0.9855	0.8788
XGB_Calibrated_Sigmoid	0.8191	0.8280	0.8235	0.8262	0.8233	0.8232	0.9138	0.9817	0.8791
RF_Calibrated_Isotonic	0.9359	0.7849	0.8538	0.8111	0.8569	0.8536	0.8924	0.9588	0.8495
LGBM_Calibrated_Isotonic	0.9070	0.8387	0.8715	0.8515	0.8720	0.8713	0.9193	0.9888	0.8807
XGB_Optimized	0.7692	0.8602	0.8122	0.8403	0.8131	0.8119	0.9299	0.9837	0.8668
RF_Optimized	0.7788	0.8710	0.8223	0.8508	0.8233	0.8220	0.9353	0.9339	0.8488
LGBM_Optimized_F2	0.9630	0.8387	0.8966	0.8609	0.8985	0.8964	0.9193	0.9888	0.8807

Simulated Transaction Scenarios



Predict Fraud Risk

To better understand the model's decision-making process, we simulate several transaction scenarios and analyze the predicted fraud probabilities.


THANK YOU