The University of Oklahoma

Intelligent Data Analytics

(DSA/ISE-5103)

Detecting Fraud on Plastic Card Transactions

Project Report Draft

December 4, 2017

Daniel Casares and Felipe Olivera

Executive Summary (1 page)

– Concise problem statement

– List of major concerns/assumptions (if any)

– Summary of findings

– Recommendations

Problem description and background

The use of plastic cards (i.e. credit and debit cards) as a payment method has grown significantly over past years, unfortunately so has fraud (Bahnson 134). Plastic card fraud is defined as an unauthorized account activity committed by means of the debit and credit facilities of a legitimate account. Some successful fraud tactics observed in the industry are lost and stolen card fraud, counterfeit card fraud, card not present fraud, mail non-receipt card fraud, account takeover fraud and application fraud (Krivko 6070). Based on the latest figures gathered in 2015, card fraud accumulated $21.84 Billion worldwide in losses (The Nilson Report 6). When banks lose money due to credit card fraud, the losses partially are passed to customers through higher interest rates, higher membership fees and reduced benefits. Hence, it is both the banks' and cardholders' interest to reduce illegitimate use of credit cards (Maes 2).

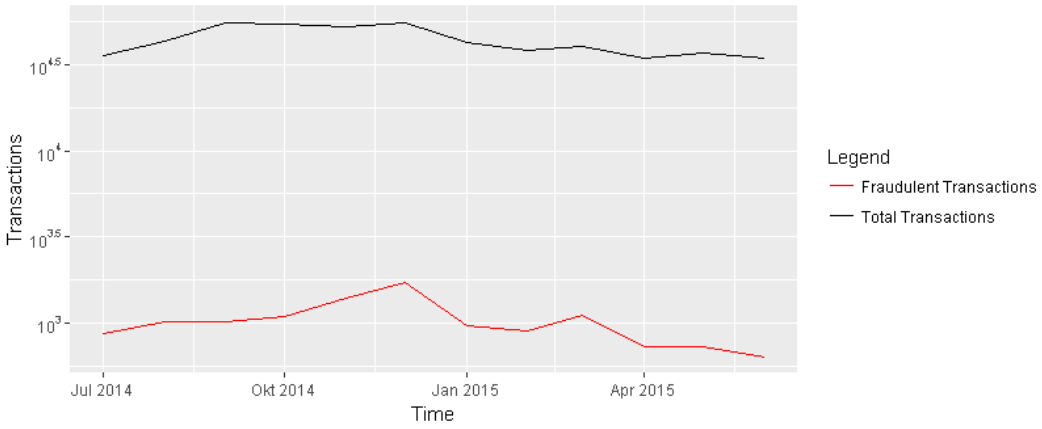
In this work, we consider the problem of identifying whether a credit or debit card account has been subject to fraudulent activity, using real-life transaction data from a Latin American credit card processing company. The goal is to construct a supervised learning model that can detect fraud on new (previously unseen) plastic card transactions. Fraud detection is, given a set of credit card transactions, the process of identifying those transactions that are fraudulent. Thus, the transactions are classified as genuine or as fraudulent transactions (Maes 2). Different detection systems that are based on machine learning techniques have been successfully used for this problem, in particular: neural networks, bayesian learning, artificial immune systems, association rules, hybrid models, support vector machines, peer group analysis, decision tree techniques such as ID3, C4.5, and random forest, discriminant analysis, social network analysis and logistic regression (Bahnson 135, Mahmoudi 2510).

Exploratory data analysis

For this project we used a dataset provided by a Latin American card processing company. The dataset consists of fraudulent and legitimate transactions made with debit and credit cards between July 2014 and June 2015. The total dataset contains 41,091,288 individual transactions, each one with 13 attributes (as shown in the table below), including a fraud label indicating whenever a transaction is identified as fraud. This label was created internally in the card processing company, and can be regarded as highly accurate. In the dataset only 12,632 transactions were labeled as fraud, leading to a fraud ratio of 0.031%.

Attribute name	Description
amount	Amount of the transaction in USD
id_issuer	Unique identifier of the bank issuer of the card
id_merchant	Unique identifier of the merchant
datetime	Date and time of the transaction
country_code	Numeric code that identifies the country of the transaction
tokenized_pan	Unique identifier of the credit card
pos_entry_mode	Numeric code that identifies the transaction entry mode (e.g. Chip and PIN, magnetic strip, etc.)
id_mcc	Identification of the Merchant Category Code (ISO 18245)
is_upscale	Indicates if the card holder is an upscale customer
mcc_group	Merchant Category Code grouping by major type of business
type	„C“ for credit cards, „D“ for debit cards
is_fraud	1 if the transaction was fraudulent, 0 otherwise

Due to the low proportion of the target class (i.e. frauds) in the given dataset, the class imbalance problem arises. Classification of imbalanced data is difficult because standard classifiers are driven by accuracy, thus the minority class may simply be ignored (Visa 67). Generally all classifiers present some performance loss when the data is unbalanced (Prati 253). Additionally, many imbalanced datasets experience problems related to its intrinsic characteristics, such as lack of density and information. To illustrate these issues, a dataset containing of 5 : 95 minority-majority examples and a dataset of 50 : 950 are compared. Though the imbalance factor is the same as in both datasets in the first case the minority class is poorly represented and suffers more from the lack of information factor than in the second case. In order to reduce these problems in our modeling, a smaller subset of transactions with a higher fraud ratio is selected from the original data. This *new* dataset contains 523,049 transactions and a fraud ratio of 2.33%. In this dataset, the total financial losses due to fraud are 1,876,697 USD. It was selected considering all the fraudulent transactions in the original dataset, in addition to all the legitimate transactions for the corresponding customers. Next, transactions for some customers that have never been victims of fraud were added. From plotting the amount of fraudulent and total transactions over time we can see that the proportion of fraudulent transactions varies over time.



Analysis plan

Explanation of modeling choice

In scientific literature three basic tried and tested classification algorithms are discussed. These are logistic regression, decision tree and random forest (Whitrow 31-51, Bahnson 134). First, we will implement those three models. The models are implemented in R and trained with the *caret* package using repeated k-fold cross validation (5 repeats of 10-fold CV). The *new* dataset is split randomly in 70% training and 30% test. Further, we want to implement another tree model, the *Extreme Gradient Boosting (XGBoost)* that is often a winning model for data science competitions on *Kaggle* .

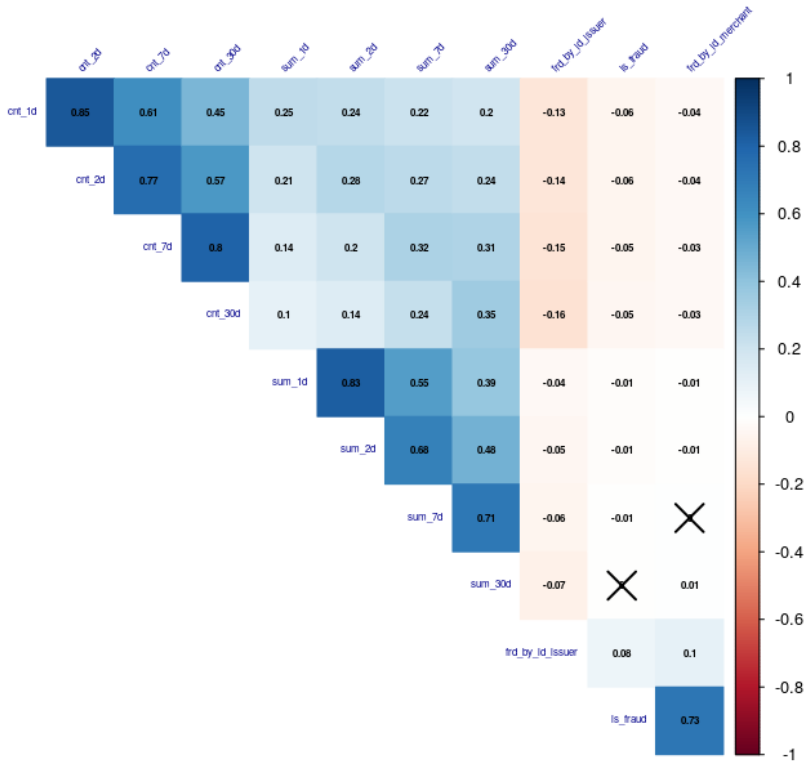
Feature Engineering

The raw data contains typical raw credit card fraud detection features for each transaction such as amount, date and time, merchant type (e.g. gas station), entry mode, among others (as stated above). Just with those attributes, fraud may be identified at the transactional level. However, a single transaction is not enough to detect a fraudulent transaction since it leaves behind the customer spending behavior. In order to fulfill this problem, Whitrow et al. propose to perform transaction aggregation (31-51).

The derivation of the aggregation features consists in grouping the transactions made during the last given number of hours by card number, followed by calculating the number of transactions and the total amount spent on those transactions. We processed those new attributes for time windows of 1 day, 2 days, 1 week and 30 days, respectively. This resulted in 8 new features for the model. When selecting the transactions related to the calculus of this feature, we took some

assumptions: (1) the own transaction is not considered; and (2) the transactions must be non-fraudulent.

In addition to the previously mentioned features, we added two more features, indicating the fraud indexes by Issuer Bank (*id_issuer*) and by Merchant (*id_merchant*), respectively. *Frd_by_Id_issuer* is the ratio of the number of frauds for each bank and overall frauds and *frd_by_id_merchant* is the ratio of the number of frauds for each merchant and overall frauds. The correlation matrix above of all engineered features and *is_fraud* reveals a high positive correlation (0.73) between *is_fraud* and *frd_by_id_merchant*. This valuable feature indicates that certain merchants are associated with fraudulent activity.



Validation

The models performance is evaluated using the standard binary classification measures Area Under the Curve (AUC), F1-Score, Log Loss and a custom cost-based metric (savings). The Receiver Operating Characteristic (ROC) curve is a typical technique for summarizing classifier performance over a range of trade-offs between true positive and false positive error rates. The Area Under the Curve (AUC) is an accepted performance metric for a ROC curve (Chawla 855). The main goal for learning from imbalanced datasets is to improve the recall (TP out of TP+FN) without hurting the precision (TP out of TP+FP). However, recall and precision goals can be regularly conflicting, since when increasing the true positive for the minority class, the number of false positives can also be

increased; this will reduce the precision. The F1-score metric is one measure that combines the trade-offs of precision and recall, and outputs a single number reflecting the "goodness" of a classifier in the presence of a minority class. While ROC curves represent the trade-off between values of TP and FP, the F1-score represents the trade-off among different values of TP, FP, and FN (Chawla 857). The expression for the F-value is as follows:

$$F1Score = 2\frac{Precision.Recall}{Precision + Recall}$$

Log Loss is a performance measure used to evaluate predictions on *Kaggle* competitions, among others. Log loss measures the uncertainty of the model and penalizes extremely wrong probabilities.

Nevertheless, these three measures may not be the right evaluation criteria when evaluating fraud detection models, because they implicitly assume that misclassification errors carry the same cost as the correct classified transactions. In practice, wrongly predicting a fraudulent transaction as legitimate usually carries a considerably higher financial cost than the opposite case (Bahnsen 136-137). The goal of companies, when it comes to fraud detection, is to take a decision to minimize the losses. Using a cost matrix (as described below) that defines the cost for both types of misclassification error, a savings metric can be computed as the difference between the cost of using no algorithm (sum of the amounts of fraudulent transactions) and the associated cost of the predictions.

	Predicted Negative	Predicted Positive
Actual Negative	*0 USD*	*20 USD*
Actual Positive	*Total transaction amount*	*20 USD*

Results and validation of analysis

Method	Package	Parameter	Selection	AUC	F1	LogLoss	Cost
Logistic Regression	stats	-	-	-	-	0.04344	-
Decision Tree	rpart	cp split prune					
Random Forest	randomForest	mtry	13				
XGBoost							

Note: cross validation was repeated k-fold (5 repeats of 10-fold CV for all models) using the caret package

Note: We are still facing some issues due to the size of datasets (long computational time, data allocation problems) and hope to solve them to be able to report some results in the upcoming days.

Conclusion

References

Bahnsen, Alejandro Correa, et al. “Feature engineering strategies for credit card fraud detection.” *Expert Systems with Applications*, vol. 51, 2016, pp. 134–142.

Chawla, Nitesh V. “Data Mining for Imbalanced Datasets: An Overview.” *Maimon O., Rokach L. (eds) Data Mining and Knowledge Discovery Handbook.* Springer, Boston, MA, 2005.

Krivko ,M. “A hybrid model for plastic card fraud detection systems.” *Expert Systems with Applications*, vol. 37, 2010, pp. 6070–6076.

Maes, Sam et al. “Credit Card Fraud Detection Using Bayesian and Neural Networks”. *Proceedings of NF*, 2002.

Mahmoudi, Nader, et al. “Detecting credit card fraud by Modified Fisher Discriminant Analysis.” *Expert Systems with Applications,* vol. 42, 2015, pp. 2510–2516.

Prati, R.C. et al. “Class imbalance revisited: a new experimental setup to assess the performance of treatments methods.” *Knowledge and Information Systems*, vol. 45, 2015, pp. 247–270.

"The Nilson Report." David Robertson, 17 Oct. 2016, www.nilsonreport.com/upload/content_promo/The_Nilson_Report_10-17-2016.pdf. Accessed 02 Dec. 2017.

Visa, Sofia. "Issues in mining imbalanced data sets – a review paper." Proc. 16th Midwest Artificial Intelligence and Cognitive Science Conference, 2005, pp. 67–73.

Withrow, C. "Transaction aggregation as a strategy for credit card fraud detection". Data Mining and Knowledge Discovery, vol. 18, 2009, pp. 30-55.

Appendix

– Data visualizations, tables, etc. which support the work, but are not of primary importance

– List of data transformations, missing value imputations, outlier treatment, etc.

– List of any important assumptions not otherwise included

– Important code excerpts or algorithms used / developed if any.