Predicting Fraudulent Online Transactions

Applying Feature Engineering and Boosting Algorithm on High Dimensional Data

Abstract

In a recent study by Fortune Insights and published on numerous international news organization, "the market (fraud prevention) is likely to witness accelerated growth in the coming years with the advent of artificial intelligence (AI) integrated systems and increasing operational efficacy" [(1)]. Visa, world's largest retail electronic payments processing network, alone handles more than 150 million transactions per day [(2)]. "The Federal Trade Commission's online database of consumer complaints has compiled 13 million complaints from 2012 to 2016, with 3 million in 2016 alone. Of those, 42 percent were fraud related" [(3)].

The number of daily transactions will increase as more people ventures into the world of internet. Subsequently, fraudulent transactions will rise as well. This paper is about applying machine learning algorithms to detect fraudulent transactions helping hundreds of thousands of businesses reduce their fraud loss and increase their revenue.

Introduction

A business loses money when a) a financial transaction initiated by wrong cardholder is approved. Or b) a financial transaction initiated by right cardholder is declined. In both situations, an algorithm has mislabeled the transactions. The basis of this paper is to improve the algorithm to label transactions more accurately.

History and Present

For detection of fraudulent activities on the large scale, massive use of data mining is required to find patterns. “Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior. These non-conforming patterns are often referred to as anomalies, ... Anomaly detection finds extensive use in a wide variety of applications such as fraud detection for credit cards …” [(4)].

“However, there is a lack of published literature on credit card fraud detection techniques, due to the unlabeled credit card transactions dataset for researchers.” [(5)]. The dataset for anomaly detection consists of multiple variables such as credit card details, amount transaction, location, time, and personal details of the cardholders that are anonymized. This type of dataset is called high dimensional data or imbalanced data. This type of data will often introduce biases which makes the accuracy of the prediction not accurate.

Techniques

The data comes from Vesta's real-world e-commerce transactions and contains a wide range of features from device type to product features [(6)]. There are several techniques that are applied to find patterns in a high dimensional data such as this. “Feature selection, as a preprocessing step to machine learning, is effective in reducing dimensionality, removing irrelevant data, increasing learning accuracy, and improving result comprehensibility” [(7)]. Another way to reduce dimensionality is to create new feature from existing ones and the process is called feature extraction. It can be done manually with domain knowledge [(8)].

Once the data is prepared, machine learning algorithms are applied to the data to find the model that is most accurate at predicting fraudulent transaction. The type of algorithm depends on the type of data and the algorithms that works best with high dimensional data is gradient boosting.

The accuracy of a predictive model can be increased by applying boosting algorithms. The most popular boosting algorithm is Gradient Boosting. "Gradient Boosting (GB) is an iterative algorithm that combines simple parameterized functions with "poor" performance (high prediction error) to produce a highly accurate prediction rule. In contrast to other statistical learning methods usually providing comparable accuracy (e.g., neural networks and support vector machines), GB gives interpretable results, while requiring little data preprocessing and tuning of the parameters" [(9)].

Finally, the best algorithm will be chosen using ROC (Receiver Operating Characteristic) curve that tells us about how good the model can distinguish between two things (e.g. If a transaction is fraudulent or not) [(10)].

Data Collection

The data is separated into two files *identity* and *transaction* which will be joined by *TransactionID* variable. Not all transactions have corresponding identity information.

Transaction Table
TransactionDT: timedelta from a given reference datetime (not an actual timestamp)
TransactionAMT: transaction payment amount in USD
ProductCD: product code, the product for each transaction
card1 - card6: payment card information, such as card type, card category, issue bank, country, etc.
addr: address
dist: distance
P_ and (R__) emaildomain: purchaser and recipient email domain
C1-C14: counting, such as how many addresses are found to be associated with the payment card, etc. The actual meaning is masked.
D1-D15: timedelta, such as days between previous transaction, etc.
M1-M9: match, such as names on card and address, etc.
Vxxx: Vesta engineered rich features, including ranking, counting, and other entity relations.
Categorical Features *ProductCD* *card1 - card6* *addr1, addr2*

*P*emaildomain R*emaildomain* *M1 - M9*

Identity Table
Variables in this table are identity information – network connection information (IP, ISP, Proxy, etc) and digital signature (UA/browser/os/version, etc) associated with transactions. They're collected by Vesta's fraud protection system and digital security partners. (The field names are masked and pairwise dictionary will not be provided for privacy protection and contract agreement)
Categorical Features *DeviceType* *DeviceInfo* *id*12 - id*38*

```
Train dataset has 590540 rows and 434 columns.
Test dataset has 506691 rows and 433 columns.
```

Figure 1: Dimension of the datasets

Data Preparation

In order to feature engineer and run machine learning models, the data needs to be prepared because this dataset is messy. First, the dataset has a lot of missing values. "Missing data can reduce the statistical power of a study and can produce biased estimates, leading to invalid conclusions" [(11)]. Figure 2

Next, the dataset is imbalanced. Most of the transactions are not fraud. "A dataset is imbalanced if the classification categories are not approximately equally represented … Predictive accuracy, a popular choice for evaluating performance of a classifier, might not be appropriate when the data is imbalanced and/or the costs of different errors vary markedly" [(12)]. Figure 3

```
isFraud              0                    id_01        0
TransactionDT        0                    id_02     3361
TransactionAmt       0                    id_03    77909
ProductCD            0                    id_04    77909
card1                0                    id_05     7368
card2             8933                    id_06     7368
card3             1565                    id_07   139078
card4             1577                    id_08   139078
card5             4259                    id_09    69307
card6             1571                    id_10    69307
dtype: int64                              dtype: int64
% of missing data =  41.17794374769424    % of missing data =  36.47062392101669
```

Figure 2: Missing Values

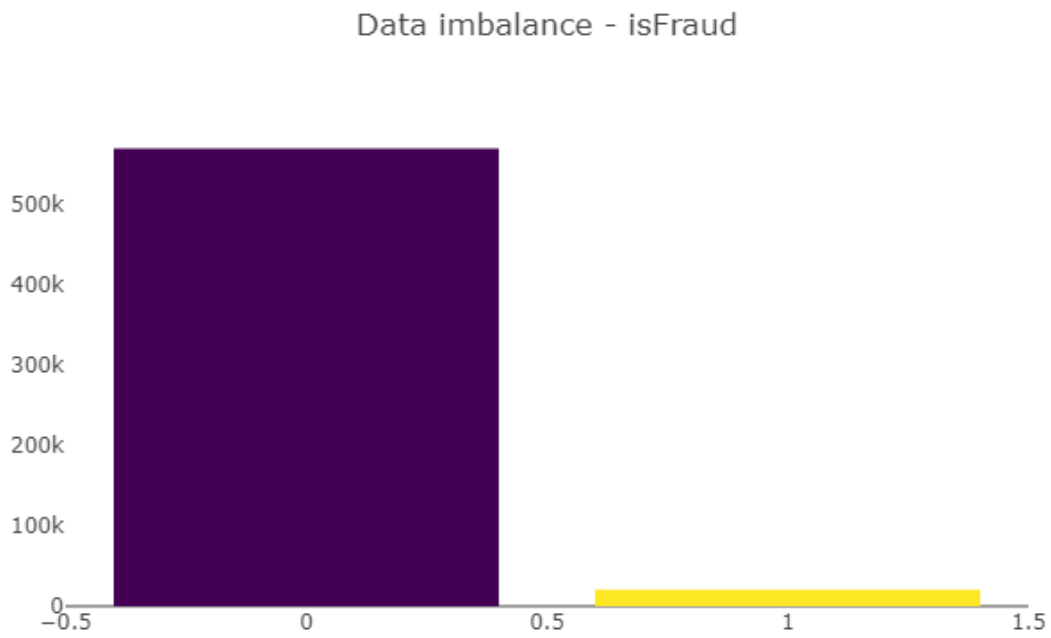


Figure 3: Imbalanced data

Feature Engineering

Selecting the right variable will yield the best model however we should avoid overfitting or underfitting the model. “Overfitting refers to the phenomenon where the validation error

increases while the training error decreases. This occurs because the model learns the expected output for every input data instead of learning the real data distribution. In contrast, underfitting problems occur when a model cannot learn enough because of insufficient training data" [(13)]. With that in mind, the following are the ways features will be engineered.

Feature Encoding – As seen in the data dictionary, many of the variables are categorical. Most of the machine learning algorithms cannot handle categorical variables unless they are converted to numerical values. Frequency encoding is a powerful technique that allows to see whether column values are rare or common. This will also help with outliers by using frequency encoding of a variable, all values that appear less than 0.1% can be removed by replacing them with a new value like -9999 [(14)].

Combining/Aggregating – Two (string or numeric) variables can be combined into one variable. For example, card1 and card2 may not correlate with the target variable. However, the combination of these variables may correlate with target variable.

Feature Split - Splitting features is a good way to make them useful in terms of machine learning. By extracting the utilizable parts of a column into new features we enable machine learning algorithms to comprehend them and improve model performance by uncovering potential information. [(15)]

Outlier Detection with Percentiles - We can assume a certain percent of the value from the top or the bottom as an outlier.

With the feature engineering performed on the dataset, 84 variables were removed, and some new variables were created which better represents the data. Figure 4

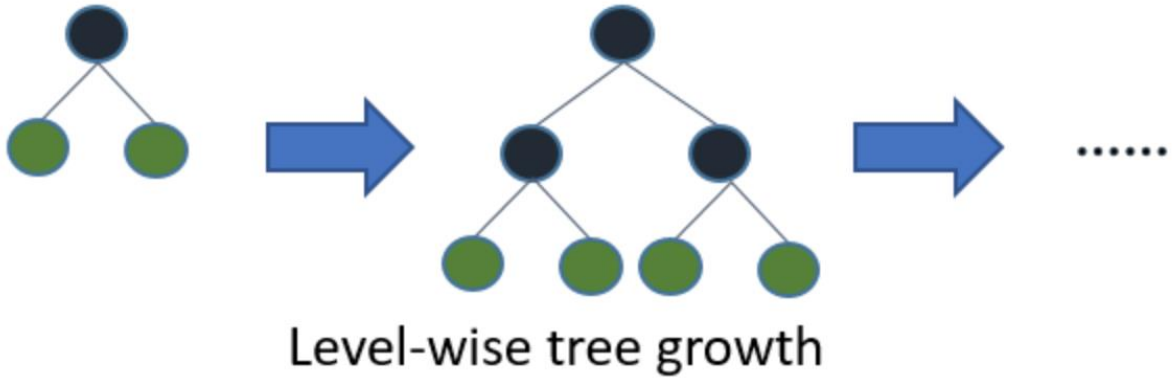
```
Train dataset has 590538 rows and 374 columns.
Test dataset has 506691 rows and 373 columns.
```

Figure 4: Dimension of the Clean data

Model

Light GBM is a gradient boosting framework that uses tree-based learning algorithm. The volume of data is increasing every minute and it is becoming difficult for traditional data science algorithms to give faster results. Light GBM is prefixed as 'Light' because of its high speed. Light GBM can handle the large size of data and takes lower memory to run. Another reason of why Light GBM is popular is because it focuses on accuracy of results. LGBM also supports GPU learning and thus data scientists are widely using LGBM for data science application development [(16)].

Gradient boosting is a machine learning algorithm that produces a prediction model in the form of an ensemble of weak classifiers, optimizing for a differentiable loss function. One of the most popular types of gradient boosting is gradient boosted trees, that internally is made up of an ensemble of week decision trees. There are two different ways to compute the trees: level-wise and leaf-wise as illustrated by the diagram below: Figure 5



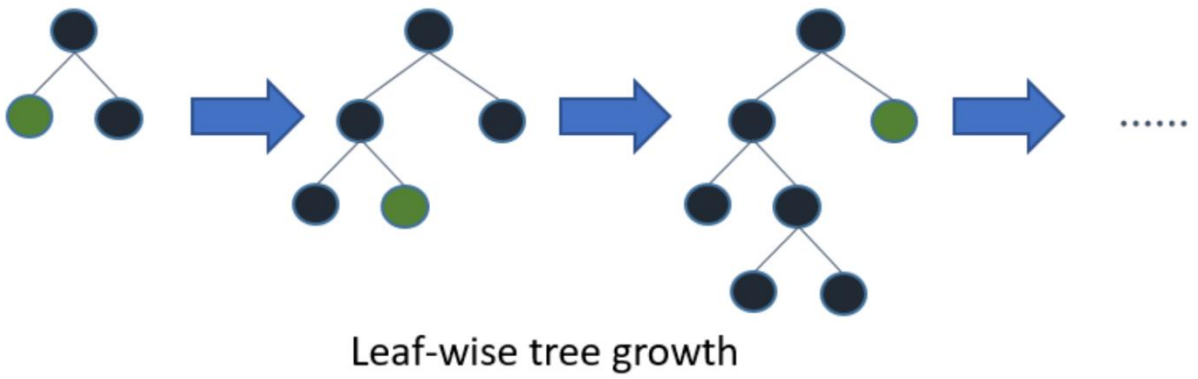


Figure 5

The level-wise strategy grows the tree level by level. In this strategy, each node splits the data prioritizing the nodes closer to the tree root. The leaf-wise strategy grows the tree by splitting the data at the nodes with the highest loss change. Level-wise growth is usually better for smaller datasets whereas leaf-wise tends to overfit. Leaf-wise growth tends to excel in larger datasets where it is considerably faster than level-wise growth.

A key challenge in training boosted decision trees is the computational cost of finding the best split for each leaf. Conventional techniques find the exact split for each leaf and require scanning through all the data in each iteration. A different approach approximates the split by building histograms of the features. That way, the algorithm doesn't need to evaluate every single value of the features to compute the split, but only the bins of the histogram, which are bounded. This approach turns out to be much more efficient for large datasets, without adversely affecting accuracy.

LightGBM is a fast, distributed, high performance gradient boosting that was open source by Microsoft around August 2016. The main advantages of LightGBM includes:

- Faster training speed and higher efficiency: LightGBM use histogram-based algorithm i.e. it buckets continuous feature values into discrete bins which fasten the training procedure.
- Lower memory usage: Replaces continuous values to discrete bins which result in lower memory usage.
- Better accuracy than any other boosting algorithm: It produces much more complex trees by following leaf wise split approach rather than a level-wise approach which is the main factor in achieving higher accuracy. However, it can sometimes lead to overfitting which can be avoided by setting the max_depth parameter.
- Compatibility with Large Datasets: It can perform equally good with large datasets with a significant reduction in training time.
- Parallel learning supported.

The significant speed advantage of LightGBM translates into the ability to do more iterations and/or quicker hyperparameter search, which can be very useful if we have a limited time budget for optimizing your model or want to experiment with different feature engineering ideas. [17]

Implementation of Light GBM is easy, the only complicated thing is parameter tuning. Parameter tuning is the selection of parameter values, which are optimal in some desired sense. The parameters are the weights and biases of the tree nodes.

Results

After fine tuning the parameters and training the model, the model is trained to perform prediction on data. The accuracy of the prediction is measured using ROC curve as mentioned

above. The AUC score below informs that the model accurately labeled more than 90% of the data as fraudulent. Figure 6

```
Training on fold 1
Training until validation scores don't improve for 500 rounds
[1000]  training's auc: 0.999997        valid_1's auc: 0.901271
Early stopping, best iteration is:
[1464]  training's auc: 1       valid_1's auc: 0.901718
Fold 1 finished in 0:02:23.904714
Training on fold 2
Training until validation scores don't improve for 500 rounds
[1000]  training's auc: 0.99995 valid_1's auc: 0.920378
Early stopping, best iteration is:
[934]   training's auc: 0.99991 valid_1's auc: 0.920636
Fold 2 finished in 0:03:48.754140
Training on fold 3
Training until validation scores don't improve for 500 rounds
[1000]  training's auc: 0.999418        valid_1's auc: 0.911114
Early stopping, best iteration is:
[874]   training's auc: 0.998797        valid_1's auc: 0.911216
Fold 3 finished in 0:05:16.312283
Training on fold 4
Training until validation scores don't improve for 500 rounds
[1000]  training's auc: 0.997652        valid_1's auc: 0.933295
Early stopping, best iteration is:
[1218]  training's auc: 0.999011        valid_1's auc: 0.933608
Fold 4 finished in 0:08:26.476478
Training on fold 5
Training until validation scores don't improve for 500 rounds
[1000]  training's auc: 0.995677        valid_1's auc: 0.929563
Early stopping, best iteration is:
[1176]  training's auc: 0.997443        valid_1's auc: 0.929968
Fold 5 finished in 0:09:31.453626
------------------------------
Training has finished.
Total training time is 0:29:26.915204
Mean AUC: 0.9194289866227334
------------------------------
```

Figure 6: Accuracy Score of LGBM

Discussion

AI computational power is doubling every three months or so, according to a report by Stanford University. That means AI is getting more powerful at a much faster rate than what

Moore's Law expects for computer chips. [18] LightGBM is one of the innovations that came from this progression. Dealing with high dimensional masked datasets is extremely difficult as demonstrated by this project. Except for few categorical variables which can be used to build domain knowledge, considerable numbers of variables are continuous. LightGBM is an exceptionally good at dealing with such datasets. It requires very little preprocessing. It can handle missing values out of the box, it is robust to all types of distribution among the variables, it does not require one-hot encoding and it is very fast to train.

Conclusion

Fraud detection and analysis accompanies high dimensional data that are anonymized. Therefore, the underlying problem this paper addresses is how to work with dataset that has high number of variables; the feature engineering to reduce dimensionality; and applying machine learning algorithm – Light Gradient Boosting Machine to accurately predict whether a transaction is fraudulent or not. The model's accuracy is evaluated on area under the ROC curve.

References:

1. "Fraud Detection and Prevention Technology Market 2019 Global Industry Size, Demand, Growth Analysis, Share, Revenue and Forecast 2026." Reuters, Thomson Reuters, https://www.reuters.com/brandfeatures/venture-capital/article?id=129567.
2. "Small Business Retail." Visa, https://usa.visa.com/run-your-business/small-business-tools/retail.html.
3. "Credit Card Fraud and ID Theft Statistics." CreditCards.com, 10 June 2019, https://www.creditcards.com/credit-card-news/credit-card-security-id-theft-fraud-statistics-1276.php.

4. Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." ACM computing surveys (CSUR) 41.3 (2009): 15.
5. Soh, W. W., and R. Yusuf. “Predicting Credit Card Fraud on an Imbalanced Data”. International Journal of Data Science and Advanced Analytics, Vol. 1, no. 1, Apr. 2019, pp. 12-17, http://ijdsaa.com/index.php/welcome/article/view/3.
6. “IEEE-CIS Fraud Detection.” Kaggle, https://www.kaggle.com/c/ieee-fraud-detection/overview.
7. Yu, Lei, and Huan Liu. "Feature selection for high-dimensional data: A fast correlation-based filter solution." Proceedings of the 20th international conference on machine learning (ICML-03). 2003.
8. “Feature Extraction in High-Dimensional Data - CMIH - The Centre for Mathematical Imaging in Healthcare.” CMIH, https://www.cmih.maths.cam.ac.uk/projects/feature-extraction-high-dimensional-data/.
9. Guelman, Leo. “Gradient Boosting Trees for Auto Insurance Loss Cost Modeling and Prediction.” Expert Systems with Applications, Pergamon, 16 Sept. 2011, https://www.sciencedirect.com/science/article/pii/S0957417411013674.
10. D'Souza, Jocelyn. “Let's Learn about AUC ROC Curve!” Medium, GreyAtom, 15 Mar. 2018, https://medium.com/greyatom/lets-learn-about-auc-roc-curve-4a94b4d88152.
11. Kang, Hyun. “The Prevention and Handling of the Missing Data.” Korean Journal of Anesthesiology, The Korean Society of Anesthesiologists, May 2013, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3668100/.

12. Chawla, Nitesh V. “Data Mining for Imbalanced Datasets: An Overview.” SpringerLink, Springer, Boston, MA, 1 Jan. 1970, https://link.springer.com/chapter/10.1007/0-387-25465-X_40.
13. Nusrat, et al. “A Comparison of Regularization Techniques in Deep Neural Networks.” MDPI, Multidisciplinary Digital Publishing Institute, 18 Nov. 2018, https://www.mdpi.com/2073-8994/10/11/648/htm.
14. Roy, Baijayanta. “All about Categorical Variable Encoding.” Medium, Towards Data Science, 21 Oct. 2019, https://towardsdatascience.com/all-about-categorical-variable-encoding-305f3361fd02.
15. Rencberoglu, Emre. “Fundamental Techniques of Feature Engineering for Machine Learning.” Medium, Towards Data Science, 1 Apr. 2019, https://towardsdatascience.com/feature-engineering-for-machine-learning-3a5e293a5114.
16. Mandot, Pushkar. “What Is LightGBM, How to Implement It? How to Fine Tune the Parameters?” Medium, Medium, 1 Dec. 2018, https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc.
17. “Lightgbm.” Lightgbm, http://ethen8181.github.io/machine-learning/trees/lightgbm.html.
18. Peers, Martin. “Briefing: AI Power Doubling Every Three Months, Study Finds.” The Information, https://www.theinformation.com/briefings/809dfe.