

```
1 from google.colab import files
2 uploaded = files.upload()
```



Choose Files Fraud Dete... Dataset.csv

- **Fraud Detection Dataset.csv**(text/csv) - 3692523 bytes, last modified: 4/2/2025 - 100% done
Saving Fraud Detection Dataset.csv to Fraud Detection Dataset.csv

```
1 import pandas as pd
2 import numpy as np
3
4 # Load the CSV file (update filename accordingly)
5 df = pd.read_csv("Fraud Detection Dataset.csv")
6
7 # Display first few rows
8 print(df.head())
```



	Transaction_ID	User_ID	Transaction_Amount	Transaction_Type	\
0	T1	4174	1292.76	ATM Withdrawal	
1	T2	4507	1554.58	ATM Withdrawal	
2	T3	1860	2395.02	ATM Withdrawal	
3	T4	2294	100.10	Bill Payment	
4	T5	2130	1490.50	POS Payment	

	Time_of_Transaction	Device_Used	Location	\
0	16.0	Tablet	San Francisco	
1	13.0	Mobile	New York	
2	NaN	Mobile	NaN	
3	15.0	Desktop	Chicago	
4	19.0	Mobile	San Francisco	

	Previous_Fraudulent_Transactions	Account_Age	\
0	0	119	
1	4	79	
2	3	115	
3	4	3	
4	2	57	

	Number_of_Transactions_Last_24H	Payment_Method	Fraudulent
0	13	Debit Card	0
1	3	Credit Card	0
2	9	NaN	0
3	4	UPI	0
4	7	Credit Card	0

```
1 # Extract numerical columns as NumPy arrays
2 transaction_amounts = df["Transaction_Amount"].fillna(0).to_numpy()
3 fraud_labels = df["Fraudulent"].to_numpy()
4 account_ages = df["Account_Age"].to_numpy()
5
```

```
1 # Fixed type structured array
2 structured_array = np.array(
3     list(zip(transaction_amounts, fraud_labels, account_ages)),
4     dtype=[("Transaction_Amount", "f8"), ("Fraudulent", "i4"), ("Account_Age", "i4")]
5 )
6
```

```
1 # Indexing
2 indexed_values = transaction_amounts[:5]
```

```
1 # Slicing
2 sliced_values = fraud_labels[:10]
```

```
1 # Reshaping
2 reshaped_array = transaction_amounts[:15].reshape(5, 3)
```

```
1 # Concatenation
2 concatenated_array = np.concatenate((transaction_amounts[:5], account_ages[:5]))
```

```
1 # Splitting
2 split_arrays = np.split(transaction_amounts[:10], 2)
3
```

```
1 # Universal functions (UFUNCs)
2 mean_transaction = np.mean(transaction_amounts)
3 sum_transactions = np.sum(transaction_amounts)
4
```

```
1 # Broadcasting
2 broadcasted_array = account_ages + 10

1 # Boolean Masking
2 high_value_transactions = transaction_amounts[transaction_amounts > 2000]

1 # Fancy Indexing
2 fancy_indexing_example = transaction_amounts[[0, 5, 10]]

1 # Sorting
2 sorted_transactions = np.sort(transaction_amounts)

1 # Partial Sorting (Top 5 highest transactions)
2 top_5_transactions = np.partition(transaction_amounts, -5)[-5:]

1 # Additional NumPy Operations
2 # Mathematical Functions
3 sqrt_transactions = np.sqrt(transaction_amounts)
4 log_transactions = np.log(transaction_amounts + 1)
5

1 # Statistical Functions
2 median_transaction = np.median(transaction_amounts)
3 std_transaction = np.std(transaction_amounts)

1 # Linear Algebra
2 dot_product = np.dot(account_ages[:5], fraud_labels[:5])

1 # Random Numbers
2 random_values = np.random.normal(loc=50, scale=10, size=5)
3

1 # Advanced Indexing
2 high_value_indices = np.where(transaction_amounts > 2000)
3 taken_values = np.take(transaction_amounts, [0, 5, 10])
4

1 # Stacking Arrays
2 hstacked_array = np.hstack((account_ages[:5].reshape(-1, 1), fraud_labels[:5].reshape(-1, 1)))

1 # Unique and Counting
2 unique_fraud_labels, fraud_counts = np.unique(fraud_labels, return_counts=True)
3

1 # Clipping and Rounding
2 clipped_transactions = np.clip(transaction_amounts, 0, 5000)
3 rounded_transactions = np.round(transaction_amounts, 2)
4

1 # Finding Min/Max Locations
2 max_index = np.argmax(transaction_amounts)
3 min_index = np.argmin(transaction_amounts)
4

1 # Tile & Repeat Functions
2 repeated_array = np.repeat(transaction_amounts[:5], 3)
3 tiled_array = np.tile(transaction_amounts[:5], 3)
4

1 # Correlation and Covariance
2 correlation = np.corrcoef(df["Account_Age"], df["Fraudulent"])[0, 1]
3 covariance = np.cov(transaction_amounts, fraud_labels)[0, 1]
4

1 # Generating Custom NumPy Arrays
2 zeros_array = np.zeros(10)
3 ones_array = np.ones(10)
4 linspace_array = np.linspace(0, 100, 10)
5 logspace_array = np.logspace(1, 3, 10)
6
```

```

1 # Printing results
2 print("Indexed Values:", indexed_values)
3 print()
4 print("Sliced Values:", sliced_values)
5 print()
6 print("Reshaped Array:\n", reshaped_array)
7 print()
8 print("Mean Transaction Amount:", mean_transaction)
9 print()
10 print("Sum of Transactions:", sum_transactions)
11 print()
12 print("High Value Transactions:", high_value_transactions[:5])
13 print()
14 print("Top 5 Transactions:", top_5_transactions)
15 print()
16 print("Square Root Transactions:", sqrt_transactions[:5])
17 print()
18 print("Log Transactions:", log_transactions[:5])
19 print()
20 print("Median Transaction Amount:", median_transaction)
21 print()
22 print("Standard Deviation:", std_transaction)
23 print()
24 print("Dot Product:", dot_product)
25 print()
26 print("Random Values:", random_values)
27 print()
28 print("High Value Indices:", high_value_indices)
29 print()
30 print("Taken Values:", taken_values)
31 print()
32 print("Horizontally Stacked Array:\n", hstacked_array)
33 print()
34 print("Unique Fraud Labels and Counts:", unique_fraud_labels, fraud_counts)
35 print()
36 print("Clipped Transactions:", clipped_transactions[:5])
37 print()
38 print("Rounded Transactions:", rounded_transactions[:5])
39 print()
40 print("Max Transaction Index:", max_index)
41 print()
42 print("Min Transaction Index:", min_index)
43 print()
44 print("Repeated Array:", repeated_array)
45 print()
46 print("Tiled Array:", tiled_array)
47 print()
48 print("Correlation:", correlation)
49 print()
50 print("Covariance:", covariance)
51 print()
52 print("Zeros Array:", zeros_array)
53 print()
54 print("Ones Array:", ones_array)
55 print()
56 print("Linspace Array:", linspace_array)
57 print()
58 print("Logspace Array:", logspace_array)

```

➡ Indexed Values: [1292.76 1554.58 2395.02 100.1 1490.5]

Sliced Values: [0 0 0 0 0 0 1 0 0 0]

Reshaped Array:
 [[1292.76 1554.58 2395.02]
 [100.1 1490.5 2372.04]
 [544.81 635.75 2318.87]
 [3656.17 0. 2733.84]
 [2376.37 1924.48 968.78]]

Mean Transaction Amount: 2848.1997950980394

Sum of Transactions: 145258189.55

High Value Transactions: [2395.02 2372.04 2318.87 3656.17 2733.84]

Top 5 Transactions: [49997.8 49997.8 49997.8 49997.8 49997.8]

Square Root Transactions: [35.95497184 39.42816252 48.93894155 10.00499875 38.60699418]

Log Transactions: [7.16530799 7.34960375 7.78156431 4.61611013 7.3075376]

Median Transaction Amount: 2392.0600000000004

Standard Deviation: 4960.376536284364

Dot Product: 0

Random Values: [38.20658443 31.03091709 49.78132788 40.27913498 57.15971118]

High Value Indices: (array([2, 5, 8, ..., 50996, 50997, 50998]),)

Taken Values: [1292.76 2372.04 0.]

Horizontally Stacked Array:

```
[[119  0]
 [ 79  0]
 [115  0]
 [  3  0]
 [ 57  0]]
```

Unique Fraud Labels and Counts: [0 1] [48490 2510]

Clipped Transactions: [1292.76 1554.58 2395.02 100.1 1490.5]

Rounded Transactions: [1292.76 1554.58 2395.02 100.1 1490.5]

Max Transaction Index: 166

Min Transaction Index: 10

Repeated Array: [1292.76 1292.76 1292.76 1554.58 1554.58 1554.58 2395.02 2395.02 2395.02
100.1 100.1 100.1 1490.5 1490.5 1490.5]

Tiled Array: [1292.76 1554.58 2395.02 100.1 1490.5 1292.76 1554.58 2395.02 100.1
1490.5 1292.76 1554.58 2395.02 100.1 1490.5]