Fraud detection using various machine learning models

Xin Gu

University of Pennsylvania
guxin@seas.upenn.edu

May 11, 2020

Abstract

This report discusses credit card fraud detection by means of machine learning. Credit card fraud detection has become more and more important to both institutions and individuals in an era of digitalization. After perform sampling on unbalanced data, various machine learning techniques are built and hyper-parameters are tuned, including supervised learning, unsupervised learning and deep learning. Logistic regression trained with oversampled dataset shows highest recall. Finally, future directions are indicated to improve the performance.

I. Introduction

Credit card has become more and more important in our daily life over decades. This evolution of payment is an example of the digitalization of our society, yet it has also caused the problem of credit card fraud. Credit card fraud is increasing dramatically with the expansion of modern technology, resulting in the loss of billions of dollars worldwide each year. Methodologies for the detection of fraud are essential if we are to catch fraudsters once fraud prevention has failed. Statistics and machine learning provide effective technologies for fraud detection and have been applied successfully to detect activities such as money laundering, e-commerce credit card fraud, telecommunications fraud and computer intrusion, to name but a few.

In this project, I performed feature engineering, built various machine learning models and compare models under certain criteria. The structure of this article is as follows: first I introduce the credit card fraud. Then I briefly introduced the machine learning techniques used and the experiment procedure. Finally, I analyze the results, followed by a conclusion and possible future directions of research.

II. Credit Card Fraud Detection

i. What is Credit Card Fraud Detection

The advent of credit card has not only given people more comfort but have also caused many crimes. Credit card is a good target of fraud, since a lot of money can be earned in a very short time and the crime is only discovered a few weeks later. At that time, it is hard to catch the criminals. Two common credit card fraud techniques are: 1. Copying a card in some way and getting to know the pin-code. 2.Vendors charging more money than agreed to the customer.

When credit card fraud happens, both banks and card holders will partially pay for losing money, so it is both the banks' and card holders' interest to reduce the fraud and that is why financial institutions started to do research on fraud detection. Fraud Detection is, given a set of transactions, the process of identifying fraudulent transaction. In another word, a classification process.

ii. Challenges with Credit Card Fraud Detection

The biggest problem with credit card fraud detection is data. There are two major challenges with the data problem. One is the lack of real-world transaction data, which is kept confidential for the reason of custom privacy. So, there is no authentic data to perform experiments on. Another challenge is that even some processed transaction data is released for study, only a small number of transactions are fraudulent. The imbalance of two classes increases the difficulty of identifying frauds.

III. Data

i. Data source

The dataset is downloaded from Kaggle. It contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, which includes 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172 percent of all transactions. It contains only numerical input variables which are the result of a PCA transformation. Due to confidentiality issues, original features and more background information cannot be provided. Features V1, V2, . . . V28 are the principal components obtained with principle component analysis (PCA), the only features which have not been transformed with PCA are 'Time' and 'Amount'.

ii. Data precessing

First, I visualized the relationship between time and label, and found out 'Time' had no obvious impact on the label, so I eliminated this feature. Next, I performed standardization on 'Amount'. Finally, I computed the correlation of all features and verified that no features have high correlation, which is no surprise since they are outcomes of PCA. So I kept all features expect for 'Time' to next steps.

iii. Sampling

Highly unbalanced dataset needs sampling. There are two common sampling method, under-sampling and over-sampling. Undersampling is the selection of a subset of samples from the majority class. Oversampling is creating new samples in the minority class. In this project, for under-sampling, I randomly selected a subset of samples from the majority class to make the two classes have samples of same number. For over-sampling, I performed Synthetic Minority Oversampling TEchnique (SMOTE) on the minority class so that the two classes have samples of same number. SMOTE works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line.

In the following steps, I trained models with original unbalanced dataset, undersampled dataset and over-sampled dataset.

IV. Models

i. Model selection

In this project, I applied several machine learning models based on some articles, including supervised learning, unsupervised learning and deep learning.
Supervised learning:

- Logistic Regression
- Linear Discriminant Analysis (LAD)
- k-NearestNeighbor (KNN)
- Decision Tree
- Random Forest
- XGBoost

  Random forests and XGBoosting are both a set of decision trees. The difference between them is that random forests builds each tree independently while gradient boosting builds one tree at a time. And random forests combine results at the end of the process (by averaging or "majority rules") while gradient boosting combines results along the way.

Unsupervised learning:

- Isolation Forest

  Isolation forest is an unsupervised learning algorithm for anomaly detection that works on the principle of isolating anomalies, instead of the most common techniques of profiling normal points.The main advantage of this approach is the possibility of exploiting sampling techniques to an extent that is not allowed to the profile-based methods, creating a very fast algorithm with a low memory demand.

Deep learning:

- Fully connected neural network

ii. Hyperparameters tuning

In this project, I applied ‘GridSearchCV’ to tune hyper-parameters of each model. It automatically trains the models with different combinations of hyper-parameters I set and gives the best combination. Table 1 shows the hyper-parameters of each model.

V. Methodology

i. Criteria

There are two most important criteria to value the models in this project, Recall and ROC-AUC. Recall means the percentage of the truly detected fraudulent transaction out of all fraudulent transactions, which is very important for fraud detection because we try to prevent as few frauds and as less loss as possible. Secondly, ROC-AUC can be applied for cases with unbalanced samples. It is about sensitivity and specificity, and it won’t be affected by the change of class distribution. Since I will train and test models using samples of different class distributions, so ROC-AUC it is appropriate for this case. And Precision will also be considered as a complementary criterion.

ii. Procedure

After performing feature engineering and visualization, I split training and test data on original data, because test data should be similar to the real-world data distribution so that models can show trustable results. Then I performed under-sampling and oversampling on training data to get three training dataset. Then I

Table 1: *Tuned Hyper-parameters of Models*

Model	Tuned hyper-parameters
Logistic Regression	’penalty’ , ’C’, ’solver’, ’multi_class’
LAD	’solver’, ’shrinkage’, ’priors ’
KNN	’n_neighbors’, ’weights’
Decision Tree	’criterion’, ’max_ depth’
Random Forest	’criterion’, ’max_ depth’
XGBoost	’criterion’, ’max_ depth’, ’learning_rate’
Isolation Forest	’contamination’, number of features used for training
FNN	number of layers, number of neurons, activation function

p.s. Different training datasets correspond to different best hyper-parameters. For specific values of hyper-parameters, please refer to my codes.

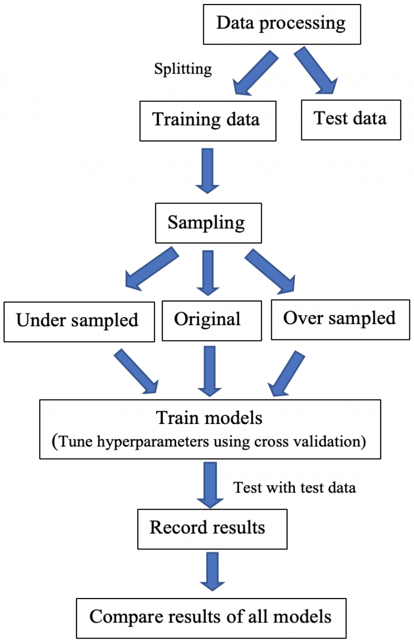

Figure 1: *Procedure*

trained models using these three training data, tuned hyper-parameters using cross validation. Finally, I tested the models using test data to get Recall and AUC-ROC of each model to compare their performance and make analysis. Figure 1 shows the project procedure.

VI. Results and Analysis

The results and comments of each model are listed on from Table 2 to Table 9. There are some conclusions can be drawn from the results:

Supervised learning:

- For all supervised learning models, including deep learning model, sampled training data always generate better recall than unbalanced data. Logistic regression has the greatest improvement of recall from unbalanced data to sampled data. For logistic regression, oversampled data has the higher recall than under-sampled data. For the rest supervised learning models, under-sampled data has the higher recall than oversampled data.

- Higher recall causes lower precision for all models, so sampled data always causes lower precision than original data. Except for Linear Discriminant Analysis, all other supervised learning model with under-sampled data have lowest precision.

- For AUC_ROC, although I list AUC_ROC as one of my criteria, I found it it wouldn't vary to big extent from model to model, because the test data is unbalanced and the percentage of positive class(Fraud) over negative class(Normal) is smaller than 0.01. So it is of little use in comparing models of credit cart fraud detection.

Table 2: *Logistic Regression*

	Unbalanced	Undersampled	Oversampled
Recall	0.6	0.92	**0.97**
Precision	0.85	0.05	0.05
AUC_ ROC	0.977	0.983	0.980
Time(s)	68.59	0.28	381.58

Sampled training data has higher Recall but much lower Precision.
Sampled data has sightly higher AUC_ ROC than unbalanced data.
Oversampled training data has highest Recall, though takes the longest.
Oversampled data is most suitable for Logistic Regression.

Table 3: *Linear Discriminant Analysis*

	Unbalanced	Undersampled	Oversampled
Recall	0.75	**0.85**	0.85
Precision	0.83	0.19	0.11
AUC_ ROC	0.982	0.983	0.978
Time(s)	53.89	0.52	105.75

Sampled training data has higher Recall but much lower Precision.
No big difference among the AUC_ROC of three cases.
Oversampled and undersampled training data have the same Recall, while the undersampled have higher precision and takes the shorter time, so undersampled data is most suitable for LDA.

Table 4: *KNN*

	Unbalanced	Undersampled	Oversampled
Recall	0.77	**0.92**	0.86
Precision	0.87	0.05	0.52
AUC_ ROC	0.885	0.945	0.929
Time(s)	2786.79	3.70	2981.52

Sampled training data has higher Recall but much lower Precision.
Sampled data has sightly higher AUC_ ROC than unbalanced data.
It takes too long to train with oversampled and original data ,
So undersampled data is most suitable for KNN.

Table 5: *Decision Tree*

	Unbalanced	Undersampled	Oversampled
Recall	0.81	**0.88**	0.79
Precision	0.871	0.01	0.21
AUC_ ROC	0.903	0.896	0.891
Time(s)	185.71	0.31	386.94

Undersampled data has highest Recall and AUC_ROC and lowest Precision among three cases.
No big difference among the AUC_ROC of three cases.
So undersampled data is most suitable for Decision Tree.

Table 6: *Random Forest*

	Unbalanced	Undersampled	Oversampled
Recall	0.76	**0.91**	0.89
Precision	0.91	0.09	0.32
AUC_ ROC	0.880	0.948	0.945
Time(s)	666.01	3.39	1466.632

Undersampled data has highest Recall and AUC_ROC and lowest Precision among three cases.

Sampled data has higher AUC_ ROC and Recall than unbalanced data.

Undersampled data is most suitable for Random Forest.

It has better performance and longer running time than Dicision Tree.

Table 7: *XGBoost*

	Unbalanced	Undersampled	Oversampled
Recall	0.83	**0.94**	0.86
Precision	0.92	0.04	0.13
AUC_ ROC	0.916	0.951	0.925
Time(s)	1996.59	7.82	9413.02

Undersampled data has obviously better Recall and AUC_ROC than other two cases but has the lowest Precision.

And running time increases dramatically as training data expend, so undersampled data is most suitable for XGBoost.

It has better performance and longer running time than Random forest.

Table 8: *Isolation Forest*

	Unbalanced	Undersampled	Oversampled
Recall	**0.93**	0.26	0.32
Precision	0.02	0.04	0.04
AUC_ ROC	0.938	0.974	0.974
Time(s)	5.16	2.06	6.23

Highest Recall happens when I choose 4 top features to train the model.

Unbalanced samples has the better Recall than balanced cases, though balanced cases have slightly higher AUC_ROC.

Precision of three cases are not as good as supervised models.

The running time is linear, so it is shorter than other models.

We should use original samples to train Isolation Forest.

Table 9: *Neural Network*

	Unbalanced	Undersampled	Oversampled
Recall	0.81	**0.94**	0.93
Precision	0.70	0.03	0.09
AUC_ ROC	0.902	0.939	0.955
Time(s)	813.26	6.69	1610.60

This neural network model has two layers with 50 neurons each layer, with activation function tanh.
Model trained with under-sampled data generates the highest recall and lowest precision, which is same as other supervised learning models.

- Decision Tree,Random Forest and XGBoost are all 'tree' models.Random Forest and XGBoost are ensemble of trees. The performance improves from Decision Tree to Random Forest and from Random Forest to XGBoost, while the training time increases as well. One benefit of XGBoost is that it can provide features and their impact in order, which can be used later in Isolation Forest.

Unsupervised learning:

- For unsupervised learning, I choose Isolation Forest model, which is an anomaly detection algorithm. Not surprisingly, Isolation Forest trained with unbalanced data has the best recall, since it is an anomaly detection model and anomaly is a small portion of samples. But it works not so well on high dimension features, so I trained this model with various number of features. After experiment, I found that it has the highest recall when I only used four top influential features to train the model.

- Isolation Forest has linear running time, which makes it the fastest model among all models in this project.

- The highest Recall of Isolation Forest(0.93) is higher than all supervised learning models, except for Logistic Regression trained with oversampled data(0.97). And the precision of Logistic Regression is also slightly higher than Isolation Forest. So Logistic Regression trained with oversampled data has better performance than Isolation Forest trained with original data.

- But for the extremely fast speed and fine performance of Isolation Forest, I would say there is much potential for unsupervised learning in the field of fraud detection.

Deep learning:

- Neural Network is of supervised learning, so sampled data generate better recall than unbalanced data. And the model trained with under-sampled data has the highest recall. Then reason why model trained with oversampled data has lower recall than under-sampled data is that the former one is more likely to be overfitting.

- But naturally, high recall is at the cost of low precision. The best recall of fully-connected neural network (0.94) is lower than Logistic regression trained with oversampled data(0.97). And the precision of Logistic Regression is also slightly higher than this neural network model. So Logistic Regression trained with oversampled data has better performance than this neural network model trained with under-sampled data, but with more training time.

VII. Conclusion and Discussion

I use recall as the major criterion to compare all models in this project. After training all models with unbalanced, under-sampled and oversampled dataset and tuning hyperparameters, I found logistic regression trained with oversampled data has the highest recall. A pitfall is that logistic regression is the only supervised model that better trained with oversampled data. So, the training time of it will not be so satisfactory if the dataset keeps expanding.

I focus on recall in this project, but precision is a potential problem in real world. Institutions need to balance recall and precision because if precision is too low, then it will affect many users who are holding their own card. In this project, random forest trained with oversampled data shows the best potential on balancing recall and precision. Another method to prevent false positive is to bring real-world regulations of transaction into force, such as requiring to show actual card.

Future effort can be paid on the following directions to peruse potential improvement of performance. First, tuning the ratio of two classes during sampling to balance recall and precision and reduce training time. Second, carrying out more research of unsupervised learning for anomaly detection, because such algorithms are built for unbalanced data so that whole information of dataset can be kept. Semi-supervised learning can also be considered. Finally, adding convolution into neural network to explore more possibilities.

References

[1] Japkowicz, N. 2000. Learning from imbalanced data sets: A comparison of various strategies. *Proceedings of Learning from Imbalanced Data*, 10–15.

[2] Maloof, M. 2003. Learning when data sets are imbal- anced and when costs are unequal and unknown. *Proc. of the ICML-2003 Workshop:Learning with Imbalanced Data Sets II*, 73–80.

[3] Visa, S., and Ralescu, A. 2005. The effect of imbalanced data class distribution on fuzzy classifiers - experimental study. *Proc. of the FUZZ-IEEE Conference.*

[4] Mohammed, Emad, and Behrouz Far. 2018. Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study. *IEEE Annals of the History of Computing.*

[5] Awoyemi, John, O. 2017. Credit Card Fraud Detection Using Machine Learning Techniques: A Comparative Analysis. *International Conference on Computing Networking and Informatics (ICCNI).*

[6] Sethi, N. and Gera, A. 2014. A revived survey of various credit card fraud detection techniques. *International Journal of Computer Science and Mobile Computing*.vol. 3, no. 4, pp. 780–791.

[7] Zojaji, Z., Atani, R. E. and Monadjemi, A. H. 2016. A survey of credit card fraud detection techniques: data and technique oriented perspective. *Cryptography and Security.*

[8] Raghavendra, P. and Lokesh, S. 2006. Credit Card Fraud Detection Using Neural Network. *International Journal of Soft Computing and Engineering (IJSCE).*Volume-1, Issue; (32-38).

[9] Siddhartha, B and Sanjeev, J. 2001. Data mining for credit card fraud: A comparative study. *Decision Support Systems.* 50 pp. 602–613.

[10] Linda, D. and Hussein, A. 2009. Credit card fraud and detection techniques: a review. *Banks and Bank Systems.* Volume 4, Issue 2.