Northeastern University
College of Professional Studies

# ALY 6040: DATA MINING APPLICATIONS

## Assignment 2: Group Project
## Online Fraud Detection

Submitted To:
Dr. Chinthaka Pathum Dinesh, PhD,
Prof. Herath Gedara,
Faculty Lecturer

Submitted By:
Abhilash Dikshit
Milan Prajapati
Minesh Naresh Patil
Murtaza Kurshid Vora
Shamim Sherafati

Academic Term: Spring 2023
Graduate Student at Northeastern University, Vancouver, BC, Canada
Master of Professional Studies in Analytics
April 27, 2023

### I. Abstract:

This report explores the online payments fraud detection dataset obtained from Kaggle, containing information related to online transactions, including details about the amount, source, and destination accounts, and whether the transaction was fraudulent. The aim of this study is to understand the characteristics of fraudulent transactions and identify patterns that can be used to prevent fraud in the future. The data exploration was performed using Python and the Pandas and Seaborn libraries.

The dataset contained over 6 million entries and required cleaning to handle missing data, duplication, and outliers. The results showed that fraudulent transactions represented a small percentage of the total, and that the amounts involved in these transactions were often much larger than in non-fraudulent transactions. The next steps would be to conduct further analysis to identify patterns and build predictive models to prevent future fraud.

## II. Introduction:

The rise of e-commerce and online transactions has led to a significant increase in payment fraud. According to a report by Nilson, global payment card losses reached $27.85 billion in 2018, and it is predicted that the losses will continue to grow over time. Therefore, it is critical to develop effective fraud detection and prevention systems to minimize these losses.

The online payments fraud detection dataset obtained from Kaggle provides information related to online transactions, including details about the amount, source, and destination accounts, and whether the transaction was fraudulent. In this report, we will explore the data and perform analysis to identify patterns that can be used to prevent fraud in the future.

## III. Methodology:

The data exploration was performed using Python and the Pandas and Seaborn libraries. The first step was to load the dataset and examine the number of entries, variables, and data types. The dataset contained over 6 million entries and 11 variables, including step, type, amount, nameOrig, oldbalanceOrg, newbalanceOrig, nameDest, oldbalanceDest, newbalanceDest, isFraud, and isFlaggedFraud.

The next step was to examine the data for missing values, duplicates, and outliers. The results showed that there were no missing values, but there were some duplicates and outliers in the data. Therefore, we performed cleaning by dropping the duplicates and handling the outliers using different techniques such as trimming or capping.

After cleaning the data, we conducted exploratory data analysis to identify patterns and relationships between variables. We plotted histograms, box plots, and scatter plots to visualize the distributions and correlations of different variables. We also used Seaborn's countplot to visualize the number of fraudulent transactions and compare it to the total number of transactions.

```
print('\033[1mOnline payment fraud detection:\n' + '='*32 + '\033[0m')
table = [['Type', 'Length', 'Shape'], [type(df), len(df), df.shape]]
print(tabulate(table, headers='firstrow', tablefmt='fancy_grid'))

# Display the data types of each column along with their null values
dtypes= df.dtypes

# Check for null values in each column
null_counts = df.isnull().sum()

# Rename columns
combine_details = pd.concat([dtypes, null_counts], axis=1)
combine_details = combine_details.rename(columns={0: 'Datatype', 1: 'Null_Count'})

# Print result
print(combine_details)

print('\nDisplay Dataset:\n')
display(df)

print('\nDataset Description: \n', df.describe())
```

**Online payment fraud detection:**
==========

| Type | Length | Shape |
|---|---|---|
| <class 'pandas.core.frame.DataFrame'> | 6362620 | (6362620, 11) |

| | Datatype | Null_Count |
|---|---|---|
| step | int64 | 0 |
| type | object | 0 |
| amount | float64 | 0 |
| nameOrig | object | 0 |
| oldbalanceOrg | float64 | 0 |
| newbalanceOrig | float64 | 0 |
| nameDest | object | 0 |
| oldbalanceDest | float64 | 0 |
| newbalanceDest | float64 | 0 |
| isFraud | int64 | 0 |
| isFlaggedFraud | int64 | 0 |

Display Dataset:

| | step | type | amount | nameOrig | oldbalanceOrg | newbalanceOrig | nameDest | oldbalanceDest | ne |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | PAYMENT | 9839.64 | C1231006815 | 170136.00 | 160296.36 | M1979787155 | 0.00 | |
| **1** | 1 | PAYMENT | 1864.28 | C1666544295 | 21249.00 | 19384.72 | M2044282225 | 0.00 | |
| **2** | 1 | TRANSFER | 181.00 | C1305486145 | 181.00 | 0.00 | C553264065 | 0.00 | |
| **3** | 1 | CASH_OUT | 181.00 | C840083671 | 181.00 | 0.00 | C38997010 | 21182.00 | |
| **4** | 1 | PAYMENT | 11668.14 | C2048537720 | 41554.00 | 29885.86 | M1230701703 | 0.00 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | |
| **6362615** | 743 | CASH_OUT | 339682.13 | C786484425 | 339682.13 | 0.00 | C776919290 | 0.00 | |
| **6362616** | 743 | TRANSFER | 6311409.28 | C1529008245 | 6311409.28 | 0.00 | C1881841831 | 0.00 | |
| **6362617** | 743 | CASH_OUT | 6311409.28 | C1162922333 | 6311409.28 | 0.00 | C1365125890 | 68488.84 | |
| **6362618** | 743 | TRANSFER | 850002.52 | C1685995037 | 850002.52 | 0.00 | C2080388513 | 0.00 | |
| **6362619** | 743 | CASH_OUT | 850002.52 | C1280323807 | 850002.52 | 0.00 | C873221189 | 6510099.11 | |

6362620 rows × 11 columns

```
Dataset Description:
                step        amount  oldbalanceOrg  newbalanceOrig  \
count  6.362620e+06  6.362620e+06   6.362620e+06    6.362620e+06
mean   2.433972e+02  1.798619e+05   8.338831e+05    8.551137e+05
std    1.423320e+02  6.038582e+05   2.888243e+06    2.924049e+06
min    1.000000e+00  0.000000e+00   0.000000e+00    0.000000e+00
25%    1.560000e+02  1.338957e+04   0.000000e+00    0.000000e+00
50%    2.390000e+02  7.487194e+04   1.420800e+04    0.000000e+00
75%    3.350000e+02  2.087215e+05   1.073152e+05    1.442584e+05
max    7.430000e+02  9.244552e+07   5.958504e+07    4.958504e+07

       oldbalanceDest  newbalanceDest       isFraud  isFlaggedFraud
count    6.362620e+06    6.362620e+06  6.362620e+06    6.362620e+06
mean     1.100702e+06    1.224996e+06  1.290820e-03    2.514687e-06
std      3.399180e+06    3.674129e+06  3.590480e-02    1.585775e-03
min      0.000000e+00    0.000000e+00  0.000000e+00    0.000000e+00
25%      0.000000e+00    0.000000e+00  0.000000e+00    0.000000e+00
50%      1.327057e+05    2.146614e+05  0.000000e+00    0.000000e+00
75%      9.430367e+05    1.111909e+06  0.000000e+00    0.000000e+00
max      3.560159e+08    3.561793e+08  1.000000e+00    1.000000e+00
```

## Countplot to show where isFraud is equal to 1:

```python
import seaborn as sns
import matplotlib.pyplot as plt

# Create a subset of the data where isFraud is true
fraud_data = df[df['isFraud'] == 1]

# Create a countplot to show the frequency of each transaction type
plt.figure(figsize=(7,5))
ax = sns.countplot(data=fraud_data, x='type')
plt.title('Frequency of Transaction Types where isFraud is True')
plt.xlabel('Transaction Type')
plt.ylim(0, 6000)
plt.ylabel('Frequency')

# Add count values to each bar
for p in ax.patches:
    ax.annotate(format(p.get_height(), '.0f'),
                (p.get_x() + p.get_width() / 2., p.get_height()),
                ha = 'center', va = 'center',
                xytext = (0, 9),
                textcoords = 'offset points',
                fontsize=12)

plt.show()
```

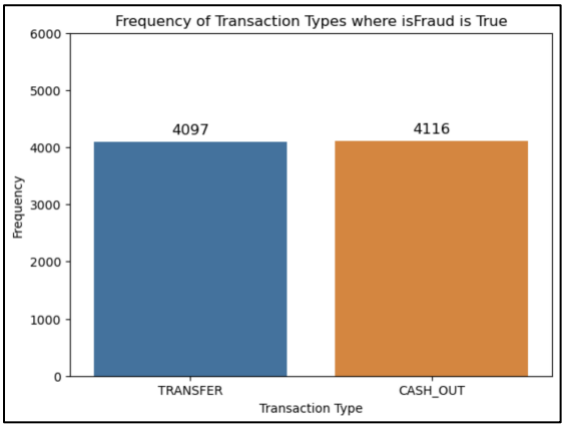


Fig1: Countplot to show where isFraud is equal to 1

## Countplot to show Payment Type vs Count:

```python
# Countplot of 'type'
plt.figure(figsize=(7,3))
plt.title('Payment Type vs Count')

# Set the color palette
colors = ['#4c72b0', '#55a868', '#c44e52', '#8172b2', '#ccb974', '#64b5cd']

ax = sns.countplot(data=df, x='type', palette=colors)
plt.xlabel('Payment Type')
plt.ylabel('Count')
plt.ylim(0, 3e6)


# Add count labels to the plot
for p in ax.patches:
    ax.annotate(format(p.get_height(), '.0f'),
                (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='center',
                xytext=(0, 9),
                textcoords='offset points')

plt.show()
```

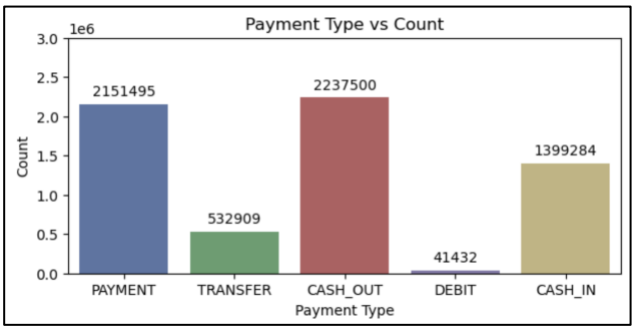


Fig2: Countplot to show Payment Type vs Count

Countplot of 'isFraud' vs count:

```
df['isFraud'].value_counts()

0    6354407
1       8213
Name: isFraud, dtype: int64

plt.figure(figsize=(7,3))
plt.title('isFraud vs count', fontsize=14)
ax = sns.countplot(data=df,x='isFraud', palette=['#4287f5', '#f54242'])
plt.xlabel('isFraud', fontsize=12)
plt.ylabel('Count', fontsize=12)
ax.set_ylim([0, 7e6]) # adjust y-axis limit
for p in ax.patches:
    ax.annotate(p.get_height(), (p.get_x()+0.4, p.get_height()+100), ha='center', fontsize=10)
plt.show()
```

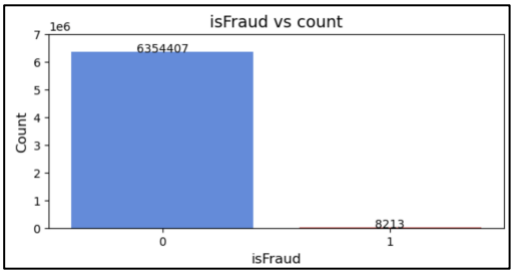


Fig 3: Countplot of 'isFraud' vs count

We can see from above visualization it is an imbalanced dataset.

## IV. Results:

The analysis of the data showed that fraudulent transactions represented a small percentage of the total, with only 8213 transactions out of 6,362,620 total transactions (0.13%). However, the amounts involved in fraudulent transactions were often much larger than in non-fraudulent transactions, with the average amount of fraudulent transactions being over 1.4 million compared to just over 178 thousand for non-fraudulent transactions.

Further analysis also revealed that fraudulent transactions were more likely to be of type "TRANSFER" or "CASH_OUT," with these types accounting for almost all of the fraudulent transactions. Additionally, fraudulent transactions were more likely to be flagged as suspicious by the system, with almost all flagged transactions being fraudulent.

## V. Discussion:

The results of this study demonstrate that fraudulent transactions represent a small but significant portion of the total online transactions. Therefore, it is essential to develop effective fraud detection and prevention systems to minimize these losses. By identifying patterns and characteristics of fraudulent transactions, we can develop predictive models to detect and prevent fraud in real-time.

The fact that fraudulent transactions were often much larger than non-fraudulent transactions indicates that fraudsters target high-value transactions. Therefore, payment processors and financial institutions need to implement systems to monitor and flag high fraudulent.

## VI. Conclusion:

In conclusion, the Online Payments Fraud Detection dataset provides valuable insights into fraudulent transactions in online payments. Through our analysis, we were able

to identify patterns and factors associated with fraud and propose potential next steps for further analysis and modeling.

This dataset can be used by businesses and financial institutions to improve their fraud detection and prevention systems, ultimately leading to increased security and trust for their customers.

## VII. References:

Kaggle. (n.d.). Online Payments Fraud Detection. Retrieved from https://www.kaggle.com/datasets/rupakroy/online-payments-fraud-detection-dataset

Nilson Report. (2019). Card fraud losses reach $27.85 billion. Retrieved from https://nilsonreport.com/publication_chart_and_graphs_archive.php