**README for Credit Card Fraud Detection Using Random Forest Classifier :**

**Overview**

This project involves detecting fraudulent transactions in credit card data using a **Random Forest Classifier**. The dataset is highly imbalanced, with only a small fraction of transactions being fraudulent. The steps in this project include data exploration, preprocessing, model training, evaluation, and visualization of results.

**Requirements**

To run this project, the following libraries are required:

* **NumPy**
* **Pandas**
* **Matplotlib**
* **Seaborn**
* **scikit-learn**

Install the libraries using pip if not already installed:

!pip install numpy pandas matplotlib seaborn scikit-learn

**Dataset**

The dataset used in this project is the **Credit Card Fraud Detection** dataset, available at:  
<https://www.kaggle.com/mlg-ulb/creditcardfraud>

**Dataset Details:**

* Contains credit card transactions labeled as:
  + **1**: Fraud
  + **0**: Valid
* Includes features like Time, Amount, and anonymized features labeled V1 to V28.
* Highly imbalanced: Fraud cases are much fewer than valid transactions.

**Steps in the Code**

**1. Data Loading and Exploration**

* Load the dataset using Pandas.
* Display the first 10 rows and dataset statistics.
* Identify class imbalance by calculating the fraction of fraudulent transactions.

**2. EDA (Exploratory Data Analysis)**

* Analyze the transaction amounts for fraudulent and valid transactions using .describe().
* Visualize the correlation matrix using a heatmap.

**3. Data Preprocessing**

* Separate features (X) and target labels (Y).
* Split the data into training (80%) and testing (20%) subsets using train\_test\_split.

**4. Model Training**

* Build a Random Forest Classifier using sklearn.ensemble.RandomForestClassifier.
* Train the model on the training data (xTrain and yTrain).

**5. Model Evaluation**

* Predict the target values for the test data (xTest).
* Calculate and display the following metrics:
  + **Accuracy**
  + **Precision**
  + **Recall**
  + **F1-Score**
  + **Matthews Correlation Coefficient (MCC)**

**6. Confusion Matrix Visualization**

* Generate and visualize the confusion matrix using Seaborn.