Bank Account Fraud Detection

Introduction

The datasets provided in this repository are related to the detection of fraudulent online bank account opening applications in a large consumer bank. The objective is to accurately predict fraudulent attempts using Machine Learning (ML) and Deep Learning (DL) models. The data is highly imbalanced, with only about 1% of applications labeled as positive (fraudulent). In addition to the base dataset, five dataset variants with specific types of data bias are included to stress test the performance and fairness of the models.

Dataset Description

Each instance (row) in the dataset represents an individual application for a bank account opening. The label for each instance is stored in the "is_fraud" column. A positive instance represents a fraudulent attempt, while a negative instance represents a legitimate application. The data spans eight months of applications, and the month of each application is identified in the "month" column. The dataset contains thirty features for each application.

Problem Statement

Given the highly imbalanced labeled dataset (Bank Account Fraud - BAF) with 30 features for each account opening application, the task is to develop ML/DL models that can accurately predict fraudulent attempts.

Steps we followed:

Step 1: Baseline Models and Hyperparameter Tuning

To start the project, we will begin with the baseline models provided in the notebook Baseline Models (ROC). We will attempt to improve the performance of these baseline models by applying hyperparameter tuning for the 4 base line models we could improve the AUC for three of the models

-RandomForestClassifier

-XGBClassifier

-Neural Network

And for LogisticRegression AUC is the same of the base line model

. Additionally, we introduced two more ML models to evaluate their performance.

The two models are decision Tree and Naïve Bayes

Step 2: Data Preprocessing Investigation

We made two preprocessing pipeline:

First preprocessing:

-Replacing -1 values with None

-Replacing null values with mean values

-Dropping irrelevant features

-Apply One-hot Encoding for categorical features

-Apply Standard Scaler

-Dropping highly correlated features

Second Preprocessing:

-Handling outliers with IQR-method

-Dropping unnecessary columns

-Handling negative values

-Taget Encoding

-Robust Scaler

Step 3: Handling Imbalanced Data - State-of-the-Art (SOTA) Approaches

We have tried five SOTA approaches:

SMOTE: giving the highest AUC =0.86 with XGB classifier

ADASYN: giving the highest AUC =0.86 with XGB classifier

Borderline SMOTE: giving the highest AUC =0.86 with Neural Network

ROS: giving the highest AUC =0.88 with Neural Network

SMOTENC: giving the highest AUC =0.85 with Neural Network and XGB classifier

Step 4: Applying the Best Approach to Variant Datasets

After finding the best approach on the base dataset which is Random Over Sampling we apply it on the 5 other variant datasets with the same preprocessing made on the baseline dataset.

Here is the Result:

-The best model for the first variant is logistic regression with AUC .86

-The best model for the second variant is logistic regression and NN with AUC = .88

- The best model for the third variant is NN with AUC = .94

-The best model for the forth variant is NN too with AUC = .84

-The best model for variant number 5 is .81 logistic regressions with AUC =.81