

GitHub link:

<https://github.com/ANAND12RAMAN>

## Guarding Transactions with AI-powered Credit Card Fraud

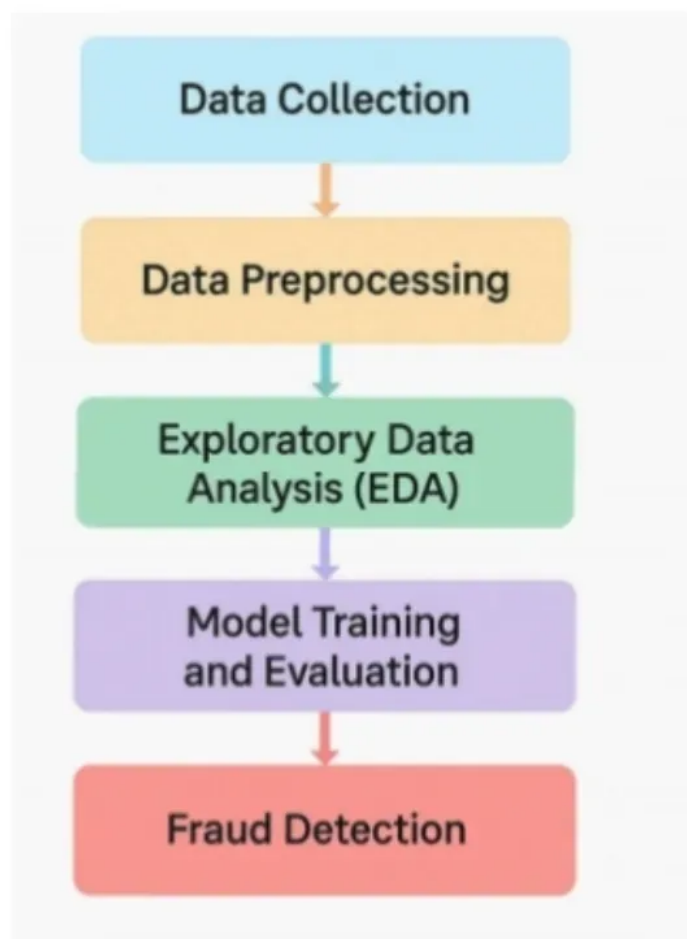
### Detection and Prevention

In today's digital economy, credit card transactions are ubiquitous, but so is fraud. Financial institutions face growing challenges in identifying and preventing fraudulent activities in real-time. AI-based fraud detection offers the potential to analyze large volumes of transaction data and detect anomalies effectively.

### 2. Project Objectives

- Detect fraudulent credit card transactions using machine learning.
- Reduce false positives and improve the accuracy of detection.
- Analyze transaction features to understand fraud patterns.
- Build a model that can predict and prevent future fraudulent activities.

### 3. Flowchart of the Project Workflow



## **4. Data Description**

The dataset contains anonymized credit card transactions with features such as time, amount, and class labels indicating fraud. Features are the result of a PCA transformation for confidentiality. Class 1 indicates fraud, and Class 0 indicates legitimate transactions.

## **5. Data Preprocessing**

- Checked for missing values and normalized the 'Amount' feature. - Performed feature scaling to improve model performance. - Handled imbalanced data using under-sampling and SMOTE.

## **6. Exploratory Data Analysis (EDA)**

- Analyzed class distribution and correlation between features. - Visualized fraud vs non-fraud transaction patterns. - Studied transaction amounts and time gaps for both classes.

## **7. Feature Engineering**

- Engineered time-based and amount-related features. - Selected important features based on correlation and model importance. - Applied PCA and autoencoders for dimensionality reduction.

## **8. Model Building**

- Trained models like Logistic Regression, Random Forest, and XGBoost.  
- Used cross-validation to improve generalization.  
- Evaluated using precision, recall, F1-score, and ROC-AUC.

## **9. Results & Insights**

- Random Forest performed best with high recall for fraud detection.  
- SMOTE significantly improved model's ability to detect rare fraud cases.  
- A real-time fraud detection system could be built using this model.

## **10. Tools and Technologies Used**

- Python libraries: pandas, numpy, sklearn, matplotlib, seaborn, imbalanced-learn.  
- Jupyter Notebook for experimentation.  
- Environmental: Google colab

## 7. Team Members and Roles

- Responsible for data cleaning and EDA:Arulirasan.G
- Leads model building and evaluation:Anandharaman.M
- Handles API development and deployment:Srimanoj.C
- Ensures timelines are met and oversees integration with client systems: Thiruneelakandan.M