Credit Card Fraud Detection

Group Presentation - CIS 9557

Group 1
Janiel Avelar
Ajish Sam
Payal Surana

Table of contents

Introduction

- ➔ Credit card fraud is a growing concern in the digital age
- ➔ Online transactions are increasing, making systems more vulnerable
- ➔ Our dataset: Kaggle Credit Card Fraud (284,807 transactions, only 492 fraud cases → ~0.2%)
- ➔ Rare fraud cases are hard to detect using traditional methods
- ➔ We aim to explore how machine learning and business analytics can improve fraud detection systems

Business Problems

- Fraud = only 0.2% of 284,807 transactions → extreme class imbalance
- Our model implementation addressed:
    - Missed frauds → by applying SMOTE and under-sampling techniques
    - False positives → reduced using cost-sensitive learning and anomaly detection
    - Real-time detection → optimized using lightweight and scalable algorithms
- Key observations from our analysis:
    - Fraud occurred most frequently during the **afternoon (12pm–6pm)**, not just late night as initially assumed and are often involved smaller amounts.
    - PCA-anonymized features challenged interpretability, so we relied on correlation-based insights
- Fraud detection proved to be not just a data science challenge, but a strategic issue involving customer trust, compliance, and real-time responsiveness

Updated Problem Statement:

"How can financial institutions design an adaptive and scalable fraud detection system that accurately identifies fraudulent credit card transactions in a highly imbalanced, anonymized dataset while ensuring low false positives and real-time responsiveness amid evolving fraud tactics?"

Analytical Techniques We Applied:

- **Value-Chain Analysis** – Helped identify where fraud impacts value the most
- **Risk Analysis** – Quantified the impact of missed fraud and false alerts
- **Customer Analysis** – Mapped behavioral patterns to fraud risk
- **Descriptive Analytics** – Revealed trends in timing, amounts, and frequencies
- **Diagnostic Analytics** – Uncovered model weaknesses and false positive causes

Data analysis and Insights

Dataset Overview

- ➔ Source: Kaggle – Credit Card Fraud Detection CSV
- ➔ Total Transactions: **284,807**
- ➔ Fraudulent Cases: **492 (0.17%)**

Cleaning and Preprocessing

- ➔ Used **Python** for further analysis
- ➔ Filtered records where **Class = 1** (fraud only)
- ➔ Kept V1–V28 (PCA-transformed features) + Amount

Why these columns?

- ➔ V1–V28: Statistically important for model training
- ➔ Amount: Helps identify transaction behavior
- ➔ No missing values – smooth processing

Tools

- **Google Colab (Python)** – Model training, evaluation, and visualization
- **Scikit-learn** – Preprocessing, classification models, performance metrics
- **XGBoost** – Advanced gradient boosting for fraud detection
- **Imbalanced-learn (SMOTE)** – Addressing extreme class imbalance
- **Matplotlib & Seaborn** – Visual analytics and data storytelling

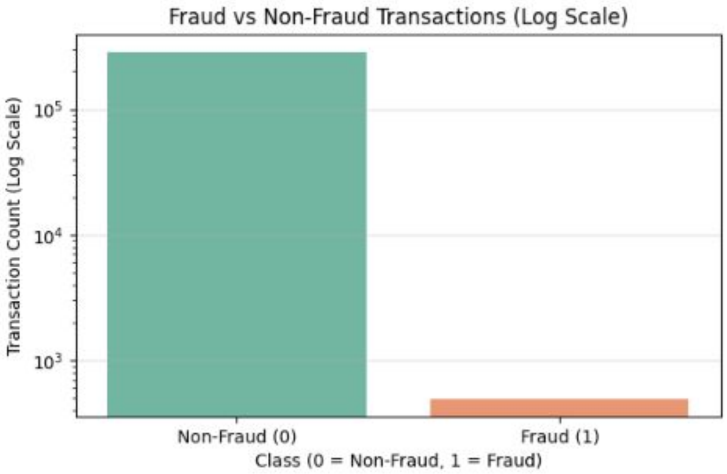
Methods

- **SMOTE (Synthetic Minority Oversampling Technique)** – Balanced fraud vs. non-fraud classes
- **Threshold Tuning** – Optimized model sensitivity for fraud prediction
- **Feature Scaling (StandardScaler)** – Improved model convergence and accuracy
- **Anomaly Detection** – Flagged unusual behavior for enhanced detection
- **PCA Feature Correlation** – Assessed relationships between anonymized features
- **Time-based Bucketing** – Identified peak fraud hours across the day

Takeaways from Our Analysis

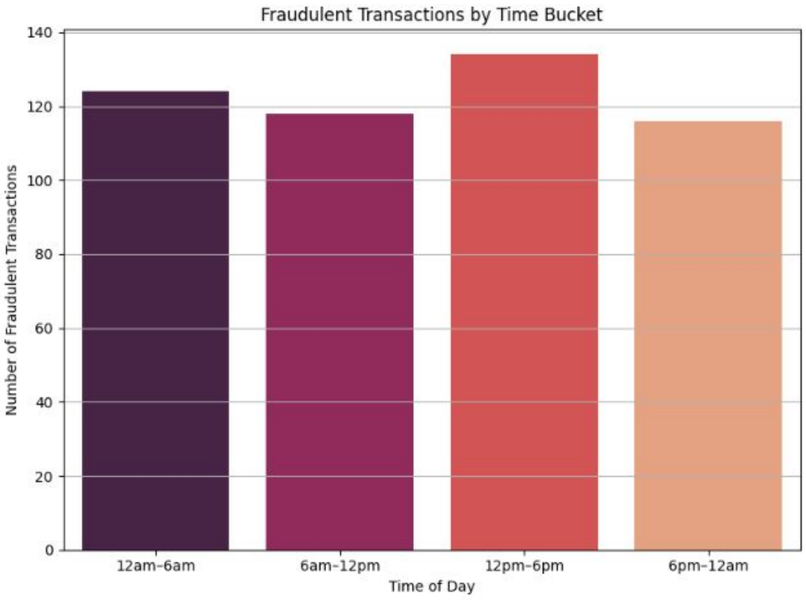
Class Imbalance Visualization

- ➔ Dataset contains **284,807 transactions**, with only **492 fraud cases (0.17%)**
- ➔ Severe **class imbalance** is a key challenge in fraud detection
- ➔ We used a **logarithmic scale** on the y-axis to make fraud cases visible in the bar chart
- ➔ This imbalance motivated the use of **SMOTE** (oversampling), **Undersampling, Cost-sensitive models**



Takeaways from Our Analysis

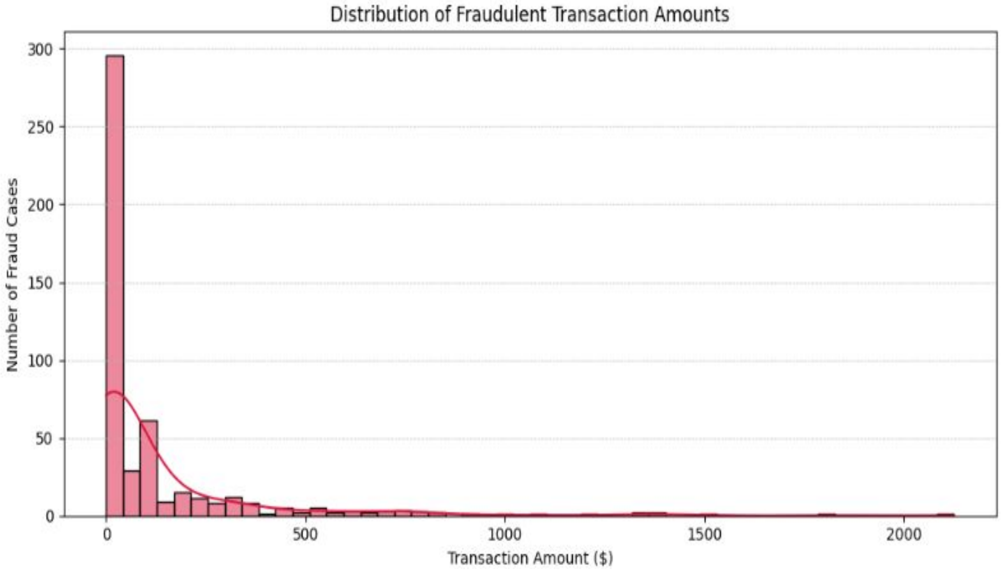
Fraudulent Transactions by Time Bucket



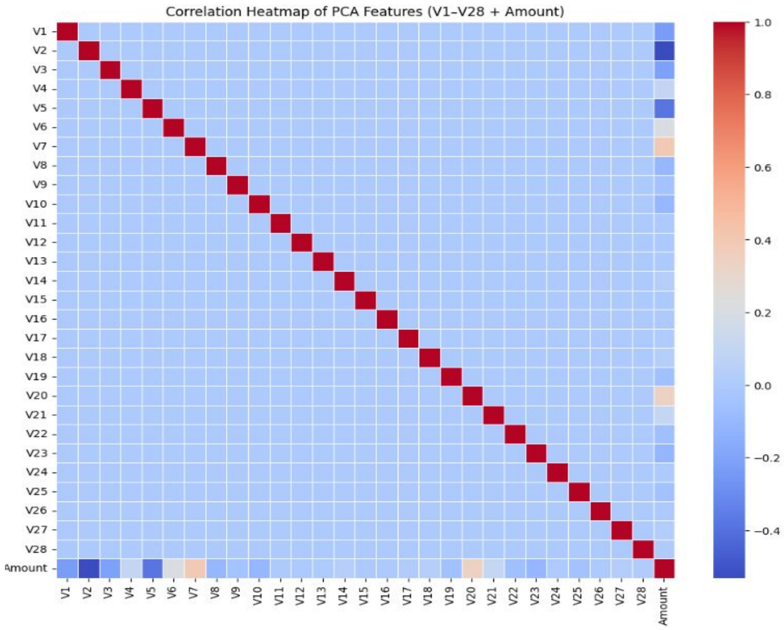
- ➔ We grouped fraud cases into 6-hour time blocks.
- ➔ Surprisingly, **most fraud occurred between 12pm and 6pm** — not late night as expected.
- ➔ Other active windows: **early morning and midnight to 6am**.
- ➔ Fraudsters may be timing attacks during busy hours to **blend in with normal transactions**.

Transaction Amount Patterns

- ➔ Focused only on **fraudulent transactions (Class = 1)**
- ➔ **Most fraud cases involve small amounts**, typically under **$200**
- ➔ **Lower transaction values** may be used to avoid detection or customer suspicion
- ➔ **Rare high-value frauds** also exist but are much less frequent
- ➔ Insights help us design models that detect **subtle and low-amount fraud patterns**



Correlation Heatmap of PCA (V1-V28 and Amount)



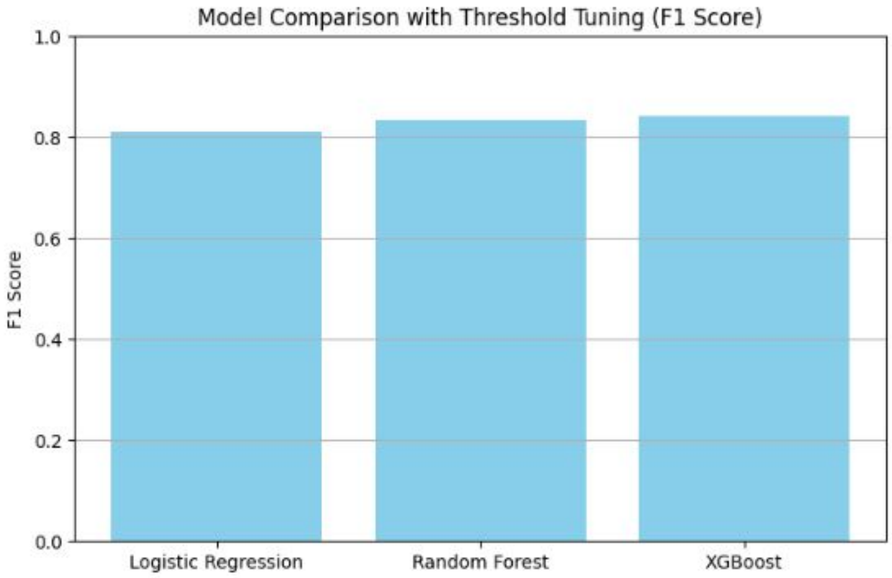
➔ As shown in the correlation heatmap, PCA-transformed features (V1–V28) do not offer clear interpretability.

➔ This aligns with our challenge of analyzing fraud risk patterns — although machine learning models can learn from PCA features, understanding their real-world meaning is limited, which makes human-driven insights or rule-based decisions difficult.

Model Comparison with Threshold Tuning (F1-Score)

- XGBoost achieved the **highest F1 score**, outperforming Logistic Regression and Random Forest.

- This was after applying **SMOTE, feature scaling**, and **threshold adjustment**, helping the model better capture rare fraud cases while minimizing false positives.



Proposed Solutions

- Use SMOTE and undersampling to fix class imbalance in the data.
- Train fraud detection models using XGBoost and other ensemble methods for better accuracy.
- Set up a real-time fraud detection system using lightweight models and live data streams.
- Apply dynamic thresholds based on time of day and transaction amount to catch more fraud.
- Add behavioral analysis to flag unusual user activity or spending patterns.
- Suggested to use tools like SHAP or LIME to explain why a transaction was flagged as fraud.
- Create a feedback loop to retrain the model regularly with new confirmed fraud cases.
- Involve manual review for edge cases to improve decision-making over time.

Implementation Plan (POAM)

ID	Task	Team	Start	End	Status	Milestones	Impact Level
10	Fraud Score Design	Business Intelligence/ Analytics	05/13	05/20	Not Started	Design Deployed and Ready for Reporting	Low
11	Model Development/Anomaly Detection	Machine Learning Team	05/13	06/13	Not Started	Temporal Features Added	Medium
12	Real-Time Fraud Alert System	Development Operations	05/13	06/13	Not Started	API in Real Time/Alert System Implemented	High
13	Implementation of XGBoost Model along with threshold tuning	Machine Learning Team	05/13	05/20	Not Started	Tuned Up and Implemented	High

Conclusion

- The Credit Card Fraud Detection Dataset from Kaggle presented a major imbalance, 492 Transactions were fraudulent.
- Due to the imbalance, excessive false positives were identified
- Techniques like SMOTE, threshold tuning, and under-sampling helped to improve the accuracy , but the XGBoost model performed the best out of all the models used
- In order to detect fraudulent transactions in real time, operational and strategic planning is necessary
- Our POAM outlines the implementation process in order to apply an API Integration, Anomaly Detection, and Reporting
- Monitoring, reporting, and training must always take place to make the Credit Card Fraud Detection system sustainable