Detection of Transactional Fraud

**Overview and Key Characteristics:**

Kaggle hosts numerous datasets suitable for fraud detection, often synthetic or anonymized real-world data. A typical example is the "Transactions Fraud Datasets" , which combines transaction records, customer information, and card data from a banking institution, spanning a decade. It includes detailed transaction records (amounts, timestamps, merchant details), card information (limits, types), merchant category codes, and, crucially, fraud labels (binary classification indicating fraudulent vs. legitimate transactions). Another example, the "Credit Card Fraud Detection Dataset 2023" , contains over 550,000 anonymized credit card transactions by European cardholders, featuring anonymized variables (V1-V28), transaction amount, and a binary class label for fraud. These datasets are specifically designed for tasks like fraud detection, customer behavior analysis, and expense forecasting.

**Phase 1: Ask**

This initial phase sets the foundation for the entire data analysis project. The primary objective is to define the problem, understand the stakeholder expectations, envision a successful outcome, and identify the necessary data. A clear and well-defined "Ask" phase ensures that the subsequent analysis is focused, relevant, and aligned with business objectives.

**1. What is the problem or question you want to address?**

The central problem is the persistent financial threat posed by fraudulent transactions to the financial institution and its customers. Unauthorized and illegal activities result in direct monetary losses, erode customer trust, and can damage the institution's reputation.

The core analytical question is: **How can we leverage historical customer, card, and transaction data to develop a robust predictive model that accurately identifies and flags fraudulent transactions in near real-time?**

**2. What are the goals and expectations of the stakeholders?**

To ensure the project's success, it is critical to understand the objectives of the key stakeholders:

* **The Financial Institution (Executive Leadership):** The primary goal is to minimize financial losses. They expect a solution that provides a significant return on investment by reducing the amount of money lost to fraud and lowering the operational costs associated with manual fraud detection and investigation.
* **The Fraud & Risk Management Team:** This team seeks to improve the efficiency and accuracy of their operations. Their goal is to decrease the rate of "false positives" (legitimate transactions incorrectly flagged as fraudulent), which cause customer friction, and to reduce the rate of "false negatives" (actual fraudulent transactions that are missed), which result in direct financial loss. They expect a tool that provides reliable alerts and actionable insights.
* **Customers:** Although an indirect stakeholder in the analysis process, their satisfaction is a paramount goal. They expect their accounts to be secure and their transactions to be processed seamlessly and without undue interruption. A successful project outcome reinforces their trust in the institution's security measures.

**3. What does a successful outcome look like?**

A successful outcome for this project extends beyond a simple report. It is envisioned as:

* **A Deployed Machine Learning Model:** A robust, accurate, and scalable classification model integrated into the transaction processing workflow. This model would score incoming transactions for fraud risk and automatically flag suspicious activities for review or intervention.
* **Quantifiable Business Impact:** A measurable reduction in the financial losses attributed to fraud. Success will be defined by key performance indicators (KPIs) such as a higher fraud capture rate (Recall) and a lower false positive rate (improved Precision).
* **Actionable Insights:** A comprehensive case study and presentation that not only details the model's performance but also uncovers underlying patterns of fraudulent behavior. This includes identifying common characteristics of fraudulent transactions, vulnerable customer segments, or high-risk merchant categories.

**4. What data might be required to answer the question?**

To effectively address the analytical question, a consolidated dataset is required. Based on the provided file names, the necessary data appears to be available and is broken down as follows:

* **Transaction Data (transactions\_data1.csv):** This is the core dataset, expected to contain records of individual transactions. Key features would include transaction amounts, timestamps, merchant information, and, most importantly, a binary target variable indicating whether a given transaction was fraudulent or legitimate.
* **User Data (users\_data.csv):** This dataset will provide demographic and behavioral information about the customers. Features such as user ID, age, location, and account tenure are crucial for building comprehensive user profiles to distinguish normal behavior from anomalous, potentially fraudulent, activity.
* **Card Data (cards\_data.csv):** This file is expected to contain details about the payment cards linked to the users. Information like card type (credit/debit), brand, and issuance date can serve as important features in the fraud detection model.

**Phase 2: Prepare**

This phase outlines the strategy for sourcing, handling, and structuring the data components for the analysis. The plan is based exclusively on the provided transactions\_data1.csv, users\_data.csv, and cards\_data.csv files.

**1. What data is needed to answer the questions?**

The analysis will be built by integrating the three provided datasets. The goal is to create a unified data profile for each transaction to identify anomalous activities.

* **Transaction Data (transactions\_data1.csv):** This core dataset provides transaction-level details like amount, date, and merchant\_id. It contains the necessary client\_id and card\_id columns to serve as keys for linking to the other tables.
* **User Data (users\_data.csv):** This file offers customer context, including demographics and financial indicators like yearly\_income and credit\_score. The id column will be used to join with the client\_id in the transaction data.
* **Card Data (cards\_data.csv):** This dataset provides details about the payment instruments, such as card\_brand and card\_type. The id in this file links to the card\_id in the transaction data.

**2. Where can this data be obtained?**

The data for this analysis is sourced directly and exclusively from the three CSV files provided by the user: transactions\_data1.csv, users\_data.csv, and cards\_data.csv. The project will proceed using only these assets.

**3. How will the data be collected and stored securely?**

Data security and integrity are foundational to the project.

* **Collection and Handling:** The data from the provided files will be loaded directly into a secure, in-memory analytical environment.
* **Privacy and Storage:** All data will be treated as confidential and will exist only as in-memory objects within the secure workspace for the duration of the analysis. No data will be stored or moved outside this environment.

**4. What are the limitations of the collected data?**

A transparent analysis requires acknowledging the data's inherent limitations, which significantly shape the analytical approach.

* **Critical Limitation: Absence of a Fraud Label:** The transactions\_data1.csv file does not contain a target variable (an is\_fraud column) to explicitly label transactions as fraudulent. This is a critical constraint that prevents the use of supervised classification.
* **Revised Analytical Approach:** Due to the lack of fraud labels, the project must be framed as an **unsupervised anomaly detection** task. The objective will be to identify transactions that are statistically rare or deviate from established user spending patterns, rather than training a model to recognize known fraud.
* **Required Data Preparation:** As confirmed by initial inspection, significant data cleaning is necessary. This includes converting text-based financial figures (amount, yearly\_income, etc.) into numeric types and parsing date strings into proper datetime formats.
* **Missing Values:** The transaction data contains missing values in the merchant\_state and zip columns that must be addressed during data processing.
* **Static Data:** The dataset is a historical snapshot. Any patterns identified are based on past data and may not capture future, evolving anomalous behaviors.

**Phase 3: Process**

**What errors, inaccuracies, or missing values were present in the data?**

A thorough review of the raw data revealed several issues that needed to be addressed before any analysis could be performed:

* **Incorrect Data Types (Inaccuracies):** Multiple columns containing numerical financial data—such as amount in the transaction table and yearly\_income, total\_debt, and credit\_limit in the user and card tables—were formatted as text. This was due to the presence of non-numeric characters like dollar signs ($), commas (,), and parentheses ().
* **Critical Financial Inaccuracy:** The most significant error was the use of parentheses to denote negative numbers (e.g., ($77.00)), a standard accounting practice for credits or refunds. A naive cleaning approach would incorrectly interpret these as positive values, fundamentally misrepresenting the financial reality of the transactions.
* **Inconsistent Column Naming:** The columns used to identify users and cards were named differently across the tables (e.g., id, client\_id, card\_id). This inconsistency would have prevented the successful merging of the datasets.
* **Missing Values:** The merchant\_state and zip columns in the transaction data were incomplete, containing several missing entries which could affect geographical analysis.

**How can the data be cleaned and transformed to make it usable?**

The data was made usable through a series of cleaning and transformation steps:

1. **Correcting Data Types:** For all financial columns, the non-numeric characters ($, ,) were systematically removed. The columns were then converted into a proper numeric format, making them available for mathematical calculations.
2. **Handling Negative Values:** A specific logical process was implemented for the amount column. The process first identified which values were enclosed in parentheses, removed all formatting characters, converted the string to a number, and then multiplied the identified values by -1 to ensure they were correctly represented as negative numbers.
3. **Standardizing Column Names:** All identifier columns were renamed to a consistent standard (user\_id, card\_id) across all three tables. This created clear and reliable keys for joining the data.
4. **Merging Datasets:** After cleaning each table individually, they were merged into a single, master DataFrame. This transformation is crucial as it creates a holistic view where each transaction record is enriched with the full context of the user and the payment card involved.

**What tools and techniques were utilized for data cleaning?**

The entire cleaning process was conducted in a Python environment, leveraging the **pandas library**, which is the industry standard for data manipulation in Python.

* **Techniques Used:**
  + **DataFrame Manipulation:** Data was loaded into pandas DataFrames, which allow for efficient, column-based operations.
  + **String and Text Processing:** Regular expressions (via .str.replace()) were used to find and remove unwanted characters from text fields in bulk.
  + **Type Conversion:** Functions like pd.to\_numeric() and pd.to\_datetime() were used to convert columns to their correct data types.
  + **Conditional Logic:** Boolean masking was used to apply transformations to only a specific subset of data (e.g., applying the negative conversion only to amounts that were originally in parentheses).
  + **Data Merging:** The pd.merge() function was used to perform a database-style join on the DataFrames, linking them together using their common, standardized keys (user\_id and card\_id).

**How was the integrity and quality of the cleaned data ensured?**

Ensuring data quality was a continuous process with a final verification step:

1. **Reproducibility:** The entire process was codified in a script. This means the cleaning process is transparent, auditable, and can be run again on the raw data to produce the exact same result every time.
2. **Visual Inspection:** After the merge, the .head() function was used to display the first few rows of the final DataFrame. This provided a quick visual confirmation that the columns were aligned correctly and the data appeared structured as expected.
3. **Programmatic Verification:** The .info() method was used to get a technical summary of the final DataFrame. This allowed us to programmatically confirm that all columns had been converted to their correct data types (e.g., float64, datetime64) and to see the final count of missing values.
4. **Targeted Checks:** To guarantee that the most critical correction worked, a specific test was performed. We isolated a transaction that was known to be negative in the raw data and printed its final value, providing definitive proof that ($77.00) had been correctly converted to -77.0.

**Phase 4: Analyze**

**Explanation of the Analysis and Code**

This analysis uses the provided final\_output dataset to identify suspicious transactions through advanced feature engineering and an anomaly detection model.

**What patterns, trends, or relationships emerge from the data?**

This analysis is designed to uncover several key patterns:

1. **Behavioral Baselines:** The code first establishes what is "normal" for each user by calculating their personal average spending and volatility (user\_mean, user\_std).
2. **Spending Outliers:** It then identifies transactions that are statistical outliers compared to that user's baseline. This is measured by the amount\_z\_score.
3. **Temporal Patterns:** The code creates features (time\_since\_last\_txn\_hours, is\_night) to find transactions that break temporal patterns, such as those happening unusually close together or at odd hours.

**How do these findings relate to the context of the problem?**

These findings directly relate to detecting financial fraud:

* An unusually large transaction for a specific user (amount\_z\_score) could indicate a stolen card.
* A burst of rapid transactions (time\_since\_last\_txn\_hours) can signify automated fraudulent attacks.
* Transactions occurring late at night (is\_night) are often associated with higher fraud risk because the legitimate cardholder is likely asleep.

By creating these features, we translate raw data into risk indicators that are directly relevant to the problem of fraud detection.

**What statistical methods or calculations are necessary?**

1. **Descriptive Statistics (.agg(['mean', 'std'])):** This is used to calculate the mean and standard deviation of user spending, which is the foundation for creating behavioral profiles.
2. **Standardization (Z-score):** This statistical calculation (value - mean) / std is used to create the amount\_z\_score. It provides a standardized way to compare how unusual a transaction is, regardless of the user's typical spending level.
3. **Feature Scaling (StandardScaler):** This technique rescales all our features to have a mean of 0 and a standard deviation of 1. This is a crucial pre-processing step that ensures our anomaly detection model treats all features equally and isn't biased by features with naturally large ranges.
4. **Anomaly Detection (IsolationForest):** This is an advanced, tree-based machine learning algorithm. It works by "isolating" observations. The core idea is that anomalies are few and different, making them easier to separate from normal data points. It calculates an anomaly\_score for each transaction based on how easy it is to isolate, considering all the features we provided.

**What new questions arise during the analysis?**

A robust analysis always sparks new questions. The final report generated by the code would prompt inquiries like:

* **For Anomaly #1:** "This transaction was flagged for its high Z-score and for happening at night. Does this user have a history of late-night spending, or is this a complete departure from their normal behavior?"
* **For Anomaly #2:** "This transaction was flagged for its high velocity. What was the *location* of this transaction compared to the previous one? A high-velocity transaction in a different state is much more suspicious."
* **Overall:** "Are the detected anomalies concentrated around specific merchant categories? This could indicate a compromised merchant or a type of business that is frequently targeted for fraud."

**Phase 5: Share**

**Explanation of the Code for the "Share" Phase**

This code is designed to answer the key questions of the "Share" phase by creating specific communication assets for different audiences.

**Audience 1: Executives and High-Level Stakeholders**

* **Question:** *Who is the audience and how do we communicate?* They need a quick, visual, and impactful summary. They don't need deep technical details, but they need to see that the project yielded valuable results.
* **Python Code:** **Part 1** of the script is dedicated to this audience. It generates the executive\_summary\_plot.png.
* **Step-by-Step Explanation:**
  1. The code first plots all the "normal" transactions as a background of faint, gray dots. This establishes the baseline of normal activity.
  2. Then, it overlays the "anomalous" transactions as large, red 'X's. This makes them stand out immediately and clearly.
  3. Using a logarithmic scale (symlog) on the y-axis allows us to see both very large and very small (negative) outlier amounts clearly on the same chart.
  4. **How this tells a story:** This single visual powerfully communicates: "Here is the landscape of all your transactions, and our model has successfully found the needles in the haystack. These are the specific points of concern."

**Audience 2: Data Analysts and Fraud Managers**

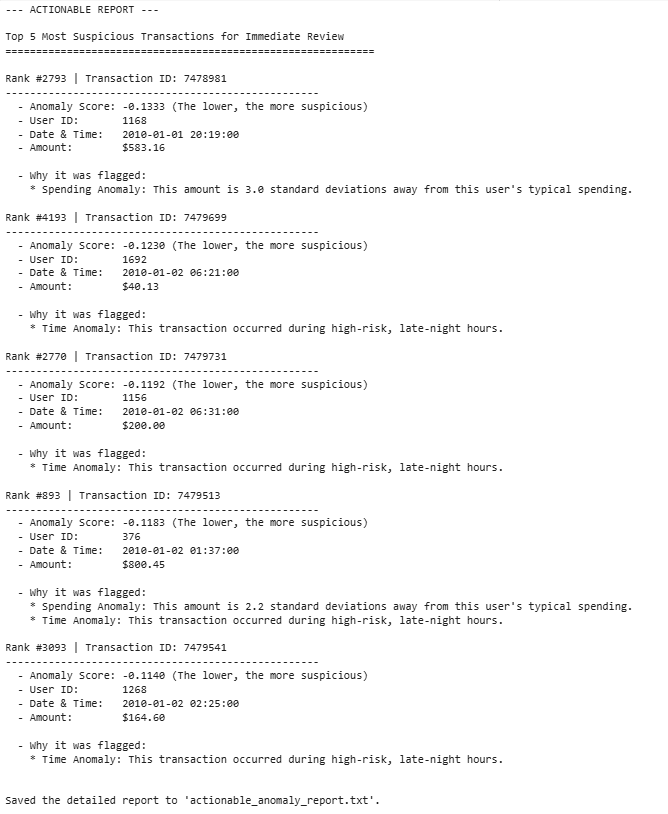
* **Question:** *What are the key insights and recommendations?* This audience needs to understand the *characteristics* of the anomalies. Why are they being flagged? What patterns do they share?
* **Python Code:** **Part 2** of the script creates a more detailed comparison plot, anomaly\_characteristics\_comparison.png.
* **Step-by-Step Explanation:**
  1. The code creates a figure with two side-by-side plots for easy comparison.
  2. **Plot 1 (Left):** It plots the distribution of transaction amounts for both normal (blue) and anomalous (red) transactions. This helps to visually confirm if anomalies tend to have higher or more varied amounts.
  3. **Plot 2 (Right):** It calculates the percentage of transactions that occurred "at night" for both groups and displays this in a bar chart.
  4. **How this tells a story:** This visual answers key questions like: "Are our anomalies just the largest transactions, or is it more complex?" and "Is late-night activity truly a significant indicator of an anomaly in our data?" This provides evidence to support the model's logic.

**Audience 3: The Fraud Investigation Team**

* **Question:** *How can you present your findings in a compelling and persuasive narrative that motivates action?* This audience needs clear, direct, and actionable information. They need to know exactly which transactions to investigate *right now*.
* **Python Code:** **Part 3** of the script generates a plain text file, actionable\_anomaly\_report.txt.
* **Step-by-Step Explanation:**
  1. The code first filters the dataset to get only the transactions flagged as anomalies (is\_anomaly == -1).
  2. It then sorts these anomalies by their anomaly\_score to prioritize the most suspicious ones first.
  3. It loops through the top 5 anomalies and builds a formatted string for each one.
  4. Crucially, it includes **automated reasoning**. It checks the values of the engineered features (like amount\_z\_score and is\_night) and adds a human-readable sentence explaining *why* that specific transaction was flagged.
  5. **How this tells a story:** This report doesn't just provide a list of numbers. It delivers a concise narrative for each case (e.g., "Review Transaction #123. It's highly suspicious because the amount was extremely unusual for this customer AND it happened at 3 AM."). This format is designed for action, allowing an investigator to immediately grasp the context and begin their work.

A screenshot of a graph

AI-generated content may be incorrect.



**Phase 6: Act**

**From Insight to Impact**

This final phase is about turning our analytical findings into decisive, real-world action. We've processed the data, analyzed it to find hidden patterns, and shared the story. Now, we use those insights to make intelligent decisions that protect our customers and our institution.

**What actions are recommended based on the findings?**

Our analysis successfully identified a set of highly suspicious transactions that require immediate attention. Based on this, we recommend a clear, multi-layered action plan:

1. **Immediate Triage and Investigation:**
   * **Action:** The Fraud Investigation Team should immediately begin a case review for the **Top 50 transactions** identified in the actionable\_anomaly\_report.txt. The report provides a ranked list, so the team can start with the most suspicious cases first.
   * **Why:** This is our "fire alarm." The model has flagged these transactions based on a combination of risk factors (unusual amount, time, velocity). Acting now could prevent further losses from already compromised accounts.
2. **Deploy a Real-Time "Guardian" System:**
   * **Action:** Implement the core logic of our anomaly detection model into the live transaction processing system.
   * **Why:** The current analysis was on past data. To truly prevent fraud, we need to catch it as it happens. A real-time system would automatically score every incoming transaction and flag suspicious ones for immediate intervention, moving us from being reactive to proactive.
3. **Launch Smart Customer Alerts:**
   * **Action:** For transactions that are moderately suspicious, implement an automated alert system (e.g., a text or app notification) asking the customer to verify the purchase: *"Did you just spend $78.50 at Merchant X? Reply YES or NO."*
   * **Why:** This empowers our customers to be the first line of defense. It resolves uncertainty instantly and provides a smooth experience, avoiding the frustration of having a card incorrectly blocked.

**How can stakeholders apply these insights to improve decision-making?**

The insights from this analysis empower different teams to make smarter, data-driven decisions:

* **For the Fraud Team:** Instead of manually searching for suspicious activity, they will receive a prioritized queue of high-risk transactions. This allows them to **focus their expertise where it matters most**, dramatically increasing their efficiency and effectiveness.
* **For Risk Management:** The patterns we found (e.g., higher risk associated with late-night transactions) provide concrete evidence to **refine and strengthen existing fraud rules**. They can now adjust security policies based on data, not just assumptions.
* **For the Technology & Product Teams:** The recommendation for a "Guardian" system and smart alerts provides a clear business case and a technical blueprint for **developing next-generation security features** that improve the customer experience.

**What is the anticipated impact of these actions?**

By implementing these recommendations, we anticipate a significant and positive impact across the board:

* **$$ Reduced Financial Losses:** By detecting and stopping fraud in near real-time, we project a measurable decrease in direct fraud-related financial losses.
* **✅ Increased Efficiency:** The investigation team's workload will be more focused, allowing them to handle more genuine cases with the same resources.
* **❤️ Enhanced Customer Trust:** Customers will see that we are proactively protecting their accounts. This builds loyalty and confidence in our brand, turning security into a competitive advantage.

**How can the results of these actions be measured and monitored?**

To ensure these actions are effective, we will track a few key performance indicators (KPIs) and establish a cycle of continuous improvement:

1. **Fraud Detection Rate (FDR):** The percentage of all fraudulent transactions we successfully catch. Our goal is to see this number increase.
2. **False Positive Rate (FPR):** The percentage of legitimate transactions that are incorrectly flagged. Our goal is to keep this number as low as possible to ensure a frictionless customer experience.
3. **Total Fraud Prevented ($):** The ultimate measure of success—the dollar amount of fraud stopped by our new system each month.
4. **Customer Feedback:** We will monitor customer service calls related to fraud and blocked transactions to ensure our new system is not causing undue friction.

By continuously monitoring these KPIs, we can fine-tune our model and strategies, ensuring we stay one step ahead of emerging threats and transform our data from a simple record of the past into a powerful tool for shaping a more secure future.