# Synthetic Dataset Generation Logic

**Dataset Generation Logic Documentation**

**Overview**

This document describes the logic for generating a synthetic credit card fraud dataset. The system simulates customer transactions, merchant profiles, and various types of fraudulent activities to generate a realistic mix of legitimate and fraudulent transactions, simulating real-world fraud patterns for training fraud detection models.

**Core Components of Python Scripts**

**Utility Class (**Utility**)**

Handles helper functions for:

- **IP Address Generation**: Creates random IPv4 addresses tied to country codes.
- **Date/Time Operations**: Converts timestamps, adjusts for holidays/weekends.
- **File Operations**: Exports data to CSV.

**Customer Class (**Customer**)**

Generates customer profiles with:

- **Income Levels**: Low, medium, high (affects transaction behavior).
- **Location**: Random (x, y) coordinates.
- **Transaction Behavior**: Mean amount, frequency, online/offline ratio.
- **Credit Cards**: Randomly assigned based on income.
- **IP Addresses**: Primary (home country) + secondary/tertiary (random countries).

**Merchant Class (**Merchant**)**

Generates merchant profiles with:

- **Categories**: From MCC (Merchant Category Codes).
- **Size**: Small/medium/large (affects POS terminals).
- **Store Credit Cards**: Large grocery stores more likely to issue them.
- **Fake Merchants**: Optionally generated for fraud scenarios.

**Transaction Class (**Transaction**)**

Generates transactions with:

- **Normal Transactions**: Based on customer habits.
- **Fraudulent Transactions**:
    - **CNP (Card Not Present) Testing**: Small transactions to validate stolen cards.
    - **CNP Monetization**: Larger fraudulent purchases after testing.
    - **CP (Card Present) Cloning**: Physical card fraud at unusual locations.

# Fraud logic

**Fraud Generation Overview**

The system simulates two types of fraud: card present (CP) and card not present (CNP)

1. **CNP (Card Not Present)** – Small transactions to validate stolen card details in first phase, then large fraudulent purchases after a certain period of time if testing is successful
2. **CP (Card Present) Cloning** – Use cloned physical card fraud at unusual locations.

Fraudulent transactions are injected into legitimate transactions based on probabilistic rules.

**Key Assumptions**

1. Fraudsters test cards before monetizing.
2. CNP fraud is more common than CP fraud.
3. Fraudsters prefer holidays/weekends.
4. Physical card fraud involves distant POS terminals.

---

**Fraud Lifecycle**

**Step 1: Compromising Customers**

- **Daily Selection**:
  - Each day, compromised_customer_nb_per_day customers are randomly selected.
  - **CNP Fraud**: 2 customers/day.
  - **CP Fraud**: 1 customer/day (rarer).
- **Holiday Influence**:
  - Fraud probability increases during holidays (65% vs. normal 35%).

**Step 2: Fraud Testing Phase (CNP & CP)**

**CNP Testing (Online Fraud)**

- **Behavior**:
  - Small transactions ($10–30) to check if the card is valid.
  - Transactions occur in quick succession (within 10–30 minutes).
- **IP Usage**:
  - 80%: Random IP (mimicking attacker).
  - 20%: Blacklisted IP (known fraudster IPs).

**CP Cloning (Offline Fraud)**

- **Behavior**:
  - Transactions at POS terminals **far from** the customer’s usual location.
  - Uses **physical card-present** transactions.

**Step 3: Monetization Phase (Large Fraud)**

After a **testing phase gap** (default: 30 days for CNP, 1 day for CP), fraudsters execute large transactions:

- **CNP Monetization**:
    - 1 large transaction/day for **14 days**.
    - Amounts up to **$3,500** (3x normal spending).
- **CP Monetization**:
    - Shorter window (**7 days**).
    - Also at distant POS terminals.

---

### Fraud Transaction Mechanics

#### Fraudulent Transaction Generation

For each compromised customer:

1. **Determine Fraud Type** (cnp_testing, cnp_monetization, cp_cloning).
2. **Generate Transactions**:
    - **If** cnp_testing:
        - Small, rapid-fire transactions.
        - Always online (card not present).
    - **If** cnp_monetization:
        - Reuses the **same card** from testing.
        - Large amounts (scaled up from normal spending).
    - **If** cp_cloning:
        - Uses far-away POS terminals (unusual location).
3. **Adjust for Seasonality**:
    - **65% of frauds** are shifted to holiday periods.
    - **Weekend Bias**: 30% shifted to weekends.

#### Fake Merchants (Future Implementation)

- Used for QR code scams.
- Transactions routed through fake merchant IDs.

---

# Generation of Dataset Output

**Data Generation Workflow**

**Step 1: Generate Customers & Merchants**

- **Customers**: n_customer profiles with randomized spending habits.
- **Merchants**: n_merchant profiles with MCC categories.

**Step 2: Simulate Transactions**

- **Normal Transactions**:
  - Generated daily per customer (Poisson-distributed counts).
  - Adjusted for weekends (20% shifted to weekends).
- **Fraudulent Transactions**:
  - **Testing**: 3–6 small transactions/day.
  - **Monetization**: 1 large transaction/day for 14 days (CNP) or 7 days (CP).

**Step 3: Post-Processing**

- **Holiday Adjustments**: 65% of frauds shifted to holiday periods.
- **Time Normalization**: Ensures transactions stay within the n_days period.

**Output Dataset Structure**

| **Column** | **Description** |
|---|---|
| transaction_date | Timestamp of transaction |
| customer_id | Unique customer identifier |
| amount | Transaction amount |
| merchant_id | Merchant ID |
| pos_id | POS terminal ID (NULL for online) |
| IP_address | IP used (NULL for in-person) |
| type_of_credit_card_used | Card category |
| card_present_or_not | card present / card not present |
| is_fraud | 1 (fraud) / 0 (legitimate) |

# Future Roadmap

1. **Dynamic Fraud Patterns**:
   - Geo-velocity checks (impossible travel detection).
2. **More Fraud Types**:
   - Account takeover
   - Fake merchant QR code scams.
3. **Code Optimization**:
   - Class structure to be optimized.
   - Replace iterrows with vectorized operations
4. **Enhanced Fraud Logic**:
   - Dynamic fraud patterns (e.g., geo-velocity checks).
5. **Unit Tests**:
   - Validate transaction/fraud generation functions for future reuse.