Credit Card Fraud Detection System

Introduction

Credit card fraud is a significant challenge in the financial industry, leading to billions of dollars in losses annually. Detecting fraudulent transactions quickly and accurately is crucial to protect consumers and reduce financial risks for businesses.

This project focuses on building a machine learning-based fraud detection system using a publicly available credit card transaction dataset. The goal is to develop models that can accurately identify fraudulent transactions despite severe class imbalance, where fraudulent cases are extremely rare compared to legitimate ones.

Through exploratory data analysis, feature engineering, and model tuning, this project demonstrates practical techniques to tackle imbalanced classification problems and improve fraud detection performance.

Dataset Description

The dataset used in this project is the Credit Card Fraud Detection dataset from Kaggle. It contains credit card transactions made by European cardholders in September 2013.

- **Number of transactions:** 284,807
- **Number of features:** 30
- **Features:** 28 anonymized features (labeled `V1` to `V28`) obtained through Principal Component Analysis (PCA) to protect sensitive information, plus `Time` (seconds elapsed between each transaction and the first transaction) and `Amount` (transaction value).
- **Target variable:** `Class` — binary indicator where 0 represents a legitimate transaction and 1 indicates fraud.

The dataset is highly imbalanced, with only about 0.17% of transactions labeled as fraud, which presents challenges for modeling.

Data Exploration & Cleaning

To understand the dataset and prepare it for modeling, initial exploratory data analysis (EDA) and cleaning were performed.

- **Basic statistics:** The dataset contains 284,807 transactions with 30 features. The `Amount` feature ranges widely, and the `Time` feature represents elapsed seconds from the first transaction.

- **Class imbalance:** The target variable is highly skewed, with only about 0.17% of transactions labeled as fraud (Class = 1). This imbalance requires careful modeling techniques.

- **Duplicate removal:** A total of 1,081 duplicate rows were detected and removed to prevent data leakage and bias. This reduced the dataset size slightly while maintaining data integrity.

- **Missing values:** No missing values were found in any features, so no imputation was necessary.

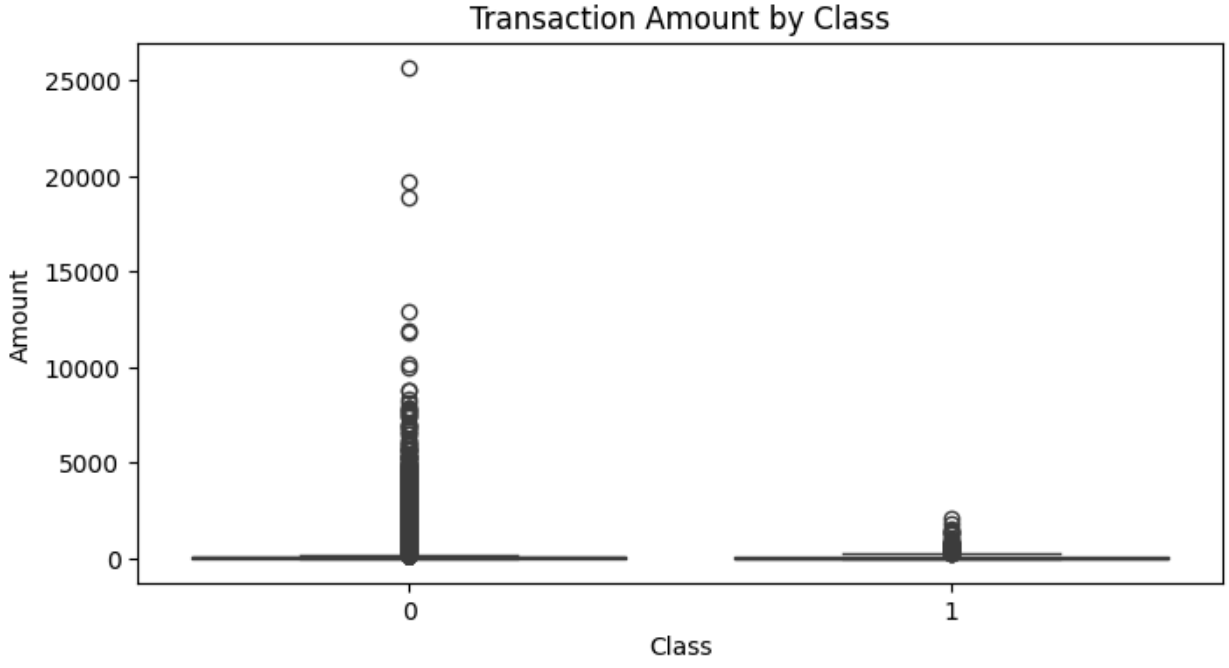
- **Visualizations:**

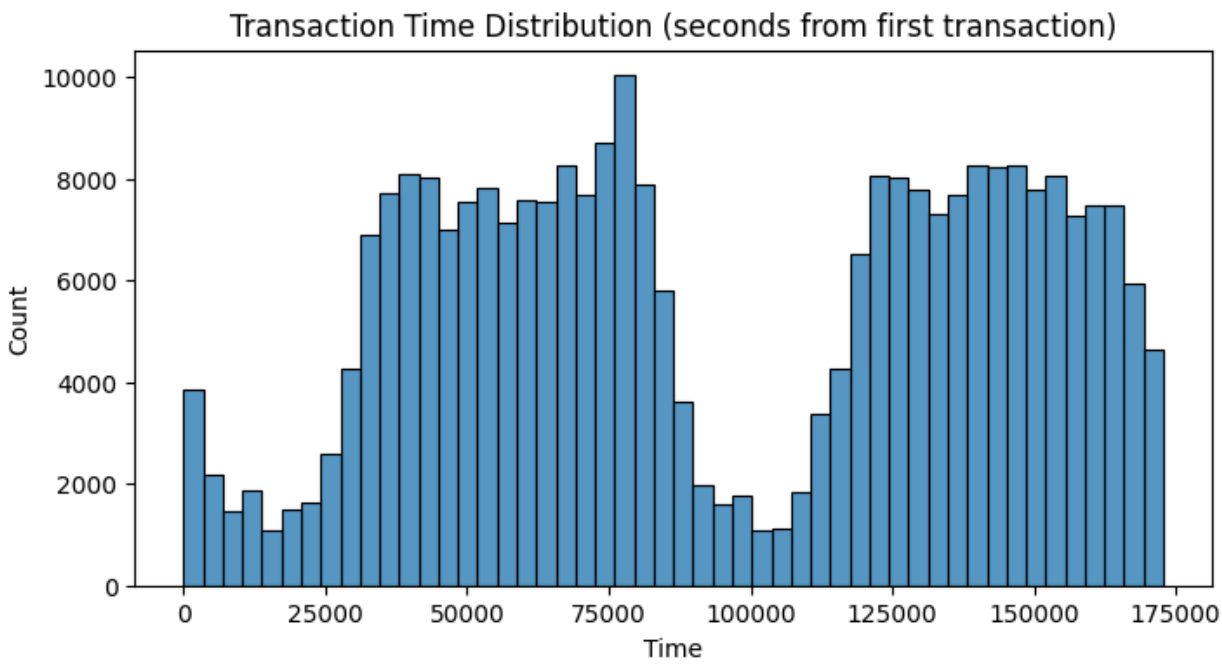
    - Class distribution was visualized with a bar plot showing the overwhelming majority of legitimate transactions.

    - Transaction amounts were compared between fraudulent and legitimate cases using boxplots, highlighting differences in distribution.

    - Transaction time distribution was explored to understand transaction patterns over time.


Class Distribution (0=No Fraud, 1=Fraud)
250000
200000
150000
100000
50000
0
count
0
1
Class


Transaction Amount by Class
25000
20000
15000
10000
5000
0
Amount
0
1
Class



Data Preprocessing

Before modeling, the dataset was prepared with the following steps:

- **Feature Scaling:** The `Amount` feature was scaled using StandardScaler because it was not transformed like the PCA features. This ensures that all features have a comparable scale, which benefits many machine learning algorithms.
- **Feature Removal:** The `Time` feature was dropped from the dataset as it did not significantly improve model performance and was less relevant for fraud detection in this context.
- **Train-Test Split:** The data was split into training and testing sets with an 80-20 ratio, using stratified sampling to maintain the original class distribution in both sets.
- **Handling Imbalance:** Due to the severe class imbalance, traditional oversampling or undersampling methods were initially avoided. Instead, threshold tuning on model prediction probabilities was employed to balance precision and recall effectively.

Model Building & Evaluation

Two machine learning models were built and evaluated to detect fraudulent transactions:

1. Logistic Regression (Baseline Model)

- A logistic regression model with class weighting was used as a baseline. Class weights helped the model pay more attention to the minority fraud class during training.

- **Performance:**

    - Accuracy was high (~98%) due to class imbalance.

    - Recall for fraud cases was 87%, meaning the model detected 87% of fraudulent transactions.

    - Precision was low (~6%), indicating many false positives.

2. Random Forest Classifier (Improved Model)

- A Random Forest classifier with balanced class weights was trained to improve performance.

- **Performance:**

    - Precision increased significantly to 99%, drastically reducing false positives.

    - Recall dropped to 71%, indicating fewer frauds detected compared to logistic regression.

    - F1-score improved, showing better balance between precision and recall.

    - ROC AUC was 0.93, indicating strong discrimination ability.

Threshold Tuning

- To improve recall without sacrificing precision, the classification threshold was tuned.

- By lowering the decision threshold from 0.5 to 0.3, recall increased to 74% while maintaining a high precision of 95%.

- This demonstrates how adjusting thresholds can tailor model sensitivity to business needs in fraud detection.

Insights & Challenges

Insights

- **The extreme class imbalance in fraud detection requires careful handling to avoid biased models that overwhelmingly predict the majority class.**
- **Logistic Regression achieved higher recall but at the cost of many false positives, which can overwhelm fraud investigation teams.**
- **Random Forest improved precision significantly, reducing false alarms, but slightly lowered recall.**
- **Threshold tuning proved effective in balancing precision and recall, showcasing the importance of adjusting decision thresholds in imbalanced classification problems.**

Challenges

- **The dataset features were anonymized PCA components, limiting interpretability of the model and feature importance analysis.**
- **Installing and managing packages like imbalanced-learn posed compatibility challenges during implementation.**
- **Handling the severe class imbalance without extensive oversampling required creative approaches like threshold tuning.**
- **Real-world fraud detection systems may require additional features, real-time processing, and more advanced algorithms.**

Conclusion

This project successfully built a machine learning-based credit card fraud detection system capable of identifying fraudulent transactions with high precision and recall. Despite the severe class imbalance and anonymized features, models like Random Forest combined with threshold tuning demonstrated strong performance, achieving a good balance between detecting fraud and minimizing false positives.

Future work could explore advanced sampling techniques, more powerful models like XGBoost, real-time fraud detection pipelines, and model interpretability tools to further enhance the system's effectiveness and practical utility.

This project highlights key challenges and strategies in handling imbalanced classification problems and provides a foundation for building scalable fraud detection solutions.

Code & Resources

- **The full Jupyter Notebook with all code, visualizations, and detailed explanations is available at:**
  [https://github.com/Saubia-Aimen/CreditCardFraudDetection.git]

- **Dataset source: Kaggle Credit Card Fraud Detection**

- **Key Python libraries used: pandas, numpy, scikit-learn, matplotlib, seaborn**