# Data Collection and Preprocessing Phase

| | |
|---|---|
| Date | 7 June 2024 |
| Team ID | 739730 |
| Project Title | Online payment Fraud Detection Using ML |
| 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 | This section provides a comprehensive summary of the data being analyzed for fraud detection. It includes information about the dataset such as the number of transactions, number of fraudulent vs. non-fraudulent transactions, and basic statistics (mean, median, mode) for each feature. |
| Univariate Analysis | This analysis focuses on each feature individually to understand its distribution and detect any anomalies. It involves visualizations such as histograms and box plots, and statistical measures like mean, variance, and skewness. |
| Bivariate Analysis | This section examines the relationships between pairs of features to identify patterns that might indicate fraud. Techniques include scatter plots, correlation matrices, and cross-tabulation. |
| Multivariate Analysis | This analysis looks at multiple features simultaneously to detect complex patterns and interactions. Techniques include PCA (Principal Component Analysis), clustering, and multivariate plots. |
| Outliers and Anomalies | This section focuses on identifying and handling outliers and anomalies that may indicate fraud or data quality issues. Techniques include IQR (Interquartile Range), Z-score, and anomaly detection algorithms. |

## Data Preprocessing Code Screenshots

### Loading Data

```
df=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/datasets/onlinefraud.csv')
df
```

| | step | type | amount | nameOrig | oldbalanceOrg | newbalanceOrig | nameDest | oldbalanceDes |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | PAYMENT | 9839.64 | C1231006815 | 170136.00 | 160296.36 | M1979787155 | 0.0 |
| 1 | 1 | PAYMENT | 1864.28 | C1666544295 | 21249.00 | 19384.72 | M2044282225 | 0.0 |
| 2 | 1 | TRANSFER | 181.00 | C1305486145 | 181.00 | 0.00 | C553264065 | 0.0 |
| 3 | 1 | CASH_OUT | 181.00 | C840083671 | 181.00 | 0.00 | C38997010 | 21182.0 |
| 4 | 1 | PAYMENT | 11668.14 | C2048537720 | 41554.00 | 29885.86 | M1230701703 | 0.0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 6362615 | 743 | CASH_OUT | 339682.13 | C786484425 | 339682.13 | 0.00 | C776919290 | 0.0 |
| 6362616 | 743 | TRANSFER | 6311409.28 | C1529008245 | 6311409.28 | 0.00 | C1881841831 | 0.0 |
| 6362617 | 743 | CASH_OUT | 6311409.28 | C1162922333 | 6311409.28 | 0.00 | C1365125890 | 68488.8 |
| 6362618 | 743 | TRANSFER | 850002.52 | C1685995037 | 850002.52 | 0.00 | C2080388513 | 0.0 |
| 6362619 | 743 | CASH_OUT | 850002.52 | C1280323807 | 850002.52 | 0.00 | C873221189 | 6510099.1 |

6362620 rows × 11 columns

### Handling Missing Data

```
df.isnull().sum()
```

```
step              0
type              0
amount            0
nameOrig          0
oldbalanceOrg     0
newbalanceOrig    0
nameDest          0
oldbalanceDest    0
newbalanceDest    0
isFraud           0
isFlaggedFraud    0
dtype: int64
```

### Data Transformation

```
def transformationPlot(feature):
  plt.figure(figsize=(12,5))
  plt.subplot(1,2,1)
  # Handle potential infinite values
  sns.distplot(feature[np.isfinite(feature)])
  plt.subplot(1,2,2)
  stats.probplot(feature[np.isfinite(feature)], plot=plt)
```

### Feature Engineering

```
le=LabelEncoder()
df['nameDest']=le.fit_transform(df['nameDest'])
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6362620 entries, 0 to 6362619
Data columns (total 11 columns):
 #   Column          Dtype
---  ------          -----
 0   step            int64
 1   type            int64
 2   amount          float64
 3   nameOrig        int64
 4   oldbalanceOrg   float64
 5   newbalanceOrig  float64
 6   nameDest        int64
 7   oldbalanceDest  float64
 8   newbalanceDest  float64
 9   isFraud         int64
 10  isFlaggedFraud  int64
dtypes: float64(5), int64(6)
memory usage: 534.0 MB
```

### Save Processed Data

```
import pickle
pickle.dump(dtc,open('model.pkl','wb'))
```

```
from google.colab import files
files.download('model.pkl')
```