**Credit Card Fraud Detection**

This repository contains a comprehensive project on credit card fraud detection using machine learning. The project includes data generation, model training, evaluation, and a Flask web application for real-time fraud detection.

**Project Overview**

This project demonstrates the following key components:

* **Data Generation**: Generate synthetic credit card transaction data.
* **Fraud Detection Model**: Build and train a machine-learning model to detect fraudulent transactions.
* **Evaluation**: Evaluate the model using various metrics.
* **Web Application**: Create a Flask web app to upload transaction files and predict fraud.

**Table of Contents**

* [Installation](https://github.com/Yash1789/Credit-Card-Fraud-Detection#installation)
* [Usage](https://github.com/Yash1789/Credit-Card-Fraud-Detection#usage)
* [Project Structure](https://github.com/Yash1789/Credit-Card-Fraud-Detection#project-structure)
* [Model Training](https://github.com/Yash1789/Credit-Card-Fraud-Detection#model-training)
* [Web Application](https://github.com/Yash1789/Credit-Card-Fraud-Detection#web-application)
* [Contributing](https://github.com/Yash1789/Credit-Card-Fraud-Detection#contributing)
* [License](https://github.com/Yash1789/Credit-Card-Fraud-Detection#license)

**Installation**

To set up this project locally, follow these steps:

1. Clone the Repository:
2. git clone https://github.com/yourusername/Credit-Card-Fraud-Detection.git
3. cd Credit-Card-Fraud-Detection
4. Create and Activate a Virtual Environment:
5. python -m venv venv
6. source venv/bin/activate # On Windows use `venv\Scripts\activate`
7. Install Dependencies:
8. pip install -r requirements.txt
9. Additional Setup: Please ensure you have the necessary files (model.pkl, scaler.pkl, etc.) in the appropriate directories.

**Usage**

**Data Generation**

Generate synthetic transaction data with the following command: python data\_generation.py

**Model Training**

Train the fraud detection model using: python model\_training.py

**Web Application**

Run the Flask web application with: python app.py

**Project Structure**

* data\_generation.py: Script to generate synthetic transaction data.
* model\_training.py: Script to train and save the fraud detection model.
* app.py: Flask application for file upload and fraud prediction.
* requirements.txt: Python packages required for the project.
* model.pkl: Serialized Random Forest model.
* scaler.pkl: Serialized StandardScaler for feature scaling.
* templates/: HTML templates for the Flask application.
* static/: Static files (CSS, JS) used by the Flask application.

**Model Training**

**Feature Engineering**

* Distance Rating: Calculates the distance between customer and merchant.
* State Rating: Checks if the customer and merchant are in different states.
* Limit Rating: Assesses transaction frequency.
* Unusual Rating: Identifies transactions occurring during unusual hours.

**Training Process**

1. **Load and Preprocess Data:** Read the data and apply feature engineering.
2. **Train Model:** Use RandomForestClassifier to train on the prepared data.
3. **Evaluate Model:** Assess the model's performance using accuracy, precision, recall, and F1 score.

**Web Application**

The Flask application allows users to:

* Upload Excel files containing transaction data.
* Predict fraud risk using the trained model.
* Download the modified file with additional columns for fraud risk and ratings.

**Contributing**

Contributions are welcome! You can help by:

* Reporting bugs or suggesting new features.
* Submitting pull requests for improvements or fixes.

Please follow the contribution guidelines if available.

**License**

This project is licensed under the MIT License. See the LICENSE file for details.

**DESIGNED & CREATED: -**

**YASH KADAM**