Cyber Data Analysis Lab1: Fraud Detection

Shiwei Bao 4506286
Zhaoyang Cheng 4620585

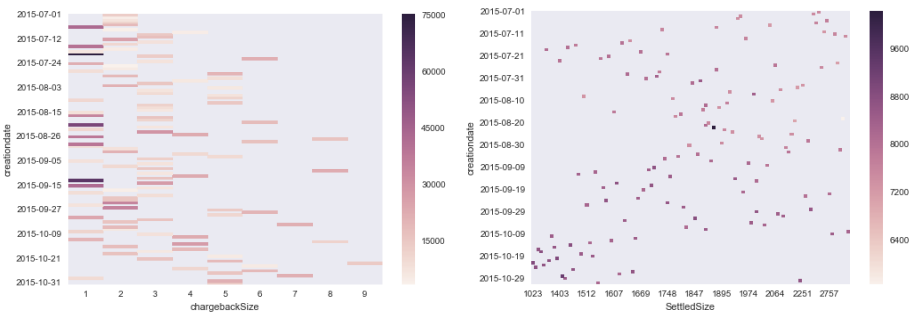
May 16, 2017

This is a report for fraud detection lab. In order to apply appropriate techniques correctly to achieve an excellent performance, we organize the report by visualization, dealing with imbalance class and classification.

1 Visualisation

First of all, we need to understand the source,the transaction data, to see that we can do. Several techniques are applied to visualize the source data from various aspects. There are 290382 records and 17 columns in total. 1.2% of these are labeled as "Chargeback" which we take it as "fraud",81.5% are "Settled" as no fraud, and the rest 18.4% are "Refused" which means not sure. First of all, we do some pre-processing on all data to study the aggregate properties(The amount in 5 different currency has been converted to EUR with new columns "amountEUR").

Intuitively, the fraud transaction does not happen frequently, but the amount per transaction is likely to be much higher than not fraud ones. We aggregate the transactions by "Creationdate" and plot it in the heatmaps,leaving out the "Refused" transactions as little info it provides. The "Chargeback" and "Setteld" data has been plotted in seperated figures. The x-ticks is the number of transactions perday against the creation date in left y-ticks, and the darker block indicating the higher average amount per day according to right y-ticks.

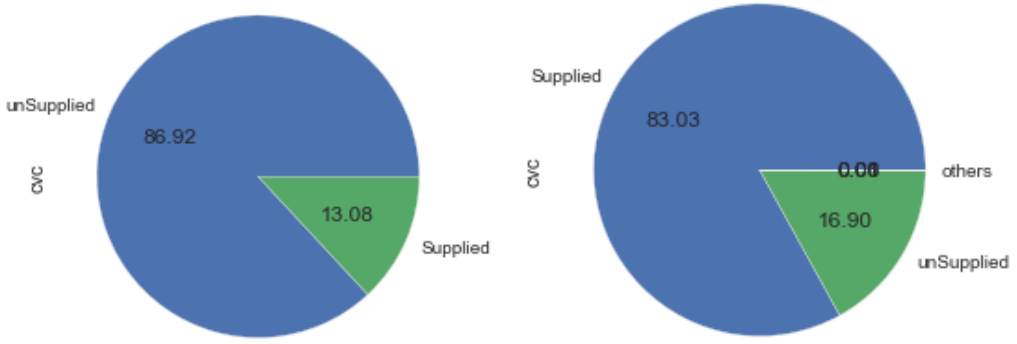


(a) Number of Chargeback deal per day (b) Number of Settled deal per day

Figure 1: Proportion of Verification Code supplied

From 1a, the "Chargeback" happens at most 9 times per day (once) and most of the time it happens less than 3 times per day. The size is much smaller than the size of "Settled" which is always more than 1000, as we expected. When it comes to the average amount, the block in "Chargeback" figure is much darker than the dots in "Settled" plot. The amount limit for "Chargeback"(75000) is almost 8 times to the one for "Settled"(9600). This big difference is coherent with our expectation as the fraud transaction tends to be with high amount.

Secondly, we notice that supplying card verification code(cvc) is a noteworthy behavior when judging a deal is fraud or not. Thus, we visualize the proportion of deal supplying their verification in Settled deal and Chargeback deal, respectively. From the pie charts below we found the big difference between the two labels. 86.92% of the transactions labeled by "Chargeback" have not supplied the verification code from 2a, but from the 2b, 83.03% of "Settled" transactions are supplied with verification code. So it's may worth to include this feature in fraud detection as transactions without verification code is likely to be fraud.



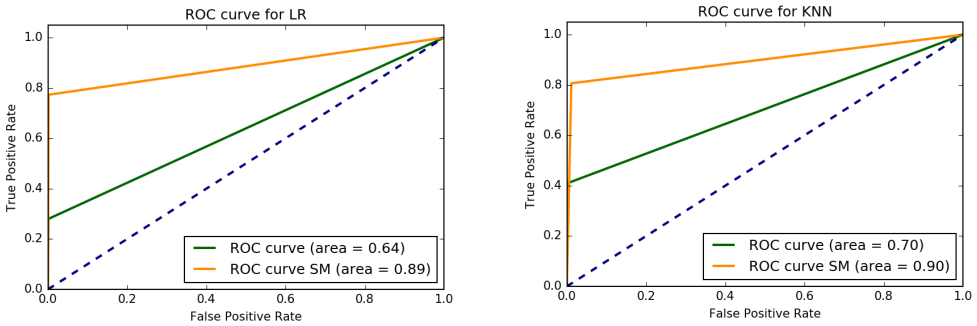
(a) Proportion of Verification Code supplied in Chargeback deal

(b) Proportion of Verification Code supplied in Settled deal

Figure 2: Proportion of Verification Code supplied

2 Imbalance

Many real-cases have imbalanced data distribution with few positives like fraud detection.But for classifiers we commonly use, they work well on balanced data, but might be less satisfying on imbalanced data. To solve this problems, we could either under sample the major class or over sample the minor class, for example using Synthetic Minority Over-sampling Technique(SMOTE). So we want to make some experiments to see whether it is a good idea to apply SMOTE by comparing the classification performance on SMOTED and UNSMOTED data. Since 2-classes transaction data are large but with few features, we use two simple and fast classifiers KNN and Linear Regression and also test with one complex Neural Networks. The ROC curves are employed to measure the performance instead of accuracy rate, because of the imbalanced source data and the imbalanced costs of misclassification. Generally, the curve in more upper left with higher AUC(area) gives better results for us. The train and test data are splitted to 8:2 and all parameters are same for all experiments to be fair. By doing the SMOTE, we oversample the train set to make the two classes in equal size. To quickly test the performance, we select only most likely relevant four features: amountEUR, bookingdate, cvc supplied, and cvc response code.

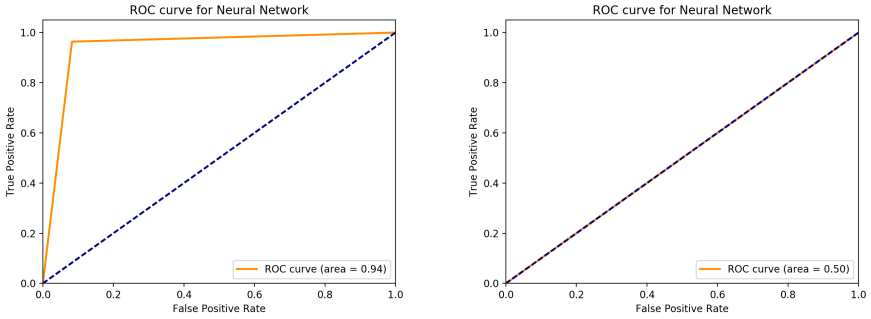


(a) ROC curve when using Linear Regression(threshold is 0.8 for smote and 0.02 for no-smote)
TP: 21 FP: 57 FN: 54 TN: 44637 TPSM: 58 FPSM: 80 FNSM: 17 TNSM: 44614

(b) ROC curve when using 3-Nearest-Neighbours
TP: 36 FP: 3 FN: 52 TN: 44678 TPSM: 71 FPSM: 504 FNSM: 17 TNSM: 44177

Figure 3: ROC Curve on 3NearestNeighbor and Linear Regression

From the 3a and 3b, we can see that for both classifiers, the curves are all at upper left to the dashed threshold line which is good. And the curves by SMOTED samples are on the top of those UNSMOTED and with larger AUC(area) indicating the average performance is better with SMOTED.However when looking at confusion matrix value, both TP and FP increase after



(a) ROC curve when using smote and Neural Network
TP: 80 FP: 3691 FN: 3 TN: 40995

(b) ROC curve when using Neural Network without smote
TP: 0 FP: 0 FN: 74 TN: 44695

Figure 4: with and without smote on neural network classifier

SMOTED for each classifier, especially for KNN, FP growing from 3 to 504. From 4, the AUC(area) in 4a with SMOTE is the highest among all, but also the FP is too high with 3691. The Neural Network does not work well according to 4b, with AUC overlapping with dashed baseline and TP, FP equal to 0.

So, by comparing these criteria, the best method here is applying KNN classifier, with higher AUC(area), (0.7 for UNSMOTED and 0.9 for SMOTED) with acceptable FP(<1000). And for whether to apply SMOTE, it depends on the cost and acceptable threshold for FP. For example,in our case, there are around 80 Positives in test set of size 45000. If FP <1000 is acceptable, then using SMOTE will improve the performance distinctively in KNN. But if the FP cost is too high and we only accept when FP <50 , then we should not use SMOTE.

3 Classification

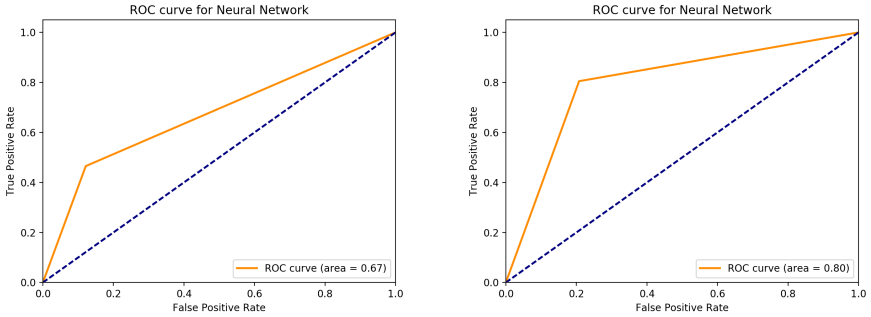
The first step in classification is preprocessing the dataset. There are 5 kinds of currency in original dataset, so we convert them all into EUR. Six features ['issuercountrycode','shopperinteraction','amountEUR','bookingc are extracted, with 'simple_journal' as label. Because we believe that these features are relevant with fraud detection. From the visualization plot above, it shows that chargeback label has strong relationship with 'cardverificationcodesupplied'. Over 80 percent deals in chargeback do not provide the verification code, while in setteld deal, the situation is completely reversed. We convert the 116 different country names in 'issuercountrycode' to numeric 0-115, likewise, ['Ecommerce','AuthCont','POS'] in shopperinteraction are converted to [0,1,2]. As for the bookingdate, we set first day 2015-07-01 as 0, and last day 2016-03-05 as 152(date not continuous). Thus we obtain a numeric array which is suitable to be input into classifiers. Since applying SMOTE improves the average classification performance in section 2, we use SMOTE in this section by default. And again, because of the imbalanced source data and imbalanced cost for misclassification, we still employ the ROC curve as our performance criteria. An ROC graph depicts relative trade offs between benefits (true positives) and costs (false positives), and the curve in more upper left side with higher AUC(area) gives better average results.

3.1 Time-based Split train-test set

Since the initial source is sequential data and the label "Chargeback" which we take as fraud is only valid after its booking date. So we split the dataset based on bookingdate which means to use the past data as train set and predict the future data as test set. To get better performance, we exclude the bookingdate feature after having split the data based on bookingdate, which means booking-date feature will not distract classifiers.

3.1.1 'Black box' classifier

The first classifier we use is the multiple layers perceptron, which is a kind of neural network as 'black box' model. After tuning the parameters several times, we set the hidden layer size as 300 through experiments, and the activation function is 'relu' at the end. As you can tell from Figure 5, increasing proportion of training set has salient improvement for neural network, which is a typical characteristic of neural network. A better average result is given by setting the 0.1 as test, the AUC(area) is 0.80.



(a) ROC curve Neural Network with 0.5 test set
TP: 114 FP: 4836 FN: 219 TN: 106586

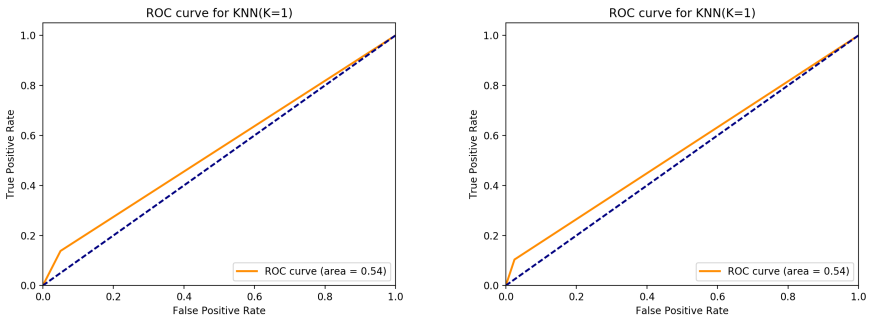
(b) ROC curve Neural Network with 0.1 test set
TP: 256 FP: 4576 FN: 62 TN: 17457

Figure 5: different test set ratio with neural network classifier

3.1.2 "White box" classifier

Besides the "Black Box" algorithm Neural Network, we choose the k-nearest neighbor with k=1 as our "white box" classifier, which is a non-parametric method. We use euclidean distance to measure how close two samples are in euclidean space. So for a certain test point, it will use the same label of its nearest neighbor in eudlidean distance. If a transaction is detected to be fraud, then it means there was one fraud transaction happened in most similar way as this one, such as no cvc code supplied or high transaction amount.

As you can see from Figure 6, nearest neighbor does not perform well even with smote, which we conclude arises from some 'noise points' brought by SMOTE. The AUC(area) is the same for two cases, and FP is much lower in 6b. However, nearest-neighbor is one of our best classifiers in experiment. So the algorithm we choose here is to set 0.1 as test with 1NN, and the AUC(area) is 0.54.



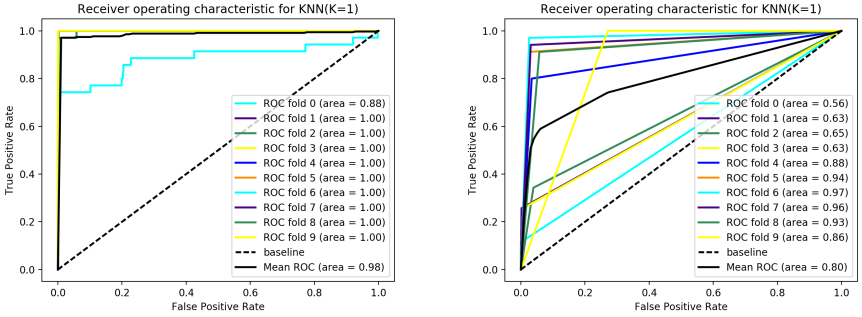
(a) ROC curve 1NN with 0.5 test set
TP: 46 FP: 5572 FN: 287 TN: 105850

(b) ROC curve 1NN with 0.1 test set
TP: 33 FP: 538 FN: 285 TN: 21495

Figure 6: different test set ratio with k Nearest Neighbor(k = 1)

3.2 10-folds cross validation

Since the time-based split training set has poor performance, we decide to conduct a random split train set and test set with 10 folds cross validation. From Figure 7, we find the performance



(a) ROC curve Neural Network (b) ROC curve nearest neighbor

Figure 7: 10 folds cross validation on classifiers

varies in each fold caused by every different SMOTE, which could lead to research on more stable SMOTE. Both the NN and KNN(K=1) gives a AUC(area) larger than 0.8 which is much better than those in section 3.1, which means splitting data set randomly gives better result than splitting set based on time. However , we insist that time-based split processing is closer to real life cases and its result is more convincing on prediction. Furthermore, it is noteworthy that neural network takes 10 times running time of nearest neighbor even though the former performs better than the latter.