**Fraud Detection in Financial Transactions**

**Overview**

This project aims to build a fraud detection system for financial transactions. The system utilizes machine learning algorithms to analyze transactional data, including customer details, transaction amounts, and merchant information. By identifying patterns indicative of fraudulent activities, the system can flag suspicious transactions for further investigation, helping financial institutions mitigate risks and protect the integrity of their systems. The project involves exploratory data analysis, data preprocessing, model training, and evaluation to develop an effective fraud detection solution.

**Author**

Saron Yaya

**Initial work**

* [Portfolio Projects](https://github.com/Saron222/PortfolioProjects)

**Released on**

* GitHub

**My professional profile on LinkedIn**

* [My LinkedIn Profile](https://www.linkedin.com/in/saron-yaya/)

**Showcase**

This project showcases the development of a fraud detection system for financial transactions using machine learning techniques. Here's how the project demonstrates various aspects:

1. **Data Analysis and Visualization**: The project begins with an extensive analysis of the dataset, including graphical representations of transaction categories, gender distribution, age distribution, and the relationship between transaction amount and fraud. These visualizations provide valuable insights into the characteristics of the data, helping to identify patterns and trends.
2. **Data Preprocessing and Balancing**: The dataset is preprocessed to handle missing values, transform data types, and balance the class distribution using techniques like Synthetic Minority Over-sampling Technique (SMOTE) and Random Under Sampling (RUS). These preprocessing steps ensure that the data is suitable for training machine learning models and mitigate the impact of class imbalance.
3. **Model Training and Evaluation**: Several machine learning algorithms are employed, including Random Forest, Logistic Regression, XGBoost, and Support Vector Machines (SVM). The models are trained on balanced datasets and evaluated using metrics such as precision, recall, F1-score, and ROC AUC score. The Random Forest Classifier is chosen for its robust performance and effectiveness in capturing complex relationships in the data.
4. **Model Deployment and Performance**: The trained Random Forest Classifier is deployed in a production environment to flag potentially fraudulent transactions in real-time. Continuous monitoring and periodic retraining are implemented to adapt to evolving fraud patterns. The model achieves satisfactory performance in terms of accuracy and ROC AUC score, contributing to the security and trustworthiness of financial systems.

**Technologies Used**

* Python
* Scikit-learn
* Pandas
* Matplotlib

**Usage Example**

This project implements a machine learning-based fraud detection system to combat fraudulent transactions:

1. **Data Collection**: Transaction data is gathered and prepared for analysis.
2. **Model Training**: Various machine learning models are trained and evaluated.
3. **Model Deployment**: The Random Forest Classifier is deployed for real-time monitoring.
4. **Real-time Detection**: Transactions are analyzed in real-time, flagging suspicious activities.
5. **Alert Handling**: Fraud detection team investigates flagged transactions promptly.
6. **Continuous Improvement**: System is updated regularly to adapt to new fraud patterns.

**Installation**

1. Clone the repository: git clone [https:// https://github.com/Saron222/PortfolioProjects.git](https://github.com/saronyaya/DSC510.git)
2. Navigate to the project directory: **cd Fraud Detection in Financial Transactions.ipynb**

**Contributing**

1. Fork the repository from [Saron222/PortfolioProjects](https://github.com/Saron222/PortfolioProjects/fork)
2. Create your feature branch: **git checkout -b feature/your-feature-name**
3. Commit your changes: **git commit -am 'Add your feature'**
4. Push to the branch: **git push origin feature/your-feature-name**
5. Create a new Pull Request in the original repository