Data Collection and Preprocessing Phase

Date	20 July 2025
Team ID	SWUID20250184320
Project Title	Online Payment Fraud Detection
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

Section	Description
Data Overview	Basic statistics, dimensions, and structure of the data.
Univariate Analysis	Exploration of individual variables (mean, median, mode, etc.).
Bivariate Analysis	Relationships between two variables (correlation, scatter plots).
Multivariate Analysis	Patterns and relationships involving multiple variables.
Outliers and Anomalies	Identification and treatment of outliers.

Data Preprocessing Code Screenshots

Loading Data

```
# Load Dataset

dataset=pd.read_csv(r"C:\Users\hp\Desktop\Fraud_Payment_Detection2\Fraud_Detection_Dataset\PS_20174392719_1491204439457_log.csv")
dataset.head(10)
```

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	1	PAYMENT	9839.64	C1231006815	170136.00	160296.36	M1979787155	0.0	0.00	0	0
1	1	PAYMENT	1864.28	C1666544295	21249.00	19384.72	M2044282225	0.0	0.00	0	0
2	1	TRANSFER	181.00	C1305486145	181.00	0.00	C553264065	0.0	0.00	1	0
3	1	CASH_OUT	181.00	C840083671	181.00	0.00	C38997010	21182.0	0.00	1	0
4	1	PAYMENT	11668.14	C2048537720	41554.00	29885.86	M1230701703	0.0	0.00	0	0
5	1	PAYMENT	7817.71	C90045638	53860.00	46042.29	M573487274	0.0	0.00	0	0

Handling Missing Data

```
# Data Preprocessing

# print("Shape:", dataset.shape)
# print(dataset.info())

print("\nChecking for null values:")
dataset.isnull().sum()
dataset.fillna(0, inplace=True)
```

Data Transformation

```
from sklearn.preprocessing import MinMaxScaler, StandardScaler

# Select the columns to scale
features_to_scale = ['amount', 'oldbalanceOrg', 'newbalanceOrig']

# Option 1: Normalization (Min-Max Scaling)
minmax_scaler = MinMaxScaler()
df[features_to_scale] = minmax_scaler.fit_transform(df[features_to_scale])
```

Feature Engineering

```
import pandas as pd

# Creating a flag for high-value transactions
df['is_high_amount'] = df['amount'].apply(lambda x: 1 if x > 100000 else 0)

# Creating a new feature: balance difference
df['balance_diff'] = df['oldbalanceOrg'] - df['newbalanceOrig']

# One-hot encoding for transaction type
df = pd.get_dummies(df, columns=['type'], drop_first=True)

# Optional: log transformation to handle skewed amounts
import numpy as np
df['log_amount'] = np.log1p(df['amount'])  # log1p avoids issues with 0
```

Save Processed Data

```
# Save model
joblib.dump(model, "best_fraud_model_xgb.pkl")
```