**Fraudulent Transaction Detector**

Part 3

Arya Goyal (ag9961)

Gitesh Chinawalkar (gc3410)

**Introduction**

With the increasing prevalence of financial transactions conducted digitally, detecting and mitigating fraudulent activities has become paramount. This project integrates a robust database schema with a state-of-the-art machine learning model to identify fraudulent transactions in real-time. The end goal is to enhance security, reduce financial losses, and improve trust in digital financial ecosystems.

The database schema was carefully designed to store transactional, user, and fraud detection log data efficiently. A machine learning model analyzes patterns within this data to flag suspicious activities. The system operates seamlessly, leveraging both structured and unstructured data for analytics and predictions.

**Machine Learning Model Selection and Training**

1. Data Selection

The dataset used for training the model consists of transactional data, including features like transaction type, amount, sender and receiver balances, and flags for suspected fraud. Missing and anomalous values were handled carefully to ensure model integrity.

Features:

* amount: Transaction amount.
* oldbalanceOrg and newbalanceOrig: Initial and post-transaction balances of the sender.
* oldbalanceDest and newbalanceDest: Initial and post-transaction balances of the recipient.
* type: Type of transaction (e.g., CASH\_IN, CASH\_OUT, PAYMENT).
* isFraud: Label indicating whether the transaction is fraudulent.

The dataset was split into training, validation, and test sets in an 80:10:10 ratio.

1. **Algorithm Selection**

The following algorithms were considered:

1. **Logistic Regression**: For its simplicity and interpretability.
2. **Random Forest**: For handling feature interactions and imbalanced data.
3. **SVM**: For its ability to handle large datasets efficiently and deliver high accuracy.

The final model selected was **Random forest**, chosen for its superior performance in fraud detection tasks.

1. **Model Training and Insights Extraction**

The model was trained using:

* Training Data: To fit the model.
* Validation Data: For hyperparameter tuning.
* Test Data: To evaluate real-world performance.

Key Metrics:

* Precision, Recall, and F1 Score for imbalanced class evaluation.
* AUC-ROC for overall model effectiveness.

Features were engineered to include:

* Differences between old and new balances.
* Ratio of transaction amount to account balances.

The trained model effectively identified patterns associated with fraudulent behavior, such as high-value transactions with zero account balances.

1. **Business Use Cases**
2. Fraud Detection: Automatically flagging high-risk transactions.
3. Risk Profiling: Assigning fraud risk scores to accounts.
4. Regulatory Reporting: Providing reports on flagged transactions for compliance.
5. **Fraud Detection Model**

* **Data Import**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* **Exploratory data analysis**

A screen shot of a computer

Description automatically generated

A screen shot of a computer screen

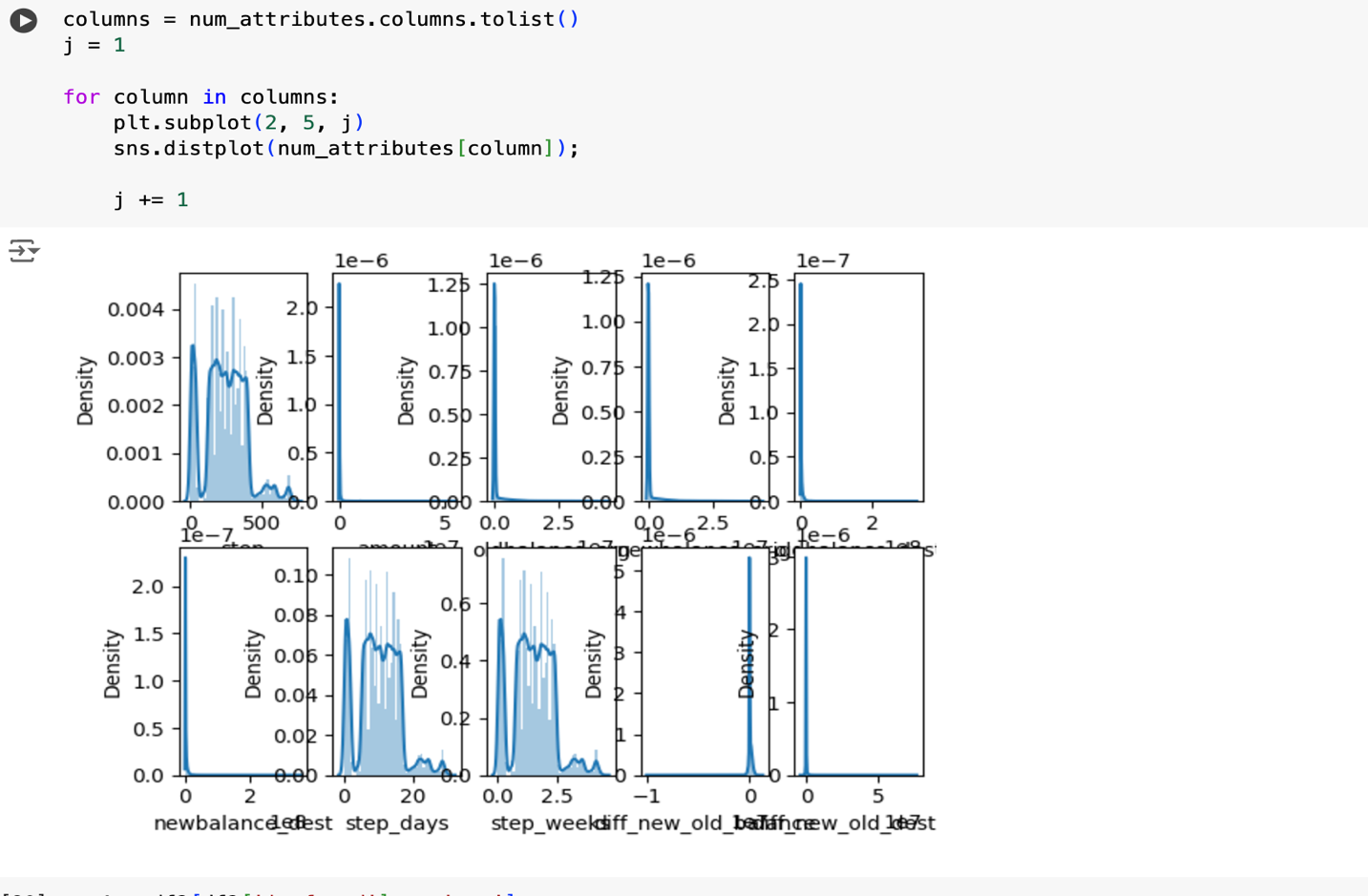
Description automatically generated

A screenshot of a graph

Description automatically generated

A screenshot of a computer

Description automatically generated



* **Data Preparation**

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

* **Machine learning model**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* **Compare results**

A screenshot of a computer

Description automatically generated

* **Github Link**

<https://github.com/Aryagoy/Fraud-transaction-detection>