BIA-678 – Big Data Technologies

Spark Sherlock:
Catching Fraudsters Faster than They Swipe!

Group Members:

Pooja Pande (20035197)

Vidhi Palan (20031836)

Pratik Kale (20036601)

Table of Contents

- Problem Statement
- Dataset Overview
- Data Dictionary
- Data Pre-processing
- Solution Approach
- Modeling Steps
- Performance Metrics
- Scaling Strategies
- Expected Outcomes

Problem Statement

Problem:

- Financial fraud in mobile money transactions is rapidly increasing.
- High volume and velocity of transactions make fraud detection challenging.
- Traditional approaches struggle to scale and lack real-time responsiveness.

Objective:

- Develop a scalable, real-time fraud detection model using big data technology (PySpark).
- Improve detection accuracy to minimize financial losses.

Dataset Overview

Dataset:

- Synthetic Mobile Money Transaction Dataset

Source:

- Synthetic data created by a wise guy with reference to African Real Money Transactions(kaggle)

Format & Size:

- CSV file format
- Size = **~ 6.3 million rows**

Domain:

- Financial services, specifically mobile money transactions.

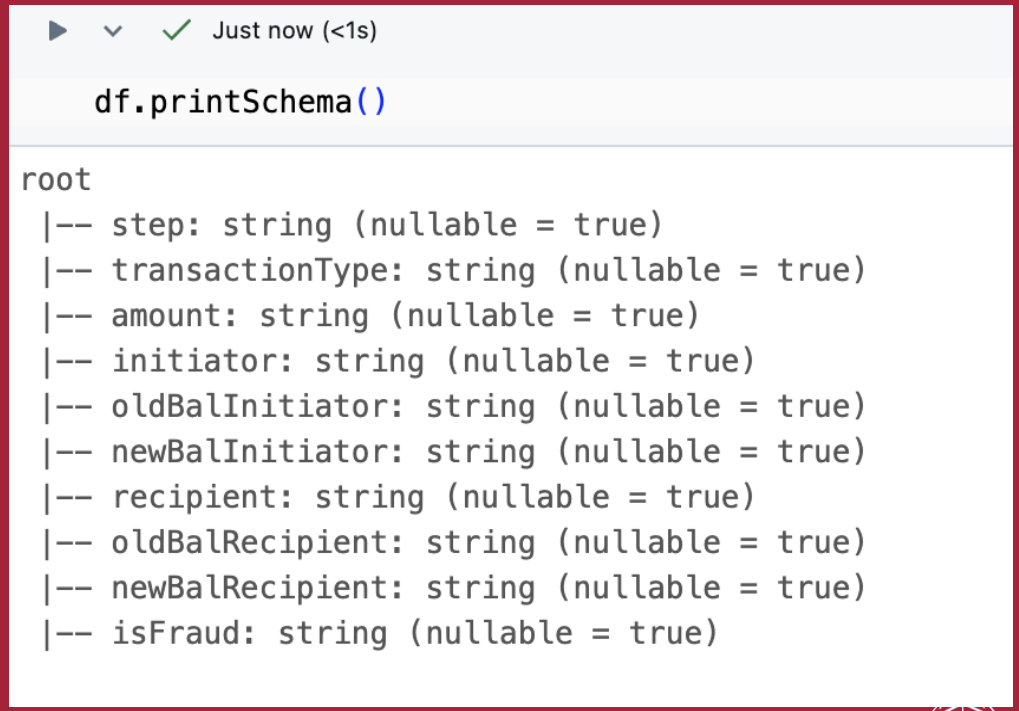
Data Dictionary

Key Columns:

- Transaction Type (e.g., deposit, withdrawal, payment)
- Initiator
- Amount
- Sender and Receiver IDs
- Sender/Receiver Balance Before and After Transaction
- Transaction Type Is Fraud (target variable)

Target Variable:

- Binary (0: Genuine transaction, 1: Fraudulent transaction)



Data Pre-processing

Data Cleaning & Transformation:

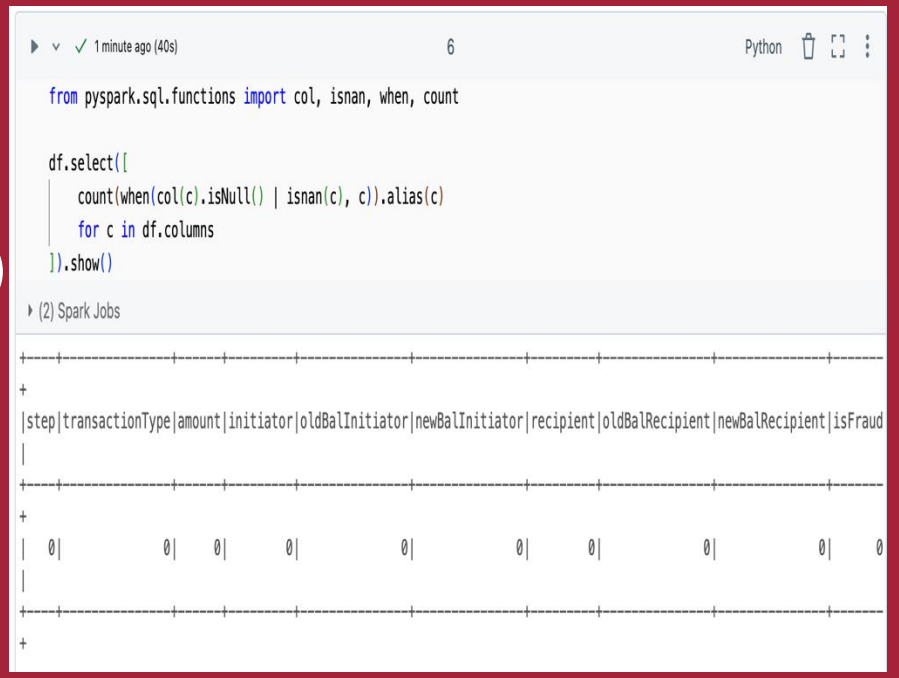
- Dropping irrelevant columns (Transaction ID, Date)
- Converting categorical variables to numerical indices (StringIndexer)

Feature Engineering:

- Created "Amount-to-Balance Ratio" to highlight anomalies.

Handling Missing Values:

- Initial dataset inspection for missing values.
- Appropriate handling strategies (imputation/removal).



Solution Approach

Modeling Algorithm:

- Random Forest Classifier

Model	Pros	Cons
Random Forest	High accuracy, handles imbalance, scalable	Less interpretable
Logistic Regression	Simple, interpretable, fast	May underperform on nonlinear data
Decision Tree	Easy to visualize and explain	Can overfit on large datasets
Gradient Boosted Trees	High performance, handles complex patterns	Slower training, harder to tune

Reason for Choosing Random Forest:

- Robustness to outliers
- Handles imbalanced datasets effectively
- Parallelizable, making it suitable for big data scenarios

Alternatives Considered:

- Logistic Regression
- Decision Trees
- Gradient Boosted Trees

Modeling Steps

Feature Selection:

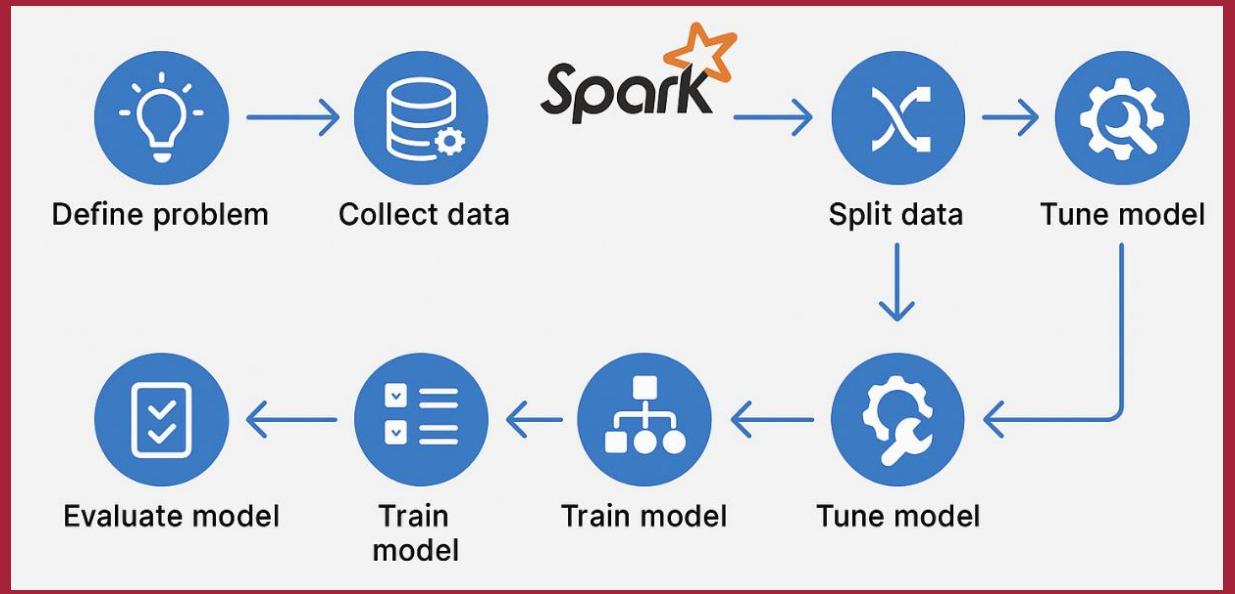
- Transaction amount, sender/receiver balances, derived ratio, transaction type.

Hyperparameter Tuning:

- Number of Trees: 100 initially
- Future scope for detailed tuning (max-depth, feature subset strategy)

Train-Test Split:

- 80% Training, 20% Testing



Performance Metrics

Primary Metric:

- F1-score

Reason for Choosing F1-score:

- Balances precision (false alarms) and recall (missed fraud)
- Effective for imbalanced fraud detection scenarios

Other Metrics Considered:

- Accuracy, Precision, Recall, ROC-AUC, PR-AUC (not primary due to imbalance and interpretation concerns)

Scaling Strategies

Data Scaling:

- Initial experiments with smaller subsets (10%, 25%, 50%, 75%) before processing full dataset
- Measure execution time and scalability trends

Cluster Scaling:

- Initially test on smaller clusters (single-node)
- Progressively scale out to multi-node clusters (2, 4, 6+ nodes)
- Demonstrate execution time reduction with cluster scaling

Expected Outcomes

- Scalable fraud detection pipeline capable of real-time analysis
- Improved detection performance
- Clearly demonstrated scalability with increased data and cluster sizes

Thank You!