Fraud Detection Using Machine Learning

1. Problem Statement

Financial institutions deal with millions of transactions each day, and fraudulent activities can lead to substantial monetary losses. The challenge lies in accurately identifying fraudulent transactions in real time, without disrupting legitimate customer activity.

In this project, we aim to build a robust and interpretable fraud detection model that can classify whether a given transaction is fraudulent based on historical financial transaction data.

2. Dataset Overview

- **Dataset Size:** 6,362,620 rows and 10 columns
- **Source:** Simulated financial transaction dataset (Link: Dataset)
- **Dictionary:** Data Dictionary

Key Features:

- step: Represents time steps (1 step = 1 hour)
- type: Type of transaction (TRANSFER, CASH_OUT, etc.)
- amount: Transaction amount
- oldbalanceOrg, newbalanceOrg: Originator balance before/after transaction
- oldbalanceDest, newbalanceDest: Receiver balance before/after transaction
- isFraud: Target variable indicating fraud
- isFFlaggedFraud: suspicious by business rule (e.g., amount > 200,000)

3. Project Aim

- To develop a machine learning model that can accurately detect fraudulent financial transactions.
- To apply domain-aware feature engineering.
- To evaluate the model using precision, recall, F1-score, and AUC-ROC.

4. Project Approach

Step 1: Exploratory Data Analysis (EDA)

- Checked for missing values, data types, and cardinality.
- No missing values found.
- Removed high-cardinality text fields (nameOrig, nameDest).

Step 2: Feature Engineering

- Created features:
  - balance_diff_of_orig, balance_diff_of_dest
  - ratio_of_amount_to_orig, ratio_of_amount_to_dest
  - Flags for zero balances
  - log_amount and day extracted from step
- One-hot encoded the type feature.

Step 3: Data Cleaning

- Removed redundant features like isFlaggedFraud, raw balance columns.
- Verified multicollinearity using heatmaps.

Step 4: Handling Class Imbalance

- Applied SMOTE **after train-test split** to balance isFraud classes.

Step 5: Model Training

- Chose **Random Forest Classifier** as the base model.

5. Why Random Forest?

Reason	Justification
Handles non-linearity	Fraud is a non-linear problem.
Works well with imbalanced data	Random Forest handles imbalance well when combined with SMOTE.
Feature importance	Allows us to interpret which features are most predictive.
Robust to outliers	Trees naturally split around unusual data.

6. Model Evaluation

Confusion Matrix

[[1270716   165]
 [2   1641]]

Classification Report

Class 0 (Non-Fraud):
- Precision: 1.00
- Recall: 1.00
- F1-Score: 1.00

Class 1 (Fraud):
- Precision: 0.91
- Recall: 1.00
- F1-Score: 0.95

AUC-ROC Score: 0.9996

ROC Curve:

- Achieved near-perfect separability.

7. Key Predictive Features

Feature	Description	Importance
ratio_of_amount_to_orig	Ratio of amount to origin balance	0.33
balance_diff_of_orig	Balance reduction after transaction	0.31
ratio_of_amount_to_dest	Ratio to destination balance	0.07
type_TRANSFER, log_amount, type_PAYMENT	High-risk transaction types	~0.05

These features logically align with fraud patterns — draining accounts, transferring all money, etc.

8. Prevention Recommendations

- Real-time rules for high amount/balance ratio.
- Flag accounts with sudden zero balances.
- Add 2FA for new transfer destinations.
- Apply rate limits and anomaly detection for new users.

9. Post-Implementation Monitoring

Metric	Expectation
Fraud Count	Decrease
False Positives	Stable or slight rise
F1-score	Stable or increase
Money Lost to Fraud	Significant drop

10. Conclusion

This project successfully applied supervised learning to financial fraud detection. With careful feature engineering, SMOTE balancing, and a Random Forest model, we achieved a **recall of 1.00** on fraudulent transactions with an **AUC of 0.9996**.

The model is suitable for deployment and can be monitored for continuous improvement.

Some Insights into the dataset and the model:

1. Data cleaning including missing values, outliers, and multicollinearity

- No missing values were present.
- Outliers were visualized using boxplots and handled via feature transformation (e.g., `log_amount`).
- Multicollinearity was assessed using a heatmap, features with high correlation were reviewed. No severe collinearity warranted dropping any variables.

2. Description of the fraud detection model in elaboration:

- A Random Forest classifier was trained after performing SMOTE oversampling.
- It uses ensemble learning with multiple decision trees, making it robust to overfitting.
- Final model achieved high recall and AUC, indicating excellent fraud detection capability.
- The model is also robust against highly imbalanced datasets.

3. Selection of variables to be included in the model:

- Used domain knowledge (e.g., balance differences and transaction types).
- Feature importance scores from Random Forest helped finalize which engineered features to retain.
- Categorical features were one-hot encoded to be ML friendly.

4. The performance of the model using best tools:

- Metrics used: Confusion matrix, Precision, Recall, F1-Score, AUC-ROC
- Tools: Scikit-learn, Seaborn, Matplotlib for visualization
- Model performance exceeded expectations, especially for detecting fraud.

5. key factors that predict fraudulent customers:

- High `ratio_of_amount_to_orig`

- Large `balance_diff_of_orig`
- High frequency of `TRANSFER` or `CASH_OUT` types

6. Do these features make sense? If yes, how?

- Yes, they make sense because fraud typically involves:
  - Transferring all available funds
  - Sudden zeroing out of account balances
  - Large transactions relative to historical behavior

7. Prevention strategies that companies should adopt are

- Adding two-factor authentication (2FA) for new payees
- Limiting the maximum transfer amount or velocity.
- Real time alerts for account draining behaviors.

8. To determine if the prevention strategies work

- Monitor fraud detection metrics post-deployment:
  - Increase in fraud recall
  - Drop in fraud-related losses
  - Stable false positives and customer complaints