Artificial Intelligence Capstone Project

Project 2: Credit card fraud detection

Auther: PV-J

Problem statement

The study aims to identify fraudulent credit card transactions on the given data set using data science solutions. We need to develop a model with knowledge of ones that turned out to be fraudulent. This model is then used to identify whether a new transaction is fraudulent or not while minimizing the incorrect fraud classifications

The aim of this project is **to identify and predict fraudulent credit card transactions using machine learning models.**

Given dataset is already PCA transformed.

The approach used for solving the problem are:

Step -1: Load the data to understand it. Data understanding is critical since we will select the subset of features to carry out model training, and it includes types of data and its patterns.

Step -2: Exploratory Data Analysis Study histogram to compare each variable's data distribution pattern and skewness of the data. Make necessary data adjustments to avoid any problems while we train the model.

Step -3: Class imbalance Study whether data is highly imbalanced between the fraudulent and non-fraudulent. We have learned four techniques to balance the data using various methods: 1. Under-sampling, oversampling, 2 SMOTE (Synthetic Minority Over Sample Technique), and ADASYN (Adaptive Synthetic). We will use ADASYN techniques to lower the bias introduced by class imbalance and adaptively shift the classification decision boundary towards complex examples.

Step -4: Data modeling and model selection We will use two classification models (RadomForest and XGBoost) for the current study. For XGBoost, we will identify the number of trees based on the accuracy levels. We will not use KNN since the data set are more than 10K (computation time is increased if data sets are more than 10K). The decision tree gives an interpretation of the flow chart. Hence, it is widely used, but we do not know when to stop building trees and tend to overfit. We will use parameters such as confusion matrix, accuracy, precision, recall, and F-score, threshold dependent. Since data is imbalanced, we will also perform a deep dive into the ROC curve data to identify the threshold value (above the threshold value is fraudulent and below the threshold is not dishonest). We will calculate the F1 score, Precision, and Recall.

Step -5: Hyperparameter Tunings the model At this time, we will have the best understanding of the type of data we have and the kind of model we were going to build. After model building, our next step will be either hyperparameter tunings (CV or Cross-validation) or grid-search cross-validation or K-Fold or stratified K-Fold, or train validation and test split. This step is essential to improve accuracy, AUC, and lower misclassification error.

Step -6: Model evaluation Evaluate the model based on AUC -ROC score, through which we will define the threshold value. We will calculate the F1 Score, Precision, and Recall. This will be key for banks to represent the business strategy to bring down the Fraud.

Step 1. Data Understanding, Data Preparation and EDA

First look at the data used here from the simplilearn LMS dataset (Train.csv, train_hidden.csv, and test.csv) suggests that it is highly imbalanced in nature. The positive class (frauds) account for only 0.172% of all transactions.

Class	0	1
Count	284315	493

Class is the target variable which have to predicted where 0 is genuine transaction and 1 is fraudulent transaction.

Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. Project shows the transaction time in ters of hours when necessary. The feature 'Amount' is the transaction Amount, this feature is used for example-dependent cost-sensitive learning.

Since the PCA transforzed variable are already Gaussian, there is no need for normalisation.

We can start with the basic EDA like correlation, boxplots etc for outliers.

Next, transformation is used to mitigate and check the skewness in the data. (Boxcox, Log transformation, Yeo-Johnson etc)