Data Science Bootcamp

Capstone Project

FindDefault (Prediction of Credit Card fraud)

Problem Statement:

A credit card is one of the most used financial products to make online purchases and payments. Though the Credit cards can be a convenient way to manage your finances, they can also be risky. Credit card fraud is the unauthorized use of someone else's credit card or credit card information to make purchases or withdraw cash.

It is important that credit card companies are able to recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase.

The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

We have to build a classification model to predict whether a transaction is fraudulent or not.

Required library versions:

Pandas: 2.1.4
Numpy: 1.26.4
Matplotlib: 3.8.0
Sklearn: 1.2.2

Data source:

Click on this link to access the raw data source for working on project: creditcard.csv
Click on this link to access the worked on data source: creditcard-working_data.csv

Steps:

1. The first step to start this project is to download the files for which links are provided above
2. Make sure to have the library already installed for the respective version if not then you can use this code:
   !pip install pandas==2.1.4 numpy==1.26.4 matplotlib==3.8.0 scikit-learn==1.2.2
3. Execute the code from Credit_card_fraud_detection file located in code folder.

Data Source:

Our data source consists following details:

- Time: Time elapsed in seconds between each transaction and the first transaction.
- V1 - V28: Principal components obtained with PCA (anonymized features).
- Amount: Transaction amount.
- Class: Indicates whether the transaction is fraudulent (1) or not (0).

Code explanation:

The Python code performs the following tasks:

1. Data loading and preprocessing: Loads the dataset, handles missing values, converts data types, and performs oversampling to address class imbalance.
2. Visualizations: Use bar chart to check the data imbalanced or not.
3. Hyperparameter Tuning: Uses GridSearchCV to find the best hyperparameters for the model.
4. Model Evaluation: Evaluates the model's performance on test data