

Interim Report: Fraud Detection System Analysis

Author: Nazrawi

Date: December 21, 2025

Project: Improved detection of fraud cases for e-commerce and bank transactions

1. Understanding and Defining the Business Objective

Business Context:

Adey Innovations Inc. operates in the high-stakes fintech sector, handling both e-commerce and banking transactions. The core business problem is the financial loss and reputational damage caused by fraudulent activities.

Primary Objective:

To develop a robust fraud detection system that minimizes financial loss (False Negatives) while preserving the user experience by not blocking legitimate customers (False Positives).

Key Challenges:

- **Class Imbalance:** Fraudulent transactions represent a tiny fraction of total activity (often < 0.2%), making standard accuracy metrics misleading.
- **Evolving Patterns:** Fraudsters constantly change tactics (e.g., changing IP locations, quick bot signups).
- **Geolocation Complexity:** E-commerce data provides IP addresses, which must be mapped to physical locations to identify cross-border anomalies.

2. Discussion of Completed Work and Initial Analysis

2.1 Data Cleaning and Geolocation Integration (Task 1a)

We successfully processed two distinct datasets: Fraud_Data.csv (E-commerce) and creditcard.csv (Banking).

- **IP Mapping Strategy:** We implemented a specialized Python module (src.data_processing) to convert string-based IP addresses into integer format. Using the merge_asof algorithm, we mapped 150,000+ transactions to their corresponding countries based on the IPAddress_to_Country range dataset.
- **Timestamp Conversion:** signup_time and purchase_time were converted to datetime objects to enable temporal feature extraction.

2.2 Exploratory Data Analysis (EDA)

Insight 1: Geographic Fraud Patterns

By analyzing the country of origin for e-commerce transactions, we identified specific regions with higher fraud frequencies.

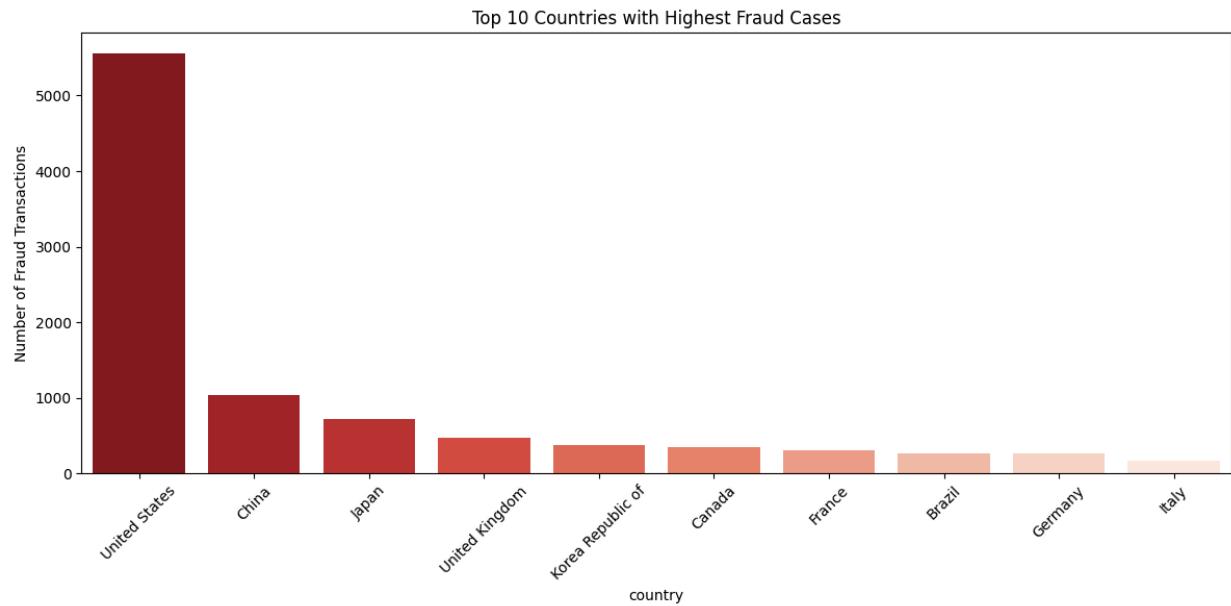


Figure 1: Top 10 countries by number of confirmed fraud cases.

Insight 2: The "Bot" Behavior (Time Since Signup)

A critical pattern emerged when analyzing the time difference between a user signing up and making their first purchase. As shown below, fraudulent transactions (Class 1) often occur almost instantly after signup, indicating automated bot activity.

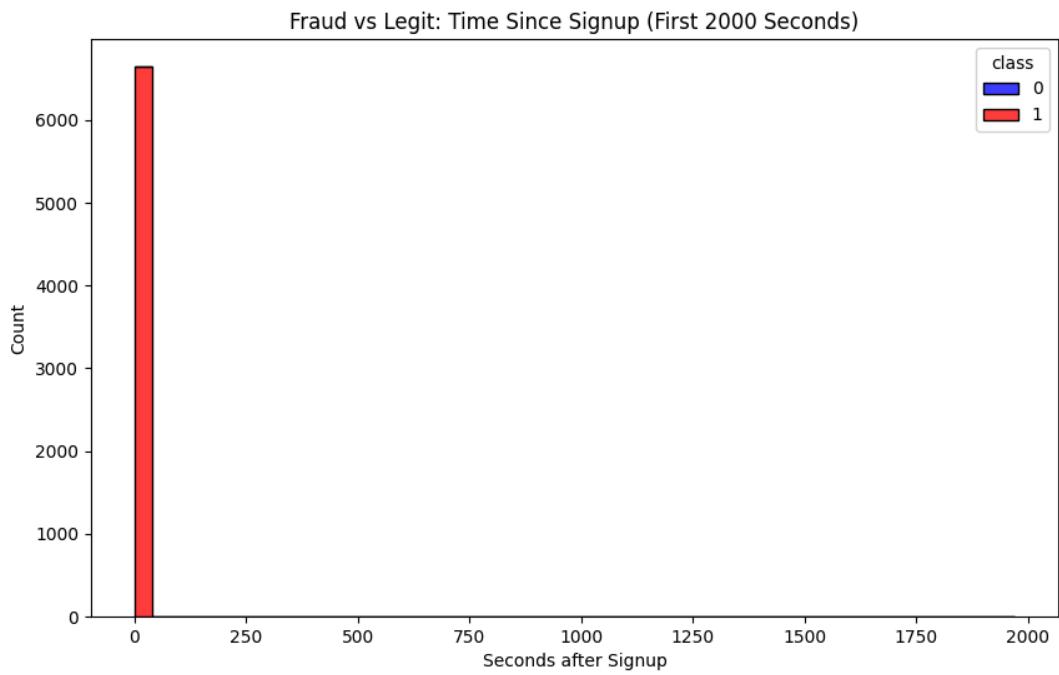


Figure 2: Distribution of time between signup and purchase. Note the spike at 0 for fraud cases.

Insight 3: Banking Data Imbalance

The banking dataset confirms the extreme class imbalance typical of fraud detection.

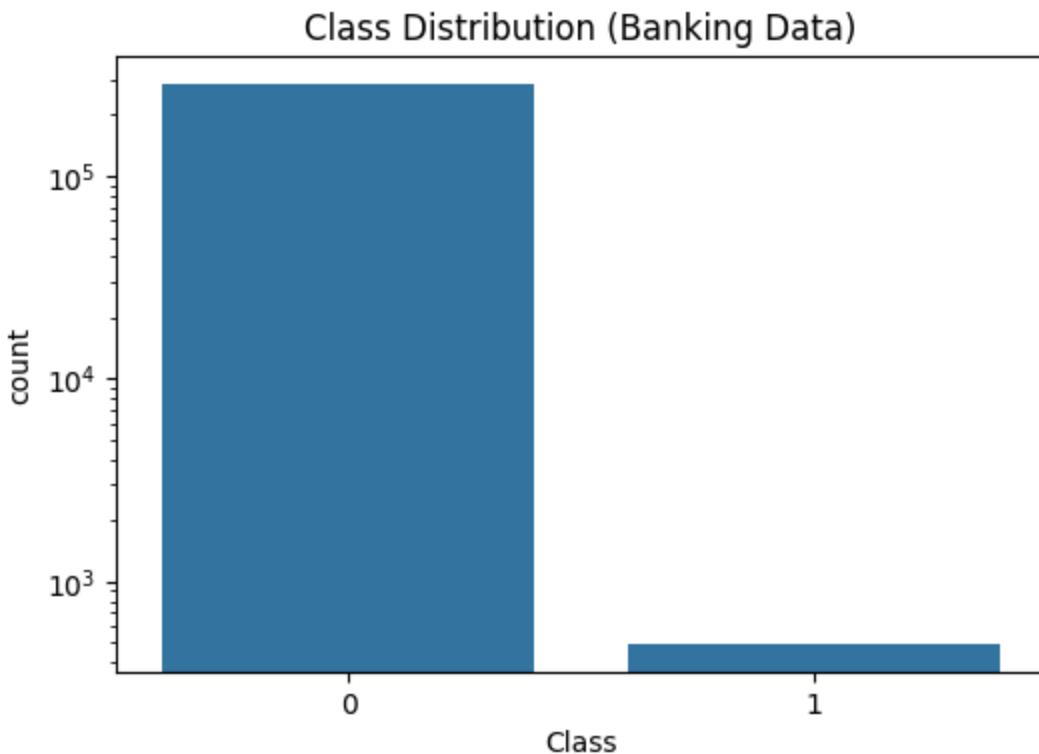


Figure 3: Class distribution in banking data (Log Scale).

2.3 Feature Engineering (Task 1b)

To prepare for machine learning, we engineered the following features:

1. **time_since_signup (Seconds)**: Derived to capture the "quick-strike" bot behavior observed in EDA.
2. **hour_of_day & day_of_week**: Extracted to capture temporal patterns (e.g., fraud occurring during non-business hours).
3. **device_transaction_count**: We calculated how many users share the same device ID. High reuse of a single device ID across multiple accounts was found to be a strong indicator of fraud rings.

2.4 Data Transformation and Class Imbalance Handling (Task 1b)

To prepare the data for machine learning models, we performed the following transformations:

- **Feature Engineering**: We created a device_transaction_count feature to capture the velocity of transactions per device, as high usage often correlates with fraud rings.
- **Encoding**: Categorical variables (source, browser, sex, country) were transformed using **One-Hot Encoding**. This expands the feature space but allows the model to interpret categorical variance without assuming ordinal relationships.

- **Scaling:** Numerical features (`purchase_value`, `time_since_signup`, `device_transaction_count`, etc.) were normalized using **StandardScaler** to ensure that features with large ranges do not dominate the gradient descent process.

Addressing Class Imbalance (SMOTE):

We applied SMOTE (Synthetic Minority Over-sampling Technique) to the training data **only**, to prevent data leakage.

- **Before Resampling:** The training set had a severe imbalance with **93,502** legitimate transactions (Class 0) and only **9,814** fraudulent ones (Class 1).
- **After Resampling:** We successfully balanced the classes, resulting in **93,502** samples for both Class 0 and Class 1. This ensures the model sees an equal representation of both classes during training, preventing it from biasing heavily toward the majority class.

3. Next Steps and Key Areas of Focus

With the data processed and key patterns identified, the next phase (Interim 2) will focus on Model Building.

1. Handling Class Imbalance:

We will employ **SMOTE (Synthetic Minority Over-sampling Technique)** on the training data to balance the classes. This prevents the model from simply predicting "Not Fraud" for every case.

2. Model Selection:

We will train and compare three specific models:

- **Logistic Regression:** As a baseline for interpretability.
- **Random Forest:** To capture non-linear relationships and interactions between features (e.g., Country + Time).
- **XGBoost:** For high performance on tabular data.

3. Evaluation Strategy:

We will strictly avoid "Accuracy" as a metric. Instead, we will optimize for **AUC-PR (Area Under the Precision-Recall Curve)** and **F1-Score**, which provide a true measure of performance on imbalanced data.