Identification of anomalous records in non-BFSI transactions

Mr. Sundara Bharathi

Senior Data Scientist
GITAA Pvt Ltd
Chennai - 600 113
sundarabharathi@gitaa.in

Motivation

For a long time now, identifying unusual records in transactions related to Banking, Financial Services, and Insurance (BFSI) has been a persistent issue, particularly as instances of fraudulent transactions have become more prevalent. As technology in the banking sector has progressed, so have the techniques used to detect anomalies, in response to fraudsters constantly adapting. It's worth noting that an anomalous transaction could refer to both a fraudulent transaction and an erroneous transaction resulting from human error. Rule-based systems were typically the only means of identifying fraudulent transactions in the past [1].

The use of rule-based detection for identifying potential anomalous transactions raised concerns due to a high incidence of false positives, where normal transactions were flagged as anomalous. Additionally, this approach is not scalable and has limitations in terms of fixed outcomes, which can be problematic in light of the constantly evolving banking trends among customers [1].

Machine learning has emerged as an augmenting technique to aid the anomalous transaction identification process by eliminating the limitations of scalability and rigidity in rule-based methods. With the help of Machine Learning, the process of identifying anomalous transactions can be automated, which reduces the need for human intervention. However, Machine Learning has not replaced rule-based methods entirely, but has instead reduced the amount of human intervention required. This is the impact that Machine Learning has had on the field of anomalous transaction identification in the context of banking transactions.

In this project, the client required us to help them identify the duplicate transactions which is a form of anomalous transactions in their internal non-banking transactions database. The duplicity in this case does not mean the exact duplicate of an existing record, a tweaked record which could skip the validations during the creation of the record in the system also counts as a duplicate record. This could be intentionally made by personnel who have complete knowledge and easy access to the system which could not be traced or detected that easily, or it could be a man-made error while entering data into the database.

Problem

GITAA had to develop a POC to detect possible anomalous transactions from the perspective of duplicity of the records.

Solution

A sample data from the client's existing system was shared with us which contained 34 features. Most of the features were categorical in nature except a few such as amount (in INR) and quantity of items. GITAA was able to reduce these into 4 features through persistent discussions with the client and analysis of the domain knowledge for duplicity identification. The features finalized were Reference number, document date, amount and credit/debit status of a transaction. The feature names are anonymized due to the sensitivity of the project. The feature descriptions are shown in the table below.

Feature Name	Description (Feature type)
Reference number	External reference number of the transaction given by the vendor to/by whom the transaction was made. (Categorical)
Document date	Date of invoice document split into Day, Month and Year (Numerical)
Debit/Credit status	Binary values indicating debit or credit status of the transaction (Categorical)
Amount	Transaction amount in INR (Numeric)

At first, GITAA's primary focus was to enhance the existing model that the client developed for detecting duplicate transactions. The problem was posed as an unsupervised learning problem of machine learning and a clustering technique was used to cluster the records over raw features, which resulted in a large number of false positives as well as true negatives.

From the metadata of the data and thorough investigation, it was suggested to encode the categorical features appropriately. Based on GITAA's suggestions, the reference number feature was one-hot encoded as it contained random unique values (not ordinal). The Credit/Debit status column was label encoded as it contained only binary values and the Day, Month, Year and Amount were used as numerical features. The data also contained transactions for many vendors and was suggested to perform clustering over data of each vendor. This resulted in an enormous reduction in false positives (by about 4 times). However, the clustering method resulted in decreasing true positives as well, i.e., many possible duplications were unidentified. Subsequently, GITAA analyzed alternative approaches for attempting to solve the problem.

Based on the approaches mentioned in the published literature [2], [3], Fuzzy Matching was performed over each of the features with a threshold identified manually. For Fuzzy Matching, Reference number, Credit/Debit and the Amount features were converted to string, and document Day, Month and Year were numerical. The features were fuzzy matched using Levenshtein distance as it measures the minimum number of single-character edits [4]. This resulted in reducing the false positives at the same time increasing the true positives.

Recommendations were then provided to deploy the model to indicate the possibility of duplicity whenever a transaction was made. While approving the transaction would be at the approver's discretion, the data would get labeled and this would happen for a brief amount of time, thus the labeled data generated would help build supervised learning models to identify the fraudulent/ anomalous record before getting into the system.

References

1. Ravelin, https://www.ravelin.com/insights/machine-learning-for-fraud-detection

2. Hussein Issa, Application of Duplicate Records detection Techniques to Duplicate Payments in a Real Business Environment, Rutgers Business School, Rutgers University.

3. Nick Gehrke, https://zapliance.com/en/duplicate-payments-i-want-my-money-back/

4. Wikipedia, https://en.wikipedia.org/wiki/Levenshtein_distance