**Credit Card Fraud Detection (ML Project):**

1.**DOWNLOADING**:

-Download the Credit Card Fraud Detection dataset from Kaggle - it contains 31 columns (time, amount, class, v1-v28)

-This v1-v28 are the anonymized features obtained by applying Principal Component Analysis (PCA) to the original transactional variables

-The original dataset contains sensitive financial information, hence to preserve the privacy while still allowing ML research, the raw features are transformed into 28 principal components.

2.**PREPROCESSING**:  
-Scikit Learn is one of the popular ML Library , preprocessing is one of the modules of Scikit-Learn   
-Standard Scaler is a class inside preprocessing which standardises values with mean=0, standard deviation=1

-ML Algorithms work better when the numerical values are on the same scale

-In our project Amount and Time are not PCA transformed , so they are on different scales when compared v1-v28 , hence we use Standard Scalar to standardise them

-X = df.drop('Class', axis=1) – we are dropping the column class from the dataframe(axis=1 indicating the column , if it were a row it would be axis=0)

-X contains the dataset except the Class Column – these are the features (input data used for prediction)

-y is the target/label (the output you want to predict -> here Class is the target/label , which is 0=genuine , 1=fraud)

**3.TEST-TRAIN SPLIT:**

-model\_selection is a module used for splitting data , cross-validation and model\_selection

-train\_test\_split converts - class of model\_selection used to split training and test datasets

X\_train , X\_test , y\_train , y\_test=train\_test\_split(X,y,test\_size=0.2 , random\_state=42)

- X-independent variables

-y- dependent variables

-test\_size= splits into 20%testing data and 80%training data

-random\_state=42 - ensures reproducibility(same split every time)

-stratify=y – preserves the fraud-to-genuine ratio in both train and test

4.RESAMPLING:

Technique used to handle imbalanced datasets , where one class (fraud in this case ) is much smaller than the other (genuine)

SMOTE (Synthetic Minority Over-Sampling Technique):

-Instead of just duplicating minority samples, SMOTE creates new synthetic examples.

How it works:

-Take a minority class sample.

-Find its k nearest neighbours in feature space.

-Randomly generate a new sample between the original and its neighbours.

**5. TRAINING THE DATASET USING LOGISTIC REGRESSION:**

-linear\_model is a module of sklearn ML Libraary and Logistic Regression is a class inside the linear\_model module

**6.EVALUATION:**

from sklearn.metrics import roc\_auc\_score, precision\_recall\_curve, auc

-roc\_auc\_score – computes Area under the ROC Curve

- precision\_recall\_curve → computes precision and recall values at different thresholds

-auc → calculates the **area under a curve** (used here for PR-AUC)

y\_proba = model.predict\_proba(X\_test)[:,1]

-model.predict\_proba(X\_test) → returns **probabilities for both classes (0 & 1)**

**-**[:,1] → selects the probability of **class 1 ( fraud)**

roc\_auc = roc\_auc\_score(y\_test, y\_proba)

-Measures how well the model **ranks positive vs negative samples.**

-For fraud detection, **high ROC-AUC** means the model assigns **higher probabilities to fraud transactions than genuine ones.**

pr\_auc = auc(recall, precision)

- Calculates **area under the precision-recall curve.**

- **PR-AUC is more informative than ROC-AUC** for imbalanced datasets.

7.SHAP explanation:

explainer = shap.LinearExplainer(model, X\_train\_res, feature\_perturbation="interventional")

-tells SHAP to explain our linear regression model

-interventional mode makes explanations more robust by perturbing inputs in a realistic way

shap\_values = explainer.shap\_values(X\_test)

-shap values tell how much the features (v1-v28, amount, time) contribute to the prediction of fraud for each transaction