## Introduction

Frauds happen in many forms, which are identity, investment, health care, etc. One of the most common frauds that experienced by 50% of Americans is payment card fraud that occurred multiple times to more than 1/3 of the card holders (to, 2004).

Although the implication of fraud can be serious, the fraud rate is incredibly low. According to Clearly Payment, the rate for CNP (Card-Not-Present) fraud in E-commerce industry is only 0.93% (Kalle Radage, 2024). Therefore, it can be tedious to spot fraudulent activities from the big volume of transaction data.

Merchants and financial institutions can prevent payment fraud by using antifraud services or software, where big data technologies are involved to detect certain patterns in fraudulent transactions. To further enhance accuracy, large datasets are needed for training and testing in machine learning.

### **Data collection**

The dataset, fraud\_data.csv, contains 15 variables and 14447 records. Data cleaning is required due to inconsistent values under several columns: trans\_date\_trans\_time, job, merchant, dob, is fraud. The highlighted values required standardization.

trans_date_trans_t •	merchant	category	amt - city -	stat •	lat 🕶	long 🔻	city_pc - j	ob -	dob -	trans_num	▼ merch_la ▼	merch_long • is	_fraud 🔻
2020-10-12 2:12	Harber Inc	gas_transport	10.97 Lakeport	CA	39.047	-122.9328	11256	Podiatrist	18-10-1972	29b86851da4ca1	8a 39.217949	-122.002342	1
2020-10-12 3:06	Gutmann-Upton	misc_pos	8.76 Lakeport	CA	39.047	-122.9328	11256	Podiatrist	18-10-1972	ecae6504633209	06 38.596368	-123.141124	1
2020-10-12 17:30	"Schroeder, Wolff and Hermisto	travel	8.96 Lakeport	CA	39.047	-122.9328	11256	Podiatrist	18-10-1972	5dd2b06961eb8e	c5 39.050171	-122.692423	1
2020-10-12 23:47	"Wuckert, Wintheiser and Friese	home	261.03 Lakeport	CA	39.047	-122.9328	11256	Podiatrist	18-10-1972	e20e4793d1662f	7e 39.873689	-122.26241	1
2020-11-12 1:51	Huel-Langworth	misc_net	890.64 Lakeport	CA	39.047	-122.9328	11256	Podiatrist	18-10-1972	b1b4b5dd6a24ce	4b 39.91141	-122.528368	1
2020-11-12 2:10	Kiehn Inc	grocery_pos	343.37 Lakeport	CA	39.047	-122.9328	11256	Podiatrist	18-10-1972	3bbec2bcebdc598	36-38.729526	-122.571755	1
2020-11-12 2:14	Stiedemann Lt	food_dining	101.04 Lakeport	CA	39.047	-122.9328	11256	Podiatrist	18-10-1972	9e0a1109a6797b	2c 38.674714	-123.544141	1
2020-11-12 3:51	"Yost, Block and Koepp"	misc_pos	8.33 Lakeport	CA	39.047	-122.9328	11256	Podiatrist	18-10-1972	858f16a84b02a8	91 38.10709	-121.933423	1
2020-11-12 3:52	"Bahringer, Schoen and Corkery	shopping_pos	770.65 Lakeport	CA	39.047	-122.9328	11256	Podiatrist	18-10-1972	65c4a80dabf9b44	10 39.721476	-122.611438	1
2020-11-12 12:36	Kris-Kertzmann	travel	7.88 Lakeport	CA	39.047	-122.9328	11256	Podiatrist	18-10-1972	6b7742d1412057	℃ 39.139593	-122.688798	1
2020-11-12 22:29	"Conroy, Balistreri and Gorczany	health_fitness	18.46 Lakeport	CA	39.047	-122.9328	11256	Podiatrist	18-10-1972	0563199dc605dc	06 39.868656	-123.337295	1
2020-11-12 23:05	Ratke and Sons	health_fitness	17.35 Lakeport	CA	39.047	-122.9328	11256	Podiatrist	18-10-1972	7e48fbbbc83523l	38.552146	-122.519483	1
2020-11-12 23:19	Thompson-Gleason	health_fitness	19.45 Lakeport	CA	39.047	-122.9328	11256	Podiatrist	18-10-1972	bfde75d978bb99	05 39.25188	-122.490946 1	2020-12-24 16:56:24"
24-12-2020 16:58	"Hahn, Douglas and Schowalte"	travel	440.56 Meadville	MO	39.7795	-93.3014	964	Fourist information centre r	23-12-1974	68a845f709866a	Of: 39.419072	-93.9479	0
24-12-2020 16:59	Erdman-Durgan	health_fitness	60.39 Crownpoint	NM	35.7206	-108.0271	5662 1	T consultant	1989-08-04	17075780fd7851	b2 36.123857	-107.164356	0
24-12-2020 16:59	"Prosacco, Kreiger and Kovacek"	home	16.13 Mound City	MO	40.1362	-95.2138	1631 /	Architect	20-01-1953	a38ea67ec77fe7e	03 39.984044	-96.203203	0
24-12-2020 16:59	"Romaguera, Cruickshank and G	shopping_net	6.7 Napa	CA	38.4549	-122.2564	94014	Airline pilot	21-08-1985	32455d6fefc982a	e2 38.229234	-122.499378	0
24-12-2020 17:00	Spencer-Runolfsson	misc_pos	55.61 Hawthorne	CA	33.9143	-118.3493	93193 '	'Editor, magazine features"	19-04-1995	ca3d23dda0ff4ec	42 33.66463	-117.730522	0
24-12-2020 17:11	Kuhn LLC	shopping_pos	2.58 Moab	UT	38.5677	-109.5271	9772 1	ocation manager	24-11-1989	67a249c83b7ada	d9 38.956696	-109.612304	0
24-12-2020 17:16	"Willms, Kris and Bergnaum"	shopping_pos	53.52 Kansas City	MO	38.9621	-94.5959	545147 (	Counsellor	18-11-1987	ee30fa8d874977	75 38.432755	-93.981096	0
24-12-2020 17:18	"Roberts, Ryan and Smith"	personal_care	81.53 Newhall	CA	34.3795	-118.523	34882	Health physicist	25-04-1971	abbf11c821a67b6	33.91462	-118.660667	0
24-12-2020 17:18	Goyette-Gerhol	kids pets	44.61 Huntington B	e CA	33.6773	-118.0051	190249	'Therapist, horticultural"	17-09-1976	7a5cb8e0529501	db 33.406105	-117.630637	0

The details of the variables and ideal data type are listed as below:

Variable	Description	Data type (ideal)	Current Values / example	
trans_date_tra	Transaction	VARCHAR(30)	4-1-2019 12:58:00 AM	
ns_time	datetime		14-01-2019 02:27	
merchant Name of the merchant		VARCHAR(255)	Predovic Inc	

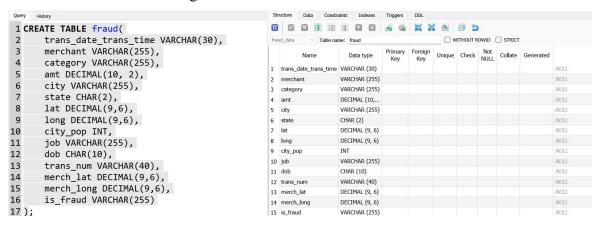
category	Merchant	VARCHAR(255)	shopping_net		
	category		grocery_pos		
amt	Transaction	DECIMAL(10, 2)	367.29		
	amount				
city	City where credit	VARCHAR(255)	Browning		
	card holder is				
	located				
state	State where credit	CHAR(2)	MO		
	card holder is				
	located				
lat	Latitude of the	DECIMAL(9,6)	40.029		
	purchase				
long	Logitude of the	DECIMAL(9,6)	-93.1607		
	purchase				
city_pop	Card holder's city	INT	602		
	population				
job	Card holder's job	VARCHAR(255)	Cytogeneticist		
			"Administrator,		
			education"		
dob	Card holder's	CHAR(10)	1939-09-11		
	birthday		18-11-1987		
trans_num	Transaction	VARCHAR(40)	8e2d2fae5319d31c887		
	number		dddbc70627ac4		
merch_lat	Merchant's	DECIMAL(9,6)	63.917785		
	latitude				
merch_long	Merchant's	DECIMAL(9,6)	-165.827621		
	logitude				
is_fraud	Indication if the	BOOLEAN	0		
	transaction is		1		
	fraud				
	• 0: Not fraud				
	• 1: fraud				

## **Problems and results**

The aim of the study is to spot fraudulent transaction patterns with the information given in the dataset. However, pre-processing works need to be done before the analysis and study begin. Steps, problems faced during every process and analysis are recorded. The database engine used to store and process the data is SQLite.

#### Load data into database

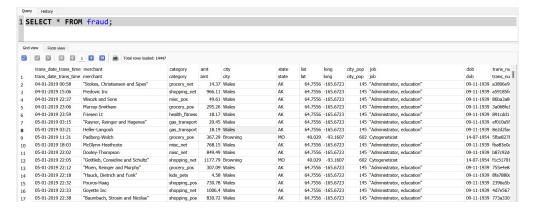
1. CREATE TABLE fraud: "trans\_date\_trans\_time" and "is\_fraud" are created as VARCHAR for data cleaning to be conducted.



Import the fraud\_data.csv into the table in SQLite CLI after setting the mode to CSV.

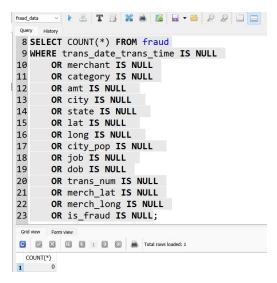
```
SQLite version 3.47.0 2024-10-21 16:30:22
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open fraud_db.db
sqlite> .tables
fraud
sqlite> .mode csv
sqlite> .import fraud_data.csv fraud
sqlite> .import fraud_data.csv
```

3. SELECT \* FROM fraud: Get a glimpse of the imported data



### **Data Cleaning**

1. COUNT(\*): Find empty cells for all columns and filter with WHERE clause. The result shown 0 null values.



2. DELETE FROM fraud WHERE amt = 'amt': Remove replicated header record.

COUNT(\*): Applied to get the total records, where 14446 is obtained.

```
59 DELETE FROM fraud
60 WHERE amt = 'amt';
61
62 SELECT COUNT(*)
63 FROM fraud;
Grid view Form view

GRID COUNT(*)
1 14446
```

3. REPLACE: Remove the double quote by updating the table columns.

```
35 UPDATE fraud
36 SET
37 merchant = REPLACE(merchant, '"', ''),
38 job = REPLACE(job, '"', '');
```

4. Standardize "is\_fraud" to 0 and 1 by editing the odd values:

Steps	Query	Result		
Find and	35 SELECT DISTINCT is_fraud 36 FROM fraud;	is_fraud		
count	49 SELECT DISTINCT is_fraud, COUNT(is_fraud) 50 FROM fraud	2 1"2020-12-24 16:56:24"		
distinct	51 GROUP BY is_fraud;	3 0		
		4 0"2019-01-01 00:00:44"		
values		is_fraud COUNT(is_fraud) 1 0 12600		
		2 0"2019-01-01 00:00:44"		
		3 1 1844 4 1"2020-12-24 16:56:24" 1		

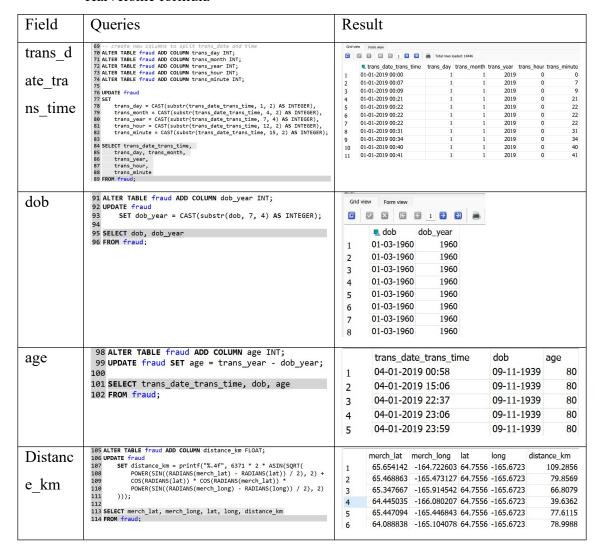
Filter "trans_num" with WHERE clause	38 SELECT trans_num, is_fraud 39 FROM fraud 40 WHERE is_fraud != 0 AND is_fraud != 1;	trans_num 1 bfde75d978bb9905a4a8c87440692 2 14392d723bb7737606b2700ac791l  The "trans_num" lead to odd "is_fra	are unique to
Update the values to 0 and 1	42 UPDAYE fraud 43 SET is_fraud = CASE 44 WHEN trans_num = 'bfder5d978bb9985a4a8c87446692a4c' THEN 1 45 WHEN trans_num = '14392d723bb7737606b2700ac791b7aa' THEN 0 46 ELSE is_fraud 47 END; 48 SELECT DISTINCT is_fraud, COUNT(is_fraud) 50 FROM fraud 51 GROUP BY is_fraud;  Change the "is_fraud" values and remain the rest accordingly	is_fraud 1 0 2 1	COUNT(is_fraud) 12601 1845

- 5. Date and time values are auto-format into "DD-MM-YYYY" upon import into SQLite, hence no cleaning nor editing required. Samples in CSV and table are filtered for comparison.
  - a) "trans\_date\_trans\_time" in "DD-MM-YYYY HH:MM"
  - b) "dob" in "DD-MM-YYYY"

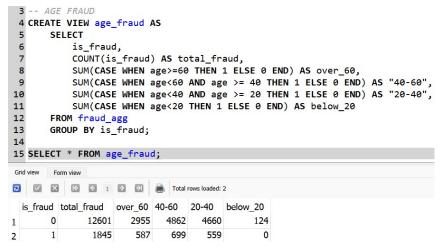
Field	CSV		Database "fraud" table			
trans_d ate_tran s_time	13-01-2019 02:29 "Raynor, Rein 14-01-2019 11:02 "Raynor, Rein	ger and Hagenes" ger and Hagenes" ger and Hagenes" ger and Hagenes" ger and Hagenes" ger and Hagenes"	55 SELECT trans_date_trans_time, merchant 56 FROM fraud 57 WHERE merchant = 'Raynor, Reinger and Hagenes';  Grid view Form view  Total rows loaded: 36  Raynor, Reinger and Hagenes 3 03-01-2019 09:29  Raynor, Reinger and Hagenes 4 04-01-2019 02:03  Raynor, Reinger and Hagenes 5 04-01-2019 02:13  Raynor, Reinger and Hagenes 7 05-01-2019 03:15  Raynor, Reinger and Hagenes 8 05-01-2019 05:10  Raynor, Reinger and Hagenes 9 11-01-2019 03:19  Raynor, Reinger and Hagenes Raynor, Reinger and Hagenes			
dob	Cytogeneticist Cytogeneticist Cytogeneticist Cytogeneticist Occupational psychologist	14-07-1954 14-07-1954 14-07-1954 14-07-1954 1954-05-07 1954-05-07 1954-05-07 1954-05-07 1954-05-07 1954-05-07	59 SELECT job,dob 60 FROM fraud 61 WHERE dob LIKE '%1954%'; 62  Grid view Form view  Job Gob Gob Gob Gob Gob Gob Gob Gob Gob G			

## **Analysis**

- 1. Aggregation:
  - a) Split "trans date trans time" and "dob" into separate columns.
  - b) Find age of card holder by taking "trans\_year" to minus "dob\_year"
  - c) Calculate distance between location of merchants and card holder by using Harversine formula



- 2. Views are created to store the analysis for studying factors affecting frauds. CASE and WHEN expressions are heavily used. Percentage fraud are calculated for better comparison with other classes.
  - a) age fraud: fraud by age groups



b) amt\_fraud: Fraud by transaction amount, the average fraudulent amount is \$517.96.

```
52 -- AMOUNT FRAUD
53 CREATE VIEW amt_fraud AS
54 SELECT
55 rounded_avg.avg_fraud_amt,
56 SUM(CASE WHEN amt >= rounded_avg.avg_fraud_amt AND is_fraud = 1 THEN 1 ELSE 0 END) AS fraud_case,
57 (SELECT COUNT(*) FROM fraud_agg WHERE is_fraud = 1) AS total_fraud,
58 ((SUM(CASE
WHEN amt >= rounded_avg.avg_fraud_amt
60 AND is_fraud = 1
61 THEN 1 ELSE 0 END)*100)/
62 (SELECT COUNT(*) FROM fraud_agg WHERE is_fraud = 1))AS fraud_pct
63 FROM fraud_agg,
64 (SELECT COUNT(*) FROM fraud_agg WHERE is_fraud=1) AS rounded_avg;
65
66 SELECT * FROM amt_fraud;

avg_fraud_amt fraud_case total_fraud fraud_pct
1 517.96 863 1845 46
```

c) cat fraud: Fraud by categories

```
39 CREATE VIEW cat_fraud AS
40 SELECT
41
42
           COUNT(category) AS total_trans,
           SUM(CASE WHEN is_fraud = 1 THEN 1 ELSE 0 END) AS fraud_cases,
ROUND((SUM(CASE WHEN is_fraud = 1 THEN 1 ELSE 0 END) * 100.0) / COUNT(category), 2) AS fraud_pct
43
       FROM fraud_agg
GROUP BY category
       ORDER BY fraud_pct DESC;
49 SELECT * FROM cat fraud;
                    total trans fraud cases fraud pct
    category
1 shopping_net
                          1408
                                         396
                                                 28.13
2 grocery_pos
                          1602
                                         444
                                                 27.72
3 misc_net
                           821
                                         223
                                                 27.16
                                         194
                                                  14.33
4 shopping_pos
                          1354
                                         159
5 gas_transport
                          1430
                                                  11.12
6 travel
                           385
                                          34
                                                   8.83
7 misc_pos
                            823
                                          64
                                                   7.78
                           474
                                          32
8 grocery_net
                                                   6.75
                           953
                                          59
                                                   6.19
9 entertainment
10 personal_care
                            990
                                          57
                                                   5.76
11 kids_pets
                           1141
                                          56
                                                   4.91
                            870
                                          39
                                                   4.48
12 food_dining
13 health_fitness
                            891
                                          37
                                                   4.15
                                                   3.91
```

d) distance\_fraud: Relationship between fraudulent records and distance (KM) between the purchaser and merchant's location.

```
distance fraud
70 CREATE VIEW distance_fraud AS
71
       SELECT
72
73
           rounded_avg.avg_fraud_km,
           SUM(CASE
74
                    WHEN distance_km >= rounded_avg.avg_fraud_km AND is_fraud = 1
75
           THEN 1 ELSE 0 END) AS fraud_case,
(SELECT COUNT(*) FROM fraud_agg WHERE is_fraud = 1) AS total_fraud,
76
77
78
                WHEN distance_km >= rounded_avg.avg_fraud_km AND is_fraud = 1
                    THEN 1 ELSE 0 END)*100)/(
79
80
                    SELECT COUNT(*)
81
                        FROM fraud_agg
82
                         WHERE is_fraud = 1))AS fraud_pct
       FROM fraud_agg,
83
           (SELECT ROUND(AVG(distance_km), 2) AS avg_fraud_km
84
                FROM fraud_agg
WHERE is_fraud=1) AS rounded_avg;
85
86
87
88 SELECT * FROM distance fraud;
Grid view Form view
🔁 🔽 🔀 🧲 1 🗗 🖼 Total rows loaded: 1
 avg fraud km fraud case total fraud fraud pct
          75.47
                      968
                                1845
```

e) hour\_fraud: Frequency of fraud by in different hours, order by highest

```
percentage.
                                                             trans_hot total_trans fraud
                                                                                 fraud_pct
 91 -- transaction hour fraud
                                                                              477
                                                                   23
                                                                         1127
                                                                                     42
 92 CREATE VIEW hour_fraud AS
                                                                   22
                                                                         1120
                                                                               468
                                                                                     41
                                                             2
 93
         SELECT
                                                                         587
                                                                               174
 94
              trans_hour,
                                                                    3
                                                                         564
                                                                               147
                                                                                     26
 95
              COUNT(trans_hour) AS total_trans,
                                                                         521
                                                                               142
                                                                                     27
                                                                   16
                                                                         653
                                                                               24
                                                                                      3
 96
              SUM(CASE
                                                                   15
                                                                         671
                                                                               24
 97
                  WHEN is_fraud = 1
                                                                         702
                                                                               23
                                                                   20
 98
                  THEN 1 ELSE 0 END) AS fraud,
                                                             10
                                                                          666
                                                                               20
                                                                                      3
                                                             11
 99
              (SUM(CASE
                                                                   21
                                                                         692
                                                                               18
                                                                                      2
                                                             12
100
                  WHEN is_fraud = 1
                                                                   12
                                                                         624
                                                                               17
                                                                                      2
                       THEN 1 ELSE 0 END) * 100 /
101
                                                             14
                                                                   14
                                                                         630
                                                                               16
                                                                         638
102
                  COUNT(trans_hour)) AS fraud_pct
                                                             15
                                                                   13
                                                                               14
                                                                               14
                                                                          418
                                                             16
103
         FROM fraud_agg
                                                             17
         GROUP BY trans_hour
104
                                                                   19
                                                                          667
                                                                               11
                                                             18
105
         ORDER BY fraud DESC;
                                                             19
                                                                   10
                                                                         418
                                                                               10
                                                                                      2
                                                             20
                                                                    5
                                                                          406
                                                                               10
                                                                                      2
106
                                                                          438
                                                             21
                                                                                9
107 SELECT * FROM hour_fraud;
                                                                          402
                                                             22
                                                                   11
                                                             23
```

f) state fraud: Fraud cases in different states.

```
state
                                                                   total_trans fraud_cases fraud_pct
20 CREATE VIEW state_fraud AS
                                                                                         65
                                                                                                  37.57
                                                       1 AK
                                                                           173
21
22
      SELECT
                                                          NF
                                                                          1460
                                                                                        238
                                                                                                   16.3
          state.
                                                       2
23
          COUNT(state) AS total_trans,
                                                       3
                                                           OR
                                                                          1211
                                                                                         197
                                                                                                  16.27
24
          SUM(CASE
                                                       4
                                                           CO
                                                                           856
                                                                                         115
                                                                                                  13.43
25
              WHEN is_fraud = 1
26
                  THEN 1 ELSE 0 END) AS fraud_cases,
                                                       5
                                                           UT
                                                                           597
                                                                                         73
                                                                                                  12.23
27
          ROUND((SUM(CASE
                                                           CA
                                                                          3375
                                                                                        411
                                                                                                  12.18
                                                       6
                       WHEN is_fraud = 1
                                                           NM
                                                                          1003
                                                                                         121
                                                                                                  12.06
                       THEN 1 ELSE 0 END) * 100.0) /
                                                       7
29
30
                      COUNT(state), 2) AS fraud_pct
                                                           MO
                                                                          2329
                                                                                        267
                                                                                                  11.46
                                                       8
31
      FROM fraud_agg
                                                       9
                                                           WA
                                                                          1150
                                                                                         126
                                                                                                  10.96
                                                       10 WY
                                                                          1100
                                                                                         119
                                                                                                  10.82
      ORDER BY fraud_pct DESC;
                                                       11 ID
                                                                           347
                                                                                         33
                                                                                                   9.51
35 SELECT * FROM state_fraud;
                                                       12 AZ
                                                                           673
                                                                                          64
                                                                                                   9.51
                                                                           172
                                                                                          16
                                                       13 HI
                                                                                                    9.3
```

## **Results**

From the analysis, some patterns of the fraud are obvious, such as the transaction hours where fraud is most active between 10PM to 3AM, highest at 11PM; occurred frequently in "shopping\_net" of 28.13%, "grocery\_pos" and "misc\_net", lowest in "home" of 3.91%. There is 46% chance of fraud when the transaction amount passed \$517.9, and 52% likelihood at average distance between purchaser and merchant of 75.47KM. Not to mention, it is saddening to see that the age group which has the highest number of fraud victims is between 40 to 60.

(762 words)

# Reference

- to, C. (2004, September 16). *inclusive term for fraud committed using a payment card, such as a credit card or debit card*. Wikipedia.org; Wikimedia Foundation, Inc. <a href="https://en.wikipedia.org/wiki/Credit\_card\_fraud#:~:text=A%20few%20examples%20of%20credit,fraudsters%20attempting%20to%20steal%20money.">https://en.wikipedia.org/wiki/Credit\_card\_fraud#:~:text=A%20few%20examples%20of%20credit,fraudsters%20attempting%20to%20steal%20money.</a>
- Kalle Radage. (2024, July 16). Fraud Risk by Industry in Payments. Credit Card Processing and Merchant Account.

  https://www.clearlypayments.com/blog/fraud-risk-by-industry-in-payments/
- Shah, J. (2022). *Online Payment Fraud Detection*. Kaggle.com. <a href="https://www.kaggle.com/datasets/jainilcoder/online-payment-fraud-detection">https://www.kaggle.com/datasets/jainilcoder/online-payment-fraud-detection</a>
- Choudhury, N. R. (2024). *Credit Card Fraud data*. Kaggle.com. https://www.kaggle.com/datasets/neharoychoudhury/credit-card-fraud-data