Capstone Project - Credit Card Fraud Detection

SOURAMRAKESH

NARESH KUMAR

- ***Problem Statement:***

  Applying different machine learning algorithms in order to classify credit card fraud from a dataset which contains transactions made by credit cards in Sep 2013 by European cardholders.

- ***Challenges****:*

  The Data set consist of 2,84,807 transactions. Out of a total of 2,84,807 transactions, 492 were fraudulent. This data set is highly unbalanced, with the positive class (frauds) accounting for 0.172% of the total transactions.in this case, it is much worse to have false negatives than false positives in our predictions because false negative mean that someone gets away with credit card fraud and on other hand false positive merely cause a complication and possible hassle when a cardholder must verify that they did, in fact, complete said transaction.

  The data set has also been modified with Principal Component Analysis (PCA) to maintain confidentiality. Apart from 'time' and 'amount', all the other features (V1, V2, V3, up to V28) are the principal components obtained using PCA.

  The feature 'time' contains the seconds elapsed between the first transaction in the data set and the subsequent transactions. The feature 'amount' is the transaction amount. The feature 'class' represents class labelling, and it takes the value 1 in cases of fraud and 0 in others.

  Perform EDA on the given Principal Components

- ***Approach for solving this problem:***

a) Data understanding:

This data set is highly unbalanced, with the positive class (frauds) accounting for just 0.172% of the total transactions2,84,807.

b) EDA:

We do analysis on data by checking missing values, data types, information of the data, skewness of the data. Post checks, we do Uni-variate and Bi-variate analysis on the features and making few transformations like scaling

c) Data imbalance:

In order to offset the imbalance in the dataset, we oversample the fraud portion of the data by adding Gaussian noise to each row using Synthetic Minority Oversampling Techniques like

1. *Under sampling*
2. *Over sampling*
3. *Synthetic Minority Over-Sampling Technique (SMOTE)*
4. *Adaptive Synthetic (ADASYN).*

d) Train/Test Split : *Now we are familiar with the train/test split, which we can perform in order to check the performance of models with unseen data. Here, for validation, we can use the k-fold cross-validation method. You need to choose an appropriate k value so that the minority class is correctly represented in the test folds.*

e) ***Model-Building/Hyperparameter Tuning:*** *Depending upon the models we build we choose the hyperparameters accordingly.*

For a logistic regression we use L1 and L2 as hyper parameters where as indecision tree we use hyper parameters as min_sample_leaves, max-depth.

Building the models like :

1. *KNN*
2. *SVM*
3. *Decision Tree*

4. *Random Forest*
5. *XGBoost*

Proceed with the model which shows the best result

f) Model Evaluation :

The classes present in the given dataset are highly imbalanced, with 99.83% of the observations being labelled as non-fraudulent transactions, and only 0.17% of observations being labelled as fraudulent. So, without handling the imbalances present, the model overfits on the training data and is, therefore, classifying every transaction as non-fraudulent and hence, achieving the aforementioned accuracy.

***Accuracy** is not always the correct metric for solving classification problems.*

There are other metrics need to calculate such as precision, recall, confusion matrix, F1 score, and the AUC-ROC score.

The ROC curve is used to understand the strength of the model by evaluating the performance of the model at all the classification thresholds.