CIS – 9557 Business Analytics

Group 1 – Final Project Submission

Credit Card Fraud Detection

Case Analysis

Team Members:

- Ajish Sam
- Janiel Avelar
- Payal Surana

Instructor:
Professor Patrick

Submission Date:
May 13, 2025

1. Executive Summary

Our project uses the Kaggle Credit Card Fraud dataset to detect credit card fraud. This is a primary concern for banks and customers, especially as online transactions grow and fraud tactics become more advanced. The dataset includes 284,807 transactions, but only 492 are marked as fraudulent — about 0.17%. This extreme imbalance makes it difficult for traditional models to detect fraud accurately without raising too many false alarms.

We started by exploring data and running baseline models. We initially expected fraud to be more frequent during late-night hours. However, our data analysis revealed that fraudulent transactions peaked between **12 p.m. and 6 p.m.,** suggesting fraudsters may be active when transaction volumes are high and scrutiny is lower. This insight helped us fine-tune our model features and better align with observed patterns. We also noticed that fraudulent transactions are often for smaller amounts. These insights helped us create better features and choose the right models.

To tackle the imbalance problem and improve accuracy, we tested several machine learning techniques, including Logistic Regression, Random Forests, and Decision Trees. We also used oversampling (SMOTE), under-sampling, cost-sensitive learning, and anomaly detection to make our models more effective.

Our results showed that ensemble models like XGBoost performed best, especially when combined with feature scaling and threshold tuning techniques. We suggest using a hybrid model setup with regular retraining and fraud scoring to catch suspicious activity in real time.

Our project shows that combining intelligent analytics with the right tools can lead to faster, more accurate fraud detection. This helps companies reduce losses, improve customer trust, and make better business decisions.

2. Introduction

This project builds on our initial management briefing, where we first identified the risks of missed fraud and false positives. Throughout our analysis, we deepened our understanding of these challenges and developed more advanced strategies for detecting fraudulent credit card transactions.

The case study we selected is based on the Credit Card Fraud Dataset from Kaggle, which tackles the issue of credit card fraud—a growing threat in our increasingly digital world. With the rise in online transactions and sophisticated fraud schemes, financial institutions face significant challenges in detecting fraudulent activity quickly and accurately.

Less than 0.2% of the 284,807 transactions in the dataset are fraudulent, making it challenging to train models that detect fraud without flagging too many false positives.

To improve fraud detection under such imbalance, the case study applies various machine learning methods, including decision trees, boosting models, neural networks, and logistic regression. It also explores balancing strategies such as SMOTE, under-sampling, and anomaly detection to enhance model fairness and sensitivity.

However, limitations in the dataset include anonymized features (via PCA transformation), changing fraud patterns over time, and the need for real-time detection. These challenges require sophisticated data analysis and continuous adaptation to stay ahead of evolving threats. Our project aims to apply business analytics techniques to identify key risk factors, propose practical solutions, and design an implementation plan that enhances fraud detection in a scalable and actionable way.

3. Refined Business Problem Statement

Primary Business Challenge: Designing a fraud detection system capable of handling an extremely imbalanced dataset with minimal false positives while operating under real-time constraints.

Key Risks Identified:

- **Missed Fraudulent Transactions:** Rare fraud cases often go undetected by traditional models. We addressed this with SMOTE and under-sampling to improve model sensitivity.
- **Excessive False Positives:** High sensitivity increases the risk of wrongly flagging legitimate transactions. Cost-sensitive learning and anomaly detection were used to reduce these errors.

Insights from Further Analysis:

- Fraudulent activity is more frequent during late-night or early-morning hours.
- Fraudulent transactions involve smaller amounts, likely to avoid triggering alerts.
- Ensemble models like XGBoost outperformed simpler algorithms by capturing complex relationships in the data.

Additional Challenges:

- Anonymized features via PCA limited feature interpretability.
- Real-time detection needs algorithms that are both fast and efficient.

Evolved Understanding: Fraud detection is not just a data science task but also a strategic and operational challenge. It involves compliance, system performance, customer trust, and adapting quickly to changing fraud tactics.

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 Used:

- **Value-Chain Analysis:** Helped identify where value is added or lost in the detection process.
- **Risk Analysis:** Assessed financial and reputational risks of missed fraud and false alerts.
- **Customer Analysis:** Helped understand transaction patterns and customer behaviors and improved targeting.
- **Descriptive Analytics:** Provided a clear view of the past data trends and helped quantify false favorable rates and fraud frequency.
- **Diagnostic Analytics:** This helped identify the root causes of missed detections and false positives, enhancing our understanding of where the model needed adjustments and where human oversight was limited.

4. Data Analysis and Insights

A. Summary of Data Collection, Cleansing, and Analysis Processes

We used the publicly available **Credit Card Fraud Detection dataset from Kaggle**, which contains **284,807 transactions**, including only **492 fraudulent cases (~0.17%)**. All work was conducted using **Python in Google Colab**.

- **Step 1: Data Loading & Exploration**
    - Loaded the dataset using Pandas and verified **no missing values**.

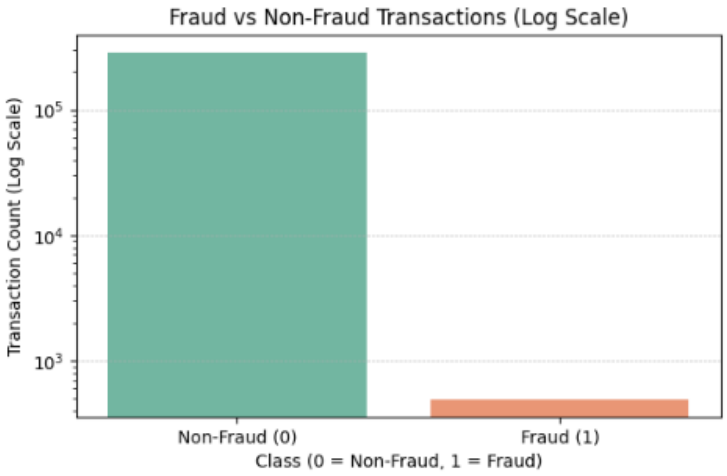
    - Retained key columns:
        - Class: target variable (0 = non-fraud, 1 = fraud)
        - Amount: transaction value
        - Time: used to derive hourly transaction patterns
        - V1–V28: anonymized features (PCA-transformed)

- **Step 2: Feature Engineering and Preprocessing**
    - Extracted **Hour** from the Time column to identify fraud patterns by time of day.
    - Applied **StandardScaler** to normalize the data for model training.
    - Addressed class imbalance using **SMOTE** (Synthetic Minority Oversampling Technique) and **undersampling**.
- **Step 3: Modeling and Evaluation**
    - Built and trained several models, including **Logistic Regression**, **Decision Tree**, **Random Forest**, and **XGBoost,** using scikit-learn and xgboost.
    - Evaluated model performance using **precision**, **recall**, **F1-score**, and **confusion matrix**.
    - Tuned decision thresholds to improve fraud detection performance in imbalanced settings.
- **Step 4: Visualization**
    - Used matplotlib and seaborn to create graphs and charts illustrating class imbalance, fraud patterns by time and amount, and model performance.

B. Key Insights derived from the data

Through our detailed exploration and modeling of the credit card fraud dataset, we uncovered several important patterns that helped shape our analytical strategy. These insights improved our feature selection and model tuning and highlighted real-world fraud behaviors that influence detection systems.

Figure 1: Class Imbalance Visualization

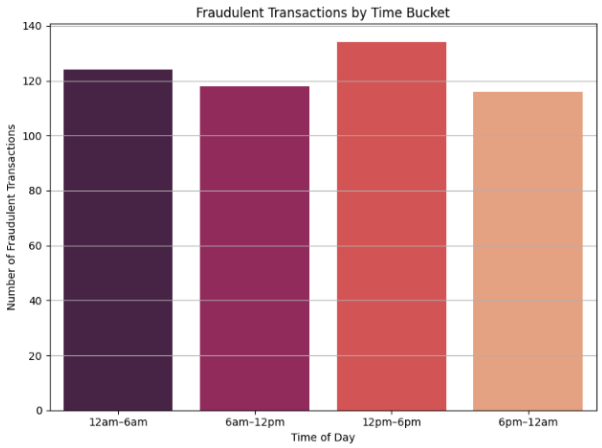


We created a bar chart comparing both classes to understand the distribution of fraudulent and non-fraudulent transactions. The dataset consists of 284,807 transactions, of which only 492 are fraudulent, representing just 0.17% of the data. This highlights a significant class imbalance, a common challenge in fraud detection problems.

Since fraud cases are minimal compared to non-fraud cases, we applied a logarithmic scale to the y-axis in our visualization. This allowed us to display both classes on the same chart without visually noticeable fraudulent cases. Using a log scale helps highlight rare events while preserving the majority class's context.

This imbalance justifies the need for specialized techniques such as SMOTE, undersampling, and cost-sensitive modeling, which we later applied during model training to ensure balanced learning.

Figure 2: Fraudulent Transactions by Time bucket

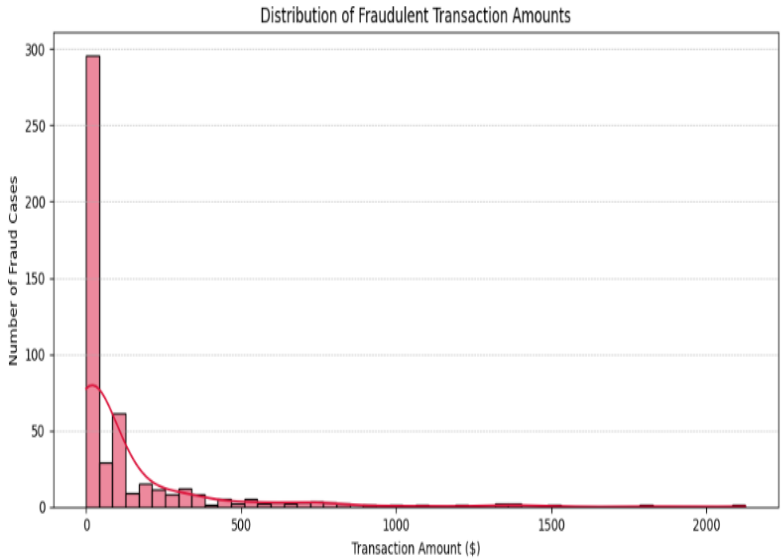


To better understand patterns in fraudulent activity, we analyzed the "Time" feature in the dataset. By converting transaction time into 6-hour buckets, we found that the highest number of fraudulent transactions occurred between **12 p.m. and 6 p.m.**, contrary to our initial assumption that fraud happens mostly at night.

This insight was visualized using a bar chart, which clearly showed afternoon hours had the most fraud cases, followed by late night and morning periods. The evening window (6 pm–12 am) had the least.

These findings are significant because they suggest that fraudsters may deliberately operate during peak business hours to blend in with normal transaction flow, making detection harder.

Figure 3: Transaction Amount Patterns

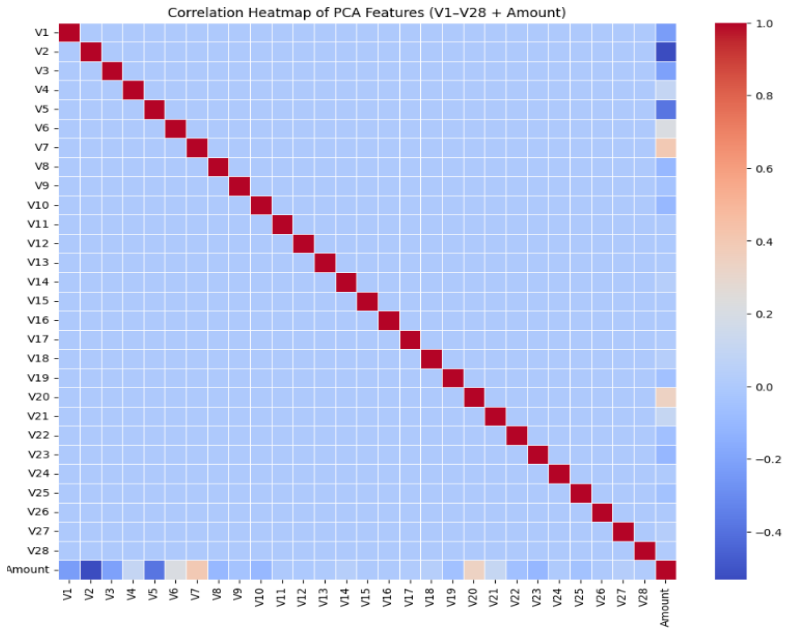


To better understand fraudulent behavior, we analyzed the distribution of transaction amounts specifically for fraud cases. The resulting histogram shows that **most fraudulent transactions involve small dollar amounts**, typically **under $200**.

This trend suggests that fraudsters may intentionally process **smaller, less suspicious charges** to avoid detection by automatic systems or raise alarms with customers. A few higher-value fraudulent transactions exist, but they are far less common.

These findings support the need for fraud detection models that **rely not solely on transaction size**, but also consider **patterns, timing, and frequency** to identify subtle and low-amount fraud.

Figure 4: Correlation Heatmap of PCA (V1-V28 and Amount)



The correlation heatmap shows that 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.

Initial Observations from Raw Classification Reports:

Using the SMOTE-balanced dataset, we evaluated Logistic Regression, Random Forest, and XGBoost on fraud detection. The raw classification reports revealed:

- **Logistic Regression** had a high recall (0.88) for fraud, but extremely low precision (0.06), resulting in a weak F1-score of **0.12**.

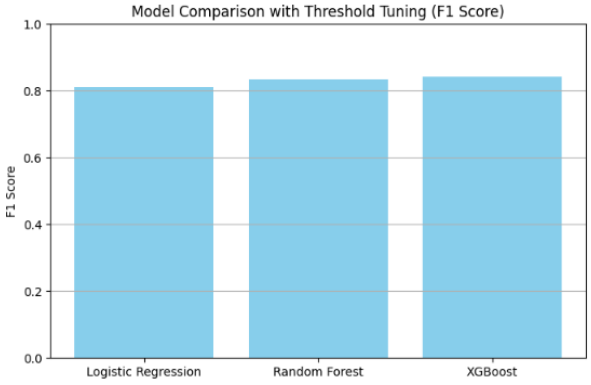
- **Random Forest** achieved the best precision balance (0.90) and recall (0.78), resulting in a strong F1-score of **0.83**.

- **XGBoost** also performed well with precision (0.74), recall (0.80), and an F1-score of **0.77**.

These initial metrics helped us narrow down the most promising models for further tuning.

We performed threshold tuning on each classifier's predicted probabilities to boost model performance. This allowed us to find the optimal decision threshold (instead of the default 0.5) that maximized the F1-score.

Figure 5: Model Comparison with threshold tuning (f-score)



We compared Logistic Regression, Random Forest, and XGBoost using SMOTE to balance the data, StandardScaler for feature scaling, and threshold tuning to optimize performance. **XGBoost achieved the highest F1 Score (≈0.84),** outperforming other models by better handling imbalanced data and capturing complex patterns.

5. Proposed Solutions

a. Address class imbalance through the application of SMOTE and undersampling techniques.

Our initial step was to tackle the dataset's primary issue: the extreme class imbalance. Models found it difficult to detect fraud because fraudulent transactions represented a mere 0.17% of all transactions. We created synthetic fraud samples with SMOTE and balanced the dataset through undersampling. This methodology improved the models' recognition of fraud patterns. When analyzing business data containing rare events, this method is a standard and successful strategy.

b. Implement XGBoost and ensemble methods to enhance fraud detection capabilities within models.

According to our analysis, XGBoost achieved the highest F1 score. Multiple weak learners work together using ensemble methods like XGBoost to achieve strong performance, particularly when dealing with imbalanced data. These models achieve high accuracy and speed while being widely adopted in industry to deliver practical and impactful solutions.

c. Create real-time fraud detection systems using models that balance light-weight design and scalability.

To ensure fraud is detected immediately, we suggest utilizing lightweight models within real-time platforms such as Kafka or Spark Streaming. Banks can analyze transactions instantly, which helps them minimize losses while protecting customers. This solution proves practical and scalable when implemented in real-world applications.

d. Establish detection thresholds that adjust according to temporal and monetary transaction patterns.

Our data analysis indicates that fraudulent activities reach their highest points between 12 PM and 6 PM and primarily involve small sums of money. Our approach recommends implementing dynamic thresholds that change model sensitivity according to time and transaction amount for more effective fraud detection with reduced false positives. This approach provides an uncomplicated yet effective method to customize detection.

e. Use behavioral analysis to identify abnormal user activity patterns.

The system monitors user behavior by identifying patterns such as spending habits and transaction locations to determine what is typical for each user. When user activity breaks typical patterns, such as making a large purchase in another country, the system flags it. This technique provides enhanced security beyond the model's foundational learning capabilities.

f. SHAP and LIME are suggested tools to enable transparency through model decision explanations.

We struggle to explain predictions because our features come from anonymized PCA values. SHAP and LIME could help us understand which model features impacted the decision-making process. The system gains transparency while supporting compliance requirements and audibility, which builds users' trust.

g. Regularly update models by incorporating input from verified cases.

Our models must adapt because fraud techniques continue to evolve. Our recommendation involves setting up a feedback loop to retrain the model with verified fraud and legitimate cases. This method, representing a leading practice in contemporary business analytics, retains the system's adaptability and relevance.

h. Implement manual reviews for uncertain cases to enhance accuracy levels.

Models have limitations that necessitate human analyst review for uncertain or borderline cases. The hybrid approach enhances transaction accuracy and builds customer trust, particularly for high-risk or ambiguous transactions.

6. Implementation Plan

To ensure the smooth execution of our proposed fraud detection framework, we developed a **Plan of Action and Milestones (POAM)**. This plan outlines specific tasks, responsible teams, timelines, and expected outcomes. The implementation requires collaboration across multiple departments, including:

- **Machine Learning Team** – responsible for model development, tuning, and integration
- **Business Intelligence & Analytics Team** – oversees data analysis, feature design, and reporting
- **Development Operations Team** – manages infrastructure, APIs, and real-time fraud alert systems

Attached below is the POAM sample:

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	

Potential Challenges & Mitigation Strategies

Implementing this POAM may involve the following key risks:

a. **Model Threshold Tuning Issues:**
Deploying the XGBoost model with an improperly tuned threshold could lead to a spike in **false positives**, overwhelming the alert system and causing operational delays.

Mitigation:

- Revalidate thresholds with updated validation data
- Introduce severity-based filtering to prioritize critical alerts

b. Model Stability Risks:
Combining anomaly detection with the XGBoost classifier may lead to unpredictable model behavior or system instability.

Mitigation:

- Conduct robust scenario-based testing
- Validate against unseen data (test sets not included during model training) before deployment

This implementation plan ensures that our fraud detection system can be deployed efficiently and with minimal disruption by defining clear tasks, assigning responsibilities, and preparing for known risks.

7. Conclusion

The Kaggle Credit Card Fraud Detection dataset posed a significant challenge due to its extreme class imbalance - only 492 out of 284,807 transactions were fraudulent. This imbalance led to many false positives, making accurate fraud detection difficult using traditional models.

We applied techniques such as SMOTE, threshold tuning, and under-sampling to address this. Among all models tested, XGBoost delivered the best performance by effectively balancing precision and recall in a complex and imbalanced setting.

We recognize that fraud detection is not just a technical problem but also a strategic and operational one. Real-time detection requires robust planning, system integration, and continuous model updates. Our POAM (Plan of Action and Milestones) outlines a structured approach for deploying fraud scoring APIs, anomaly detection models, and performance reporting systems.

Ultimately, sustainability depends on ongoing monitoring, retraining new data, and close collaboration between analytics, development, and operations teams to adapt to evolving fraud tactics.