Interactive Financial Fraud Detection Dashboard

An intuitive web application built with Streamlit and Scikit-learn to analyze financial transaction data, evaluate a pre-trained fraud detection model, and predict fraudulent activities in new, unlabeled datasets.
<!-- It's highly recommended to add a screenshot of your app here -->

🚀 Features

This project provides a comprehensive toolkit for both data scientists and financial analysts. It is split into two main functional tabs:

1. Analyze Labeled Dataset

- **Upload & Analyze**: Upload a CSV file with historical transaction data, including an isFraud label.
- **Instant Accuracy Score**: Immediately see the model's accuracy on a 20% test sample of your data.
- **In-Depth Performance Metrics**:
  - **Classification Report**: Detailed precision, recall, and F1-scores.
  - **Confusion Matrix**: A clear visual of true/false positives and negatives.
  - **ROC Curve**: Assess the model's diagnostic ability.
- **Data Exploration**: View overall statistics, transaction type distributions, and a complete list of all actual fraudulent transactions in the uploaded file.
- **Feature Importance**: Understand which transaction features the model finds most predictive of fraud.

2. Predict on Unlabeled Dataset

- **Separate Upload for Unlabeled Data**: Upload a CSV file without the isFraud column to simulate a real-world prediction scenario.
- **Intelligent Fraud Detection**: The application uses the pre-trained model to flag individual high-risk transactions.
- **Business Logic for Pattern Recognition**: A rule-based logic layer is applied on top of the model's predictions to identify and display the common fraud pattern of a high-value TRANSFER immediately followed by a corresponding CASH_OUT.
- **Clear & Actionable Output**: Displays a clean, sorted list of all transactions identified as part of a fraudulent pattern.

🛠️ How It Works

The application leverages a powerful stack for data science and web development:

- **Backend**: Python

- **Machine Learning Model**: A **Random Forest Classifier** trained on a balanced dataset using the **SMOTE** (Synthetic Minority Over-sampling Technique) to handle the severe class imbalance typical of fraud detection. The model is pre-trained and saved as model/fraud_detector.pkl.
- **Data Manipulation**: Pandas for efficient data loading and processing.
- **Web Framework**: Streamlit for creating the interactive user interface.
- **Plotting**: Matplotlib & Seaborn for generating visualizations.

The workflow is as follows:

1. A user uploads a CSV file.
2. The data is preprocessed (cleaned, and categorical features are encoded).
3. The pre-trained Random Forest model is loaded.
4. Depending on the tab, the app either evaluates the model against known labels or predicts fraud on the new data.
5. Results, metrics, and visualizations are displayed in the interactive dashboard.

⚙️ How to Run Locally

Follow these steps to set up and run the project on your local machine.

Prerequisites

- Python 3.8+
- pip and venv

1. Clone the Repository

```
git clone https://github.com/your-username/fraud_detection_app.git
cd fraud_detection_app
```

2. Create and Activate a Virtual Environment

```
# Create the environment
python -m venv venv

# Activate it
# On Windows:
venv\Scripts\activate
# On macOS/Linux:
source venv/bin/activate
```

3. Install Dependencies

All required packages are listed in requirements.txt.

```
pip install -r requirements.txt
```

4. Run the Streamlit App

streamlit run app.py

Your web browser should automatically open with the application running.

🤖 The Model

The core of this project is the fraud_detector.pkl model. It is a **Random Forest Classifier** chosen for its high accuracy and its ability to handle complex datasets. A key aspect of its development was the use of **SMOTE** to artificially balance the training data, ensuring the model learned the patterns of the minority 'fraud' class effectively without being biased towards the majority 'non-fraud' class.