# Fraud Detection

## Project Overview

This project uses machine learning models to detect fraudulent financial transactions in a large financial dataset containing over **6 million transactions**. The goal is to proactively identify fraud and provide actionable insights for the company.

## Dataset

- Total transactions: 6,362,620
- Features:
    - step – time step of transaction
    - type – transaction type
    - amount – transaction amount
    - nameOrig / nameDest – sender/receiver accounts
    - oldbalanceOrg / newbalanceOrig – sender balances
    - oldbalanceDest / newbalanceDest – receiver balances
    - isFraud – fraud label (target)
    - isFlaggedFraud – flagged transactions

For faster execution and demonstration, a **10% random sample** (~636k transactions) was used for model building while maintaining the same fraud distribution.

## Approach / Methodology

1. **Data Cleaning:** Handled missing values, outliers, and checked multicollinearity.
2. **Exploratory Data Analysis (EDA):** Visualized transaction patterns and fraud distribution.
3. **Feature Engineering:** Encoded categorical variables (type) and selected relevant features.
4. **Handling Imbalanced Data:** Used **SMOTE** to balance the minority class (fraudulent transactions).
5. **Model Building:**
    - Logistic Regression
    - Optimized Random Forest
    - Optional XGBoost (if installed)

6. **Model Evaluation:** Confusion matrix, classification report, ROC-AUC score, and feature importance.
7. **Business Insights:** Identified key predictors and suggested preventive measures.

**Tools & Technologies**

- **Programming & Notebook:** Python, Jupyter Notebook
- **Libraries:** Pandas, NumPy, Matplotlib, Seaborn, scikit-learn, imbalanced-learn (SMOTE), XGBoost (optional)

**Results**

- **Random Forest** performed best:
    - Accuracy: **99%**
    - ROC-AUC: **0.98**
- **Important Features:** amount, step, oldbalanceOrg, newbalanceOrig
- **Recommended Preventive Measures:**
    - Real-time transaction monitoring
    - Multi-factor authentication
    - Alerts for unusual transactions

**How to Run**

1. Open the Jupyter Notebook (fraud_detection.ipynb).
2. Install required Python libraries:
3. pip install pandas numpy scikit-learn matplotlib seaborn imbalanced-learn xgboost
4. Run the notebook **step by step** to reproduce analysis and results.

**Conclusion**

The project demonstrates a **scalable machine learning approach** to detect fraudulent transactions. By analyzing the key predictors and implementing preventive measures, the company can **reduce financial losses** and improve fraud detection efficiency.