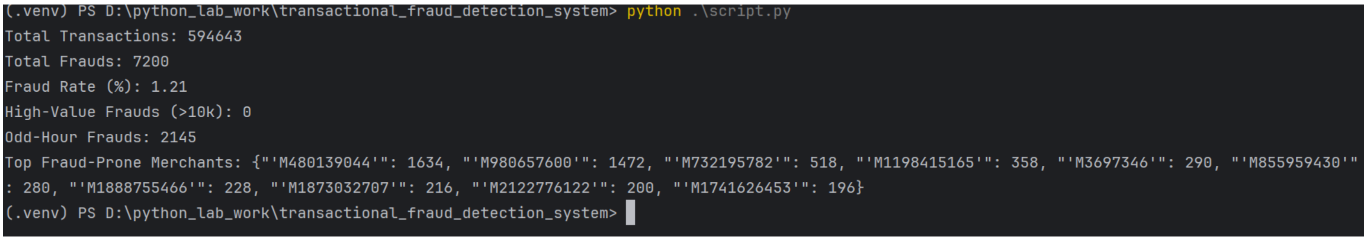
Transactional Fraud Detection Report

Project Objectives

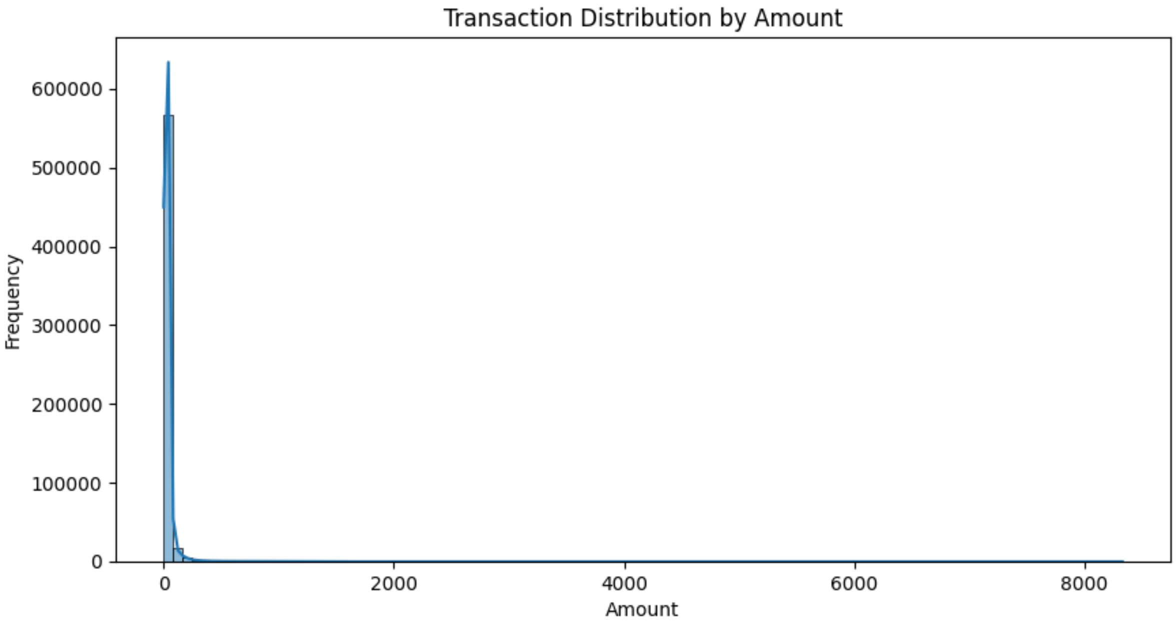
1. Data Handling & Processing: Work with large transaction datasets, clean, and preprocess the data.
   - Read and write data using Pandas
   - Handle missing values and data inconsistencies
   - Convert timestamps into useful formats (hour, day of the week)
   - Extract high-risk locations
2. Fraud Analysis: Identify high-risk transactions based on anomalies in spending patterns.
   - Compute fraud rate (%)
   - Identify high-risk transactions (e.g., over $10,000 or odd hours)
   - Find the most fraud-prone regions
3. Statistical Insights: Compute fraud percentages, transaction distributions, and high-risk regions.
4. Data Visualization: Create fraud detection dashboards using Matplotlib & Seaborn.
   - Transaction Distribution by Amount
   - Fraud vs Non-Fraud Transactions
   - Hourly Fraud Trend
   - Fraud-Prone Locations
5. Anomaly Detection Using Rules: Define fraud detection rules based on transaction behaviour.
6. Interactive Dashboard: Display fraud trends and allow filtering by time, amount, and location.

Console Output

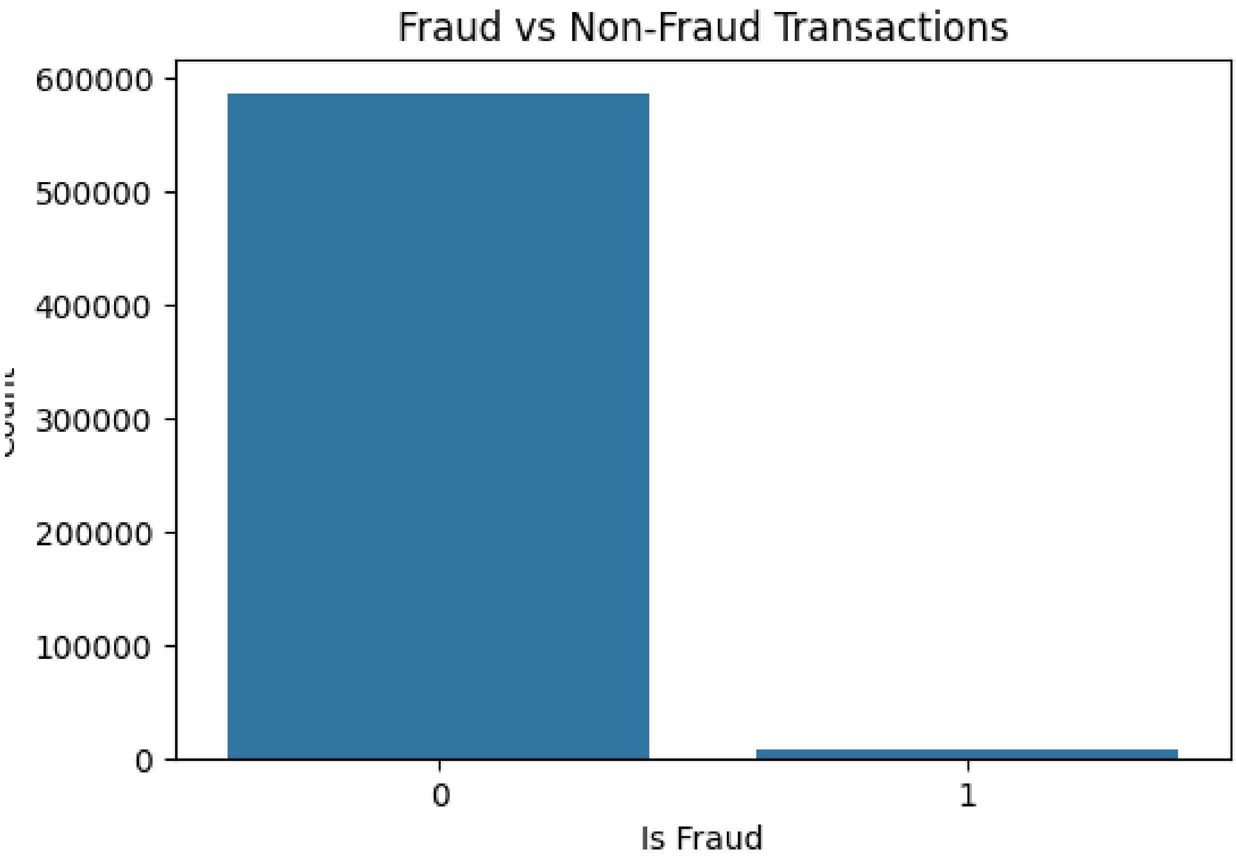


```
(.venv) PS D:\python_lab_work\transactional_fraud_detection_system> python .\script.py
Total Transactions: 594643
Total Frauds: 7200
Fraud Rate (%): 1.21
High-Value Frauds (>10k): 0
Odd-Hour Frauds: 2145
Top Fraud-Prone Merchants: {"'M480139044'": 1634, "'M980657600'": 1472, "'M732195782'": 518,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

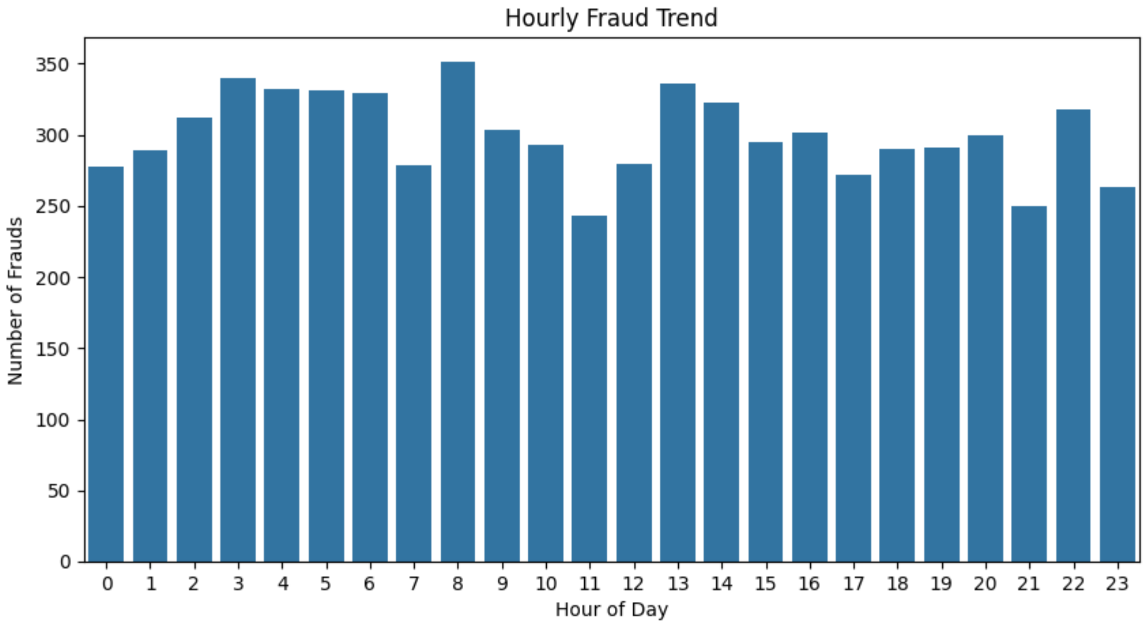
Transaction Distribution by Amount



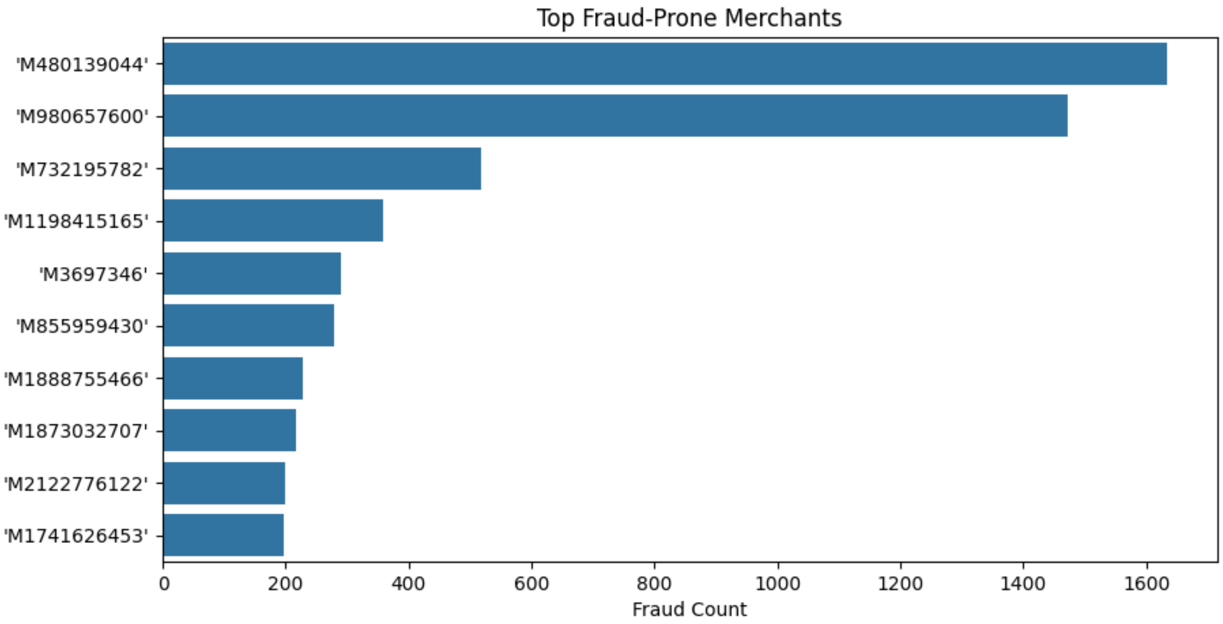
Fraud vs Non-Fraud Transactions



Hourly Fraud Trend



Top Fraud-Prone Merchants


Top Fraud-Prone Merchants
'M480139044'
'M980657600'
'M732195782'
'M1198415165'
'M3697346'
'M855959430'
'M1888755466'
'M1873032707'
'M2122776122'
'M1741626453'
Fraud Count