Sheridan

Sheridan College Institute of Technology and Advanced Learning

Course

Artificial Intelligence Applications

ENGI51071

Project Title

Fraud Prediction for Online Payments

Submitted by:

Mohamed Arslan (ID: 991708261)

Nehal Ur Rahman (ID: 991691259)

Submitted to:

Prof. Ameera Al-Karkhi

Term: Spring 2023

Table of Contents

Introduction

The beginning of the digital era has led to a profound transformation across industries, revolutionizing how transactions are conducted, information is exchanged, and business processes are streamlined. Central to this transformative wave is the financial sector, which has undergone a transformation in the way financial interactions occur. The increase of online transactions, driven by the increase of smartphones, digital wallets, and e-commerce platforms, has redefined the concept of convenience and accessibility.

However, within this digital revolution lies a persistent and threatening challenge – **the rise of credit card fraud**. As financial transactions increasingly migrate to the online domain, the boundaries that once determined the secure from the vulnerable has been blurred. The shadow of fraudulent activities has increased, threatening both consumers and financial institutions alike. The exponential growth of fraudulent transactions not only translates into colossal financial losses but also erodes the very foundation of trust upon which the financial ecosystem rests.

In this backdrop, the application of artificial intelligence (AI) emerges as a beacon of hope. The mix of data analysis and machine learning unveils the potential to mitigate the threat of credit card fraud. By harnessing the power of data-driven insights and predictive models, financial institutions can fortify their defense against fraudulent activities, safeguarding the interests of legitimate users and the integrity of the financial ecosystem.

In this project we delve into the intricacies of credit card fraud detection, unveiling the capabilities of AI in interpreting fraudulent patterns, illuminating the potential of predictive models, and unraveling the traces of feature engineering. Our aim is not only to unveil the technological power of AI but also to contribute to the collective understanding of how innovation can strengthen security in the digital age.

Background

In today's interconnected world, the transformation of financial transactions from the physical to the digital realm has not only redefined convenience but has also given rise to new challenges. The convenience of making payments, transferring funds, and conducting financial activities from the comfort of one's device is unparalleled. Yet, this very convenience is underpinned by intricate technological networks susceptible to exploitation.

The exponential growth of online transactions, particularly in the wake of the global pandemic, has propelled financial fraud into the spotlight. Fraudulent activities, ranging from unauthorized access to accounts to complex money laundering schemes, have infiltrated the digital landscape, leaving financial institutions grappling with an evolving threat.

To address this challenge, the domain of data science and machine learning has emerged as a impressive collaboration. By leveraging the power of data, algorithms, and predictive modeling, institutions can arm themselves with tools capable of distinguishing between legitimate and fraudulent transactions with remarkable accuracy. This convergence of technology and finance

holds the promise of mitigating financial losses, preserving trust, and fortifying the foundations of the digital financial ecosystem.

In the pursuit of this goal, our project delves into the "Financial Datasets for Fraud Detection" dataset curated by Edgar Lopez-Rojas. This dataset represents a fusion of real-world transaction patterns and synthesized data, providing a comprehensive list upon which we can give our analytical insights. By meticulously navigating this dataset, we aim to decode the patterns that differentiate legitimate financial interactions from fraudulent activities, thereby contributing to the collection of tools that can counter the forthcoming threat of credit card fraud.

Implementation

Project Objectives

The primary objectives of this project are as follows:

1. **Data Exploration and Preprocessing**:
    - Thoroughly explore the credit card transaction dataset to understand its structure and contents.
    - Address missing values and handle duplicates to ensure the integrity of the data.
2. **Visualize Key Features**:
    - Utilize data visualization techniques to present the distribution of transaction types, fraud vs. non-fraud transactions, and other relevant insights.
    - Generate correlation heatmaps to visualize the relationships between different features and the target variable.
3. **Model Building and Evaluation**:
    - Train a Decision Tree model to predict fraudulent transactions based on various transaction attributes.
    - Build a Random Forest model to further enhance predictive accuracy.
    - Evaluate model performance using accuracy metrics and visualizations.

Detailed Methodology

Data Collection and Preprocessing

Data Source: Financial Datasets for Fraud Detection(https://www.kaggle.com/code/netzone/eda-and-fraud-detection/input)

For our project, we selected the Synthetic Financial Datasets by Edgar Lopez-Rojas for Fraud Detection dataset from Kaggle. This dataset, which contains 6,362,620 records and 11 columns, simulates real mobile money transactions using real samples from a month's financial logs of a

mobile money service in an African country. Out of these transactions, 99.87% are legitimate, and only 0.13% are fraudulent. This dataset was scaled down from its original size for Kaggle usage.

```
In [19]: dataset_size = data.shape
         print("Dataset Size (Rows, Columns):", dataset_size)

         Dataset Size (Rows, Columns): (6362620, 11)
```

```
In [6]: # Data Exploration

        print(data.head())
        print(data.describe())

           step      type    amount     nameOrig  oldbalanceOrg  newbalanceOrig  \
        0     1   PAYMENT   9839.64  C1231006815       170136.0       160296.36
        1     1   PAYMENT   1864.28  C1666544295        21249.0        19384.72
        2     1  TRANSFER    181.00  C1305486145          181.0            0.00
        3     1  CASH_OUT    181.00   C840083671          181.0            0.00
        4     1   PAYMENT  11668.14  C2048537720        41554.0        29885.86

              nameDest  oldbalanceDest  newbalanceDest  isFraud  isFlaggedFraud
        0  M1979787155             0.0             0.0        0               0
        1  M2044282225             0.0             0.0        0               0
        2   C553264065             0.0             0.0        1               0
        3    C38997010         21182.0             0.0        1               0
        4  M1230701703             0.0             0.0        0               0
                       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
```

The dataset comprises 11 columns, each serving a specific purpose:

- step: Represents a time unit where 1 step equals 1 hour.
- type: Denotes the type of online transaction.
- amount: Indicates the transaction amount.
- nameOrig: Identifies the customer initiating the transaction.
- oldbalanceOrg: Reflects the customer's balance before the transaction.
- newbalanceOrig: Represents the customer's balance after the transaction.
- nameDest: transaction receiver.
- oldbalanceDest: The receiver's initial balance before the transaction.
- newbalanceDest: The receiver's balance after the transaction.
- isFraud: Indicates if the transaction is fraudulent.
- isFlaggedFraud: Marks transfers exceeding 200,000 in a single transaction.

Data Preprocessing

Prior to diving into the area of analysis, it is necessary to clean and shape the data to ensure its suitability for machine learning. Our preprocessing encompasses multiple stages, from handling missing values and eliminating duplicates to mapping categorical attributes to their numerical counterparts.

```
In [4]: # Data Cleaning and Preprocessing
        # Handling Missing Values

        missing_values = data.isnull().sum()
        print("Missing Values:\n", missing_values)
```

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

```
In [5]: # Handling Duplicates

        duplicates = data.duplicated().sum()
        print("Number of Duplicates:", duplicates)
```

```
Number of Duplicates: 0
```

But as we can see that dataset, we have curated has zero missing values as well duplicates. Therefore, there is no need of replacing the missing values or deleting the duplicates.

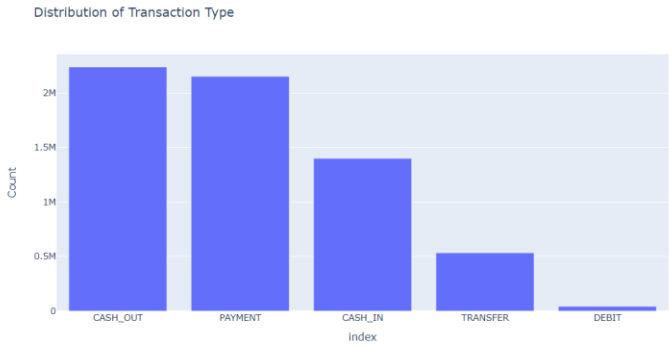
Exploratory Data Analysis

Transaction Type Distribution

Now we move forward with a visual exploration of transaction types. Utilizing a bar chart, we explain the distribution of various transaction types, showing the common types of financial interactions.

```
In [7]: # Exploring Transaction Type Distribution using a bar chart

        type_counts = data["type"].value_counts()
        fig_type_counts = px.bar(type_counts, x=type_counts.index, y=type_counts.values,
                                 labels={'x': 'Transaction Type', 'y': 'Count'},
                                 title="Distribution of Transaction Type")
        fig_type_counts.show()
```



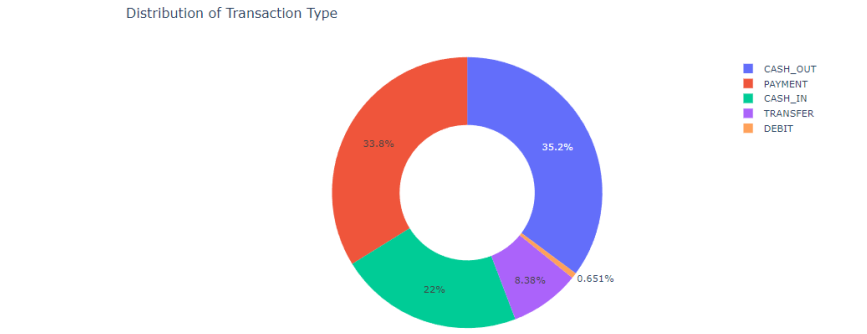
```
In [8]: # Exploring transaction type using a pie chart

        print(data.type.value_counts())

        type = data["type"].value_counts()
        transactions = type.index
        quantity = type.values

        import plotly.express as px
        figure = px.pie(data,
                        values=quantity,
                        names=transactions,hole = 0.5,
                        title="Distribution of Transaction Type")
        figure.show()
```

```
CASH_OUT    2237500
PAYMENT     2151495
CASH_IN     1399284
TRANSFER     532909
DEBIT         41432
Name: type, dtype: int64
```



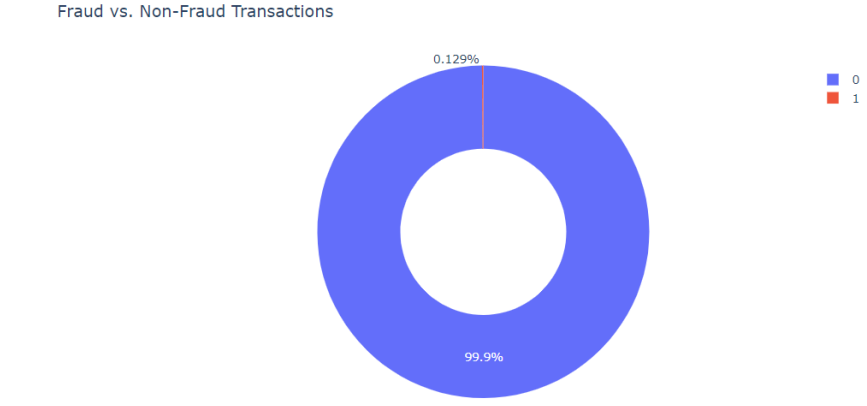
From the above graphs we can see that the highest distribution of transaction is CASG_OUT (35.2%) followed by PAYMENT (33.8%), CASHIN (22%), TRANSFER (8.38%) and DEBIT (0.651%)

Fraud vs. Non-Fraud Transactions

Delving deeper, we embark on a comparative analysis between fraudulent and non-fraudulent transactions. Employing a pie chart, we see the proportion of fraud within the dataset, laying the groundwork for our subsequent predictive models.

```
In [9]: # Exploring Fraud vs. Non-Fraud Transactions using a pie chart

        fraud_vs_nonfraud = data["isFraud"].value_counts()
        fig_fraud_vs_nonfraud = px.pie(fraud_vs_nonfraud,
                                       values=fraud_vs_nonfraud.values,
                                       names=fraud_vs_nonfraud.index,
                                       hole=0.5,
                                       title="Fraud vs. Non-Fraud Transactions")
        fig_fraud_vs_nonfraud.show()
```



Here, we can see that the number of fraud transactions accounts for 0.129% of the total transactions.

Correlation Analysis

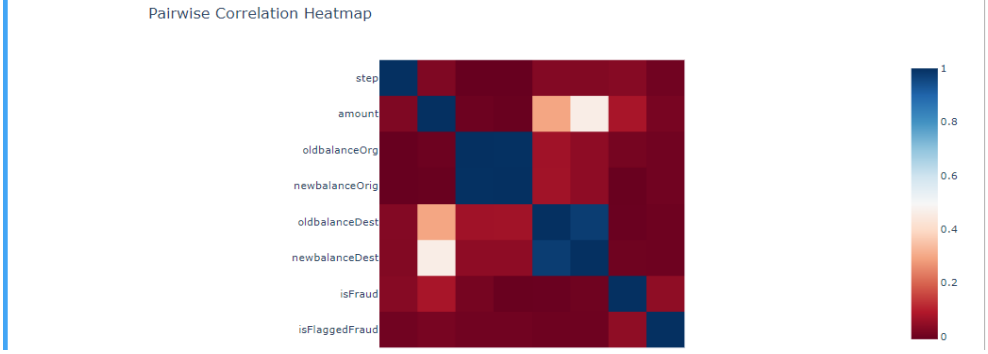
Discovering the intricate relationships between attributes is a fundamental aspect of our analysis. Our exploration extends to uncovering correlations between various attributes with respect to the FRAUD variable, contributing to a more profound understanding of the underlying dynamics that distinguish fraudulent transactions.

```
# Pairwise Correlation Heatmap
fig_correlation_heatmap = px.imshow(correlation,
                                    x=correlation.index,
                                    y=correlation.columns,
                                    color_continuous_scale='RdBu',
                                    title="Pairwise Correlation Heatmap")
fig_correlation_heatmap.show()
```

```
C:\Users\nehal\AppData\Local\Temp\ipykernel_17972\3419244696.py:3: FutureWarning:

The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only v
alid columns or specify the value of numeric_only to silence this warning.
```

```
isFraud           1.000000
amount            0.076688
isFlaggedFraud    0.044109
step              0.031578
oldbalanceOrg     0.010154
newbalanceDest    0.000535
oldbalanceDest   -0.005885
newbalanceOrig   -0.008148
Name: isFraud, dtype: float64
```



Evaluation

The success of this project will be evaluated based on the following criteria:

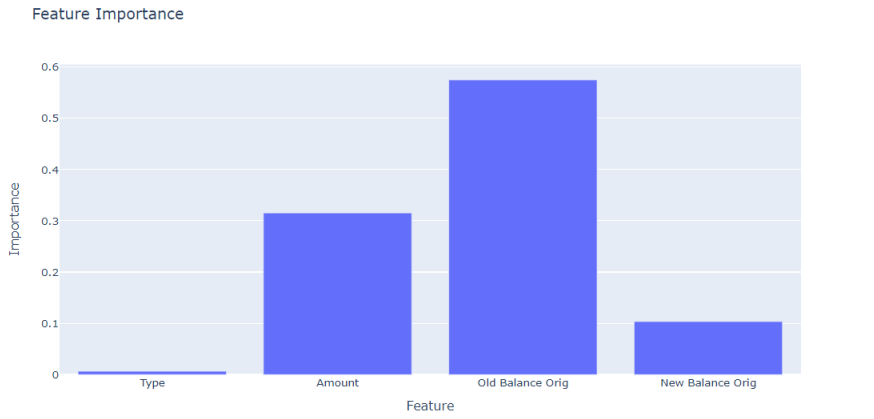
1. **Model Accuracy**:
    - The accuracy of the Decision Tree and Random Forest models in predicting fraudulent transactions will be a primary evaluation metric.
2. **Feature Importance**:
    - Understanding which transaction attributes contribute most to fraud detection will provide insights into potential indicators of fraud.
3. **Practical Applicability**:
    - The project's impact on real-world fraud detection practices and its potential to be integrated into operational systems will be considered.

Feature Importance Analysis

An integral facet of model interpretation lies in deciphering feature importance. By unraveling the significance of various attributes within the model's decision-making process, we gain insights into the contributing factors that drive accurate fraud detection.

```
In [14]: # Visualizing Feature Importance using a bar chart

feature_importance = model.feature_importances_
fig_feature_importance = px.bar(x=['Type', 'Amount', 'Old Balance Orig', 'New Balance Orig'],
                                y=feature_importance,
                                labels={'x': 'Feature', 'y': 'Importance'},
                                title="Feature Importance")
fig_feature_importance.show()
```



Interpretation of Feature Importance

Looking into the depths of model interpretation, we analyze the feature importance scores. This revelation not only enhances our understanding of the intricate dynamics of fraud detection but also empowers us to refine our predictive models for optimal accuracy.

Results and Discussion

```
In [12]: # splitting the data into training and testing sets

from sklearn.model_selection import train_test_split
x = np.array(data[["type", "amount", "oldbalanceOrg", "newbalanceOrig"]])
y = np.array(data[["isFraud"]])
```

```
In [13]: # training a machine learning model: Decision Tree model

from sklearn.tree import DecisionTreeClassifier
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.10, random_state=42)
model = DecisionTreeClassifier()
model.fit(xtrain, ytrain)
print(model.score(xtest, ytest))
```

```
0.9997391011878755
```

After training the model with Decision Tree model we get an accuracy of 99.97%. This high accuracy percentage will ensure that the predictions by this model will be highly accurate.

In [47]:
```
# Making a prediction using the trained Decision Tree model

#features = [type, amount, oldbalanceOrg, newbalanceOrig]
features = np.array([[4, 2000, 4000, 2000]])
print(model.predict(features))
```

```
['No Fraud']
```

In [43]:
```
# Making a prediction using the trained Decision Tree model

#features = [type, Amount, oldbalanceOrg, newbalanceOrig]
features = np.array([[4, 2000, 4000, 0]])
print(model.predict(features))
```

```
['Fraud']
```

Model Performance Comparison

As the final step we do a comprehensive comparison of the Decision Tree and Random Forest models. Through evaluation, we see which model is stronger. Seeing the screenshot below we conclude that the Random Forest model has a better accuracy.

In [20]:
```
# Define features (X) and target (y)
X = data[["type", "amount", "oldbalanceOrg", "newbalanceOrig"]]
y = data["isFraud"]

# Split the data into training and testing sets
xtrain, xtest, ytrain, ytest = train_test_split(X, y, test_size=0.10, random_state=42)

# Train Decision Tree model
decision_tree_model = DecisionTreeClassifier()
decision_tree_model.fit(xtrain, ytrain)
decision_tree_predictions = decision_tree_model.predict(xtest)
decision_tree_accuracy = accuracy_score(ytest, decision_tree_predictions)

# Train Random Forest model
random_forest_model = RandomForestClassifier()
random_forest_model.fit(xtrain, ytrain)
random_forest_predictions = random_forest_model.predict(xtest)
random_forest_accuracy = accuracy_score(ytest, random_forest_predictions)

# Compare model accuracies
print("Decision Tree Accuracy:", decision_tree_accuracy)
print("Random Forest Accuracy:", random_forest_accuracy)
```

```
Decision Tree Accuracy: 0.9997406728674666
Random Forest Accuracy: 0.9997689631001065
```

Benefits of the Analysis

The analysis of credit card transactions offers several key benefits:

1. **Fraud Detection and Prevention**:
    - By identifying patterns associated with fraudulent transactions, financial institutions can enhance their ability to detect and prevent fraud in real-time.
    - Improved fraud detection leads to reduced financial losses and increased customer trust.
2. **Operational Efficiency**:
    - Automated fraud detection models streamline the process of identifying suspicious transactions, reducing the need for manual review of every transaction.
    - This efficiency allows financial institutions to allocate resources more effectively.

3. **Customer Experience**:
    - Accurate fraud detection minimizes false positives, ensuring that legitimate transactions are not unnecessarily flagged as fraudulent.
    - Customers experience fewer disruptions and smoother transaction processes.
4. **Strategic Decision-Making**:

Insights gained from analyzing transaction data can inform strategic decisions related to risk assessment, security measures, and fraud prevention strategies.

Conclusion

In the world of modern finance, the rise of online transactions has brought convenience and complexity hand in hand. Our analysis into credit card fraud detection unveils the potential of technology to safeguard financial integrity.

The Random Forest model emerged as a better model with better accuracy, effectively identifying fraudulent activities. Its grouping with decision trees demonstrated the power of collaboration in predictive accuracy.

Feature importance analysis underscored the significance of transaction type, amount, and initial balances in fraud detection, providing crucial insights for the industry.

Looking forward, the path of progress extends. Real-time fraud detection and collaborative efforts hold the promise of an even more secure financial landscape. As we embrace innovation, we contribute to a strong ecosystem where digital transactions increase with confidence.

In conclusion, our project shows the combined effect of technology and finance. Therefore, with the help of AI, we strengthen the defense against fraud, contributing to a future where financial interactions are secure, efficient, and trustworthy.

References

- Shahrukh, S. (May 22). Online Payment Fraud Detection. Medium. https://medium.com/@shahrukhshl0/online-payment-fraud-detection-b8ba96304dd3

- NetZone. EDA and Fraud Detection Dataset. Kaggle. https://www.kaggle.com/code/netzone/eda-and-fraud-detection/data

- Stripe. (June 27, 2023). How Machine Learning Works for Payment Fraud Detection and Prevention. Stripe. https://stripe.com/en-ca/resources/more/how-machine-learning-works-for-payment-fraud-detection-and-prevention#:~:text=Anomaly%20detection,activities%20that%20may%20indicate%20fraud.

- SEON. Fraud Detection with Machine Learning. SEON. https://seon.io/resources/fraud-detection-with-machine-learning/