# Advanced Credit Card Fraud Detection Project

## Project Title:

Advanced Credit Card Fraud Detection Using Machine Learning and Explainable AI

## Main Project Objective:

Build a complete ML pipeline to identify fraudulent credit card transactions with a focus on data imbalance, model explainability, and performance optimization.

## Advanced-Level Project Questions:

### 1. Exploratory Data Analysis (EDA) and Visualization

- What is the distribution of the Class feature (fraud vs non-fraud)?
- Are there any correlations between features that can be visualized using a heatmap?
- How do transaction amounts differ between fraudulent and legitimate transactions?
- Can PCA component distributions help us understand fraud better?
- Do fraudulent transactions show specific patterns across time (Time column)?
- Can we visualize clusters of fraud vs non-fraud using t-SNE or UMAP?

### 2. Data Preprocessing and Handling Imbalance

- How should missing values or anomalies be handled if any?
- How can we deal with extreme imbalance in fraud labels (e.g., SMOTE, ADASYN, or class weighting)?
- Should we normalize or scale certain features like Amount or Time?

### 3. Model Training and Evaluation

- Which algorithms perform best for this task? (Try: Logistic Regression, Random Forest, XGBoost, Isolation Forest, AutoEncoders)
- How do metrics like precision, recall, F1-score, and ROC-AUC compare across models?
- Can we use cross-validation effectively despite class imbalance?
- What threshold tuning methods can improve F1-score?

### 4. Model Explainability

- Which features contribute most to predicting fraud?
- Use SHAP or LIME to explain individual predictions.
- Can we visualize SHAP summary plots, dependence plots, and force plots?

### 5. Visualization Goals

# Advanced Credit Card Fraud Detection Project

- Pie chart or bar graph of class imbalance
- Histograms of Amount and Time by class
- Correlation heatmap
- Boxplots of features by class
- t-SNE or UMAP projection to visualize high-dimensional feature clustering
- Confusion matrix heatmap
- ROC and Precision-Recall curves
- SHAP summary and force plots

## 6. Optional Deep Learning Approach

- Can an autoencoder detect anomalies without labels?
- How does its performance compare with supervised models?

## 7. Deployment (Optional)

- Can we build a lightweight web app using Streamlit to test real-time predictions?
- How can we load a saved model and preprocess user inputs dynamically?

## Tools & Technologies

- Python (Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, SHAP, XGBoost, Imbalanced-learn)
- Optional: TensorFlow/Keras for autoencoders
- Streamlit/Flask for deployment