**CREDIT CARD FRAUD DETECTION WITH**

**APPLIED DATA SCIENCE**

**Phase 3: Development Part 1**

Creating a credit card fraud detection project involves several steps, including loading and preprocessing the dataset. Here's a general outline of how you can do this using Python and some popular libraries like Pandas and Scikit-learn. Make sure to download the credit card transaction dataset from a reliable source before proceeding.

**Program:**

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

data = pd.read\_csv("credit\_card\_data.csv")

print(data.head())

print(data.info())

print(data.describe())

data.dropna(inplace=True) # If you have missing values

scaler = StandardScaler()

data['Amount'] = scaler.fit\_transform(data['Amount'].values.reshape(-1, 1))

from imblearn.over\_sampling import SMOTE

X = data.drop('Class', axis=1)

y = data['Class']

smote = SMOTE(sampling\_strategy=0.5) # Adjust the sampling strategy as needed

X\_resampled, y\_resampled = smote.fit\_resample(X, y)

* This code appears to be a python script that works with a dataset using the pandas library. Here’s a breakdown of the process:

**Explanation:**

* **Import Libraries:** Start by importing necessary Python libraries for data manipulation, preprocessing, and machine learning.
* **Load the Dataset:** Load your credit card transaction dataset using a library like Pandas.

**data = pd.read\_csv("credit\_card\_data.csv")**

* **Data Exploration:** Explore the dataset to understand its structure, check for missing values, and gain insights into the data.

**print(data.head())**

**print(data.info())**

**print(data.describe())**

* **Preprocessing:**

Preprocess the data by handling missing values, standardizing numerical features, and addressing class imbalance if necessary.

* **Handling Missing Values**: Check and handle missing values if necessary.

**data.dropna(inplace=True)**

* **Feature Scaling:** Standardizing numerical features, such as the transaction amount, is essential to ensure that features are on a similar scale. This helps the model's convergence and performance.

**scaler = StandardScaler()**

**data['Amount'] = scaler.fit\_transform(data['Amount'].values.reshape(-1, 1))**

* **Handling Class Imbalance:** In credit card fraud detection, we often deal with imbalanced datasets, with very few fraud cases. Techniques like oversampling or undersampling can be applied to balance the classes.

**from imblearn.over\_sampling import SMOTE # Example for oversampling**

**X = data.drop('Class', axis=1)**

**y = data['Class']**

**smote = SMOTE(sampling\_strategy=0.5) # Adjust the sampling strategy as needed**

**X\_resampled, y\_resampled = smote.fit\_resample(X, y)**

**Conclusion:**

These preprocessing steps are essential for setting the foundation of a credit card fraud detection model. They help ensure the data is clean, appropriately scaled, and that any class imbalances are addressed, ultimately improving the model's performance and accuracy in detecting fraudulent transactions. The preprocessed data can now be used for model training, evaluation, and further steps in the fraud detection pipeline.