Credit Card Fraud Projection

Capstone Project for the Caregivers and Machine Learning course

Vector Institute

By Catherine Ducharme
May 5, 2023

Executive Summary

We set out to determine which machine learning algorithm would work best for predicting credit card fraud. The dataset used was anonymized and largely imbalanced, illustrating the challenges faced by machine learning specialists working in cybersecurity when it comes to data availability. We have found that tree-based algorithms worked best when performance was measured with evaluation metrics that are well-suited for imbalanced datasets, namely precision-recall and F1 score. We singled out the XGBoost algorithm as having the best performance on this specific dataset.

We invite you to also review the code associated with this project.

Introduction

Credit and debit card fraud is a much bigger problem than most people expect before looking at the data, costing the world economy a staggering amount every year. In 2020 alone, $28.58 billion in credit card fraud loss was reported, and this number is projected to increase significantly in the coming decade.[1]

One would think that with the recent advances in machine learning, credit card fraud would become easier to prevent. However, it is also true that cybercriminals are coming up with increasingly creative ways to steal our money, highlighting the importance for cybersecurity and machine learning specialists to keep improving their techniques and models.

Extensive research has been, and is still, being done in order to better use machine learning to help prevent credit card fraud. The reason is, of course, that machine learning algorithms are evolving quickly and new capabilities are constantly being unlocked. Kaggle alone hosts several credit card fraud detection datasets. The one we will use for this project contains credit card transactions made by European cardholders and has been published by the Machine Learning Group at the Université Libre de Bruxelles.

Problem Definition

Dataset Presentation

This dataset presents transactions that occurred over two days in September 2013 and, as can be expected, is highly imbalanced: it contains 492 entries classified as fraudulent transactions out of a total of 284,807 transactions (0.172% positivity).

[1] Nikolovska, Hristina. Shocking Credit Card Fraud Statistics & Facts for 2023. *Moneytransfers.com*. Online: https://moneytransfers.com/news/2022/09/21/credit-card-fraud-statistics, last updated February 16, 2023.

It is also important to note that we will not be working with regular labelled data. For confidentiality reasons, the dataset only contains numerical input variables that are the result of a principal component analysis (PCA) transformation (Apart from the variables “Time”, “Amount” and “Class”). While we do not have any background information on the data these PCA variables represent, we do not expect this transformation to affect the classification accuracy of our model.

The variable “Time” represents the seconds elapsed between each transaction and the first transaction in the dataset, and the variable “Class” takes a value that is either 0 (no fraud) or 1 (fraud). The variable 'Amount' is the transaction amount.

Problem Definition

We will attempt to build a machine learning model that can predict if a credit card transaction is fraudulent or not. This is expected to be somewhat challenging: despite widespread machine learning implementation in the field of cybersecurity, there are still massive amounts of fraudulent financial transactions taking place every day online, illustrating the difficulty for even the highest-performing machine learning models to flag fraudulent transactions.

Part of the problem, of course, is that it can be challenging to find publicly-available, good-quality training data to train a supervised learning model, in part because of privacy and security concerns. We are lucky to have access to this dataset from Kaggle, even if we are lacking information about the variables. However, given the very small proportion of fraudulent transactions contained in the dataset, we expect training the algorithm to be a challenge.

Proposed model and approach

We propose to study our data to determine which ones of the variables in the dataset are the most influential in determining whether or not a transaction will be fraudulent (i.e. have the highest correlation to the Class outcome variable), and to see if these variables will be the same ones chosen by our best performing model to predict the Class outcome.

We will attempt to predict fraudulent credit card transactions using logistic regression, random forest and XGBoost algorithms, before choosing the most efficient algorithm based on their precision-recall and F1 scores. Those metrics were selected for their ability to better represent the performance of a model when working with an imbalanced dataset.[2]

Logistic regression seems like an obvious choice when presented with a binary outcome problem. However, according to recent research[3], a random forest algorithm does best at predicting credit card fraud rates when using a genetic algorithm for feature selection. While the variables in our dataset were probably not selected this way, we will still test this option. We will also test the XGBoost algorithm, a similar tree-based algorithm that uses gradient boosting instead of bagging.

[2] Brownlee, Jason. “How to Use ROC Curves and Precision-Recall Curves for Classification in Python”, https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/, last updated on January 13, 2021.

[3] See Ileberi, E., Sun, Y. & Wang, Z. A machine learning based credit card fraud detection using the GA algorithm for feature selection. *J Big Data* 9, 24 (2022). https://doi.org/10.1186/s40537-022-00573-8

Data Exploration and Description

General Description
The dataset contains 31 columns (Time, Amount, Class, and 28 PCA-transformed variables we do not have any information about) and 284,807 rows. Of these rows, 99.83% (284,315) represent normal, non-fraudulent transactions and 0.17% (492) represent fraudulent transactions.

Correlation
Using Pandas' corr() function, we calculated the correlation between all the variables in the dataset and the Class outcome (0 for non-fraudulent transactions, 1 for fraudulent ones). We found that no single variable has a meaningful correlation to the Class variable. A correlation coefficient of at least 0.6, or -0.6, could have been considered meaningful, but the highest coefficient (in absolute terms) was -0.326481 for V17.

We also saw that some variables, including Time and Amount, have very little correlation to the Class outcome.

A heatmap was produced to illustrate correlation between all variables in the dataset. It has shown that PCA variables are not correlated to each other, but that many of them are correlated not only to class outcome but also to the Time and Amount variables. The highest correlation in the dataset, according to this map, is between Amount and V7.

PCA Variables
In order to better understand how these variables work, we compared the mean of all variables depending on their Class outcome. We found that the mean of the PCA variables tend towards 0 when the transaction is normal (Class 0) but takes different values when the transaction is fraudulent (Class 1). This effect is not as strong for the variables that were shown to have little correlation to the Class outcome. We also produced a graphical representation of this effect.

Amount
We compared statistics for this variable when transactions are fraudulent vs normal and found that the fraudulent transactions have a higher mean (€122 versus €88) but a lower median (€9 versus €22). This means that more than half of all fraudulent transactions are for amounts smaller than €10, which was unexpected. The higher mean for the fraudulent transactions, however, shows that the fraudulent set of transactions is more sensitive to outliers since it is much smaller than the set of normal transactions.

A graph of the relation between Amount and Class seems to show a much higher likelihood for low amount transactions to be fraudulent. However, it is worth remembering that, given the smaller amount of both high amount transactions and fraudulent transactions in our dataset, this impression might be due to a lack of data points.

Time

We did not expect the Time variable to be related to the Class outcome. We created a visual representation of the relation between Time and Class, and found that, indeed, those two variables do not seem to be meaningfully correlated.

Missing values and outliers

We verified that the dataset has no missing values. Also, as most of the dataset is transformed with PCA, we are assuming that any outlying value for these variables has already been taken care of. We therefore chose not to perform any outliers treatment on the dataset.

Imbalance[4]

We have considered several approaches to address the high class imbalance in our dataset:

- Undersampling (randomly removing instances from the majority class to balance the dataset). Since we have a limited number of instances in the minority class, this approach seemed problematic.
- Oversampling (generating synthetic instances of the minority class to balance the dataset).
- Using ensemble methods (such as bagging and boosting) to create a balanced dataset, using a combination of under- and oversampling techniques.
- Using evaluation metrics that are suitable for imbalanced datasets such as precision-recall and F1 score.

We have chosen to choose evaluation metrics adapted to our dataset (precision-recall and F1 score) rather than modifying it.

Model

“Limited” dataset

We created a limited version of our dataset based on the correlation coefficients obtained for each variable during data exploration. This limited dataset does not contain variables that were found to have very little correlation with the Class outcome. Our logistic regression model was then tested with both the full and limited datasets, and the limited dataset was found to perform better. It was not used, however, with the two tree-based algorithms, given that the max_feature and max_depth hyperparameters already ensure that only the most relevant variables are used in our models.

Test models

We tested three algorithms on our dataset split into training, validation and test sets, and then recorded their precision-recall and F1 scores and outputted their precision-recall curves:

- Logistic regression
- Random forest: we selected the hyperparameters n_estimators=100, max_features=8 and random_state=42

[4] We acknowledge ChatGPT’s help for the content of this section.

- XGBoost: we selected the hyperparameters n_estimators=100, max_depth=5 and random_state=42.

Logistic regression performance for the minority class on the test set:

- Precision: 0.85
- Recall: 0.56
- Precision-Recall: 0.74
- F1 score: 0.67

The low recall score indicates that our classifier outputs a lot of false negatives. It seems to fail at recognizing many of the fraudulent transactions.

Random forest performance for the minority class on the test set:

- Precision: 0.98
- Recall: 0.81
- Precision-Recall: 0.86
- F1 score: 0.88

XGBoost performance for the minority class on the test set:

- Precision: 0.98
- Recall: 0.81
- Precision-Recall: 0.89
- F1 score: 0.88

Results and Findings

Both the random forest and XGBoost algorithms performed well on predicting Class outcomes on our dataset, with XGBoost consistently getting slightly better precision-recall. It would be our chosen option as the best model to predict credit card fraud with this dataset.

An F1 score of 0.88 is considered good. Given the great precision rate of our model, it would seem that recall, at 0.81, is the reason our F1 score is not higher. This means that our model's weakness is that it sometimes does not recognize fraudulent transactions as such and produces false negatives.

The 5 most important features in our XGBoost model were the following, in descending order of importance:

1. V14
2. V4
3. V26
4. Amount
5. Time

We were surprised to see how much the variables Amount, Time and V26 influenced the output of our XGBoost model, given that our initial exploration of the data did not find any significant correlation between these variables and the Class outcome. It is a powerful reminder that tree-based models do not only rely on correlation, but also on entropy, to select the variables that will yield the best performance when our model is tested on unseen data.

Conclusions and Future Work

It is very difficult to establish conclusions on credit card fraud based on a set of anonymized data. That V14 turned out to be the most influential feature in predicting credit card fraud does not actually tell us much. We understand the difficulties for researchers to make their results public when relying on datasets containing such sensitive information, and for this reason we expect most future research in the area will not make their code and datasets accessible to the public.

Because the data is anonymised and that we have no background information on it, it is also difficult to evaluate the likelihood of bias in our model's predictions. Bias can happen at all steps of the machine learning process, from data collection to model tuning. Not knowing how the data was collected, for instance, means that, even if we knew what our variables represented, we would hesitate to draw firm conclusions on their respective weight when trying to predict credit card fraud.

Moreover, while machine learning is an important tool in a credit card company's toolbox in order to help prevent fraud, streamline its security operations and protect its clients' money, it is important to remember that the classifications of a machine learning algorithm will not be perfect: as we have shown, it is difficult to get both good precision and good recall when working with such a small proportion of positive cases.

References

Bajaj, Aayush. What does your classification metric tell about your data? https://towardsdatascience.com/what-does-your-classification-metric-tell-about-your-data-4a8f35408a8b, December 25, 2019.

Banu, R. V. and N. Nagaveni, "Preservation of Data Privacy Using PCA Based Transformation," *2009 International Conference on Advances in Recent Technologies in Communication and Computing*, Kottayam, India, 2009, pp. 439-443, doi: 10.1109/ARTCom.2009.159.

Brownlee, Jason. “How to Use ROC Curves and Precision-Recall Curves for Classification in Python”, https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/, last updated on January 13, 2021.

Cheng, Casey. Principal Component Analysis (PCA) Explained Visually with Zero Math. https://towardsdatascience.com/principal-component-analysis-pca-explained-visually-with-zero-math-1cbf392b9e7d, February 3, 2022.

Ileberi, E., Sun, Y. & Wang, Z. A machine learning based credit card fraud detection using the GA algorithm for feature selection. *J Big Data* 9, 24 (2022). https://doi.org/10.1186/s40537-022-00573-8

Lev, Alon. XGBoost versus Random Forest. *Qwak*. Online: https://www.qwak.com/post/xgboost-versus-random-forest, December 19, 2022.

Machine Learning Group, Université Libre de Bruxelles. Credit Card Fraud Detection (Kaggle dataset). https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud?resource=download&select=creditcard.csv, last updated 5 years ago.

Nikolovska, Hristina. Shocking Credit Card Fraud Statistics & Facts for 2023. *Moneytransfers.com*. Online: https://moneytransfers.com/news/2022/09/21/credit-card-fraud-statistics, last updated February 16, 2023.