Fraud Detection for Online Transaction

Group 19

GR5291 Advanced Data Analysis Fall 2019

Chen, Pengyuan pc2845

Chen, Xingyu xc2457

Ji, Kaiwen kj2476

Li, Zhiying zl2697

Liu. Siwei sl4224

Su, Feng fs2658

Wang, Tianchen tw2665

Wang, Zhaoyang zw2551

Yang, Zeyu zy2327

Zhang, Cheng cz2532

Zhang, Liwei lz2655

Zhang, Yue yz3383

Zhang, Zhicheng zz2555

Zheng, Kaiyan kz2324

Table of Contents

1 Introduction

Getting credit declined is not something uncommon in our life. It brings safety to our account balance when a real fraud happens to you, but also brings inconvenience when you're trying to pay your bills. In this project, we will use the transaction and identity data from Vesta to explore an accurate way to make fraud detection with machine learning methods.

The project has mainly five parts: exploratory data analysis, data processing, feature selection, model building and evaluation.

For data processing, we firstly dropped features with over 85% missing points and then used multiple imputation as well as mode to fill in missing data. We handled imbalanced data with SMOTE.

After that, we chose the method "Time Consistency Test + Correlation Test + Lasso" to select features and shrunk feature number down to 272 from 360 features. Notice that, for all the parts above, Logistics Regression was used as our baseline model to test performances of data processing and feature selection methods. Also, we chose AUC, precision and recall as our evaluation metrics for the whole project.

Eventually, we conducted 5 classification models, including Logistics Regression, SVM, Random Forest, XGBoost and LightGBM. LightGBM gave us the best result by having AUC 0.96, recall 0.75 and precision 0.49.

All our works have been uploaded to GitHub at *https://github.com/fs2658/ADA-Project-Fraud-Detection*.

2 Project Objective

The objective of our project is to detect potential fraud in each transaction of clients. We make use of transaction information and personal identity information. The improvement of fraud detection accuracy can

- protect customers from being the false positive of fraud detection;
- prevent the company from fraud loss.

3 Data Source and Description

The datasets are provided by Vesta Corporation and obtained from Kaggle (IEEE, 2019). There are two parts, which are *Identity* and *Transaction*. The target variable *isFraud* is in the Transaction dataset. We checked the basic situations of the datasets.

First, most features are anonymous, which means their real column names are replaced by some meaningless string. Secondly, not all transaction observations have corresponding identity information. Thirdly, both original datasets have no duplication observations. Finally, the primary key, *TransactionID*, is unique.

Since the target variable *isFraud* is in Transaction data, and we want to use as many observations as possible, we left joined the 'Transaction' table with the 'Identity' table by 'TransactionID' as our dataset.

In the joined dataset, there are 600,000 observations and 433 features. 393 features are from *Transaction* and 41 features are from *Identity*. In *Transaction*, categorical features include *ProductCD, card1 - card6, addr1 - addr2, P_emaildomain, R_emaidomain*, and other anonymous features. In *Identity*, categorical features include *DeviceType, DeviceInfo*, and other anonymous features.

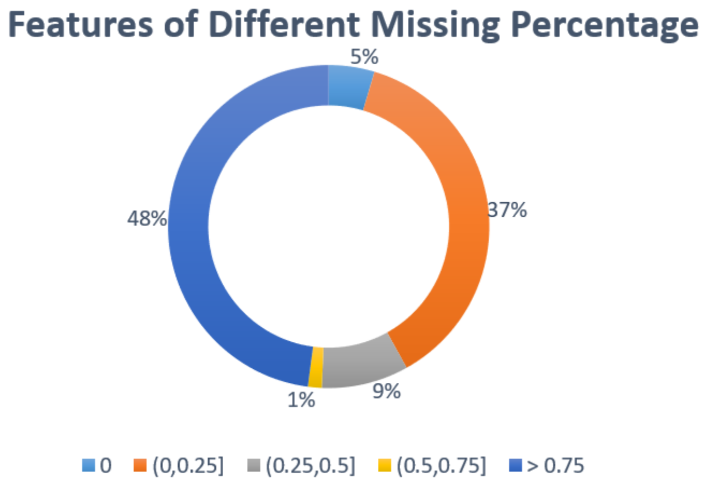
4 EDA

Before going any deeper into the data, we conducted EDA first to check the data quality. There are two major problems in our dataset: large number of missing for many features and extremely imbalance target variables.

4.1 Missing data

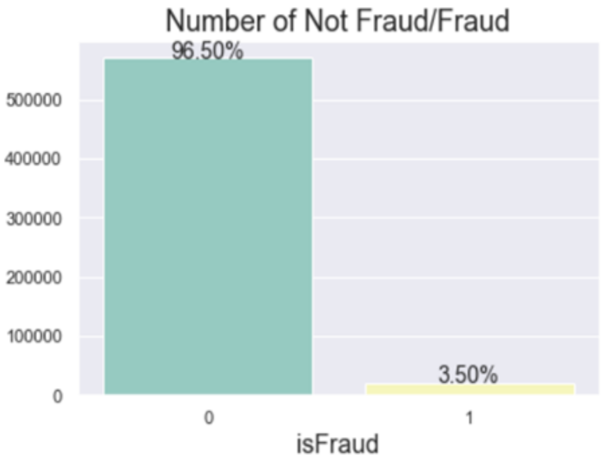
Firstly, most features have missing values and there is a large number of features with a high percentage of missing. Among 433 features, we find that only 5% of the features are without missing values and there are 48% features with a missing value higher than 0.75. What's more, there are even 12 (2.8%) features with missing values higher than 0.9. We need to be very careful with these missing situations when doing the feature selection.

The situation is as follows. The percentage in the plot is the percentage of a certain type of feature. All the features are grouped based on the missing proportion shown in the legend.



4.2 Imbalanced data

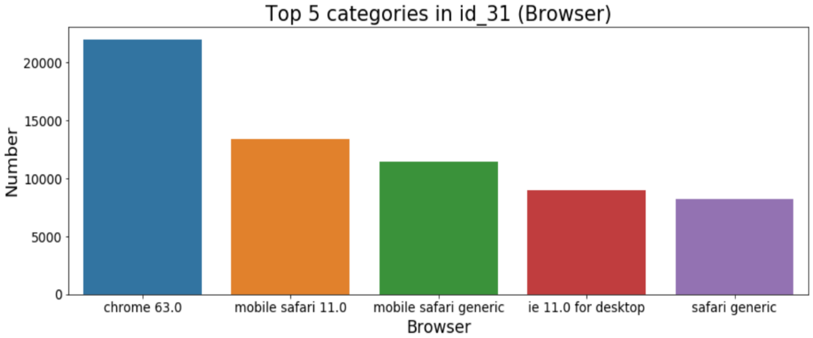
Secondly, the target variables are severely imbalanced. More than 96% of the data are not fraud which makes a lot of sense since most transactions are not fraud. But imbalanced data may cause sever model overfitting.



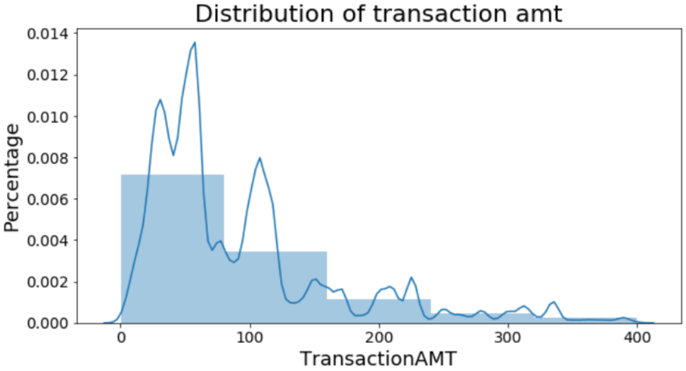
4.3 Other findings

Finally, even though most features are anonymous, we tried to explore the meaning of some anonymous variables and find some interesting facts. Besides, we also explore some patterns from meaningful features. We'll briefly show some of them below to take a glimpse of the data itself.

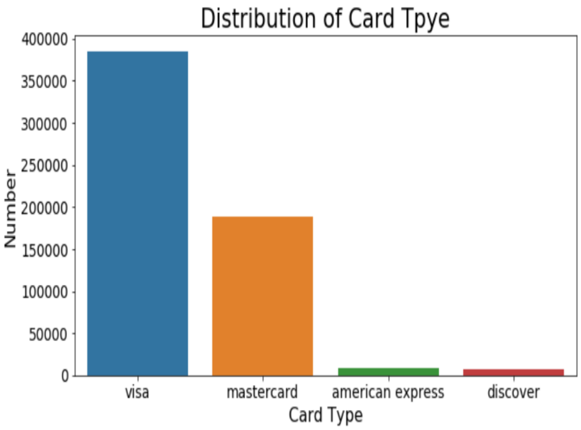
(1) *id_31* should be the type of browser. The missing rate is 76.25%. Except for the missing value, the most popular browser is chrome 63.0.

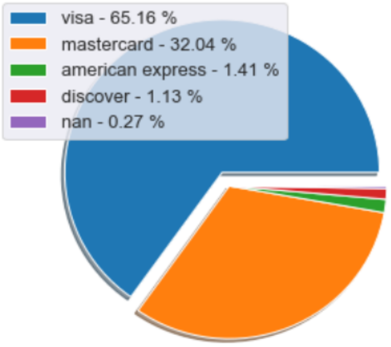


(2) The *transaction amt* shows the number of money in every transaction and there are no missing values. The number of amount smaller than 400 is 556,759 (94.28%), and the distribution of this group is as follows. Most transactions just have a small amount.

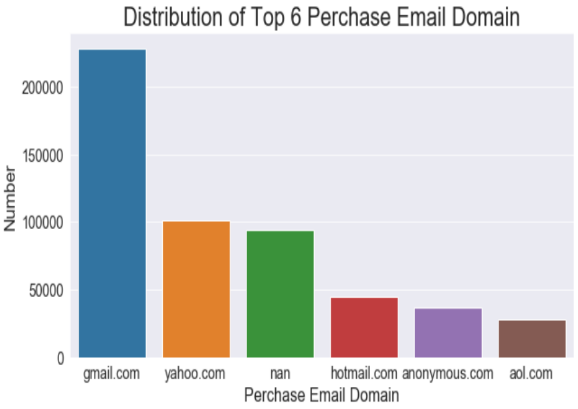


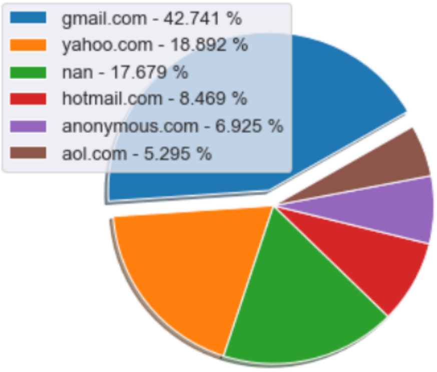
(3) The *Card4* shows the type of card. The missing rate is 0.27%. There are just four types of cards.





(4) The *P_emaildomain* shows the purchase email domain. The top 6 (including null) counts for more than 90% of total domains and the distribution is as follows





5 Evaluation Criterion

For the whole project, we will focus on 3 metrics to evaluate our results, which are precision, recall and auc score. A brief review for those will be given below.

- Precision(Higher precision: less chance of “fake” alert)

$$Precision = \frac{True\ Positive}{True\ Positive\ +\ False\ Positive}$$

In our case, True Positive means a fraud transaction is correctly classified as *isFraud*. False Positive means a non-fraud transaction is incorrectly classified as *isFraud*. In general, precision implies, among all cases classified as fraud, the rate of those cases correctly classified.

- Recall(Higher Recall: more ability to identify fraud)

$$Recall = \frac{True\ Positive}{True\ Positive\ +\ False\ Negative}$$

Here, False Negative means a fraud transaction is wrongly classified as not *isFraud*. Thus, recall implies, among all fraud cases, the rate of successfully detected ones.

Notice that there is a tradeoff between precision and recall given by the definition. For fraud detection, the company would rather wrongly catch non-fraud cases than not catch fraud cases.

- AUC

AUC(Yufeng Wang, 2019), is when we randomly pick a positive case and a negative case, the probability that the positive case outranks the negative one according to the classifier is given by the AUC. Mathematically, it equals the area under the ROC curve. Notice that AUC is independent of the threshold set for classification because it only considers the rank of each prediction and not its absolute value.

In our case, AUC would be taken as the most important metric and helps us decide which model is the most optimal one.

6 Data Processing

6.1 Missing data

Step 1: For features with over 85% missing data (>500,000 missing data), we simply dropped them.

Step 2: We mainly used two methods to handle those remaining missing values and selected the one which generate better results.

(1) Mean and Mode

For this method, we filled numerical(continuous) missing data with the mean of other non-null values of that specific columns(features). We also filled categorical missing data with the mode of other non-null values.

(2) Multiple imputation

Multiple imputation is essentially an iterative form of stochastic imputation. Instead of filling in a single value, the distribution of the observed data is used to estimate multiple values that reflect the uncertainty around the true value. In our case, we ran linear regression for 5 times and used the mean to impute missing values.

To test which method is better, we used Logistic Model as testing model and Fisher Criteria, Stepforward Selection as our feature selection procedure (the choice of testing model and testing feature selection procedure can vary, as long as they are the same in both cases. In addition, Logistic Regression, Fisher Criteria and SFS will be discussed later in this report). The result we obtained is in the table.

	Precision	Recall	AUC
Mean & Mode	0.372	0.17770	0.6524
Multiple Imputation	0.373	0.17777	**0.6525**

Comments: Logistic Regression is used as the testing model.

From the table, although the results are similar to each other, multiple Imputation did give us a generally better result with higher precision, recall and AUC. So, we can believe that if the size of dataset is getting larger, multiple imputation will return a better result. In summary, we chose Multiple Imputation to handle missing data.

6.2 Imbalanced data

For the imbalanced data, we tried to use down sampling method to resolve this issue. However, we found this procedure largely decreases the size of our dataset.

In this case, we used an oversampling method called SMOTE to solve this problem. It is Synthetic Minority Over-sampling Technique. This is an up-sampling approach in which the minority class is up-sampled by creating synthetic examples rather than by over-sampling with replacement.

The advantage of using SMOTE compared to traditional oversampling method is that, SMOTE can generate more information which helps our model to learn better. While on the other side, the disadvantage is also obvious -- that information is not real data. Hence, the choice of method is dependent on the company's requirements and goals.

Also, note that we cannot apply any remedial method for imbalance data to the whole dataset. We have to make sure that the testing dataset stay as imbalanced, by which it means that we can only make our training set become balanced. Otherwise, the testing dataset will differ from the reality and become meaningless.

6.3 Feature Selection

In this part, we tried 9 types of methods to select features, including Time Consistency, Correlation Test, Fisher Criterion, Stepforward Selection, PCA, Lasso. Before we dig into the details of those features selection methods we actually used. We will first briefly go through the methodology behind them.

Note: based on the above section, from now on we use multiple imputation to compute missing value and SMOTE for imbalance data issue. The testing model is still Logistic Regression.

6.3.1 Explanation of different tests

(1) Correlation Test

Correlation Test (Pearson) is used to test if two input variables (x_i) are highly correlated with each other. Here, if the correlation is more than 90%, we define those two input variables as highly correlated and delete the one with more missing value. Interestingly, the choice of 90% is tricky. We don’t want those variables that provide similar information. However, at the same time, we also do not want to delete any variable at least containing some useful information.

(2) Time Consistency Test

Time consistency is to train a single model using a single feature (or small group of features) on the first 150,000 of train dataset and predict *isFraud* for the last 150,000 of train dataset. This evaluates whether a feature by itself is consistent over time. For example, if they have training AUC around 0.60 and validation AUC 0.40, we discard them (how we define the threshold of a big difference is up to the company itself and reality).In other words, the feature finds patterns in the present that does not exist in the future.

Of course the possible of interactions complicates things but we double checked every test with other tests. In this project, we selected 0.2 as our threshold and we found all of the features fit time consistency, by which it means that we don't need to discard any of them.

(3) Fisher Criteria

Fisher's linear discriminant is a method that maps high-dimensional data onto a line and performs classification in this one-dimensional space. The basic theory of Fisher criterion can be applied to feature selection and feature extraction. According to Xiang Fang (2017) , the Fisher criterion is as follows.

$$r = \frac{\sigma_{between}}{\sigma_{in}}$$

Where r represents a Fisher ratio of feature, $\sigma_{between}$ represents a variance between classes and σ_{in} represents a variance within classes. The higher r is, the better discrimination between classes is.

The advantages of using Fisher Criteria: efficient, low computational cost, easy to understand.

The disadvantages of using Fisher Criteria: it cannot consider the combination effect of different variables. For example, x_1 may have higher Fisher score with the output variable *IsFraud*, however, the combination of x_2 and x_3 may outperform x_1, even through the Fisher score of x_2 and x_3 separately is lower than x_1's.

(4) Stepwise forward Selection (SFS)

Forward selection is a kind of feature selection methods based on stepwise regression which begins with an empty model and adds in variables one by one. The model will add the variables until it gives you the best improvement for your model. We believe that SFS is a very robust feature selection procedure(Greedy algorithm) with really high computational cost.

(5) PCA

Principal component analysis (PCA) is a statistical procedure that simplifies the complexity in high dimensional data set into a set of low dimensional uncorrelated variables called Principal Components. PCA is relatively more efficient, which helps us “explain” that information by only several of components. However, it is also not easy to explain those principal components, especially to those clients without Mathematics/Statistics background.

(6) Lasso Logistic Regression

The Lasso (L1 penalty) is a shrinkage and selection method for linear regression. It performs the variable selection while training and regularization to minimize the sum of squared error and improve the model performance and interpretability of the statistical models it produces. It is an embedded method which can reach a balance point between the model performance and efficiency.

6.3.2 Methodology

Next, we would like to give out more details about how we combined these methods to perform feature selection and why we made that specific choice.

The first feature selection procedure we selected was to perform time consistency and correlation test at the beginning, which helped us eliminate those inconsistent and “duplicate”(feature provide same information) features.

Then we performed Fisher Criteria to select 20 most important features and applied SFS to further select the 6 most robust features among them. The reason we used Fisher Criteria right after time consistency test and correlation test is mainly based on the fact that, although Fisher Criteria cannot consider the combination effect between variables, it is a very efficient way to help us to firstly shrink down the size of feature set before the application of SFS and it is also a very nice starting point for a baseline feature selection procedure, which can provide us with more insights of this feature set.

In addition, the reason we manually shrunk the feature set to a relatively small one (20 after Fisher Criteria, 6 in the end) is based on the fact that, in the financial industry, overfitting can be easily triggered. So in this case, we want to start from a relatively small feature set and make adjustments according to the result. And from the corresponding result, we can see that AUC = 0.656, Recall = 0.378, Precision = 0.178. It indicates that we are far from overfitting. Therefore, according to this result, we decided to try all of the features and some specific algorithms (statistical models) that can help us automatically select features.

Based on this idea, we tried Lasso, PCA and the full feature set after time consistency test and correlation test. From the result we can see that applying Lasso and PCA returned similar and better results compared to using the full feature set.

However, interestingly, we found out that lasso only shrunk one feature down to zero. Does this indicate that the correlation test might eliminate some of useful information(features)? In order to find out the "truth", we decided to use PCA and LASSO from the first place without correlation test and time consistency. At this time, they returned only slightly better results compared to previous one and Lasso only shrunk two features down to zero. However, the time used to run this procedure has been largely increased. Thus, in this case, we believe that the correlation test did help us get rid of useless features and decrease the running time significantly.

Details of results are demonstrated as follows:

	Precision	Recall	AUC
Time Consistency + Correlation Test	0.098	0.687	0.730
Time Consistency + Correlation Test + Fisher Criterion (Top 20)+ SFS(Top 6)	0.178	0.378	0.656
Correlation Test + PCA	0.110	0.707	0.747
Time Consistency + Correlation Test + Lasso	**0.14**	**0.71**	**0.78**
PCA	0.10	0.697	0.742
Lasso	**0.15**	**0.72**	**0.78**
Fisher Criterion (Top 20)+ SFS(Top 6)	0.373	0.178	0.653

Comments: 1. Logistic Regression is used as the testing model. 2. The original feature number is 433.

6.3.3 Summary

From the table, the two methods “Time Consistency + Correlation Test + Lasso” as well as “Lasso” can give us the best recall and auc. However, using Lasso directly will take around 10 hours while the other one will take only a few hours to accomplish and the results are similar.

Therefore, we selected Time Consistency(original number of features: 360, remaining features: 360) + Correlation Test(original number of features: 360, remaining features: 273) + Lasso(original number of features: 273, remaining features: 272) as our final feature selection procedure. But the choice is really depend on company’s favour (Higher Recall: more ability to identify fraud, Higher precision: less chance of “fake” alert, or a balance point between them) and reality.

7 Model Selection

In this stage, we tired 5 models below to select the one brings the best performance.

(Reminder: procedure before model selection is: Drop features with over 85% missing points + Multiple Imputation/ Mode (for missing value) ⟹ Time Consistency test + Correlation Test + Lasso (for feature selection) ⟹ SMOTE (for imbalanced training data))

- Logistic Regression (Testing/Baseline model)

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable. The function will give a probability. In our case, we used the default threshold 0.5. If the probability is greater than 0.5, it will be classified as fraud and vice versa. In addition, we used this model as our testing and baseline model.

- Support Vector Machine (SVM)

Support Vector Machine constructs a hyperplane or set of hyperplanes in a high-dimensional or infinite-dimensional space, which can be used for classification and regression. In our case, we used it as a classifier.

- Random Forest

Random forest is made from decision trees. Each decision tree is a weak classifier and can easily generates overfitting problems. To solve this, random forest applies bagging to those trees and produce a more stable result by averaging the major votes or regression results. Notice that, compared to boosting models, random forest is faster because of its paralleled calculation.

- XGBoost

XGBoost is one of the most classic boosting methods. Differently from bagging, it trains one decision trees over and over again by looking at its result and retrain the tree by lifting the weights of the observation incorrectly classified. For calculating thresholds, it splits the data by pre-sorting, which means enumerate all features and instances. It treats categorical variables with one-hot-encoding.

- Light GBM

LGBM is a faster boosting method than XGBoost due to its different structure. It uses gradient-based one-sided sampling to filter out the data instances for finding a split value without enumeration. Besides, unlike XGBoost, LGBM can use categorical variables directly without any variable conversion. Compare to XGBoost, LGBM usually brings a similar and good result with a faster speed. Here in our case, it actually led to the best auc scores and outperformanced XGBoost a lot.

Notice all tree models mentioned above (a.k.a. Random forest, XGBoost and Light GBM) have the nature the handle missing data and imbalanced data. They often lead to a good result when applied to an actual classification or regression problem in reality.

Here is our results:

	Precision	Recall	AUC
Logistic Regression	0.0947	0.7038	0.7230
SVM	0.2427	0.6252	0.7772
Random Forest	0.8997	0.6196	0.8684
XGBoost	0.8159	0.6075	0.8308
Light GBM	0.4905	0.7521	0.9603

8 Future Developments

For further developments, we can try more advanced methods and different combinations to resolve the problems imposed by missing value, imbalanced data and feature selection. Especially for feature selection, due to the time limitation and functionality of our computers, we haven't tried any greedy algorithm (most robust) with large subset of features(eg: use SFS at the first place or use SFS to select best 100 features).

In addition, some of the feature selection procedures may have different performances over different models. For example, in our project, those features selected by LASSO might only be the best choice for regression model. In the financial risk industry, it is normal to try out plenty of models and we will also constantly push our model to its best.

9 Bibliography

[1] IEEE Computational Intelligence Society (2019). IEEE-CIS Fraud Detection. Retrieved from https://www.kaggle.com/c/ieee-fraud-detection/overview

[2] Xiang Fang1, L. W. (2017). Feature Selection Based on Fisher Criterion and Sequential Forward Selection for Intrusion Detection.

[3] Yufeng Wang (2019). Rethinking the Right Metrics for Fraud Detection. Retrieved from https://medium.com/datadriveninvestor/rethinking-the-right-metrics-for-fraud-detection-4edfb629c423