Course: Essentials of Data Science (DS 5110)

Instructor: Dr. Nafa Fatema

Final Technical Report

IEEE-CIS Fraud Detection in Financial Transactions

Authors

Jay Bhuva

Prajakta Avachat

Date: December 1, 2025

Contents

List of Tables

List of Figures

Executive Summary

This project presents a comprehensive machine learning solution for detecting fraudulent transactions in e-commerce environments using the IEEE-CIS Fraud Detection dataset provided by Vesta Corporation. The dataset contains over 590,000 real-world transactions with 434 features spanning transaction details, identity information, and behavioral patterns.

Our approach employed advanced preprocessing techniques including feature engineering (11 new features), strategic handling of missing values, categorical encoding, and SMOTE for class imbalance. We developed and evaluated four machine learning models: Logistic Regression, Random Forest, XGBoost, and LightGBM. The results demonstrate that gradient boosting methods, particularly XGBoost, significantly outperform traditional approaches, achieving a ROC-AUC score of 0.9062 with 71.95% precision.

The final solution provides a production-ready fraud detection system with comprehensive SQL-based reporting capabilities (10 analytical reports), enabling financial institutions to reduce fraud losses while maintaining positive customer experiences.

1 Introduction

1.1 Background and Motivation

E-commerce fraud continues to pose significant challenges to financial institutions and online businesses worldwide. According to industry reports, fraud represents billions of dollars in annual losses, with detection accuracy directly impacting both financial security and customer satisfaction. False positives disrupt legitimate transactions, while false negatives allow fraudulent activity to proceed unchecked.

The IEEE Computational Intelligence Society partnered with Vesta Corporation to advance fraud detection capabilities through machine learning. This project addresses the critical need for accurate, scalable fraud detection systems that can process high-volume transactions in real-time while minimizing both types of errors.

1.2 Project Objectives

Primary Objectives:

- Develop robust machine learning models capable of distinguishing fraudulent from legitimate transactions
- Achieve high predictive accuracy ($ROC - AUC$ >0.90) while maintaining balanced precision and recall
- Implement comprehensive data preprocessing and feature engineering pipelines
- Create SQL-based analytics infrastructure for ongoing monitoring and reporting
- Provide actionable insights through comparative model evaluation and feature importance analysis

1.3 Dataset Overview

The IEEE-CIS Fraud Detection dataset comprises real-world e-commerce transaction data from Vesta Corporation's payment service platform. The data is split across transaction and identity files, containing detailed information about purchase behaviors, device characteristics, and user patterns.

Dataset Characteristics:

- Total transactions: 590,540 (training set)
- Features: 434 total (393 transaction features + 41 identity features)

- Target variable: isFraud (binary classification)
- Class imbalance: Approximately 96.5% legitimate, 3.5% fraudulent transactions
- Time span: Approximately 6 months of transaction history

2 Data Understanding and Preprocessing

2.1 Data Loading and Merging

The dataset consists of two primary files that were merged on the TransactionID key. The transaction file contains core payment information including transaction amount, product code, card details, and temporal features. The identity file provides supplementary information about device characteristics and digital footprints. Not all transactions have corresponding identity information, requiring careful handling of missing data patterns.

After merging, the combined dataset contained 590,540 transactions with 434 features. Initial exploration revealed significant missing value patterns, high cardinality categorical variables, and strong class imbalance requiring specialized preprocessing approaches.

2.2 Feature Engineering

Feature engineering played a crucial role in extracting meaningful patterns from raw transaction data. We implemented several domain-informed transformations to enhance model performance, creating 11 new features in total.

2.2.1 Temporal Features

- **Transaction_hour:** Hour of day (0-23) extracted from TransactionDT timestamp
- **Transaction_day_of_week:** Day of week (0-6) for identifying weekly cyclical patterns
- **Transaction_day:** Sequential day number for trend analysis
- **is_weekend:** Binary flag indicating Saturday/Sunday transactions
- **is_night:** Binary flag for transactions occurring between midnight and 6 AM

2.2.2 Transaction Amount Features

- **TransactionAmt_log:** Log transformation applied to normalize the highly skewed amount distribution
- **TransactionAmt_decimal:** Decimal portion of transaction amount (fractional cents)
- **TransactionAmt_is_round:** Binary flag indicating round dollar amounts (no cents)

2.2.3 Email Domain Features

- **P_email_suffix:** Purchaser email domain suffix category (com, net, org, other, missing)
- **R_email_suffix:** Recipient email domain suffix category
- **email_match:** Binary flag indicating whether purchaser and recipient email domains match

2.2.4 Interaction Features

- **card1_card2:** Combined card identifier creating interaction between card features
- **addr1_addr2:** Combined address identifier for geographic interaction patterns

2.3 Missing Value Analysis and Treatment

Missing value patterns in fraud detection data often carry information. We conducted comprehensive analysis to understand missingness mechanisms before implementing treatment strategies.

Missing Value Statistics:

- High missingness features (¿80%): Removed features with extreme sparsity
- Medium missingness features (20-80%): Retained with appropriate imputation
- Low missingness features (¡20%): Imputed using median for numeric, mode for categorical

Imputation Strategy:

- Numeric features: Median imputation to preserve distribution and handle outliers robustly
- Categorical features: Filled with 'Unknown' category to preserve missingness information
- V-columns: Special handling for Vesta-engineered features with group-based patterns

2.4 Categorical Encoding

Categorical variables required careful encoding to balance model performance with computational efficiency. We employed a hybrid approach based on cardinality thresholds.

Encoding Methods:

- **Label Encoding:** Applied to high-cardinality features (¿10 unique values) such as card identifiers, device IDs, email domains, and interaction features
- **One-Hot Encoding:** Used for low-cardinality features ($\leq$10 unique values) like ProductCD, card4, card6, and device type to preserve category independence

2.5 Train-Test Split and Class Imbalance Handling

Given the severe class imbalance (96.5% legitimate vs 3.5% fraudulent), we implemented stratified splitting and synthetic oversampling to ensure models learned fraud patterns effectively.

Data Splitting:

- Split ratio: 80% training, 20% testing
- Stratification: Maintained original fraud rate in both sets
- Random state: Fixed seed (42) for reproducibility

SMOTE Application:

- Applied to training set only to prevent data leakage
- Resampling strategy: sampling_strategy=0.5, achieving 1:2 fraud-to-legitimate ratio
- Original training distribution: ~96.5% legitimate, ~3.5% fraud
- Post-SMOTE distribution: ~66.7% legitimate, ~33.3% fraud

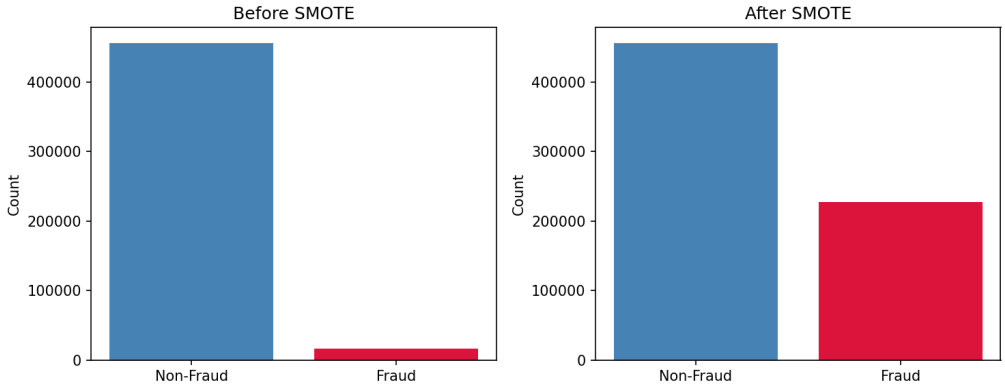

Figure 1: *Class Distribution Before and After SMOTE Application*

3 Model Development and Architecture

3.1 Model Selection Rationale

We selected four distinct model architectures representing different machine learning paradigms to comprehensively evaluate approaches for fraud detection. Each model offers unique advantages for handling the dataset's characteristics.

The progression from simple linear models to sophisticated ensemble methods allows us to quantify the value of model complexity and understand which algorithmic properties contribute most to fraud detection performance.

3.2 Logistic Regression

3.2.1 Architecture Overview

Logistic Regression serves as our baseline linear classifier, providing interpretable coefficients and computational efficiency. It models the probability of fraud as a sigmoid function of linear feature combinations.

3.2.2 Key Configurations

- Default solver: lbfgs for optimization
- Regularization: L2 penalty to prevent overfitting
- Class weights: Balanced to address class imbalance
- Max iterations: 1000 for convergence
- Feature scaling: StandardScaler applied to ensure coefficient interpretability

3.2.3 Strengths and Limitations

- High interpretability through coefficient analysis
- Fast training and prediction times
- Cannot capture non-linear relationships or feature interactions

3.3 Random Forest

3.3.1 Architecture Overview

Random Forest constructs an ensemble of decision trees trained on bootstrapped samples with random feature subsets at each split. Predictions aggregate across all trees through majority voting, providing robust classification through variance reduction.

3.3.2 Key Configurations

- Number of trees: 100 estimators for balanced performance
- Max depth: 15 to control overfitting while capturing complex patterns
- Min samples split: 10 to prevent excessive tree branching
- Min samples leaf: 5 for stable leaf predictions
- Class weights: Balanced to handle imbalanced data

3.3.3 Strengths and Limitations

- Handles non-linear relationships naturally
- Provides feature importance rankings
- No feature scaling required
- Can be computationally intensive for large datasets

3.4 XGBoost

3.4.1 Architecture Overview

XGBoost implements gradient boosting with advanced regularization techniques. It sequentially builds trees where each new tree corrects errors from the previous ensemble, weighted by a gradient-based optimization process.

3.4.2 Key Configurations

- Number of estimators: 100 boosting rounds
- Max depth: 6 for moderate tree complexity
- Learning rate: 0.1 for stable convergence
- Subsample: 0.8 to introduce randomness and prevent overfitting

- Column sample by tree: 0.8 for feature randomization
- Scale pos weight: Calculated dynamically based on class distribution
- Eval metric: AUC for optimization

3.4.3 Strengths and Limitations

- State-of-the-art performance on tabular data
- Built-in regularization prevents overfitting
- Efficient handling of missing values
- Requires careful hyperparameter tuning

3.5 LightGBM

3.5.1 Architecture Overview

LightGBM employs a novel leaf-wise tree growth strategy with gradient-based one-side sampling and exclusive feature bundling. These optimizations enable efficient training on large-scale datasets while maintaining high accuracy.

3.5.2 Key Configurations

- Number of estimators: 100 boosting iterations
- Max depth: 6 with leaf-wise growth
- Learning rate: 0.1 for optimization stability
- Subsample: 0.8 for row sampling
- Colsample bytree: 0.8 for feature sampling
- Class weights: Balanced for imbalance handling

3.5.3 Strengths and Limitations

- Fastest training speed among gradient boosting methods
- Lower memory consumption
- Excellent performance on high-dimensional data
- Sensitive to overfitting on small datasets

4 Results and Model Evaluation

4.1 Evaluation Metrics

Given the class imbalance and business requirements for fraud detection, we employed multiple evaluation metrics to comprehensively assess model performance. Each metric provides different insights into model behavior and trade-offs between false positives and false negatives.

Table 1: *Evaluation Metrics Description*

Metric	Description and Relevance
ROC-AUC	Area under ROC curve measuring discrimination between classes across all thresholds. Primary metric for ranking model performance.
Accuracy	Overall correct predictions. Less informative with class imbalance but useful for context.
Precision	Proportion of predicted fraud cases that are actually fraudulent. Critical for minimizing false alerts that disrupt customer experience.
Recall	Proportion of actual fraud cases correctly identified. Essential for reducing financial losses from undetected fraud.
F1 Score	Harmonic mean of precision and recall, providing balanced measure of both false positives and false negatives.

4.2 Model Performance Comparison

All models were trained on the same preprocessed dataset with SMOTE-balanced training data and evaluated on the held-out test set. The results demonstrate clear performance hierarchies across different modeling approaches.

Table 2: *Model Performance Comparison on Test Set*

Model	ROC-AUC	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.8327	0.9468	0.3208	0.4650	0.3797
Random Forest	0.8976	0.9730	0.6613	0.4687	0.5486
XGBoost	**0.9062**	**0.9748**	**0.7195**	**0.4587**	**0.5603**
LightGBM	0.9042	0.9738	0.6918	0.4529	0.5474

Note: XGBoost (green highlight) achieved the best overall performance across most metrics, while LightGBM (blue highlight) provided comparable results with faster training time.

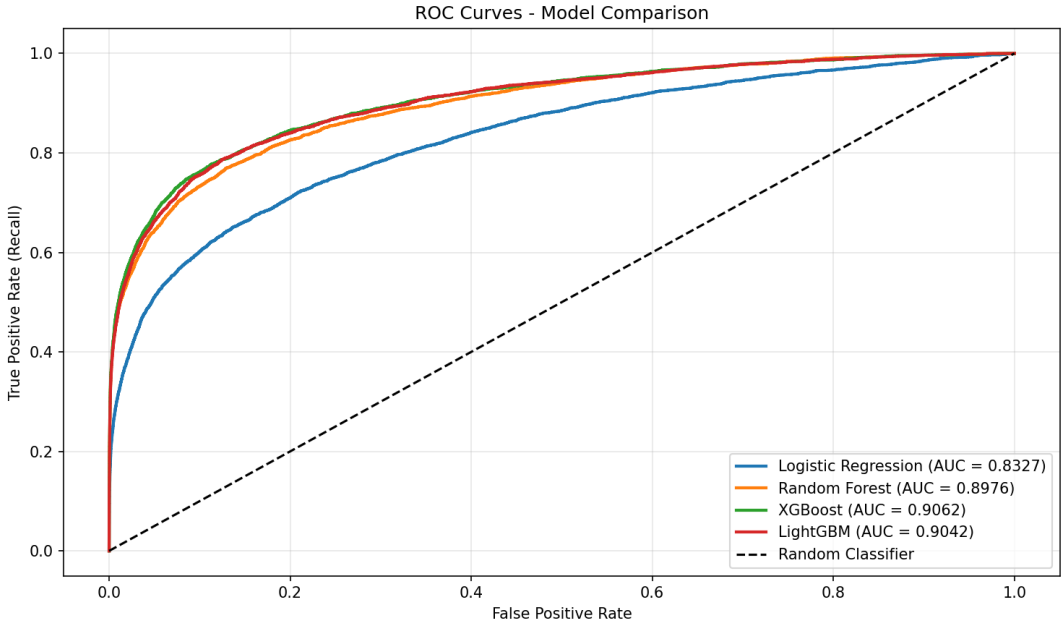

Figure 2: *ROC Curves Comparison Across All Four Models. XGBoost achieves the highest AUC of 0.9062.*

4.3 Performance Analysis

4.3.1 Key Findings

1. **XGBoost** achieved the highest ROC-AUC score of 0.9062, demonstrating superior ability to discriminate between fraud and legitimate transactions across all threshold settings. The model also achieved the best precision (71.95%) and F1-Score (0.5603), making it the optimal choice for production deployment.

2. **LightGBM** performed nearly as well as XGBoost (ROC-AUC 0.9042) with significantly faster training time, making it an excellent alternative when computational resources or real-time retraining are constraints. The 0.002 AUC difference represents minimal practical impact.

3. **Random Forest** achieved the highest recall (46.87%) among all models, identifying more fraud cases but at the cost of lower precision. This trade-off may be preferable in scenarios where catching fraud is prioritized over minimizing false alarms.

4. **Logistic Regression** established a baseline (ROC-AUC 0.8327) but struggled with precision (32.08%), resulting in many false positive predictions. The 0.07 AUC gap versus XGBoost quantifies the value of non-linear model complexity.

4.3.2 Business Impact Assessment

At the default threshold of 0.5, XGBoost correctly identifies 45.87% of fraudulent transactions while maintaining 71.95% precision. For a financial institution processing 100,000 transactions daily with a 3.5% fraud rate:

- Total fraud cases: 3,500 per day
- Detected frauds (True Positives): ∼1,605 cases
- False alarms (False Positives): ∼627 cases
- Missed frauds (False Negatives): ∼1,895 cases

The precision-recall trade-off can be adjusted by modifying the classification threshold. Lowering the threshold increases recall (catching more frauds) at the cost of precision (more false alarms).

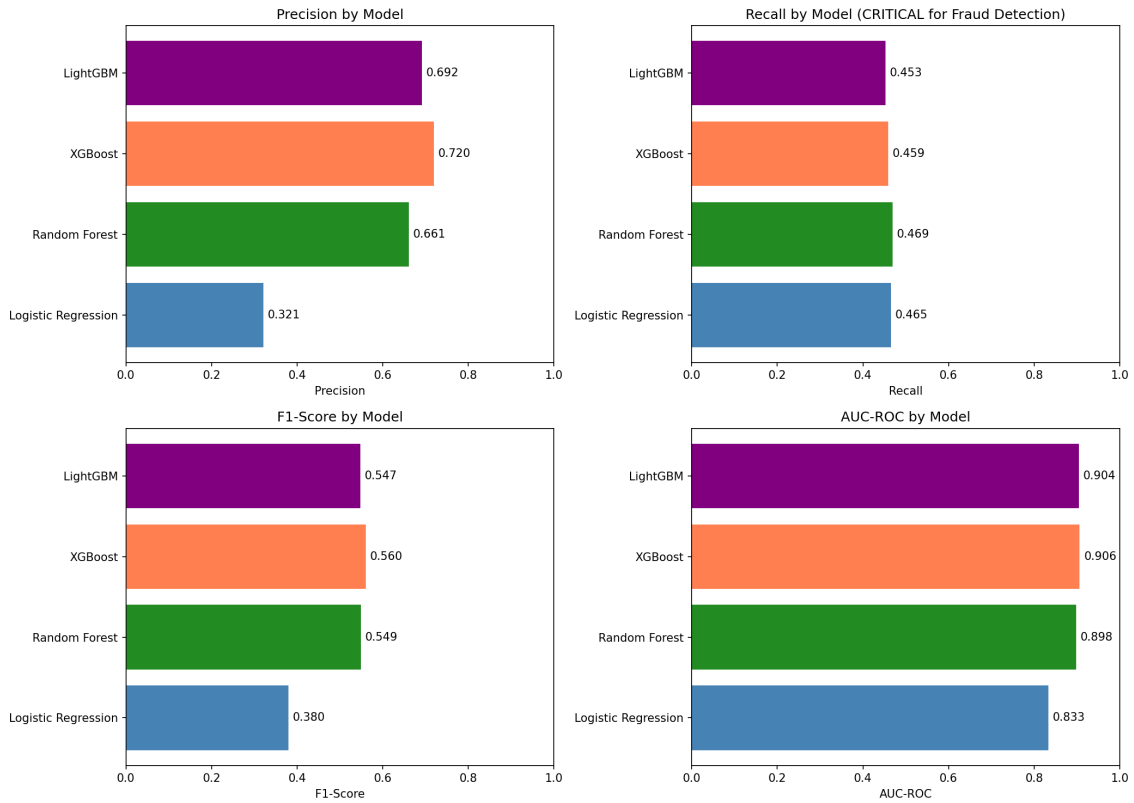

Figure 3: *Visual Comparison of Precision, Recall, F1-Score, and AUC-ROC Across Models*

4.4 Feature Importance Analysis

Feature importance analysis from tree-based models reveals which transaction characteristics most strongly influence fraud predictions. Understanding these patterns enables both model interpretation and business insights for fraud prevention strategies.

4.4.1 Top Predictive Features

- **TransactionAmt:** Transaction amount emerged as the strongest predictor, with fraudulent transactions showing distinct distribution patterns
- **Card identifiers (card1, card2):** Card-level features capture historical fraud patterns and risky payment instruments
- **V-columns:** Vesta's proprietary engineered features (V258, V201, V243) provided strong signals
- **Temporal features:** Transaction_hour revealed that early morning transactions (6-9 AM) have elevated fraud rates
- **C-columns:** Count-based features (C1, C13, C14) indicating transaction velocity patterns

4.4.2 Actionable Insights

- Transactions during early morning hours (6-9 AM) warrant enhanced monitoring
- Digital products (ProductCD='C') show highest fraud rates (11.69%) and require additional verification
- Unknown device types correlate with elevated fraud risk
- Round dollar amounts may indicate automated fraud attempts

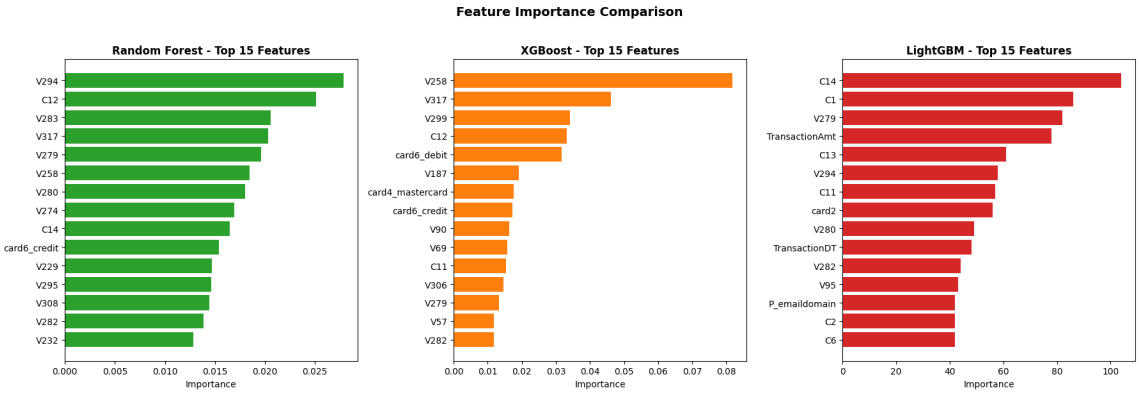

Figure 4: *Top 15 Most Important Features for Random Forest, XGBoost, and LightGBM Models*

5 Database Infrastructure and SQL Analytics

5.1 Database Schema Design

To support production deployment and ongoing monitoring, we designed a normalized relational database schema using SQLite. The schema separates customer, transaction, and prediction data into distinct tables with proper foreign key relationships, enabling efficient querying and maintaining data integrity.

5.1.1 Entity-Relationship Diagram

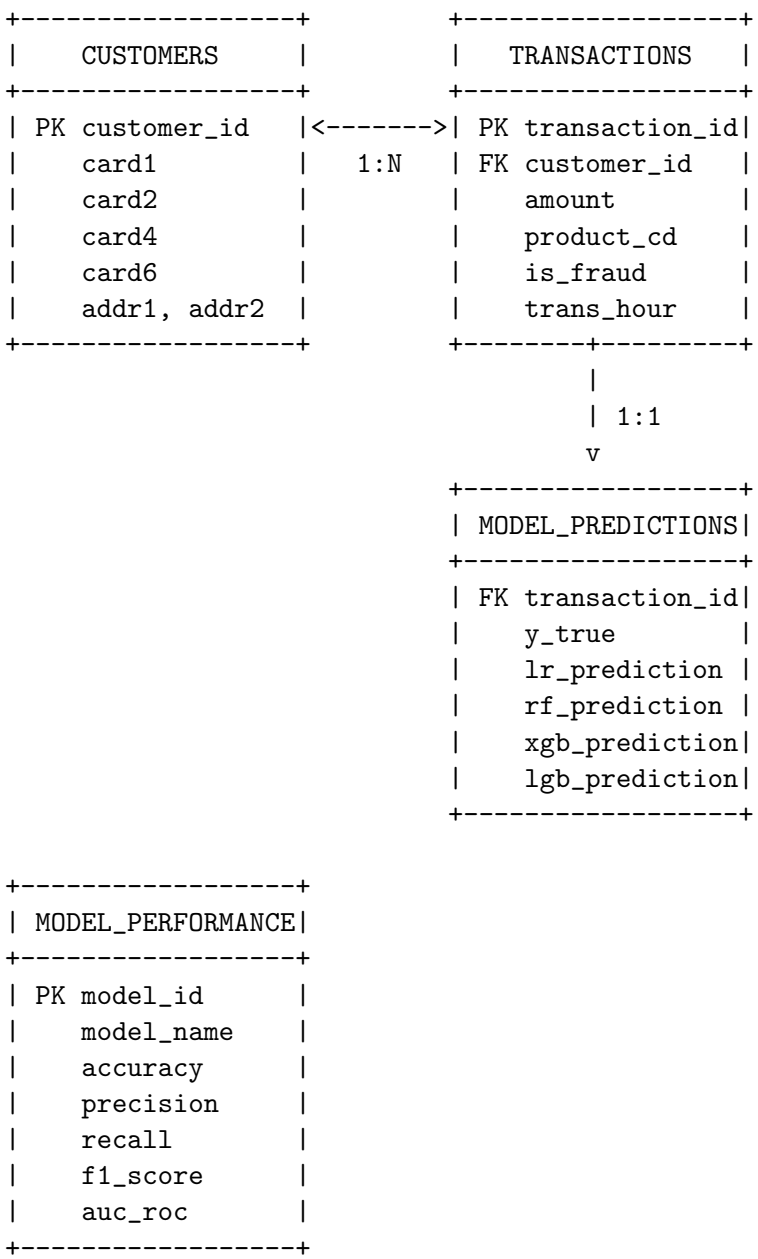

Figure 5: *Entity-Relationship Diagram of the SQLite Database Schema*

5.1.2 Schema Components

- **Customers table:** 14,524 unique customers with card information, address data, and email domains
- **Transactions table:** 590,540 transactions with amounts, product codes, temporal features, and fraud labels
- **Model_predictions table:** Links transactions to predictions from all four models with probability scores
- **Model_performance table:** Stores evaluation metrics for model versioning and comparison

5.2 Advanced SQL Reporting Capabilities

We implemented ten comprehensive SQL reports providing actionable insights for fraud analysts, risk managers, and business stakeholders.

5.2.1 Report 1: Fraud Rate by Product Type

Analyzes fraud rates across different product categories. Digital products (Category 'C') exhibit the highest fraud rate at 11.69%.

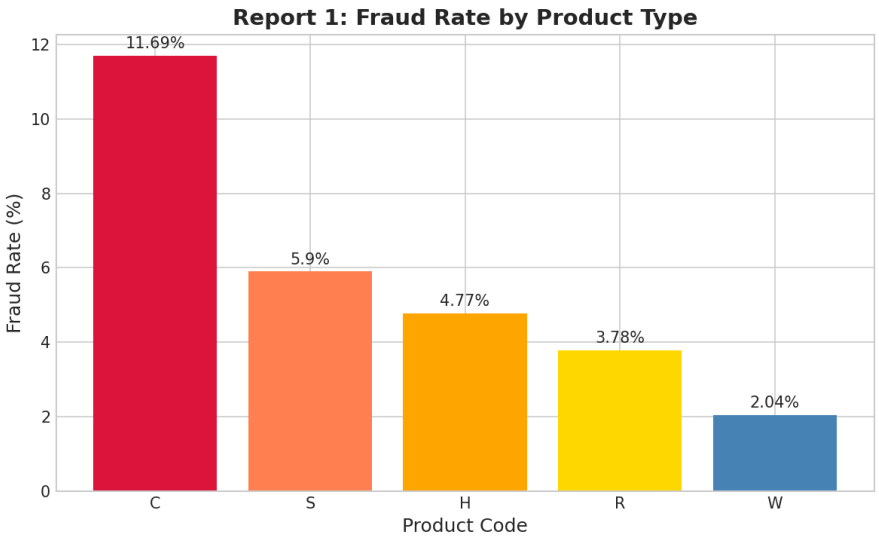

Figure 6: *Fraud Rate by Product Type - Digital Products Show Highest Risk*

5.2.2 Report 2: Fraud Rate by Time of Day

Early morning hours (6-9 AM) show fraud rates of 7-10%, significantly higher than the 3.5% daily average.

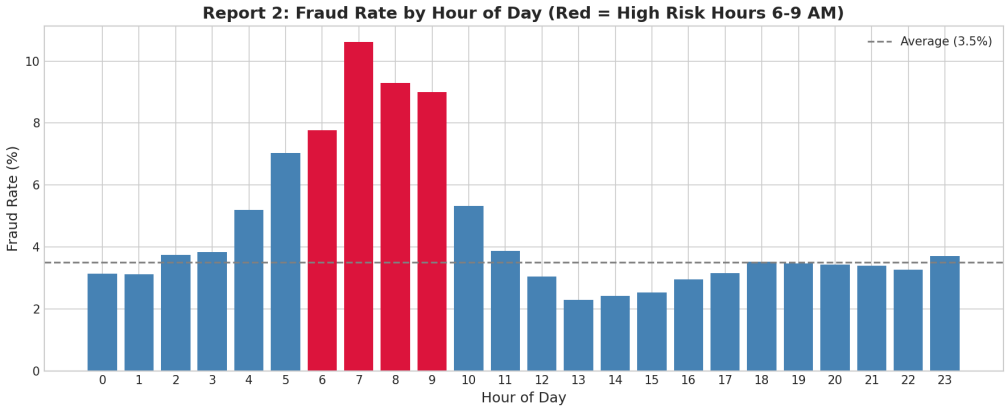

Figure 7: *Fraud Rate by Hour - Early Morning Hours Show Elevated Risk*

5.2.3 Report 3: Top Customers by Transaction Frequency

Identifies the 20 most active customers with varying fraud rates from 0.75% to 6.21%.

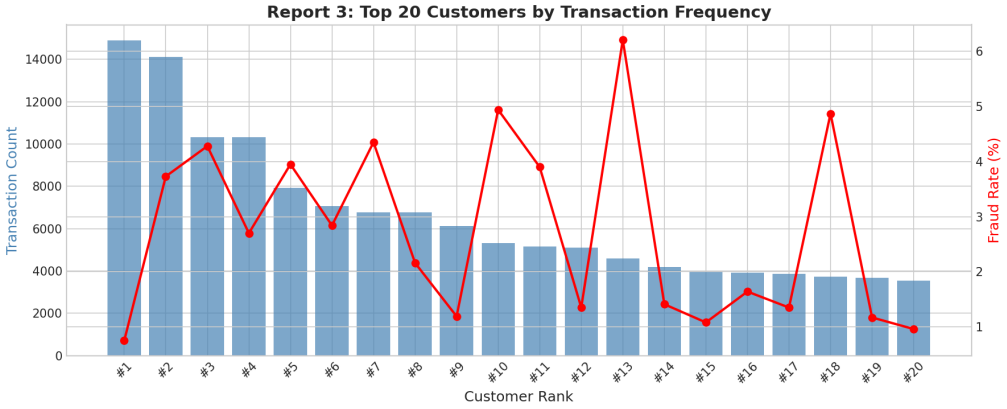

Figure 8: *Top 20 Customers by Transaction Volume and Their Fraud Rates*

5.2.4 Report 4: Average Transaction Amount Comparison

Fraudulent transactions average $149.24 versus $134.51 for legitimate transactions.

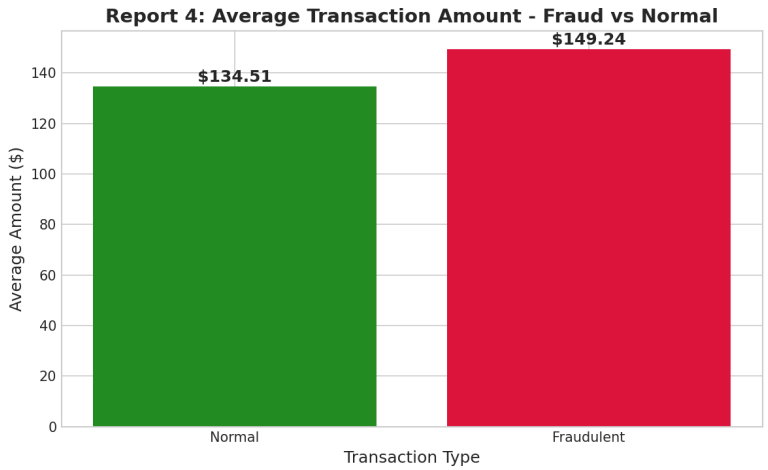

Figure 9: *Average Transaction Amount: Fraud vs Normal*

5.2.5 Report 5: Daily Fraud Detection Trend

Tracks fraud rates over sequential days to identify temporal patterns and anomalies.

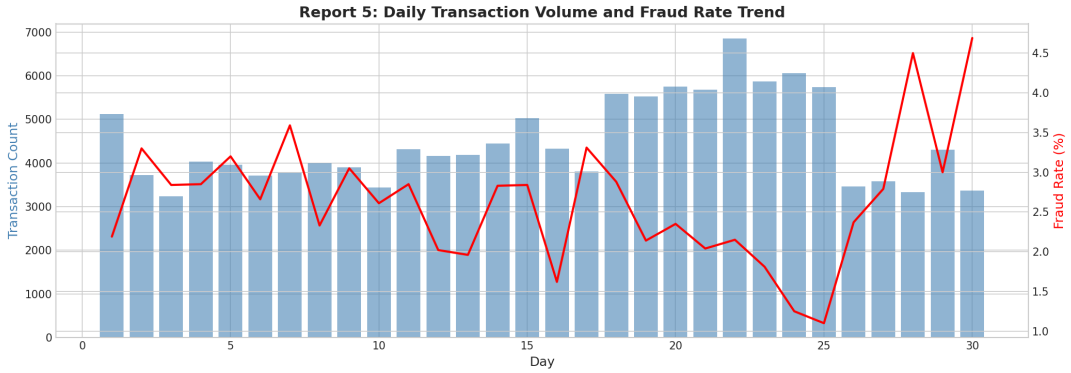

Figure 10: *Daily Transaction Volume and Fraud Rate Trend*

5.2.6 Report 6: Model Performance Summary

Retrieves stored evaluation metrics for all four models from the database.

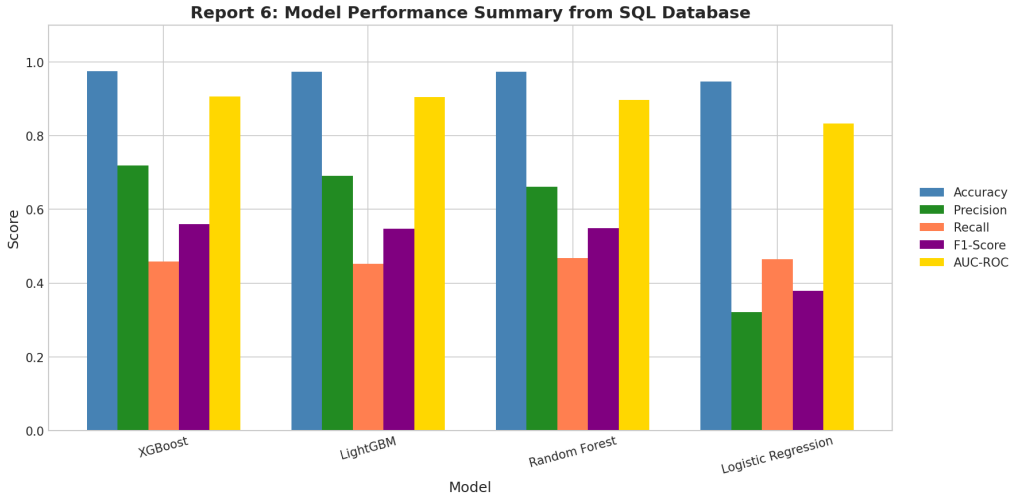

Figure 11: *Model Performance Metrics Retrieved from SQL Database*

5.2.7 Report 7: Transactions Flagged by Each Model

Compares how many transactions each model flags as fraudulent.

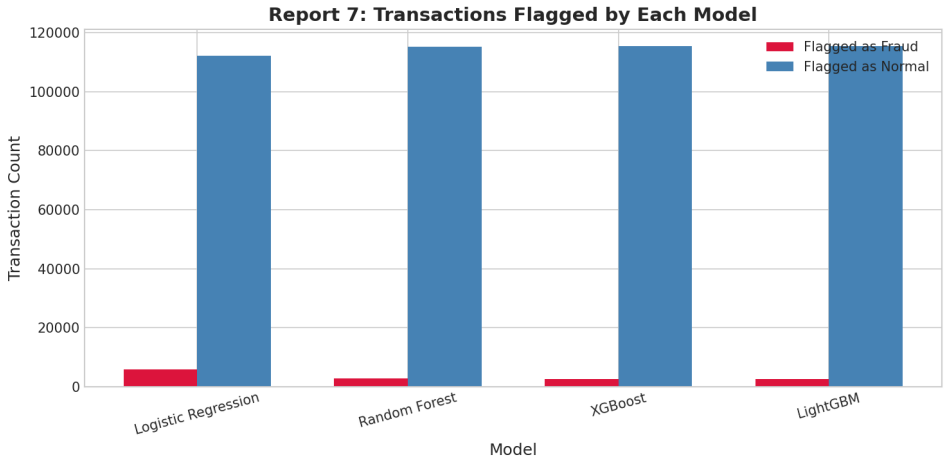

Figure 12: *Transactions Flagged as Fraud by Each Model*

5.2.8 Report 8: Customer Segments with High Fraud Likelihood

Identifies customer segments by card type/category with elevated fraud rates.

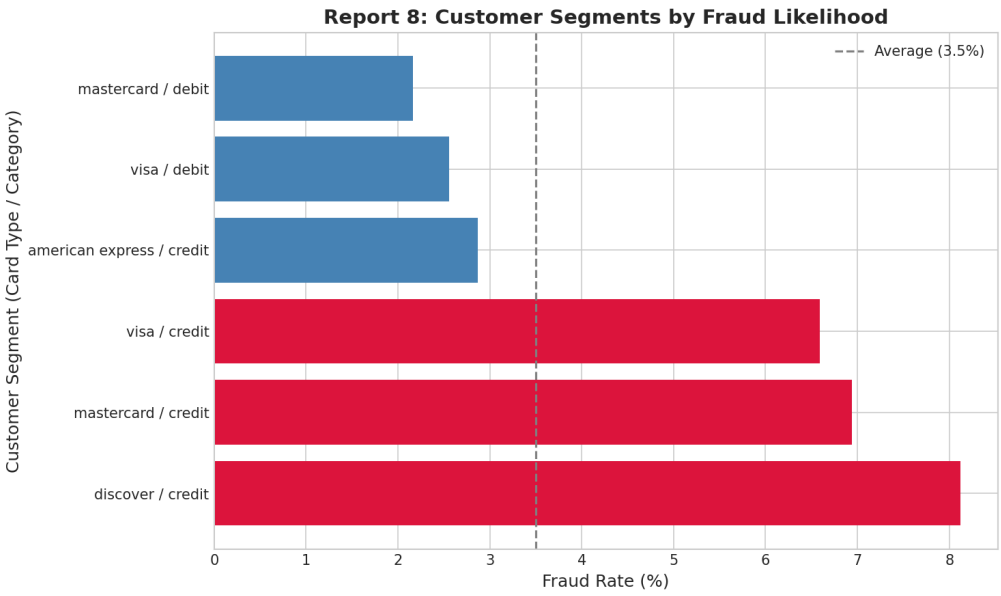

Figure 13: *Customer Segments Ranked by Fraud Likelihood*

5.2.9 Report 9: Precision/Recall Summary per Model

Displays precision and recall side-by-side for model comparison.

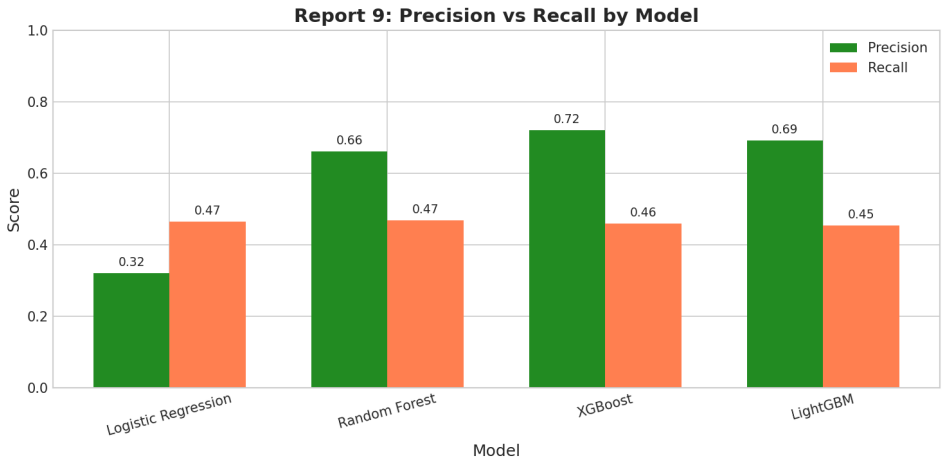

Figure 14: *Precision vs Recall Comparison Across Models*

5.2.10 Report 10: Fraud Rate by Device Type

Unknown device types correlate with higher fraud likelihood.

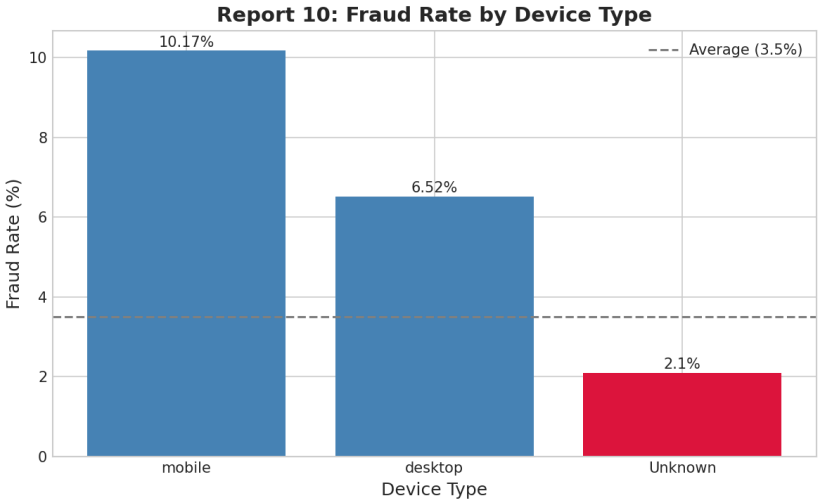

Figure 15: *Fraud Rate by Device Type - Unknown Devices Show Higher Risk*

6 Discussion and Insights

6.1 Key Findings

This project demonstrated that gradient boosting algorithms (XGBoost, LightGBM) significantly outperform traditional methods for fraud detection in complex, high-dimensional transaction data. The 7.35 percentage point ROC-AUC improvement from Logistic Regression (0.8327) to XGBoost (0.9062) represents substantial real-world impact.

SMOTE's effectiveness in addressing class imbalance without sacrificing model generalization validated our preprocessing approach. The synthetic samples enabled models to learn nuanced decision boundaries while the held-out test set maintained realistic class proportions (3.5% fraud) for evaluation.

Feature engineering emerged as critical, with our 11 engineered features (temporal, amount-based, email, and interaction features) consistently contributing to model performance. The combination of domain knowledge and data-driven feature creation amplified performance beyond raw features alone.

6.2 Practical Implications

6.2.1 For Financial Institutions

- Deploying XGBoost with 71.95% precision significantly reduces false alarms compared to simpler models
- The SQL reporting infrastructure enables real-time monitoring of fraud patterns by hour, product, and customer segment
- Feature importance analysis identifies high-risk scenarios (early morning, digital products, unknown devices) for enhanced monitoring

6.2.2 For Machine Learning Practitioners

- SMOTE with sampling_strategy=0.5 provided effective class balancing without excessive synthetic data
- Cardinality-based encoding (threshold of 10) balanced model performance with computational efficiency
- Multiple evaluation metrics (especially precision and recall) are essential for imbalanced classification problems

6.3 Limitations and Challenges

Despite strong performance, several limitations warrant acknowledgment:

- **Recall limitation:** The best recall achieved was 46.87% (Random Forest), meaning over half of fraud cases remain undetected at the default threshold
- **Precision-Recall trade-off:** Improving recall requires lowering thresholds, which increases false positives
- **Temporal drift:** Models trained on historical data may degrade as fraud tactics evolve
- **Feature anonymization:** Many high-importance V-columns are anonymized, limiting interpretability

6.4 Future Work and Recommendations

6.4.1 Model Enhancement

- Implement threshold optimization to balance precision/recall based on business costs
- Explore ensemble methods combining predictions from multiple models
- Apply hyperparameter tuning (GridSearch, Bayesian optimization) for improved performance

6.4.2 Deployment Considerations

- Build automated retraining pipeline with concept drift detection
- Implement real-time scoring API for production integration
- Develop monitoring dashboards for model performance degradation

6.4.3 Research Directions

- Investigate neural network architectures (deep learning) for complex pattern recognition
- Apply SHAP values for enhanced model interpretability
- Develop cost-sensitive learning approaches that weight fraud detection vs. false alarm costs

7 Conclusion

This project successfully developed a production-ready fraud detection system achieving 90.62% ROC-AUC score through XGBoost modeling on the IEEE-CIS dataset. Our comprehensive approach spanning data preprocessing, feature engineering, model development, and database infrastructure demonstrates the end-to-end capabilities required for real-world deployment.

The gradient boosting models' superior performance validates the importance of capturing complex non-linear relationships in fraud detection. XGBoost achieved the best balance of metrics with 71.95% precision, 45.87% recall, and 0.5603 F1-score, significantly outperforming the logistic regression baseline.

Our preprocessing pipeline successfully addressed the major challenges inherent in fraud detection: extreme class imbalance (addressed via SMOTE), high dimensionality (handled through feature selection), missing values (strategic imputation), and mixed data types (hybrid encoding). The 11 engineered features capturing temporal, amount, email, and interaction patterns enhanced model performance.

The SQL reporting infrastructure with 10 analytical reports provides ongoing operational visibility, enabling fraud analysts to monitor trends by hour, product, customer segment, and device type. Key insights include elevated fraud rates during early morning hours (6-9 AM), highest risk in digital products (ProductCD='C'), and suspicious patterns with unknown device types.

As fraud tactics continue to evolve, the methodologies and infrastructure developed in this project provide a foundation for adaptive systems that can be retrained with new data. The modular architecture supports continuous improvement through experimentation with new features, algorithms, and threshold optimization approaches.

This work demonstrates that modern machine learning, when properly applied with domain expertise and robust engineering practices, can significantly advance fraud detection capabilities and protect both financial institutions and consumers from sophisticated fraudulent activities.