Fraud Detection in Digital Transactions

By – Prasad Somvanshi

Problem Statement:

The increasing prevalence of fraudulent activities in digital transactions poses a significant challenge for financial institutions. The project aims to develop an efficient algorithm to identify and flag fraudulent transactions, reducing the risk of letting them go undetected.

About Data:

Data Source: The data used for this analysis is a synthetically generated digital transactions dataset using the PaySim simulator. The dataset simulates mobile money transactions based on real transaction logs from a mobile money service in an African country.

Dataset Description:

- step: Represents a unit of time in the real world, where each step corresponds to 1 hour.
- type: Represents the type of transaction (CASH-IN, CASH-OUT, DEBIT, PAYMENT, TRANSFER).
- amount: Amount of the transaction in the local currency.
- nameOrig: Customer who initiated the transaction.
- oldbalanceOrg: Initial balance before the transaction for the originator.
- newbalanceOrig: New balance after the transaction for the originator.
- nameDest: Customer who is the recipient of the transaction.
- oldbalanceDest: Initial balance before the transaction for the recipient.
- newbalanceDest: New balance after the transaction for the recipient.
- isFraud: Indicates whether the transaction is fraudulent (1 for fraudulent, 0 for not fraudulent).
- isFlaggedFraud: Flags illegal attempts according to the business model (1 for flagged, 0 for not flagged).

Model Used:

Machine Learning Model: Gaussian Naive Bayes

Justification: The Naive Bayes model has been chosen for its performance in the initial evaluation, showcasing a high recall in identifying fraudulent transactions, which is crucial to minimize false negatives.

Approach:

Feature Engineering:

1. Difference in Balance: Detects deviations between the amount debited from the sender and credited to the receiver.
2. Surge Indicator: Flags transactions with amounts greater than the 75th percentile threshold.
3. Frequency Indicator: Flags receivers who receive money from more than 20 different individuals.
4. Merchant Indicator: Flags transactions with merchant receivers.

Pre-processing Steps:

1. Balanced the target labels to address the imbalance issue.
2. Applied one-hot encoding to the 'type' feature.
3. Split the data into training and testing sets.
4. Standardized numerical columns.
5. Applied tokenization to customer and merchant names.

Model Building: Utilized various classification algorithms (Logistic Regression, Decision Tree, KNN, SVC, Naive Bayes, Random Forest) and selected Gaussian Naive Bayes as the final model.

Tools:

Programming Language: Python

Libraries and Frameworks: Pandas, NumPy, Seaborn, Matplotlib, Plotly, TensorFlow, Scikit-learn

Visualization Tools: Plotly, Matplotlib, Seaborn

ML Libraries: TensorFlow, Scikit-learn

Metrics Libraries: Scikit-learn

Outcome:

- The Naive Bayes model demonstrates high accuracy and recall in identifying fraudulent transactions.
- Confusion matrix and classification metrics showcase the model's performance, emphasizing a 100% recall for fraud transactions.

Personal Contribution:

- **Roles and Responsibilities:** Led the feature engineering process, implemented data pre-processing steps, and played a significant role in model selection.
- **Innovations or Unique Contributions:** Introduced novel features (Difference in Balance, Surge Indicator, Frequency Indicator, Merchant Indicator) to enhance model performance.

Project Demo/Visualizations:

- **Pivot Table Analysis:** Examined overall numbers using pivot function to identify patterns in fraudulent transactions.
- **Distribution of Amount:** Visualized the distribution of transaction amounts using box plots.
- **Confusion Matrix:** Plotted the confusion matrix to showcase model performance on the test data.

Conclusion:

The project successfully addressed the challenge of fraud detection in digital transactions, achieving a high recall in identifying fraudulent activities. The Naive Bayes model, with its tuned parameters, demonstrated efficiency in reducing false negatives while maintaining a reasonable level of accuracy. Automation of fraud detection with machine learning has the potential to significantly reduce investigation time and enhance customer experience. Ongoing efforts focus on further improving model performance and adaptability to evolving patterns in fraudulent activities.