Credit Card Fraud Detection Capstone Project -by Srilakshmi KB

Given all possible hypotheses and considering the feasibility and customer time, the most suitable solution is to implement a fraud detection system. This does not affect the customer's time with extra OTP checks on all transactions and is also quite feasible, as educating all customers on various fraudulent techniques is a challenging task. Building a fraud detection system is a one time procedure and deploying this would be a permanent resolution to the long time blocker that the banks have been facing since years.

In the banking industry, detecting credit card fraud using machine learning is not just a trend; it is a necessity for the banks, as they need to put proactive monitoring and fraud prevention mechanisms in place. Machine learning helps these institutions reduce time-consuming manual reviews, costly chargebacks and fees, and denial of legitimate transactions.

Suppose you are part of the analytics team working on a fraud detection model and its cost-benefit analysis. You need to develop a machine learning model to detect fraudulent transactions based on the historical transactional data of customers with a pool of merchants. You can learn more about transactional data and the creation of historical variables from the link attached here. You may find this helpful in the capstone project while building the fraud detection model. Based on your understanding of the model, you have to analyse the business impact of these fraudulent transactions and recommend the optimal ways that the bank can adopt to mitigate the fraud risks.

Data Understanding

This is a simulated data set taken from the Kaggle website and contains both legitimate and fraudulent transactions. The data set contains credit card transactions of around 1,000 cardholders with a pool of 800 merchants from 1 Jan 2019 to 31 Dec 2020. It contains a total of 18,52,394 transactions, out of which 9,651 are fraudulent transactions. The data set is highly imbalanced, with the positive class (frauds) accounting for 0.52% of the total transactions. Now, since the data set is highly imbalanced, it needs to be handled before model building. The feature 'amt' represents the transaction amount. The feature 'is_fraud' represents class labelling and takes the value 1 the transaction is a fraudulent transaction and 0, otherwise.

Project Pipeline

The project pipeline can be briefly summarised in the following steps:

- Understanding Data: In this step, you need to load the data and understand the features present in it. This will help you choose the features that you need for your final model.
- Exploratory data analytics (EDA): Normally, in this step, you need to perform univariate and bivariate analyses of the data, followed by feature transformations, if necessary. You can also check whether or not there is any skewness in the data and try to mitigate it, as skewed data can cause problems during the model-building phase.
- Train/Test Data Splitting: In this step, you need to split the data set into training data and testing data in order to check the performance of your models with unseen data. You can use the stratified k-fold cross-validation method at this stage. For this, you need to choose an appropriate k value such that the minority class is correctly represented in the test folds.
- Model Building or Hyperparameter Tuning: This is the final step, at which you can try different models and fine-tune their hyperparameters until you get the desired level of performance out of the model on the given data set. Ensure that you start with a baseline linear model before going towards ensembles. You should check if you can get a better performance out of the model by using various sampling techniques.
- Model Evaluation: Evaluate the performance of the models using appropriate evaluation metrics. Note that since the data is imbalanced, it is important to identify which transactions are fraudulent transactions more accurately than identifying non-fraudulent transactions. Choose an appropriate evaluation metric that reflects this business goal.
- Business Impact: After the model has been built and evaluated with the appropriate metrics, you need to demonstrate its potential benefits by performing a cost-benefit analysis which can then be presented to the relevant business stakeholders.

To perform this analysis, you need to compare the costs incurred before and after the model is deployed. Earlier, the bank paid the entire transaction amount to the customer for every fraudulent transaction which accounted for a heavy loss to the bank.

Now after the model has been deployed, the bank plans to provide a second layer of authentication for each of the transactions that the model predicts as fraudulent. If a payment gets flagged by the model, an SMS will be sent to the customer requesting them to call on a toll-free number to confirm the authenticity of the transaction. A customer experience executive will also be made available to respond to any queries if necessary. Developing this service would cost the bank $1.5 per fraudulent transaction. For the fraudulent transactions that are still not identified by the model, the bank will need to pay the customer the entire transaction amount as it was doing earlier. Thus, the cost incurred now is due to the left out fraudulent transactions that the model fails to detect and the installation cost of the second level authentication service. Hence, the total savings for the bank would be the difference of costs incurred after and before the model deployment. You need to perform the following calculations sequentially to arrive at the final savings that your model can potentially provide to Finex.

Cost-Benefit Analysis

Let us take a look at what you need to do in order to perform the cost-benefit analysis step by step. Part I: Analyse the dataset and find the following figures: Average number of transactions per month Average number of fraudulent transactions per month Average amount per fraudulent transaction

Part II: Compare the cost incurred per month by the bank before and after the model deployment: Cost incurred per month before the model was deployed = Average amount per fraudulent transaction * Average number of fraudulent transactions per month Cost incurred per month after the model is built and deployed: Let TF be the average number of transactions per month detected as fraudulent by the model and let the cost of providing customer executive support per fraudulent transaction detected by the model = $1.5 Total cost of providing customer support per month for fraudulent transactions detected by the model = 1.5 * TF. Let FN be the average number of transactions per month that are fraudulent but not detected by the model Cost incurred due to these fraudulent transactions left undetected by the model = Average amount per fraudulent transaction * FN Therefore, the cost incurred per month after the model is built and deployed = 1.5*TF + Average amount per fraudulent transaction * FN Final savings = Cost incurred before - Cost incurred after.