Exploratory Data Analysis (EDA) Report: Credit Card using fraud Detection

Exploratory Data Analysis (EDA) is a crucial step in understanding transaction data, identifying fraudulent behaviour patterns, and generating insights that support fraud detection systems. In this project, we perform EDA on the credit_card_fraud_10k.csv dataset to explore transaction characteristics, detect anomalies, and understand relationships between transaction features and fraud occurrence

Procedure Exploratory Data Analysis:

1.Understand the Problem and the Data

2.Import and Inspect the Data

3.Handling Missing Values

4.Explore Data Characteristics

5.Perform Data Transformation

6.Visualize Data Relationships

7.Handling Outliers

8. Communicate Findings and Insights

Dataset: credit_card_fraud_10k.csv from Kaggle

project Structure:

1.Understand the Problem and the Data

The objective of this project is to analyse credit card transaction data and identify patterns that distinguish fraudulent transactions from legitimate ones. Fraud detection is a critical task in financial systems to prevent monetary losses and enhance customer security.

- **Numerical:** Transaction Amount,Transaction Hour,Account Age,Transaction Velocity,Daily Transaction Count.
- **Categorical:** Foreign Transaction,Online Transaction,Chip Used,High-Risk Merchant
- **Target Variable:** is_fraud (0 – Non-Fraud, 1 – Fraud)

2.Import and Inspect the Data

File Name: mapping.py / visualization.py

The dataset was loaded into a Pandas DataFrame using the read_csv() function. Initial inspection was performed using:

head() to display the first five records

info() to examine column names, data types, and non-null counts

shape to identify the total number of rows and columns

This step helped in understanding the structure of the dataset and identifying potential issues such as missing values or incorrect data types

```
First 5 rows of the dataset:
   transaction_id  amount  transaction_hour merchant_category  ...  device_trust_score  velocity_last_24h  cardholder_age  is_fraud
0               1   84.47                22       Electronics  ...                  66                  3              40         0
1               2  541.82                 3            Travel  ...                  87                  1              64         0
2               3  237.01                17           Grocery  ...                  49                  1              61         0
3               4  164.33                 4           Grocery  ...                  72                  3              34         0
4               5   30.53                15              Food  ...                  79                  0              44         0

[5 rows x 10 columns]
transaction_id
```

```
Dataset Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 10 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   transaction_id       10000 non-null  int64
 1   amount               10000 non-null  float64
 2   transaction_hour     10000 non-null  int64
 3   merchant_category    10000 non-null  object
 4   foreign_transaction  10000 non-null  int64
 5   location_mismatch    10000 non-null  int64
 6   device_trust_score   10000 non-null  int64
 7   velocity_last_24h    10000 non-null  int64
 8   cardholder_age       10000 non-null  int64
 9   is_fraud             10000 non-null  int64
dtypes: float64(1), int64(8), object(1)
memory usage: 781.4+ KB

Total number of entries:
10000
(md) PS D:\project1.py>
```

3. Handling Missing Values and Outliers

File Name:

3.1 Detection of Missing Values

The dataset was checked for missing values using isnull().sum().It was observed that most columns contained no missing values, indicating good data quality.

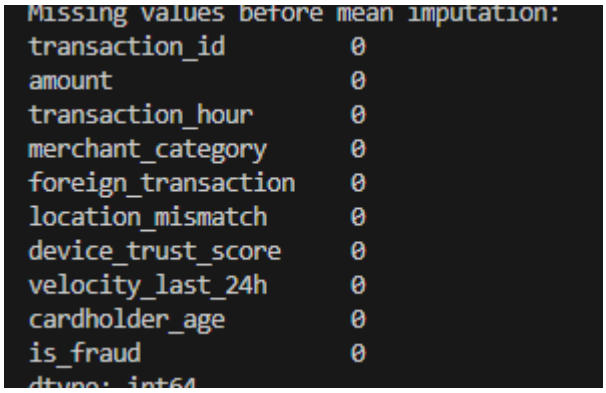
```
Missing values before handling:
transaction_id         0
amount                 0
transaction_hour       0
merchant_category      0
foreign_transaction    0
location_mismatch      0
device_trust_score     0
velocity_last_24h      0
cardholder_age         0
is_fraud               0
dtype: int64
```

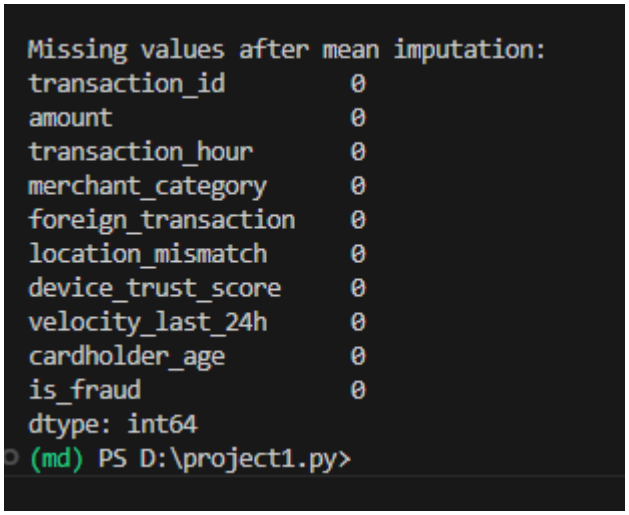
```
Missing values after handling:
transaction_id         0
amount                 0
transaction_hour       0
merchant_category      0
foreign_transaction    0
location_mismatch      0
device_trust_score     0
velocity_last_24h      0
cardholder_age         0
is_fraud               0
dtype: int64

Dataset shape after removing outliers:
(7782, 10)
(md) PS D:\project1.py>
```

3.2 Mean Imputation:

Any minor missing values in numerical columns were handled using mean imputation to maintain dataset size and preserve transaction distribution.





3.3 Duplicate Removal:

Duplicate records were checked using drop_duplicates()

No significant duplicate transactions were found, confirming the integrity of the dataset.

```
Missing values after cleaning and removing duplicates:
transaction_id         0
amount                 0
transaction_hour       0
merchant_category      0
foreign_transaction    0
location_mismatch      0
device_trust_score     0
velocity_last_24h      0
cardholder_age         0
is_fraud               0
dtype: int64
```

```
Non-missing values after cleaning and removing duplicates:
transaction_id         10000
amount                 10000
transaction_hour       10000
merchant_category      10000
foreign_transaction    10000
location_mismatch      10000
device_trust_score     10000
velocity_last_24h      10000
cardholder_age         10000
is_fraud               10000
dtype: int64
(md) PS D:\project1.py>
```

4. Explore Data Characteristics:

File Name:

Using descriptive statistics, we calculated the following for all numeric features

- Mean – Average value of the dataset
- Median – Middle value when data is arranged in order
- Mode – Most frequently occurring value
- Standard Deviation – Measure of how spread out the values
- Minimum – Smallest value in the dataset
- Maximum – Largest value in the dataset

Descriptive statistics were computed using describe() to understand the range, central tendency, and dispersion of numerical features.

Key Observations:

Transaction Amount showed right-skewed distribution,Majority of transactions involved small monetary values,Transaction Hour indicated higher activity during certain hours,Fraud transactions were significantly fewer compared to non-fraud transactions, indicating class imbalance.

```
Minimum values:
transaction_id          1.0
amount                  0.0
transaction_hour        0.0
foreign_transaction     0.0
location_mismatch       0.0
device_trust_score     25.0
velocity_last_24h       0.0
cardholder_age         18.0
is_fraud                0.0
dtype: float64

Maximum values:
transaction_id         10000.00
amount                  1471.04
transaction_hour          23.00
foreign_transaction        1.00
location_mismatch          1.00
device_trust_score        99.00
velocity_last_24h          9.00
cardholder_age            69.00
is_fraud                   1.00
dtype: float64

Average (Mean) values:
transaction_id         5000.500000
amount                  175.949849
transaction_hour         11.593300
foreign_transaction       0.097800
location_mismatch         0.085700
device_trust_score       61.798900
velocity_last_24h         2.008900
cardholder_age           43.468700
is_fraud                  0.015100
dtype: float64
```

5. Visualize Data Relationships

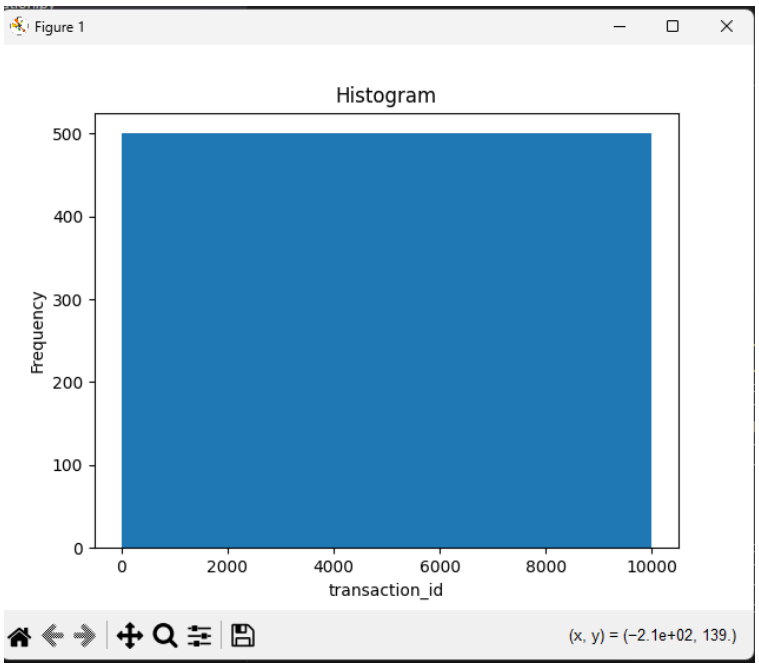
File Name:

Univariate Analysis

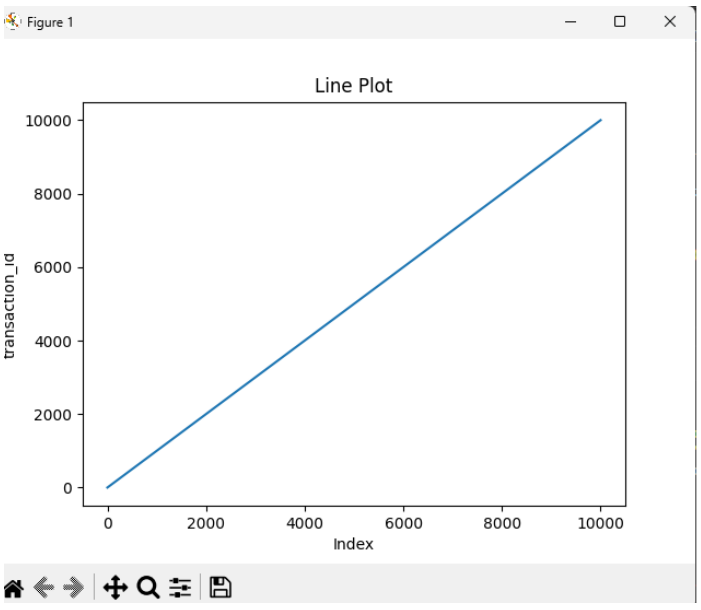
This analysis focuses on **one variable at a time** to understand its distribution and individual behavior.

Charts:

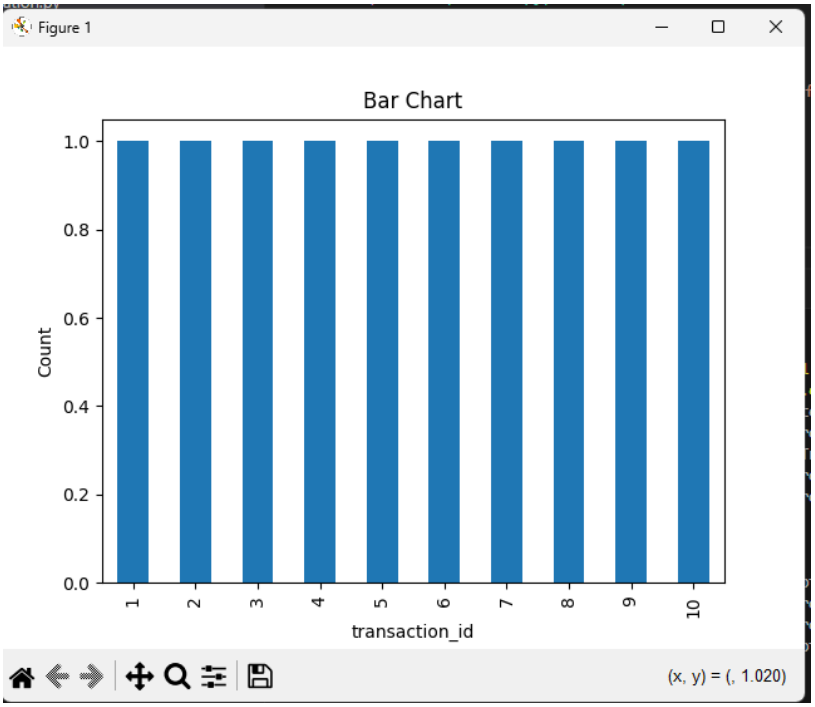
Histogram: Transaction_id,Frequency



Line Plot: Transaction id,Index



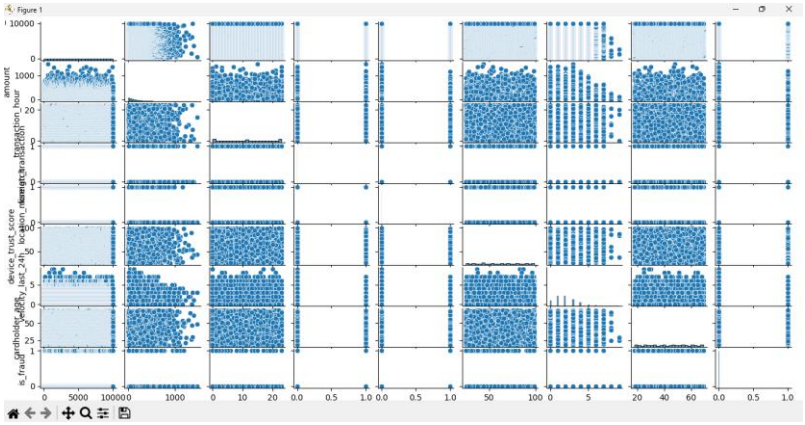
Bar Chart: Transaction_id,counts



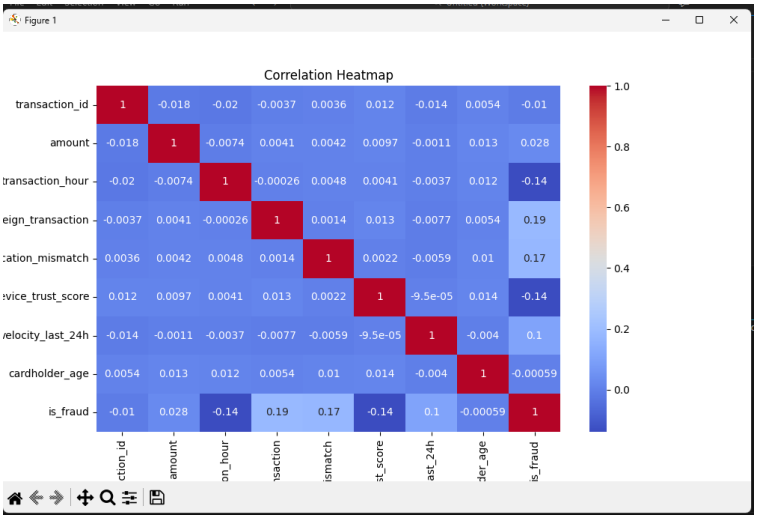
6.Multivariate Analysis:

Multivariate Analysis studies how multiple transaction features together help identify fraud in the credit card datast.

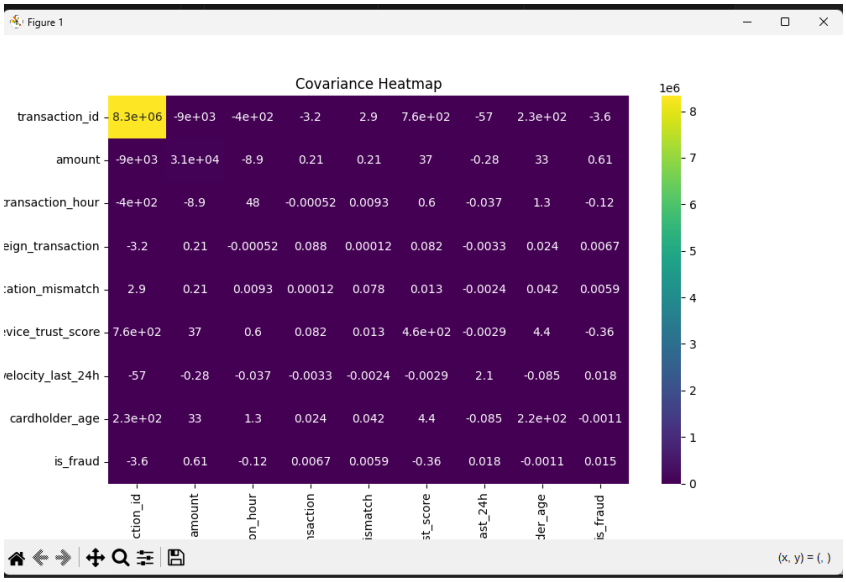
Pair Plot:



Correlation Heatmap:



Covariance Heatmap:



7. Handling Outliers

File Name:

Outliers were detected using the Interquartile Range (IQR) method.

Findings:

Transaction Amount contained extreme values representing unusually large transactions

These outliers were either capped or removed to avoid skewed analysis

This step improved data consistency and visualization clarity.

Logic:

Calculate Q1(25th percentile) and Q3(75th percentile).

Define bounds: Lower = Q1 - 1.5 *IQR; Upper = Q3 + 1.5*IQR

```
PS D:\project1.py> & D:/project1.py/md/Scripts/Activate.ps1
(md) PS D:\project1.py> & D:/project1.py/md/Scripts/python.exe "d:/project1.py/7)Handling_outliers.py"
      transaction_id  amount  transaction_hour merchant_category  ...  device_trust_score  velocity_last_24h  cardholder_age  is_fraud
0                  1   84.47               22       Electronics  ...                  66                  3              40         0
1                  2  541.82                3            Travel  ...                  87                  1              64         0
2                  3  237.01               17           Grocery  ...                  49                  1              61         0
3                  4  164.33                4           Grocery  ...                  72                  3              34         0
4                  5   30.53               15              Food  ...                  79                  0              44         0
...              ...     ...              ...               ...  ...                 ...                ...             ...       ...
9995            9996  350.91               22              Food  ...                  99                  4              37         0
9996            9997  410.04                5          Clothing  ...                  70                  3              25         0
9997            9998  527.75               21       Electronics  ...                  44                  2              45         0
9998            9999   91.20                2       Electronics  ...                  38                  0              37         0
9999           10000   44.06                2          Clothing  ...                  38                  0              66         0

[10000 rows x 10 columns]
Dataset shape before outlier handling: (10000, 10)
Dataset shape after outlier handling: (7782, 10)
   transaction_id  amount  transaction_hour merchant_category  ...  device_trust_score  velocity_last_24h  cardholder_age  is_fraud
0               1   84.47               22       Electronics  ...                  66                  3              40         0
2               3  237.01               17           Grocery  ...                  49                  1              61         0
4               5   30.53               15              Food  ...                  79                  0              44         0
5               6   30.53               13          Clothing  ...                  90                  2              46         0
6               7   10.77               18            Travel  ...                  48                  1              28         0

[5 rows x 10 columns]
(md) PS D:\project1.py>
```

8.Communicate Findings and Insights

File Name:

Key Insights:

Fraud transactions are rare but highly impactful

Foreign transactions show higher fraud probability

High-risk merchants are strongly associated with fraud cases

Transaction timing and velocity play a crucial role in detecting fraud

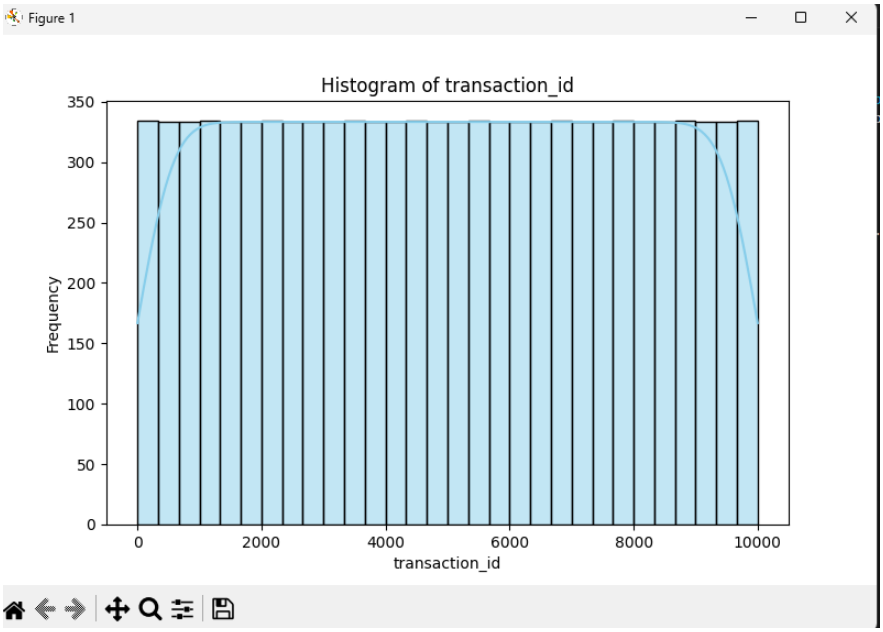
Dataset is highly imbalanced, requiring special handling during modelling

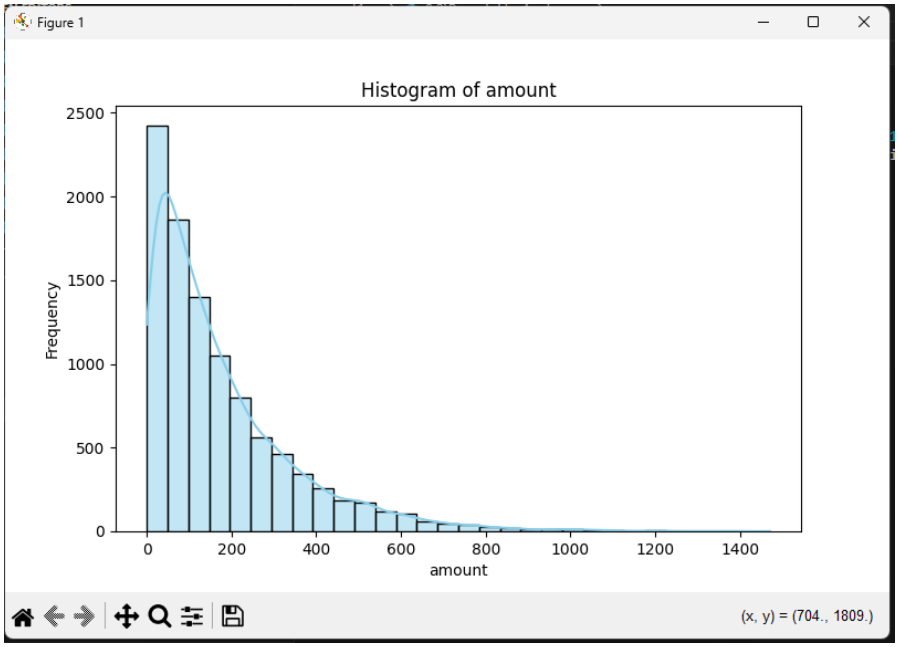
```
Dash is running on http://127.0.0.1:8050/

 * Serving Flask app '8.1)Dash'
 * Debug mode: on
```

8.Probability Analysis:

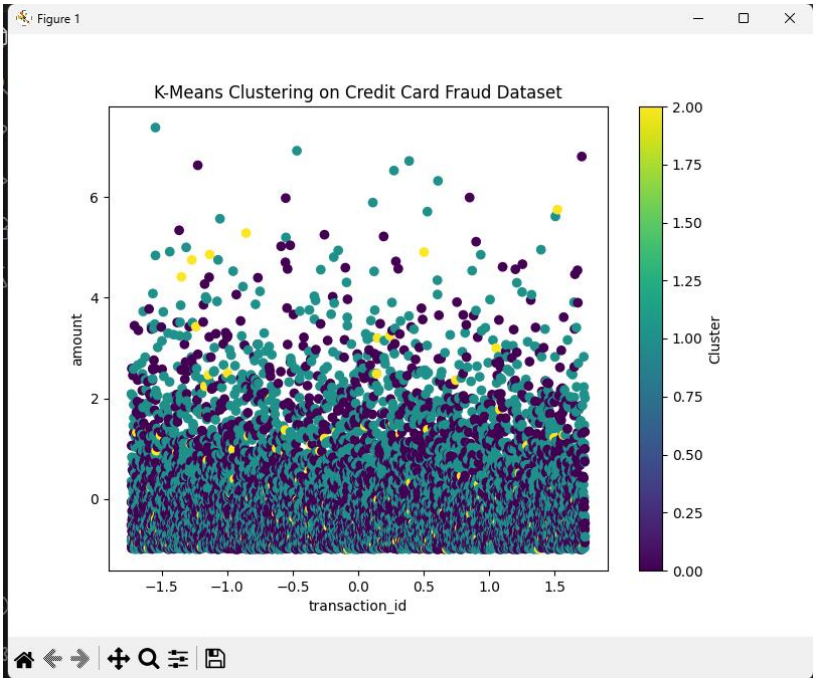
Range and variance were calculated for key numeric variables to understand their spread and variability, and histograms with KDE curves were plotted to observe probability distributions**.**





9.K-Means Clustering:

The standardized numeric dataset was clustered into three segments using the K-Means algorithm.

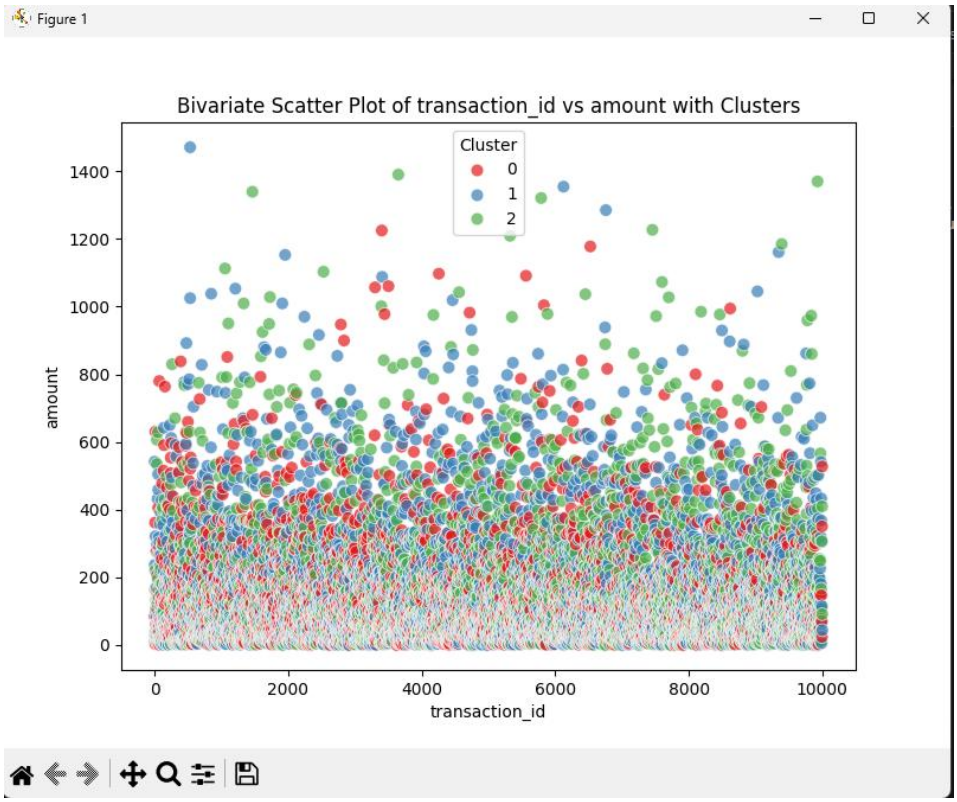


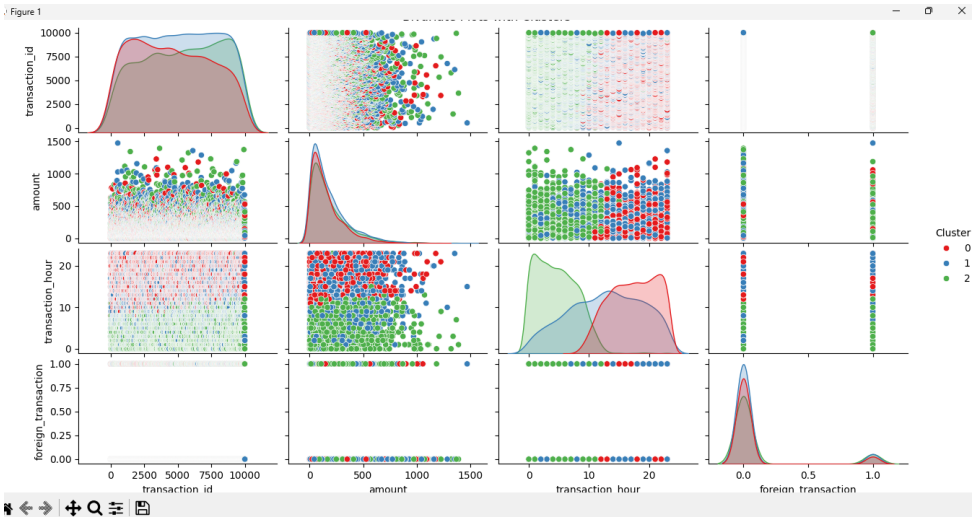
9.1 Cluster Visualization:

A scatter plot and pair plot were generated to visually distinguish clusters and observe inter- feature relationships.

Scatter plot - Bivariate Analysis:

This analysis examines the relationship between two variables simultaneously.





KMeans Pairplot

10. Conclusion:

This project successfully performed an end-to-end Exploratory Data Analysis on the Credit Card Fraud dataset. The data was inspected, cleaned, transformed, and visualized to uncover meaningful patterns related to fraudulent behavior. EDA revealed strong indicators such as foreign transactions, transaction amount, and velocity that contribute to fraud detection. These insights provide a strong foundation for building predictive models and real-time fraud monitoring systems.