NEW YORK UNIVERISTY CENTER FOR DATA SCIENCE

Credit Card Fraud Detection

Xiao Tan, Xinyang Qiu , Nimi Wang

{xt437, xq303, nw212}@nyu.edu

February 21, 2019

Contents

ABSTRACT

This paper focuses on detecting the fraudulent credit card transactions, and dealing with imbalanced data. We proposed several supervised learning methods to predict the fraudulent credit card transactions. Specifically, Logistic Regression, Random Forest, XGboost are trained on the data. The performance of the model is then measured by F1-score. The result indicates that XGboost performs the best with under-sampled data among all the models, achieving a 82.5% F1-score.

1 INTRODUCTION AND MOTIVATION

Credit card fraud is the most common form of identity theft. It means people illegally obtain goods or services with others' credit card and personal information, for example, name, address, birthday and social security number(SSN), etc. These behaviors will cause great turmoil in people's life and bring trouble to companies issuing the credit cards. Therefore, it is important for credit card issuing companies to recognize fraudulent credit transactions to secure the customers' assets.

In this report, we use the data set of transaction records by European cardholders. The biggest difficulty we face is the data set being highly imbalanced, referring to huge difference in the number of instances available for two classes. To solve this problem, we then implement different re-sample methods and propose several supervised classification models to identify fraudulent credit card transactions.

The following part of the report is arranged as following: in Section 2, we will cover all methods that we apply in our task. In Section 3, we will explain our experiment results. The detailed comparison regarding each methods and conclusion will be in Section 4 and 5.

2 DATA DESCRIPTION AND METHODOLOGY

2.1 DATA-SET

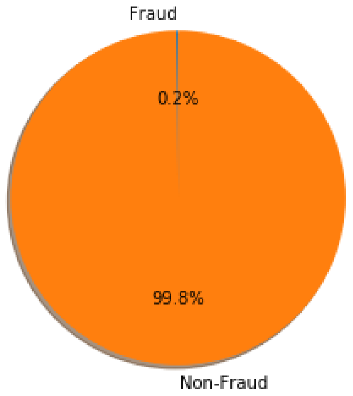


Figure 1: Pie chart describing the type of credit card transactions

Our data is provided by Machine Learning Group of ULB[1]. As mentioned previously, it covers more than 284,807 credit transaction by European cardholders in September, 2013. All features are numerical, including transaction time, transaction amount, and 28 attributes generating from principal component

[1] http://mlg.ulb.ac.be

analysis(PCA) transformation. The target variable, type of credit card transaction, is categorical data, with 1 meaning fraud and 0 meaning non-fraud. As shown in Figure 1, fraudulent transaction is the minority class, accounting for about 0.17% of the whole transactions. Due to the extreme low percentage of observations with duplicated records and unknown values, we don't replace them with any other values and directly drop them.

2.2 Evaluation metrics

One big problem that we should solve is the choice of evaluation metrics. For most classification problems, accuracy, focusing on the true predictions,

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + False\ Positive + True\ Negative + False\ Negative}$$[2]

is a commonly used criterion. To maximize the overall accuracy, most of the machine learning algorithms tend to predict that most instances belong to the majority class. However, the recall, a measure of how many truly relevant results being predicted,

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

could be very low. To prove this situation, we run logistic regression on our original data set and use the accuracy as criterion. We get 99.89% accuracy, however, 42% recall.

According to the reason stated above, we should evaluate the learning results appropriately. Nguyen, Bouzerdoun, Phung(2010)[1] introduced three metrics called *"Precision, recall, and F-measure"* to use in the situations when performance for the minority class is preferred. With recall defined previously, precision is a measure of result relevancy. From the definition and the formula above, it is clear that there is a trade-off between precision and recall. F-measure is the introduced as the harmonic mean of precision and recall. Therefore, in the following models, we will use F-measure as the evaluation metrics.

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

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

2.3 Re-sampling data-set

In addition to choose a suitable evaluation metrics, we also solve the imbalanced problem by re-sampling data sets. Three methods, demonstrated in Figure 2, are applied to the original data sets.

2.3.1 Under-sampling

Under-sampling, as shown in first part of Figure 2, refers to methods of concentrate a subset of majority class to into the whole minority class as new data set. We use under-sampling to reduce the number of non-fraudulent instances in the training set. According to Liu (2014)[2], random under-sampling outperforms most of the other under-sampling techniques. Hence, we will use random under-sampling in this report.

[2] True: *prediction = Truth* ; False: *prediction ≠ Truth*; Positive: *prediction being positive*; Negative: *prediction being negative*

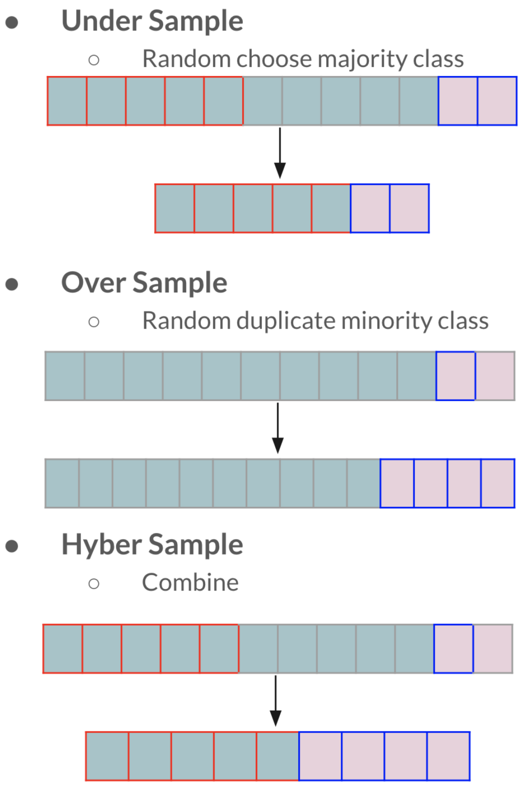


Figure 2: Figure explanation of re-sampling methods

We tried out two under-sampling rates: 0.0015, 0.002,(we cut the majority data instances to 0.0015 and 0.002 of the original size respectively),which makes the fraud and non-fraud ratio 52.7% and 45.5%. One drawback of under-sampling methods is that the number of data instances decreases significantly, which may eliminate some useful information.

2.3.2 Over-sampling

Shown in the second part of Figure 2, over-sampling is a sampling technique that increases the minority class without changing the majority class. However, as Wang (2008)[3] pointed out, over-sampling may cause over-fitting problem since there exist replicated data instances after re-sampling. Another major drawback is that it increases the training time.

Our initial choices for over-sampling rates are 500,1000,and 1500 (we duplicated the minor data instances to 500, 1000, and 1500 times of the original size). However, due to the large number of duplicated data, our model tends to over-fit and is only capable of predicting 0, which makes the f-scores ill-defined. Hence, we switched the over-sampling rates to 2 and 50.

2.3.3 Hybrid-sampling

b Hybrid-sampling, the last part in Figure 2 is a combination of random over-sampling and random under-sampling.We used the package "SMOTETomek" to achieve this. We set the fraud and non-fraud ratio as 0.45,0.6,and 0.65 respectively.

3 EXPERIMENT AND RESULTS

3.1 TRAINING AND PREDICTION

With more than 280,000 data instances included in the project, 80% of the them are randomly chosen as training set and the rest of the 20% as test set. We used 3-fold cross-validation when fitting the training set. Only training set is re-sampled with the methods mentioned previously. To test the performance of models in real life, we don't re-sample in validation and test set

We tried three classifiers: Logistic Regression, Random Forest,and XGBoost. For each classifier, we used cross validation with parameter tuning using grid search to find the best model architecture. All the parameters we tried out can be found in Table 1 in Appendix.

3.2 RESULT

	Resampled Rate	Fraudulent Rate	Logistic Regression	Random Forest	XGBoost
Original Dataset	N/A	0.1727%	0.7144	0.8298	0.8362
Under Sample	0.0015	52.6796%	0.9457	0.9434	**0.9546**
	0.002	45.5024%	0.9404	0.9453	0.94667
Over Sample	2	0.3328%	0.9017	N/A	0.9232
	50	7.7060%	N/A	N/A	N/A
Hybrid Sample	0.45	25.3356%	0.9233	0.9222	0.9385
	0.6	19.0024%	0.9201	0.9302	0.9289
	0.65	17.7204%	0.9100	0.9222	0.9300

Table 1: Results for different models

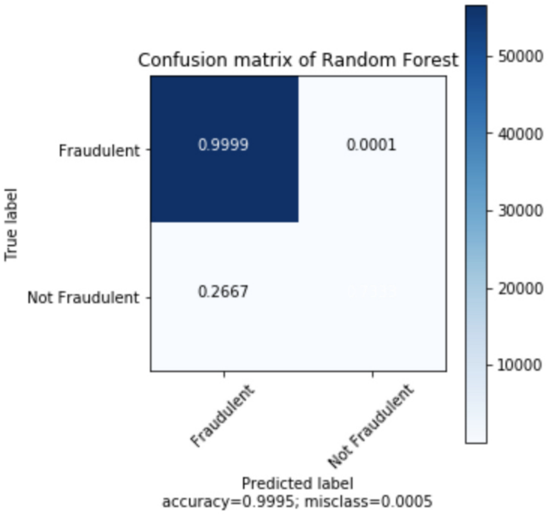
Observing the results suggested in Table 1, we came to three conclusions:

1. Classifiers do not give good performance when using original data due to the issue of imbalanced data.

2. When using oversampling, F-score is sometimes ill-defined, especially when the resampled rate is high. The reason is that when we duplicated data too many times, the model tends to overfit,and is not capable of predicting the minority case.

3. We got the best performance when using XGBoost Classifier on the under-sampled data when fraud rate is 52.6796%, with a F-score of 0.9546.

For next steps, we used our best estimator with the under sampling method for the testing data and got a F-1 score of 0.825 which is decent since the testing data is supposed to resemble the actual data se we are going to deploy our model on. Further, we want to look at the confusion matrix to find out the true positives and false negative we got with the best estimator. As we can see, we got a 99.99% true positive rate, which means we are able to detect the fraudulent transactions 99.99% of the time. However, we got a relatively higher false positive rate, which means we might sometimes predict non-fraudulent transactions as fraudulent. However, for our project, it is more important to get a high true positive rate than a low false positive rate since it is more expensive to afford false negative predictions.



matrix.png

Figure 3: Confusion Matrix for XGBoost Classifier prediction on Testing Data

4 DISCUSSION

Our general methodology is to use under sampling method to randomly select data from the majority case, and use all samples from the minority case, and apply XGBoost classifier as our machine learning model. Because we used under sampling method, this poses some advantages and disadvantages. Under sampling can help improve run time and storage problems by reducing the number of training data samples when training data is huge,which is our case. However, at the same time, it can discard potentially useful information from the majority samples which could be important for building rule classifiers. This might lead to bias to our model since it will not be an accurate representation of the population. In addition, because our data will be constantly updating since new transaction records come in on a daily basis, our whole data set will be constantly growing. This poses computational stress to modeling if we want to update our training data from time to time especially when we use tree models for modeling.For the future work, we can train the model on more data in order to solve the problem of losing information due to down-sampling.

5 CONCLUSION

In this report, we proposed several supervised machine learning methods to detect the fraudulent transactions. We also tried out three re-sampling methods in order to deal with imbalanced data. The results indicate that XGBoosting performs the best with down-sampled data, achieving a 2.5% F1-score. We also found out that over-sampling data can be problematic due to the duplicated data. However, we expect investigation on more data could be further tested, since our model may lack generality due to the limited amount of data we used (especially after we use down-sampling). More details about our project can be found via `https://github.com/nimiw0821/1007-Final-Project.git`.

REFERENCES

[1] Son Lam Phung, Abdesselam Bouzerdoum,Giang Hoang Nguyen (2010)
Learning pattern classification tasks with imbalanced data sets Retrieved 26 July, 2012 from http://cdn.intechweb.org/pdfs/9154.pdf

[2] Liu, Y.C. (2014)
The Effect of Oversampling and Undersampling on Classifying Imbalanced Text Datasets. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.101.5878 rep=rep1type=pdf

[3] Wang, S. (2008).
Class Imbalance Learning. Retrieved 28 July, 2012 from http://www.cs.bham.ac.uk/syw/documents/progressreports/Thesis

Appendices

Logistic Regression	
Parameters	Values
C	10**i for i in range(-5,5)
penalty	l1, l2
Random Forest	
criterion	entropy
bootstrap	TRUE
n_estimators	300, 500, 1000
max_features	auto, sqrt
max_depth	10, 20, 30
min_samples_split	2, 10, 100
min_samples_leaf	1, 5, 10
XGBoosting	
eta	0.01, 0.1
max_depth	range(3,10,2)
min_child_weight	range(1,6,2)
gamma	0, 1
subsample	i/10.0 for i in range(6,10)
colsample_bytree	i/10.0 for i in range(6,10)
n_estimators	300, 500, 1000

Table 2: List of all the parameters we tried out