Credit Card Fraud Analysis with Machine Learning

Tackling Class Imbalance for Robust Detection

Problem

Credit card fraud: Significant financial losses for banks, merchants, and consumers.

Erodes trust in financial systems.

Traditional rule-based systems struggle with evolving fraud patterns.

Core Problem: Highly imbalanced datasets – fraud is a rare event (e.g., 0.17% of transactions).

Our Goal: Build accurate ML models to identify fraud, minimizing both False Positives (customer inconvenience) and False Negatives (financial loss).

Understanding the Data

Source: Kaggle credit card transaction dataset (Sept 2013, 48 hours).

Features:

- Time, Amount (original scale)
- V1-V28 (PCA-transformed for confidentiality)

Target (Class): Binary (0: Non-Fraud, 1: Fraud).
Key Challenge: Extreme Class Imbalance: Fraud is only 0.17% of transactions (e.g., ~492 fraud vs. ~284k non-fraud in full dataset).

Initial Data Insights

Confirmed severe class imbalance.

Fraud Distribution: More evenly spread across the 48-hour period.

Non-Fraud Distribution: Peaks during typical working/daytime hours.

Transaction Amount: Fraudulent transactions tend to be for smaller amounts.

Correlation: Some PCA features show correlation with the 'Class' variable.

Separability (t-SNE): Visual exploration with t-SNE suggested distinct clusters, indicating potential for classification.

Data Preprocessing

Data Cleaning: Removed 1,081 duplicate rows (no missing values found).

Train-Test Split:

- 80% Training, 20% Testing.
- **Crucial:** Used **stratification** to preserve class proportions in both sets.

Feature Scaling:

- Time: StandardScaler (consistent range).
- Amount: RobustScaler (less sensitive to outliers in transaction values).
- **Key Prevention:** Scalers fitted **only on training data** then transformed on both, preventing **data leakage**.

Data Preprocessing

Why resample? Prevent models from being biased towards the majority class (leading to poor fraud recall).

1. Random Oversampling:

- Duplicates random instances of the minority (fraud) class.
- Simple, but risks overfitting by creating exact copies.

2. SMOTE (Synthetic Minority Over-sampling Technique):

- Creates *synthetic* new minority samples by interpolating between existing ones.
- Introduces more diversity than simple duplication, helping generalization.

Data Preprocessing

3. Random Undersampling:

- Randomly removes instances from the majority (non-fraud) class.
- Reduces dataset size (faster training), but risks losing valuable information.

4. SMOTE + Tomek Links (Hybrid):

- Combines SMOTE oversampling with Tomek Links undersampling.
- SMOTE creates synthetic samples, then Tomek Links removes "noisy" majority samples close to minority ones.
- Aims to create a cleaner decision boundary.

Data Modeling

1. Logistic Regression (Baseline):

- Simple, interpretable, computationally efficient.
- Used as a baseline for comparison.
- class_weight='balanced' parameter to handle imbalance internally.

2. Random Forest Classifier:

- Ensemble of decision trees.
- Handles high-dimensional data well.
- class_weight='balanced' parameter.

3. XGBoost Classifier:

- Gradient Boosting Machine (strong ensemble method).
- Highly efficient and performs well on structured data.
- scale_pos_weight parameter to address imbalance (weights positive class).

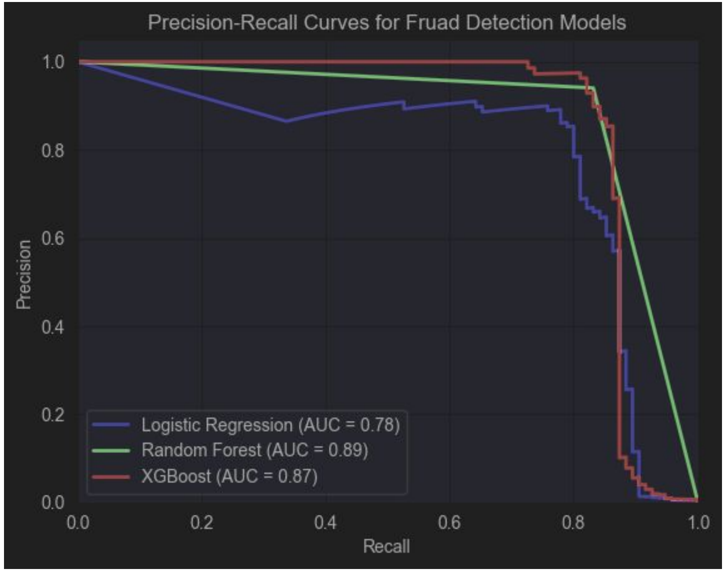
Precision-Recall Curves

X-axis: Recall (True Positive Rate): Ability to find *all* actual fraud cases (minimize false negatives).

Y-axis: Precision (Positive Predictive Value): Proportion of identified fraud cases that are *actually* fraudulent (minimize false positives).

AUC (Area Under Curve): Higher AUC indicates better overall performance.

Interpretation: The closer the curve is to the top-right corner, the better the model's performance trade-off.



Evaluation & Key Findings

Logistic Regression: Decent Recall (~0.91) but very low Precision (~0.06), flagging many legitimate transactions. Resampling had minimal impact.

Random Forest: Strong performance with original (class_weight='balanced') and Random Oversampled data.

- *Example (Random Oversampled):* Recall 0.82, Precision 0.98, F1-Score 0.89.

XGBoost: Consistent strong performance across sampling methods, particularly Random Oversampling.

- *Example (Random Oversampled):* Recall 0.84, Precision 0.93, F1-Score 0.88.

Key Insight: Both Random Forest and XGBoost performed best when trained on **randomly oversampled data**, despite SMOTE/SMOTETomek being theoretically more advanced. **Recommendation: XGBoost** is preferred due to its comparable performance to Random Forest but significantly faster training time.

Challenges & Limitations

Dataset Limitations: Anonymized PCA features limit deep interpretation and rich feature engineering.

"Advanced" Sampling Surprises: SMOTE / SMOTE + Tomek links did not universally outperform random oversampling; requires further investigation for optimal application.

Overfitting Risk (Synthetic Data): Care must be taken to ensure synthetic data doesn't lead to overfitting to artificial patterns.

Concept Drift: Fraud patterns constantly evolve; models degrade over time. Requires continuous learning/adaptation.

Data Privacy & Sharing: Real-world data is highly sensitive, limiting cross-institutional research.

Conclusion

Successfully developed and evaluated machine learning models for credit card fraud detection.

Implemented and compared various data imbalance handling techniques (oversampling, undersampling, hybrid).

Identified **XGBoost** as the top-performing model for this dataset, achieving strong Recall, Precision, and F1-Score, especially with **randomly oversampled training data**, while also being computationally efficient.

Highlighted critical real-world challenges: extreme imbalance, concept drift, data privacy, and interpretability.

Reinforced the importance of balancing false positives and false negatives for real-world application.