Fraud Detection with Light Gradient Boosting Model (LGBM)

Xiaoxuan Shi (xiaoxuan_shi@berkeley.edu)

Abstract

In this project, I implemented and compared the performance of logistic regression, random forests, and gradient boosting in the binary classification problem. It turns out lgbm greatly beats logistic regression and random forests in predicting fraud transactions. With a lot work on unmasking the muted features, the model taking in all cleaned features correctly classified more than 95% of each class. To better interpret fraud transaction behaviors, the final lgbm model takes in ten features and gives over 94% accuracy in both class. Compared to the whole model, these 10 features capture most information in identifying fraud cases among 52 unmasked meaningful features.

INTRODUCTION

Fraud detection is getting popular with the thriving e-commerce industry. There are more than 3.2-million fraud cases reported to the Federal Trade Commission (FTC) in 2019[1]. As more people turn to e-commerce for convenience, more criminals are attracted to easy money. And companies who need effective prediction of fraud transactions are of different social roles — online shopping website like Amazon needs alert when potential fraud comes; bank like Chase needs to secure customer accounts; e-payment like Paypal needs payment verification.

Multiple machine learning methods are mainly used to work out a decision rule nationwide, mainly SVM and Random Forest [2]. An effective fraud detection system should be able to reach a balance of false positive rate and false negative rate. The goal of fraud detection is to prevent every fraud detection so many people focus on reducing false positive rate. But at the same time, customer experience should be taken into consideration since a false negative will cause customers some troubles.

A major challenge in applying ML to fraud detection is presence of highly imbalanced data, as is the case in most available financial datasets. Intuitively, in real world, fraud transactions are far less than normal ones, and that's why we need specific model feeding big data to catch the crimes. The usual technique is to upsample minority class or downsample majority class, which is used in this project.

And another challenge is feature engineering since many new features can be separated from a blend of information and many "noisy" features can be dropped to get a concise model. Different companies collect different information, and our data contains information of identity, payment and product within 435 features. To reduce dimensions, I drop features with missing values more than 60% and use PCA in one series of more than 150 features. Also I make reasonable hypothesis on the date-related features to uncover their meaning in real world.

Our goal is to build an effective and efficient classifier to separate fraud transactions from non-fraud ones. In our project, multiple models like logistic regression, random forest and lgbm are implemented on the labeled data. We compare performance of these models by true positive rate and a risk function given by Altendorf [3].

DATASET AND ANALYSIS

The data is collected by Vesta's fraud protection system and digital security partners [4]. But the field names of original data are masked for privacy protection and contract agreement. IEEE-CIS is partnering with Vesta Corporation to seek the best solutions for fraud prevention industry.

Two datasets are provided. There are 394 features in "transaction" and 41 features in "identity". Variables in "transaction" table are transaction details – product code, device information, payment method etc. Variables in "identity" table are identity information – network connection information (IP, ISP, Proxy, etc) and digital

signature (UA/browser/os/version, etc) associated with transactions. After merging the two datasets on “TransactionID”, the whole dataset contains 590,541 observations and 435 features.
And here are some important features in the raw data.

Table1 Brief summary of important features

Feature Name	Feature Meaning	Comments
isFraud	Target	Highly unbalanced (2.6% fraud)
TransactionAmt	Transaction Amount in USD	Different distribution in two classes (Figure1) Some new features like decimal part of amount, log amount can be drawn. Also group by categorical features to get statistical values.
Card1-6	Payment card info	Combined with card-open&first-transaction time to make a card-id
D1-D15 (*)	Time-related	Corresponding to the hypothesis that D1 is the open time of card, D2 is time of first transaction. D3 is time delta since the previous transaction on the same card. And the others are still beyond understanding.
TransactionDT (*)	Time-related	Corresponding to the hypothesis that data starts from 2017-12-01, and it ends at 2018-12-31. Some new features like month, day, weekday, hour, is_holiday, can be drawn..
DeviceInfo	Device Information	Combined with id30-35 and Device Type to make a device-id

The response "isFraud" is unbalanced binary value, with number of normal transactions 30 times more than fraud ones. And in some features even over 90% are missing values. There are 208 meaningless columns with more than 60% missing values each. And in the features left, 60 features share a >0.95 correlation with at least one of the others.

1)Time-related features

After analyzing the maximum and minimum values of "TransactionDT", it supports the hypothesis that our data starts at the 12/01/2017 and ends at 12/31/2018. The time span is 395 days, 13 months. It's reasonable because the kaggle competition is released at 09/2019. Thus "TransactionDT" records the time delta of transaction time and 12/01/2017 in seconds. Then many new features like date, hour, day, weekday, month, is_holiday, is_December can enrich the dataset. And transactions within one hour and one day are calculated.

With similar hypothesis and techniques, D1 can be the open time of card and D2 can be time of first transaction. D3 is time delta since the previous transaction by the same card_id (Table 1) . And the others are still beyond scope.

2)Product-related features

“ ProductCD" feature classifies products into five types (Table2), among which the fraud rate differs a lot. And I calculated the mean of transaction amount in each product type to get a general sense of price of each product.

Product Type	Fraud Rate
C	0.116873
H	0.047662
R	0.037826
S	0.058996
W	0.020399

Table2 Fraud Rate of different types of Product

It's highly likely that "TransactionAmt" plays an important role in fraud detection in view of different distribution in two classes (Figure1). To better analyze its impact, I take decimal_amt and log_amt as new features.

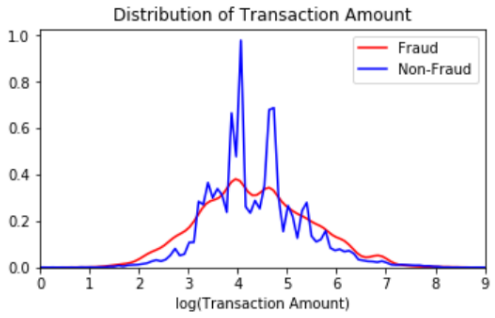

Figure1 Distribution of "TransactionAmt"

3)Payment-related features

Card1-Card6 gives the card information in one trasactiom, including bank, credit/debit and some muted numbers, maybe some verification codes. Combining the analysis of date-related features, I combined card1-card6 with open-card-time and first-transaction-time to construct a card_id feature, which roughly identify a unique card. And there are 189,905 "unique" cards used in the 590,541 transactions.

4)Device-related features

The raw data includes information including device brand, device version, browser, browser version etc. I combined them as a device_id to analyze transactions on one device. There are 9815 unique device_id, and as a result of many missing values, 449,680 out of 590,541 transactions have no device_id.

I calculate how many devices a card has as a new feature "card_id_number_of_device", and it turns out one card could have up to 109 devices to complete all the transactions. In spite of no knowledge of completeness of those identity features, our assumptions are reasonable after feature engineering.

METHOD

1) Reduce Dimensions

After feature enrichment, there are nearly 500 features in all, so the first step is to drop some features. Firstly I dropped 208 features with missing values more than 60%. Secondly by calculating correlations of each column, I dropped 60 features above the line of 0.95 correlation to all others. Thirdly, to deal with V1-V339, three directions are chosen by PCA because the three directions account for over 80% of the variance. After those steps, 52 features are left.

2) Categorical features encoding

Obviously categorical features cannot be put directly into lgbm model. Considering dimensionality, I choose target encoding instead of one-hot encoding. The missing values are filled with global mean. In all 22 features are encoded.

3) Downsampling

As mentioned in the first section, unbalanced data is a challenge for fraud detection. We solve the problem by downsampling the majority class instead of upsampling the minority class. In lgbm the stop criteria is decided by validation set, so it may lead to bad result on test data if there are identical fraud samples in train and validation data. Leaving out most of normal transaction data, 41,326 observations are left. And we randomly split 20% of them as validation data.

4) Preliminary Models

Before the final model, we tried different models to catch a baseline performance. All raw features are put into those toy models. And they all did bad on predicting fraud ones while true negative rate can be over 99%.

The only linear model is Logistic Regression. The true positive rate can be as low as 11% (Table3). By every metric its performance is much worse than Random Forest and Lgbm. Thus linear model doesn't work in fraud detection.

Table3 Performance of different toy models

Metric	Logitic Regression	Random Forest	LightGBM
Total accuracy	0.92	0.96	0.94
TNR	0.996	0.99	0.99
TPR	0.11	0.64	0.63
ROC	0.56	0.81	0.97

The Random Forest Model identifies similar amount fraud transactions as Lgbm and RF shows a 0.02% higher accuracy. But according to AUC, RF doesn't perform better than lgbm.

Lgbm is a raising machine learning method these days with higher efficiency. For our big dataset, it hits better ROC score using lower memory and shorter running time than RF. As a result we choose the best performance type of model--Lgbm as final model.

5)Final Model

The cleaned data has 52 features with known or guessed meanings. Fit them in a lgbm model and tune parameters by GridSearch.

With feature importance of this whole model, I fit a smaller lgbm with top 10 featrues to better analyse the fraud behaviours.

RESULT

1)With all cleaned features

The whole model converges in 566th iteration, with 99.3% validation accuracy and 0.113 logloss. The parameters are shown in Table4.

Table4 Tuned LightGBM parameters

Parameters	Values
seed	2020
metric	auc
num_leaves	256
learning_rate	0.01
n_estimators	20000
min_split_gain	0.0
min_child_weight	0.001
min_child_samples	20
early_stopping_rounds	100

This model has a great performance. AUC is 0.953. Although AUC is a bit lower than the preliminary model which takes in all raw features, the final model far outweighs the preliminary in predicting fraud transactions. This final model correctly predicts 3884 nonfraud transactions and 3992 fraud ones. True negative rate is 94.6%. And true positive rate is even higher, 95.9%. As I mentioned in section one, a balance between false positive and false negative rate is important for both the sell side and buy side in one transaction. Obviously, this model reaches a balanced prediction ability between fraud and normal transaction, or to say it is a fair classifier.

The result tells the efficiency/sufficiency of these 52 features. It beats the performance of preliminary model with all raw features. The information it contains shines light on further discussion. To analyze the predictors of fraud transaction, it's important to look into their feature importance (Figure2). It gives top 20 features in this

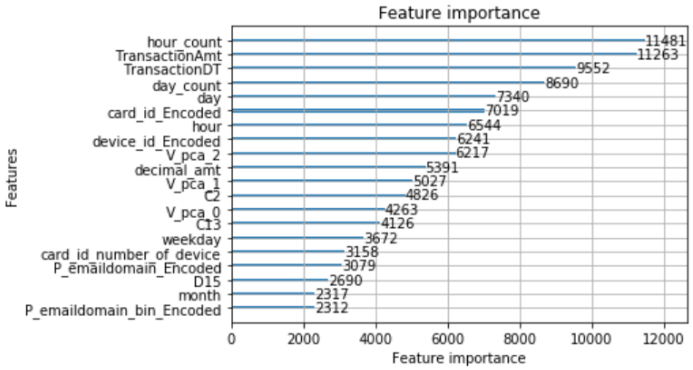

Figure2 Feature Importance of top20 features in Lgbm

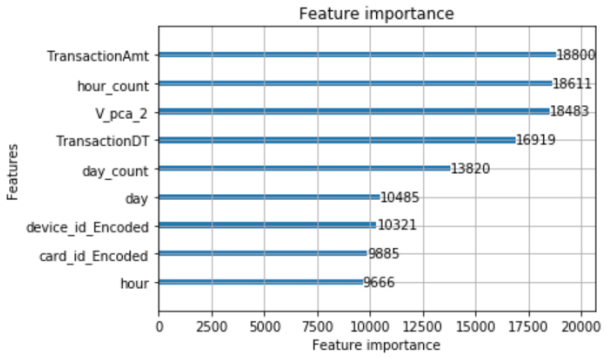

Figure3 Feature Importance of smaller lgbm

model, and most of them are categorical features.

Comparing this model with preliminary model, it's easy to draw the conclusion that there are a bunch of noisy features which hinder the classifier in predicting fraud transactions. To further interpret the fraud behavior, a smaller lgbm with top10 features is fitted.

2) With top10 features

These 10 most important features are transactions in one hour, transaction amount, transaction time, transactions in one day, the day of date, card_id, the hour in transaction time and device_id, V_PCA_2 and decimal part of transaction amount.

The smaller model converges in 498th iteration, with 98.8% validation accuracy and 0.140 logloss. AUC is 0.941. Parameters remain the same. This model also gives a satisfying performance in predicting both classes. True positive rate and true negative rate are both over 93%, far above the baseline of 70% given by preliminary model.

Table5 Comparison of two final models

Result	Lgbm (All cleaned features)	Lgbm (Top10 features)
Validation Accuracy	0.993	0.988
Validation Logloss	0.113	0.140
Stopping Round	566	498
True Positive Rate	0.959	0.945
True Negative Rate	0.946	0.937

From the comparison in Table5, these top features almost covers all useful information in identifying fraud transactions. There is only a slight difference which is ignorable. With much less time and memory cost to hit almost same great performance, the smaller model can be called an elite model. Thus this concise model of ten features can be enough to analyze fraud behaviors.

The feature importance order (Figure3) is different from the whole model (Figure2), but all ten figures have high importance score.

.

DISCUSSION

As mentioned in introduction part, for a fraud detection system, the balance between false negatives and positives is important. Thus a metric to access different model can be made through a linear combination of them.We can define a risk function r=a*FP+(1-a)*FN with only one parameter a, while the risk function Altendorf 's[3] has two parameters. In our toy models, the difference is big enough to know which model performs best. And in the two final models, smaller lgbm shows both higher FN and higher FP, so it's also easy to know which one is better. However, sometimes it's hard to choose between highFP+lowFN and lowFP+highFN, and that's the time to use risk function as a metric.

The fraud behaviors can be analyzed through top features (Figure3) in the "elite" lgbm. The most important feature is hour_count (Figure4), which gives how many transactions are completed in the hour of one transaction.

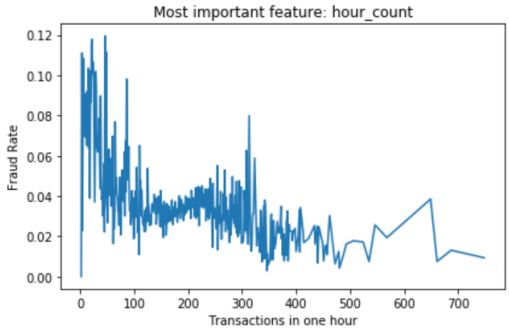

Figure4 Fraud rate vs. transaction in one huor

Fraud rate reaches 10% when number of transactions in one hour is less than 50, in other words, fraud is highly likely to happen when most people don't buy things. This is probably because transaction always

happens from 6am to 9am, and reaches peak at 7am (Figure5), when most people are sleeping or on the way to work. After 10pm, when people start working, it goes back to normal.

There can be many reasons for that. For example, the bank staff didn't start working at that time, and people cannot be reminded when they sleep. Another probability is oversea criminal in different time zone, but information in this dataset cannot validate this assuption.

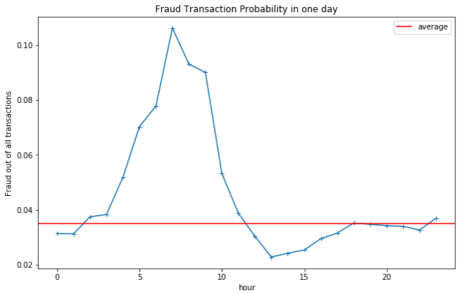

Figure5 Fraud rate in one day

Card_id (Figure6) is also an important feature. Most cards have no fraud record. And for a card has once fraud transaction, it's less possible for the second fraud transaction to happen. But if a card have two fraud transaction records, it's highly likely that more fraud transactions will be done on this card.

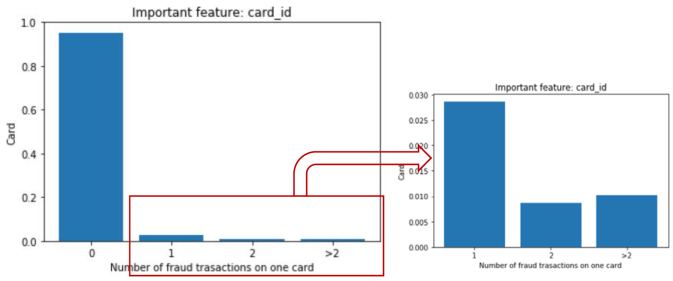

Figure6 Number of Transactions on one card

Transaction amount is an important feature in all toy models and final models. We can see its different shape of distribution in two classes (Figure1). Fraud class has a more flattened normal-like shape, and it seems more frequent in small amount(e^3) or large amount(e^6) while nonfraud class is condensed in the mean area. In view of financial loss, it's always safer to give large amount transaction a frequent check system.

Since date-related features like transactionDT, hour, day, hour_count etc. are important in final models, it worths some work to explore the characteristics of fraud transactions.

Considering special dates for people to buy things, I constructed two features called is_holiday and is_December. 2.3% of fraud transactions happen in holidays and 16.5% in December, while 2.7% of nonfraud transactions happen in holidays and 23.0% in December. As a result, it's reasonable for companies to put more attention to the vast transaction need and less to criminals.

In the scope of one month (Figure7), 29th,31st,1st are most frequent of fraud transactions, far above average. Just guess many cards may get money at the end or first day of a month so that criminals are attracted.

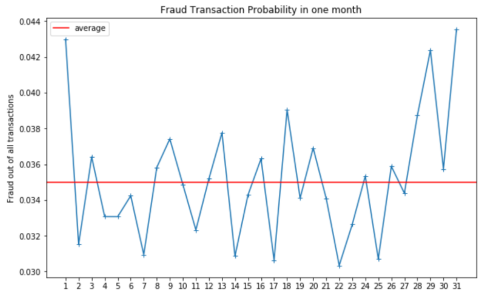

Figure7 Fraud rate in one month

In the scope of one week (Figure8), from Fri. to Sun., number of fraud transactions is increasing and it reaches bottom on Tue.

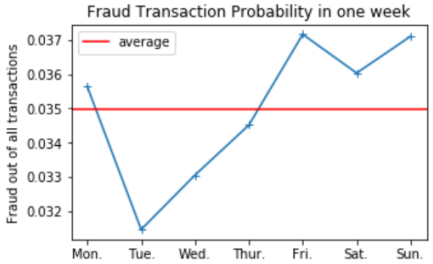

Figure8 Fraud rate in one week

From the date-related features, it seems fraud transactions mostly happen out of normal working time, which is in the early morning and on weekends.

CONCLUSION

Fraud detection is difficult given the low rate out of all transactions. But it's significant because financial loss can be destructive to unlucky victims. In our project, properly tuned Light Gradient Boosting model is robust to study the fraud behaviors and give accurate classification. It's true that payment methods are expanding constantly so the application of reinforcement learning to a real-time data stream could be an extension of this project. It's also true that fraudsters will never retire, so I hope the fraud detection techniques will always beat the cheating algorithms.

REFERENCE

[1] Federal Trade Commission (2020). "Consumer Sentinel Network Data Book 2019."

[2] van Liebergen, Bart, 2017. “Machine learning: A revolution in risk management and compliance?” Journal of Financial Transformation, Capco Institute, vol. 45, pages 60-67.

[3] Altendorf, Eric et al. “Fraud Detection for Online Retail using Random Forests.” (2005).

[4] https://www.kaggle.com/c/ieee-fraud-detection/data