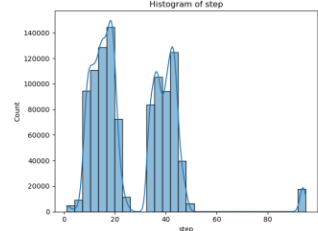


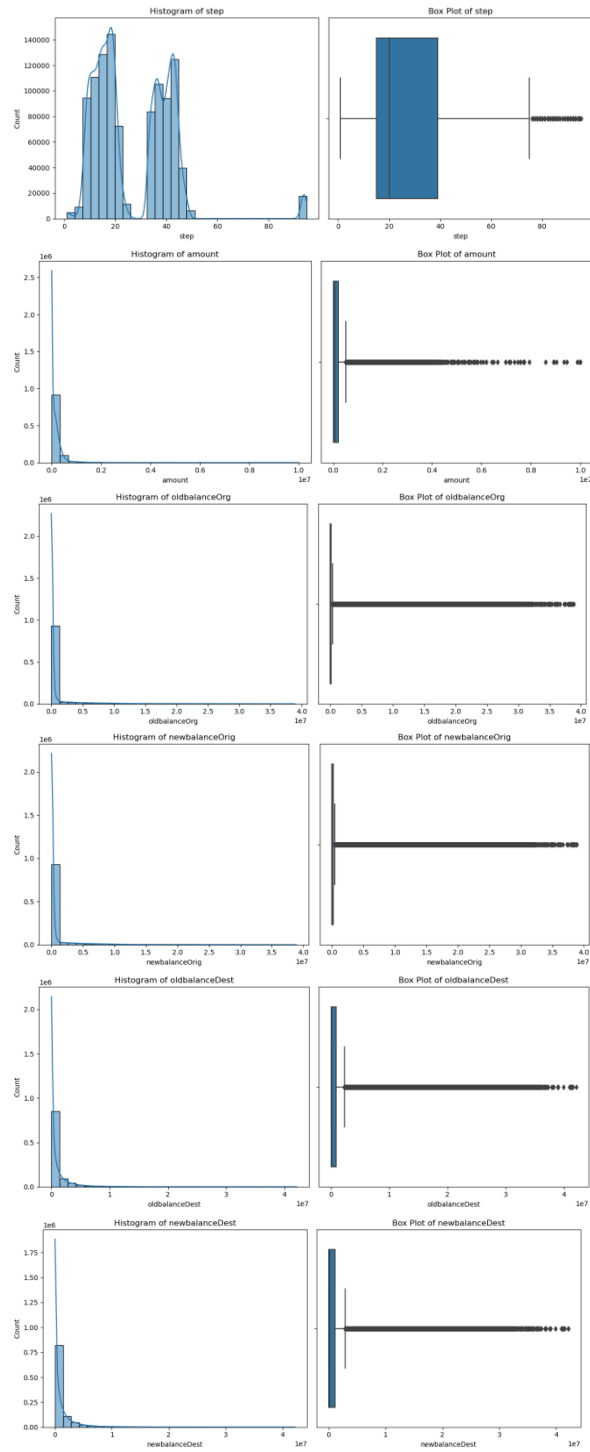
## Data Collection and Preprocessing Phase

Date	15 July 2024
Team ID	team-740137
Project Title	Online Payments Fraud Detection
Maximum Marks	6 Marks

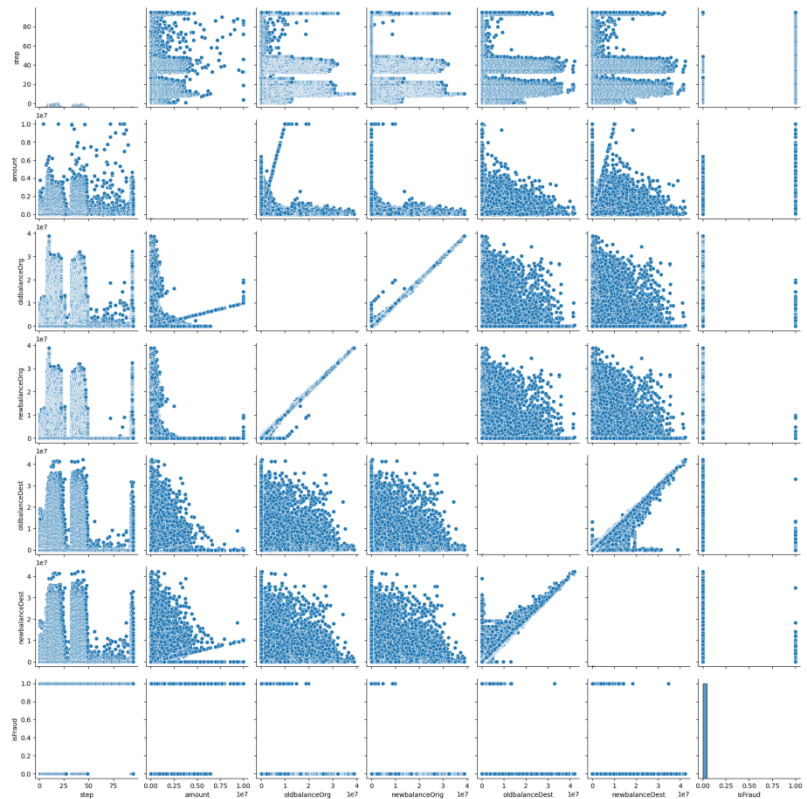
## Data Exploration and Preprocessing Report

Dataset variables will be statistically analyzed to identify patterns and outliers, with Python employed for preprocessing tasks like normalization and feature engineering. Data cleaning will address missing values and outliers, ensuring quality for subsequent analysis and modeling, and forming a strong foundation for insights and predictions.

Section	Description																																																																								
Data Overview	<div><pre>[10]: df.describe()</pre><pre>[10]:</pre><table><thead><tr><th></th><th>step</th><th>amount</th><th>oldbalanceOrig</th><th>newbalanceOrig</th><th>oldbalanceDest</th><th>newbalanceDest</th><th>isFraud</th></tr></thead><tbody><tr><td>count</td><td>1.048575e+06</td><td>1.048575e+06</td><td>1.048575e+06</td><td>1.048575e+06</td><td>1.048575e+06</td><td>1.048575e+06</td><td>1.048575e+06</td></tr><tr><td>mean</td><td>2.696617e+01</td><td>1.586670e+05</td><td>8.740095e+05</td><td>8.938089e+05</td><td>9.781600e+05</td><td>1.114198e+06</td><td>1.089097e-03</td></tr><tr><td>std</td><td>1.562325e+01</td><td>2.649409e+05</td><td>2.971751e+06</td><td>3.008271e+06</td><td>2.296780e+06</td><td>2.416593e+06</td><td>3.298351e-02</td></tr><tr><td>min</td><td>1.000000e+00</td><td>1.000000e+01</td><td>0.000000e+00</td><td>0.000000e+00</td><td>0.000000e+00</td><td>0.000000e+00</td><td>0.000000e+00</td></tr><tr><td>25%</td><td>1.500000e+01</td><td>1.214907e+04</td><td>0.000000e+00</td><td>0.000000e+00</td><td>0.000000e+00</td><td>0.000000e+00</td><td>0.000000e+00</td></tr><tr><td>50%</td><td>2.000000e+01</td><td>7.634333e+04</td><td>1.600200e+04</td><td>0.000000e+00</td><td>1.263772e+05</td><td>2.182604e+05</td><td>0.000000e+00</td></tr><tr><td>75%</td><td>3.900000e+01</td><td>2.137619e+05</td><td>1.366420e+05</td><td>1.746000e+05</td><td>9.159235e+05</td><td>1.149808e+06</td><td>0.000000e+00</td></tr><tr><td>max</td><td>9.500000e+01</td><td>1.000000e+07</td><td>3.890000e+07</td><td>3.890000e+07</td><td>4.210000e+07</td><td>4.220000e+07</td><td>1.000000e+00</td></tr></tbody></table></div>		step	amount	oldbalanceOrig	newbalanceOrig	oldbalanceDest	newbalanceDest	isFraud	count	1.048575e+06	1.048575e+06	1.048575e+06	1.048575e+06	1.048575e+06	1.048575e+06	1.048575e+06	mean	2.696617e+01	1.586670e+05	8.740095e+05	8.938089e+05	9.781600e+05	1.114198e+06	1.089097e-03	std	1.562325e+01	2.649409e+05	2.971751e+06	3.008271e+06	2.296780e+06	2.416593e+06	3.298351e-02	min	1.000000e+00	1.000000e+01	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	25%	1.500000e+01	1.214907e+04	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	50%	2.000000e+01	7.634333e+04	1.600200e+04	0.000000e+00	1.263772e+05	2.182604e+05	0.000000e+00	75%	3.900000e+01	2.137619e+05	1.366420e+05	1.746000e+05	9.159235e+05	1.149808e+06	0.000000e+00	max	9.500000e+01	1.000000e+07	3.890000e+07	3.890000e+07	4.210000e+07	4.220000e+07	1.000000e+00
	step	amount	oldbalanceOrig	newbalanceOrig	oldbalanceDest	newbalanceDest	isFraud																																																																		
count	1.048575e+06	1.048575e+06	1.048575e+06	1.048575e+06	1.048575e+06	1.048575e+06	1.048575e+06																																																																		
mean	2.696617e+01	1.586670e+05	8.740095e+05	8.938089e+05	9.781600e+05	1.114198e+06	1.089097e-03																																																																		
std	1.562325e+01	2.649409e+05	2.971751e+06	3.008271e+06	2.296780e+06	2.416593e+06	3.298351e-02																																																																		
min	1.000000e+00	1.000000e+01	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00																																																																		
25%	1.500000e+01	1.214907e+04	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00																																																																		
50%	2.000000e+01	7.634333e+04	1.600200e+04	0.000000e+00	1.263772e+05	2.182604e+05	0.000000e+00																																																																		
75%	3.900000e+01	2.137619e+05	1.366420e+05	1.746000e+05	9.159235e+05	1.149808e+06	0.000000e+00																																																																		
max	9.500000e+01	1.000000e+07	3.890000e+07	3.890000e+07	4.210000e+07	4.220000e+07	1.000000e+00																																																																		
Univariate Analysis	<div><h3>Data Visualization</h3><pre>[20]: numerical_columns = df.select_dtypes(include=['int64', 'float64']).columns</pre><pre>for column in numerical_columns:</pre><pre>    plt.figure(figsize=(12, 5))</pre><pre>    # Histogram</pre><pre>    plt.subplot(1, 2, 1)</pre><pre>    sns.histplot(df[column], bins=30, kde=True)</pre><pre>    plt.title(f'Histogram of {column}')</pre><pre>    # Box Plot</pre><pre>    plt.subplot(1, 2, 2)</pre><pre>    sns.boxplot(x=df[column])</pre><pre>    plt.title(f'Box Plot of {column}')</pre><pre>plt.tight_layout()</pre><pre>plt.show()</pre><div><div><h4>Histogram of step</h4></div><div><h4>Box Plot of step</h4></div></div></div>																																																																								



	<div><div>Histogram of isFraud</div><div>Box Plot of isFraud</div></div>																																																								
Bivariate Analysis	<div><div>Heat Map</div><div><pre>[22]: if len(numerical_columns) &gt; 1: plt.figure(figsize=(12, 8)) correlation_matrix = df[numerical_columns].corr() sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=0.5) plt.title('Correlation Heatmap') plt.show()</pre></div><div><div>Correlation Heatmap</div><table><tr><td>step</td><td>1</td><td>-0.026</td><td>-0.0068</td><td>-0.0072</td><td>-0.0023</td><td>-0.02</td><td>0.045</td></tr><tr><td>amount</td><td>-0.026</td><td>1</td><td>0.0049</td><td>-0.0011</td><td>0.22</td><td>0.31</td><td>0.13</td></tr><tr><td>oldbalanceOrig</td><td>-0.0068</td><td>0.0049</td><td>1</td><td>1</td><td>0.093</td><td>0.064</td><td>0.0038</td></tr><tr><td>newbalanceOrig</td><td>-0.0072</td><td>-0.0011</td><td>1</td><td>1</td><td>0.095</td><td>0.064</td><td>-0.0094</td></tr><tr><td>oldbalanceDest</td><td>-0.0023</td><td>0.22</td><td>0.093</td><td>0.095</td><td>1</td><td>0.98</td><td>-0.0076</td></tr><tr><td>newbalanceDest</td><td>-0.02</td><td>0.31</td><td>0.064</td><td>0.064</td><td>0.98</td><td>1</td><td>-0.0005</td></tr><tr><td>isFraud</td><td>0.045</td><td>0.13</td><td>0.0038</td><td>-0.0094</td><td>-0.0076</td><td>-0.0005</td><td>1</td></tr></table></div></div>	step	1	-0.026	-0.0068	-0.0072	-0.0023	-0.02	0.045	amount	-0.026	1	0.0049	-0.0011	0.22	0.31	0.13	oldbalanceOrig	-0.0068	0.0049	1	1	0.093	0.064	0.0038	newbalanceOrig	-0.0072	-0.0011	1	1	0.095	0.064	-0.0094	oldbalanceDest	-0.0023	0.22	0.093	0.095	1	0.98	-0.0076	newbalanceDest	-0.02	0.31	0.064	0.064	0.98	1	-0.0005	isFraud	0.045	0.13	0.0038	-0.0094	-0.0076	-0.0005	1
step	1	-0.026	-0.0068	-0.0072	-0.0023	-0.02	0.045																																																		
amount	-0.026	1	0.0049	-0.0011	0.22	0.31	0.13																																																		
oldbalanceOrig	-0.0068	0.0049	1	1	0.093	0.064	0.0038																																																		
newbalanceOrig	-0.0072	-0.0011	1	1	0.095	0.064	-0.0094																																																		
oldbalanceDest	-0.0023	0.22	0.093	0.095	1	0.98	-0.0076																																																		
newbalanceDest	-0.02	0.31	0.064	0.064	0.98	1	-0.0005																																																		
isFraud	0.045	0.13	0.0038	-0.0094	-0.0076	-0.0005	1																																																		
Multivariate Analysis	<div><div><pre>[21]: if len(numerical_columns) &gt; 1: sns.pairplot(df[numerical_columns]) plt.show()</pre></div></div>																																																								



## Data Preprocessing Code Screenshots

### Loading Data

#### Reading the Dataset

```
[3]: df = dfpd.read_csv('OFDdata.csv')
df
```

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrg	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	1	PAYMENT	9839.64	C1231006815	170136.00	160296.36	M1979787155	0.00	0.00	0	0
1	1	PAYMENT	1864.28	C1666544295	21249.00	19384.72	M2044282225	0.00	0.00	0	0
2	1	TRANSFER	181.00	C1305486145	181.00	0.00	C553264065	0.00	0.00	1	0
3	1	CASH_OUT	181.00	CB40083671	181.00	0.00	C38997010	21182.00	0.00	1	0
4	1	PAYMENT	11668.14	C2048537720	41554.00	29885.86	M1230701703	0.00	0.00	0	0
...	...	...	...	...	...	...	...	...	...	...	...
1048570	95	CASH_OUT	132557.35	C1179511630	479803.00	347245.65	C435674507	484329.37	616886.72	0	0
1048571	95	PAYMENT	9917.36	C1956161225	90545.00	80627.64	M668364942	0.00	0.00	0	0
1048572	95	PAYMENT	14140.05	C2037964975	20545.00	6404.95	M1355182933	0.00	0.00	0	0
1048573	95	PAYMENT	10020.05	C1633237354	90605.00	80584.95	M1964992463	0.00	0.00	0	0
1048574	95	PAYMENT	11450.03	C1264356443	80584.95	69134.92	M677577406	0.00	0.00	0	0

1048575 rows x 11 columns

## Handling Null Values

### Handling Null Values

```
[11]: df.isnull().sum()
[11]: step      0
     type      0
     amount    0
     oldbalanceOrig  0
     newbalanceOrig  0
     oldbalanceDest  0
     newbalanceDest  0
     isfraud      0
     dtype: int64

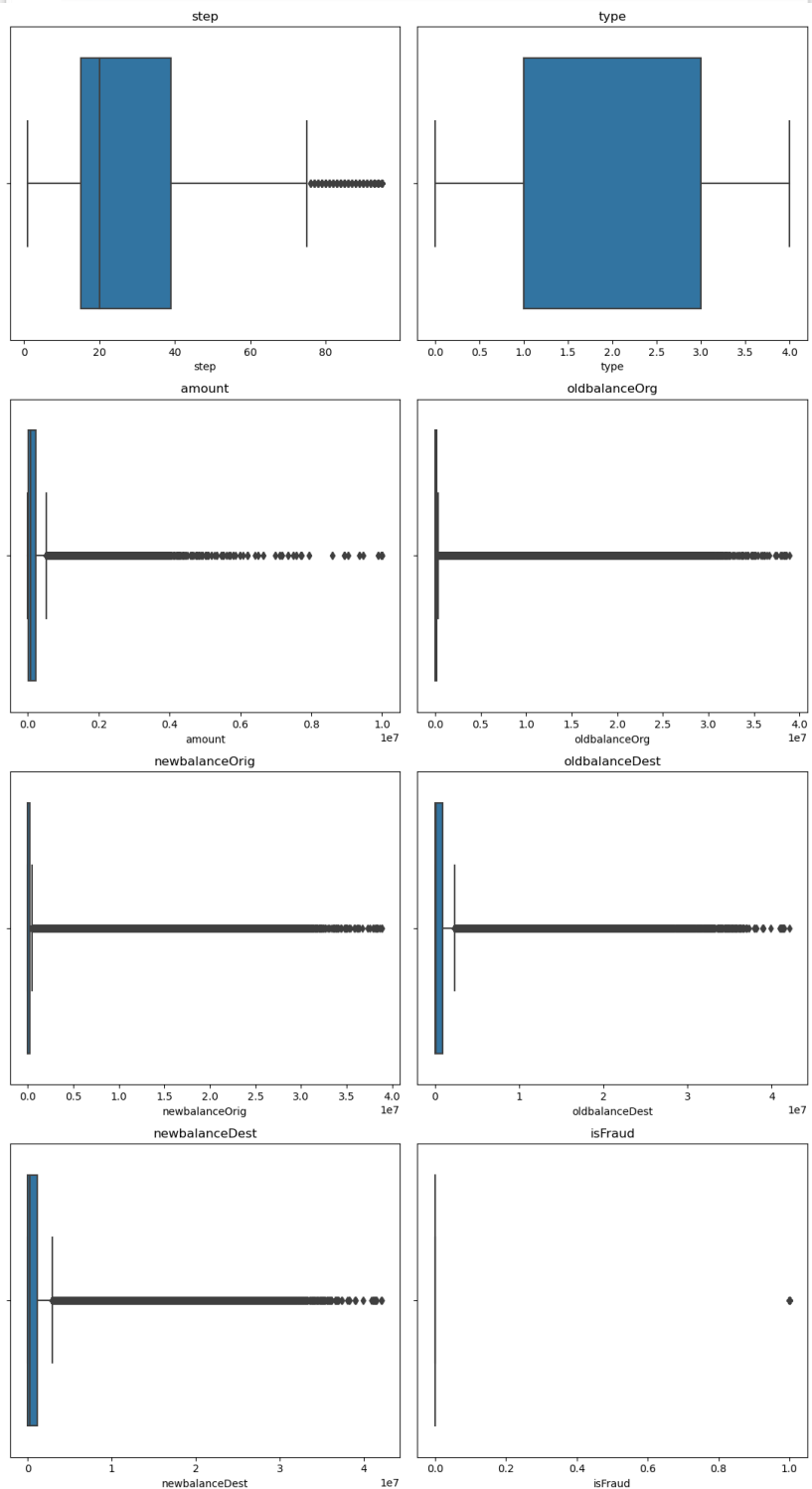
[12]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1048575 entries, 0 to 1048574
Data columns (total 8 columns):
#   Column              Non-Null Count  Dtype
---  -
0   step                1048575 non-null  int64
1   type                1048575 non-null  object
2   amount              1048575 non-null  float64
3   oldbalanceOrig      1048575 non-null  float64
4   newbalanceOrig      1048575 non-null  float64
5   oldbalanceDest      1048575 non-null  float64
6   newbalanceDest      1048575 non-null  float64
7   isfraud              1048575 non-null  int64
dtypes: float64(5), int64(2), object(1)
memory usage: 64.0+ MB
```

## Viewing outliers

### Viewing Outliers

```
[16]: num_columns = df.shape[1]
      num_rows = (num_columns + 1) // 2

      plt.figure(figsize=(11, num_rows + 5))
      for i, column in enumerate(df.columns):
          plt.subplot(num_rows, 2, i + 1)
          sns.boxplot(x=df[column])
          plt.title(column)
      plt.tight_layout()
      plt.show()
```



## Handling outliers

### Handling Outliers

```
[17]: Q1 = df.quantile(0.25)
      Q3 = df.quantile(0.75)
      IQR = Q3 - Q1

      outliers = ((df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 * IQR))).any(axis=1)
      print("Number of outliers:", outliers.sum())

Number of outliers: 332172
```

## Saved Processed Data

```
[18]: df_cleaned = df[~outliers]
      print("Data shape after removing outliers:", df_cleaned.shape)

Data shape after removing outliers: (716403, 8)

[19]: df.describe(include='all')
```

	step	type	amount	oldbalanceOrig	newbalanceOrig	oldbalanceDest	newbalanceDest	isfraud
count	1.048575e+06	1.048575e+06	1.048575e+06	1.048575e+06	1.048575e+06	1.048575e+06	1.048575e+06	1.048575e+06
mean	2.696617e+01	1.713400e+00	1.586670e+05	8.740095e+05	8.938089e+05	9.781600e+05	1.114198e+06	1.089097e-03
std	1.562325e+01	1.345007e+00	2.649409e+05	2.971751e+06	3.008271e+06	2.296780e+06	2.416593e+06	3.298351e-02
min	1.000000e+00	0.000000e+00	1.000000e-01	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	1.500000e+01	1.000000e+00	1.214907e+04	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
50%	2.000000e+01	1.000000e+00	7.634333e+04	1.600200e+04	0.000000e+00	1.263772e+05	2.182604e+05	0.000000e+00
75%	3.900000e+01	3.000000e+00	2.137619e+05	1.366430e+05	1.746000e+05	9.159235e+05	1.149808e+06	0.000000e+00
max	9.500000e+01	4.000000e+00	1.000000e+07	3.890000e+07	3.890000e+07	4.210000e+07	4.220000e+07	1.000000e+00