

# FRAUD DETECTION ANALYSIS — PROJECT REPORT

## 1] Introduction :

Fraud detection is one of the most critical challenges for the financial and banking sector. With the increase in digital transactions, identifying fraudulent activities early can prevent major financial losses.

This project focuses on analyzing transaction data to detect fraud patterns using Python, SQL, and Power BI.

The output includes a fully interactive 3-page dashboard, providing insights on fraud transactions, high-risk users, and fraud trends.

## 2] Objectives:

*The main objectives of this fraud detection project are:*

- Identify legitimate vs fraudulent transactions
- Detect high-risk customers
- Analyze fraud patterns by location, transaction type, and time
- Provide business insights using interactive visual dashboards
- Help decision-makers detect and prevent future fraudulent activities

## 3] Dataset Discription :

**Total Rows: 500 , Total Columns: 13**

*The dataset contains transactional information with the following key fields:*

Column Name	Description
transaction_id	Unique ID for each transaction
customer_id	ID of the customer

Column Name	Description
amount	Amount of transaction
Transaction time	Time of transaction
Transaction type	ATM , Payment, Transfer, Online
Location	City of transaction
Device_type	ATM, Web, Mobile
is_international	Whether transaction is international (yes/no)
is_fraud	1 = Fraud, 0 = Legitimate
hour	Hour at which transaction occurred
day_of_week	Day of the transaction
high_amount_flag	1 = High amount transaction
international_flag	Whether transaction is international

#### 4. Data Cleaning & Preparation (Python) :

Data preprocessing was performed using Python (Pandas, NumPy).

Steps performed:

1. Removed duplicates
2. Handled missing values
3. Converted data types
4. Created new features:
  - *high\_amount\_flag*
  - *Risk Score*

- *international\_flag*

**Example code :**

```
df['hour'] = df['transaction_time'].dt.hour
df['day_of_week'] = df['transaction_time'].dt.day_name()

df['high_amount_flag'] = (df['amount'] > 50000).astype(int)
df['international_flag'] = (df['is_international'] == 'Yes').astype(int)

# Show new columns
df[['transaction_id', 'amount', 'transaction_time', 'hour', 'day_of_week', 'high_amount_flag', 'international_flag']].head()
df.head()
```

Python

## 5. complete Deep fraud analysis.

```
fraud_rate = df['is_fraud'].mean() * 100
print("Fraud Rate (%) :", fraud_rate)

fraud_by_hour = df.groupby('hour')['is_fraud'].mean().sort_values(ascending=False)
fraud_by_hour

fraud_by_day = df.groupby('day_of_week')['is_fraud'].mean().sort_values(ascending=False)
fraud_by_day

fraud_by_type = df.groupby('transaction_type')['is_fraud'].mean().sort_values(ascending=False)
fraud_by_type

fraud_by_location = df.groupby('location')['is_fraud'].mean().sort_values(ascending=False)
fraud_by_location

high_amount_fraud_rate = df[df['high_amount_flag'] == 1]['is_fraud'].mean()
high_amount_fraud_rate

international_fraud_rate = df[df['international_flag'] == 1]['is_fraud'].mean()
international_fraud_rate

df.groupby('is_fraud')['amount'].describe()

df
```

Python

## 6. EDA by using Matplotlib.

### 5] SQL Analysis :

*SQL queries were created to extract important fraud insights.*

**Top 5 high-risk customers:**

```
-- Q12. Top 5 highest amount fraud transactions.
```

```
• SELECT *  
FROM fraud_transaction  
WHERE is_fraud = 1  
ORDER BY amount DESC  
LIMIT 5;
```

	transaction_id	customer_id	amount	transaction_time	transaction_type	location	device_type	is_international	is_fraud	hour
▶	147	1000	99297.18	10/07/2024 07:34	ATM	Delhi	Web	No	1	7
	62	1059	99051.46	12/07/2024 04:16	Payment	Bangalore	Mobile	No	1	4
	377	1045	98997.03	20/06/2024 07:33	Online	Hyderabad	ATM	Yes	1	7
	190	1088	98728.89	15/02/2024 22:30	Payment	Delhi	Mobile	No	1	22
	333	1019	98652.87	08/06/2024 16:27	Online	Mumbai	Web	No	1	16

**Which day has the most fraud cases:**

```
32 -- Q7. Which day has the most fraud cases?  
33 • SELECT day_of_week, COUNT(*) AS fraud_count  
34 FROM fraud_transaction  
35 WHERE is_fraud = 1  
36 GROUP BY day_of_week  
37 ORDER BY fraud_count DESC;  
--
```

	day_of_week	fraud_count
▶	Saturday	25
	Thursday	25
	Sunday	23
	Friday	20
	Monday	19
	Wednesday	19
	Tuesday	16

**Fraud by transaction type:**

```
--  
54 -- Q11. Total amount lost due to fraud  
55 • SELECT SUM(amount) AS total_fraud_loss  
56 FROM fraud_transaction  
57 WHERE is_fraud = 1;  
--
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
total_fraud_loss			
▶ 10419974.459999997			

And much more.....

## 6] Power BI Dashboards :

*The dashboard consists of **three pages**, each providing different levels of insights.*

### Page 1: Fraud Overview KPIs + Charts + Filters:

#### Key visuals :

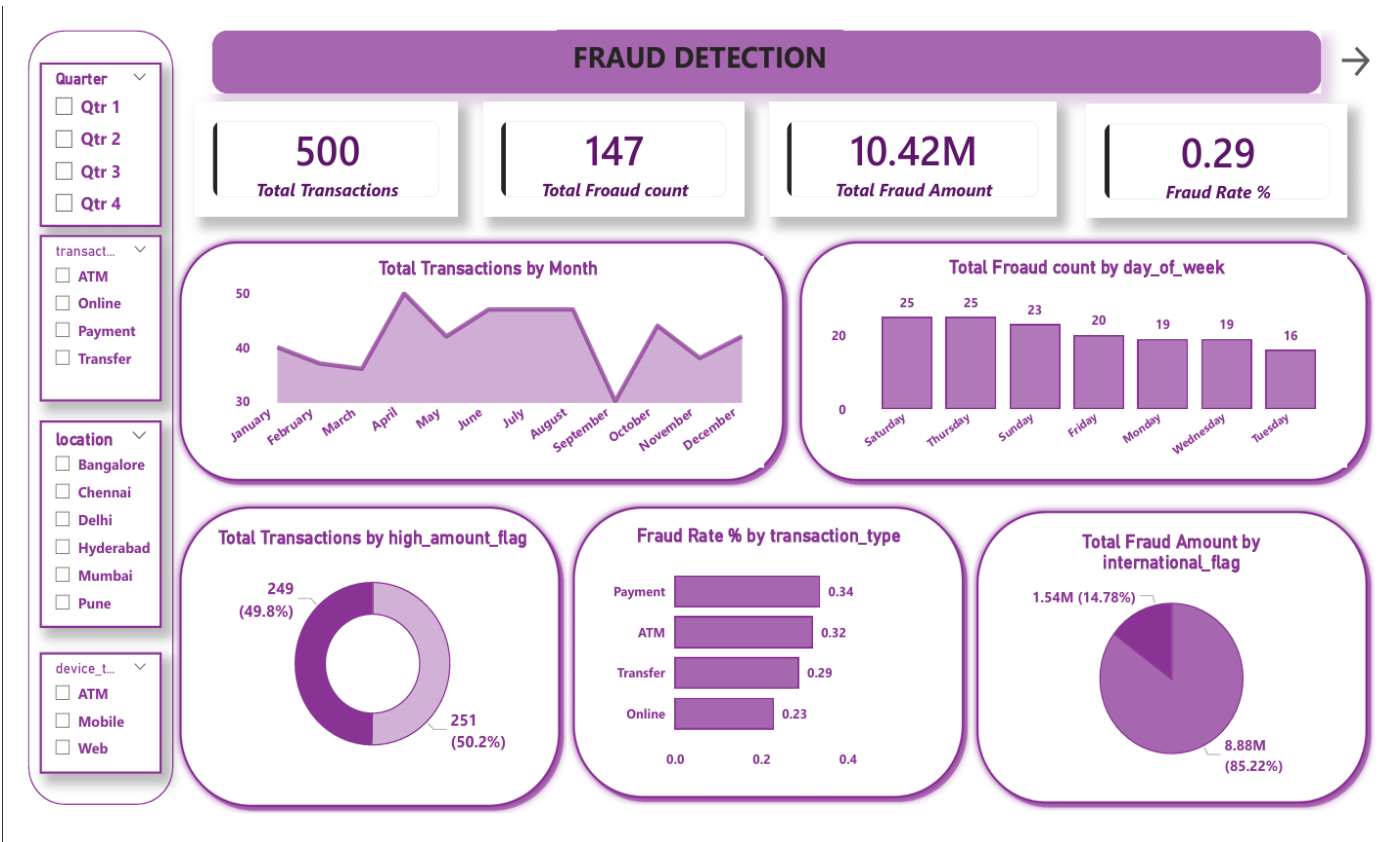
- ☐ Total Transactions
- ☐ Total Fraud Count
- ☐ Total Fraud Amount
- ☐ Fraud Rate (%)

#### Filters :

- a] Quarters
- b] Transaction Type
- c] Location
- d] Device\_type

#### Charts :

- a] Total transaction by months
- b] Total transaction count by day\_of\_week
- c] Total transaction by high amount flag
- d] Fraud rate % by transaction type
- e] Total fraud amount by international\_flag



## Page 2: Fraud Insights Table + Filters:

### Filters:

- a] transaction\_type
- b] location
- c] is\_fraud
- d] high\_amount\_flag
- e] is\_international

**Table :** Created single table for best visualization that includes,

- 1] transaction\_id
- 2] customer\_id

3] amount

4] transaction\_type

5] location

6] is\_fraud

7] high\_amount\_flag

8] international fla

←

→

transaction\_type

All

location

All

is\_fraud

All

high\_amount\_flag

All

is\_international

All

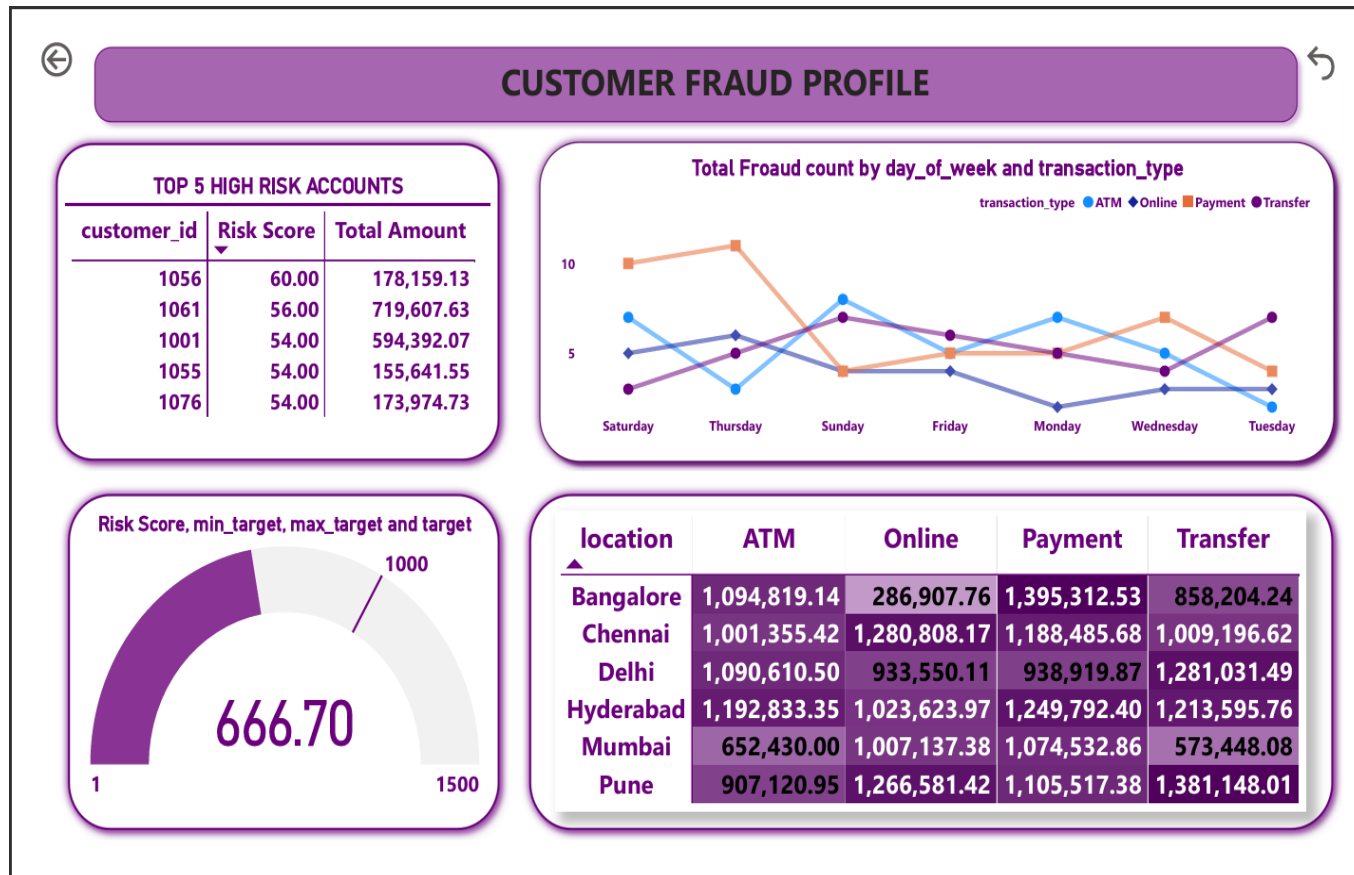
FRAUD INSIGHTS TABLE							
transaction_id	customer_id	amount	transaction_type	location	is_fraud	high_amount_flag	international_flag
1	1051	24848.33	Payment	Pune	0	0	0
2	1092	35661.67	Payment	Hyderabad	0	0	0
3	1014	75808.83	Transfer	Delhi	0	1	0
4	1071	1537.91	Payment	Hyderabad	0	0	0
5	1060	11695.66	Payment	Chennai	0	0	1
6	1020	4695.66	Transfer	Chennai	0	0	0
7	1082	4168.81	Transfer	Chennai	0	0	0
8	1086	85560.51	Online	Mumbai	0	1	0
9	1074	70395.42	Payment	Delhi	1	1	0
10	1074	47469.97	ATM	Pune	0	0	0
11	1087	9873.63	Transfer	Pune	0	0	0
12	1099	49212.43	Online	Mumbai	0	0	0

## Page 3: Deep Fraud Insights:

### Key visuals:

- Total Fraud count by day\_of\_week and transaction\_type
- Risk Score Gauge

- **Amount Heatmap by Location & Transaction Type**
- **Top 5 High-Risk Accounts Table**



## 7. Key Insights & Findings

## Fraud Amount Distribution

- **High fraud concentration in Online & Payment transactions**
- **International transactions have significantly higher fraud amounts**

## Location Insights

- **Cities like Bangalore, Chennai, Hyderabad show higher fraud amounts**



- Metro cities are at higher risk

### Time-based Fraud Trends

- Fraud peaks occur on weekends
- Evening hours show more suspicious transactions

### Customer Risk Insights

- Top 5 high-risk customers contribute majority of the fraud amount
- High amount transactions strongly correlate with higher fraud probability

### Fraud Rate Summary

- Total fraud rate = 29.40%
- This indicates potential system vulnerabilities

## 8] Business Recommendations

### ✓ Strengthen authentication for Online and Payment channels

*Add 2-factor authentication, device binding, behavioral analysis.*

### ✓ Monitor high-risk locations

*Set up real-time alerts for suspicious activity.*

### ✓ Improve weekend transaction monitoring

*Add rules for weekend-based fraud pattern detection.*

### ✓ Flag high amount transactions instantly

*Set dynamic thresholds based on customer history.*

### ✓ Customer Risk Scoring

*Use the risk score model to prioritize investigations.*

---

## 9. Conclusion

**This project successfully demonstrates how data can be used to detect fraud patterns.**

**Using Python, SQL, and Power BI:**

- **Fraud trends were identified**
- **High-risk customers were classified**
- **Business insights were extracted**
- **Final dashboards were created for fraud analysts**

This dashboard enables financial institutions to identify fraud faster and take preventive actions.

---

## 10. Deliverables

- *Clean dataset*
- *Python preprocessing code*
- *SQL queries*
- *Power BI dashboard (.pbix)*
- *PDF dashboard*
- *Full project report (this document)*