Transactional Fraud Detection Report

1. Data Handling & Processing

- Read and write data using Pandas

  -> Dataset with 594,643 transactions handled using Pandas.

- Handle missing values and data inconsistencies

  -> Missing values were addressed during preprocessing.

- Convert timestamps into useful formats (hour, day of the week)

  -> Hour extracted from timestamps for hourly fraud analysis.

- Extract high-risk locations

  -> Top fraud-prone merchants identified.

2. Fraud Analysis

- Compute fraud rate (%)

  -> Total Frauds: 7200 out of 594,643. Fraud Rate: 1.21%

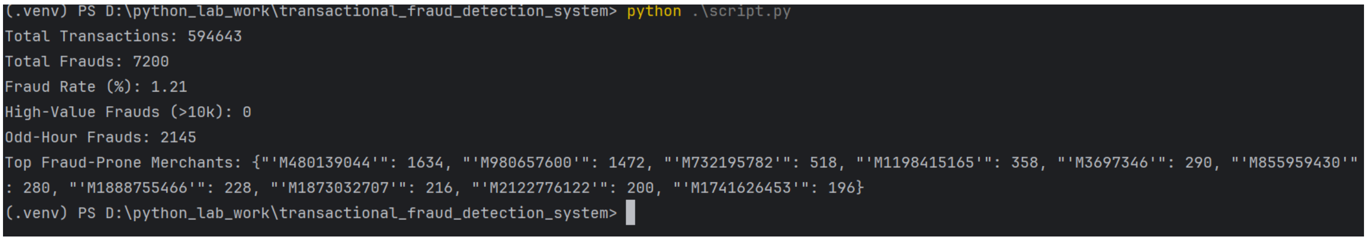
- Identify high-risk transactions

  -> High-Value Frauds (over $10,000): 0

  -> Odd-Hour Frauds: 2,145

- Find the most fraud-prone regions

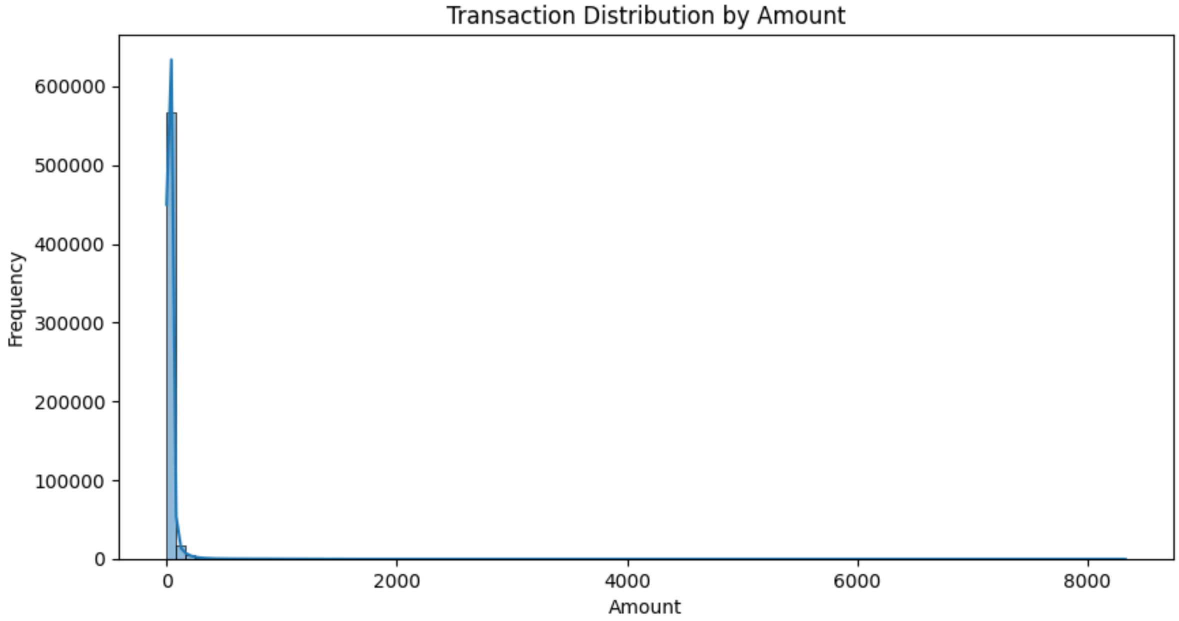
  -> Top Fraud-Prone Merchants listed below:

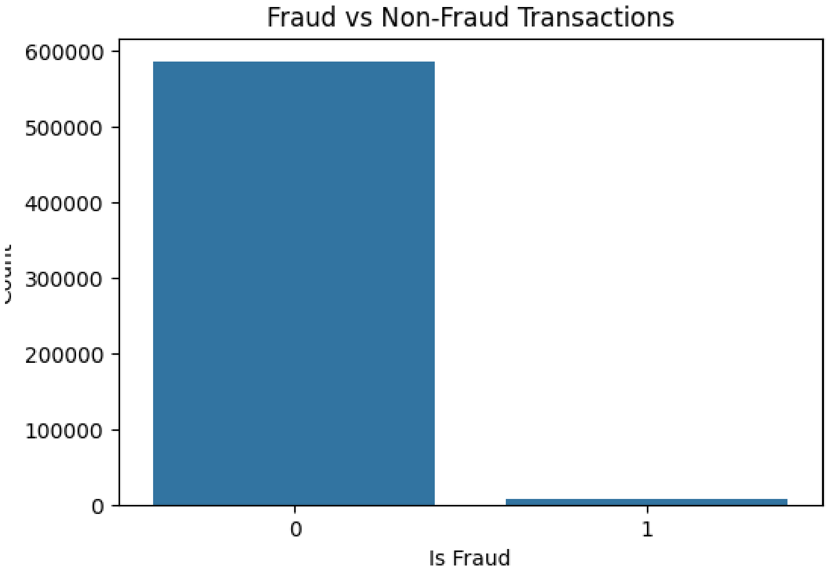


```
(.venv) PS D:\python_lab_work\transactional_fraud_detection_system> python .\script.py
Total Transactions: 594643
Total Frauds: 7200
Fraud Rate (%): 1.21
High-Value Frauds (>10k): 0
Odd-Hour Frauds: 2145
Top Fraud-Prone Merchants: {"'M480139044'": 1634, "'M980657600'": 1472, "'M732195782'": 518, "'M1198415165'": 358, "'M3697346'": 290, "'M855959430'"
: 280, "'M1888755466'": 228, "'M1873032707'": 216, "'M2122776122'": 200, "'M1741626453'": 196}
(.venv) PS D:\python_lab_work\transactional_fraud_detection_system>
```

3. Statistical Insights

Insights were derived through visualizations of transaction amounts and fraud counts.





4. Data Visualization

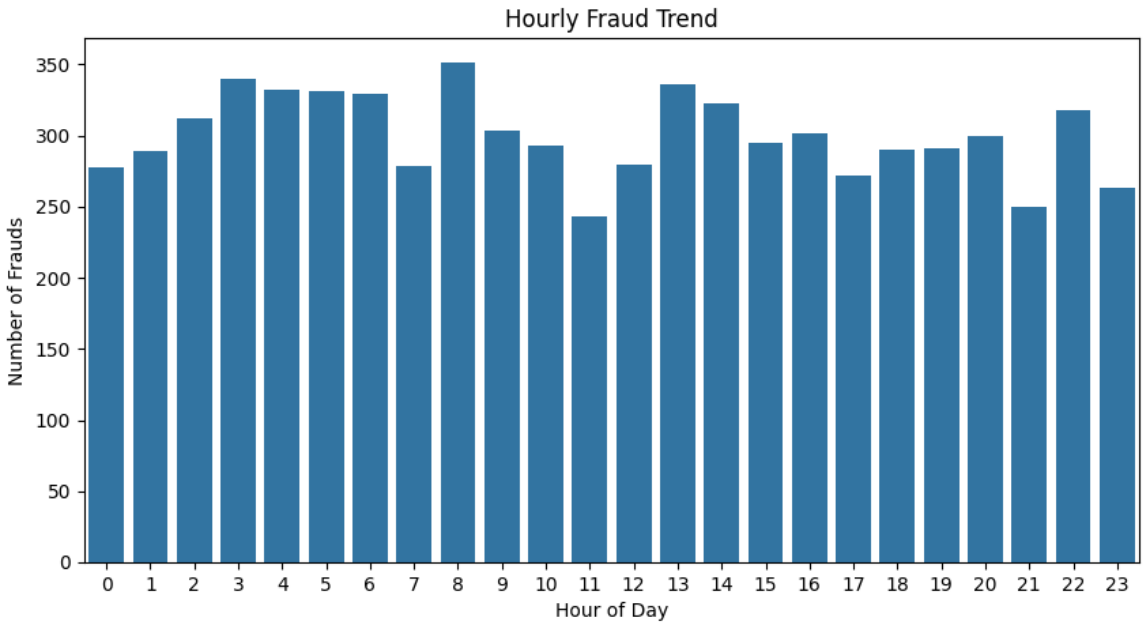
Fraud trends and distributions visualized using Matplotlib and Seaborn:

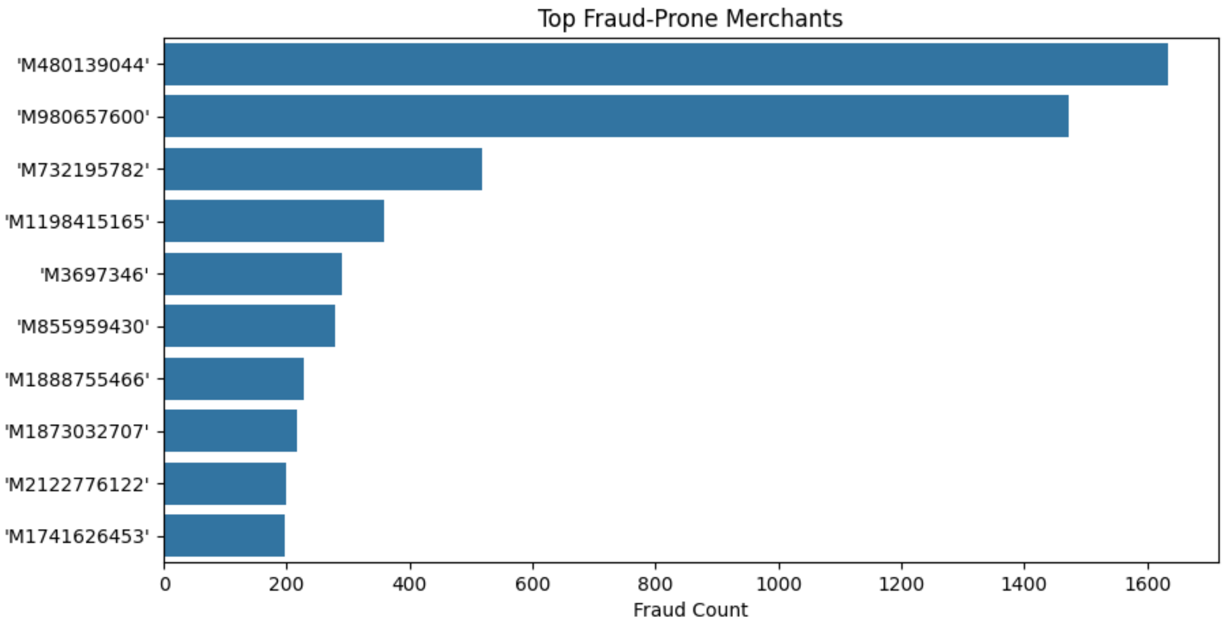
- Transaction Distribution by Amount

- Fraud vs Non-Fraud Transactions

- Hourly Fraud Trend

- Top Fraud-Prone Merchants





5. Anomaly Detection Using Rules

Rules such as odd-hour transactions and high-value thresholds were used for fraud detection.

6. Interactive Dashboard

Interactive visualization and filtering by amount, time, and location is planned for implementation.