Effectiveness of Weighted ML Models for Credit Card Fraud Detection

TASHRISH PATEL, University of Waterloo, Canada

1 INTRODUCTION

Credit cards make up an annual $100 Billion global market. As credit cards become more and more popular in the digital age, they also come with their inherent risks. Credit card fraud is the act of a fraudulent credit card transaction in order for criminals to acquire money or goods and services not belonging to them. Criminals committing credit card fraud are rarely apprehended and therefore, automatic credit card fraud detection can be of great significance. The task of automation credit card fraud classification is not easy, since the data-sets for credit card transactions are severely imbalanced. An imbalanced data-set is one where the proportion of one class is significantly higher than the rest of the classes. Classification tasks on imbalanced data-sets are difficult since it tends to be possible to gain a high level of accuracy despite misclassifying most of the data points from the minority class. A plausible way of attempting to classify on an imbalanced data-set is to apply class weights such that each data point from the minority class is given more importance and each data point from the majority class is given less importance. This study will attempt to see how class weights can help tackle the task of credit card fraud classification. This will be done by comparing Binary Logistic Regression model and Random Forest Classifier model without weights to those with class weights. This study will aim to find if a weighted classification will be able to improve upon the unweighted classification of credit card fraud detection.

2 METHODOLOGY

2.1 Data

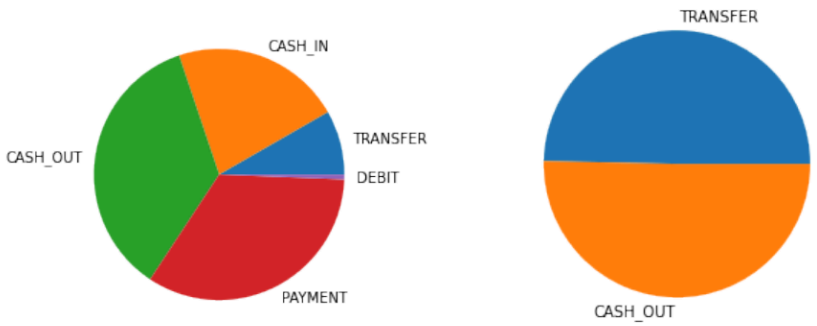
Credit Card transactions data tend to contain confidential information and therefore, they are scarcely available to the public. Hence, the data-set for this study is one created from a simulator, PaySim, that simulates credit card transactions. The simulator utilizes a private data-set to generate a simulated data-set that mimics ordinary credit card transactions along with fraudulent activity. The data-set contains 3,223,223 synthetic transactions.

Table 1. Fields of the Data-set

Field	Data Type	Description
type	STRING	type of transaction
amount	DOUBLE	amount of the transaction
oldbalanceOrg	DOUBLE	account balance before the transaction
newbalanceOrig	DOUBLE	account balance after the transaction
oldbalanceDest	DOUBLE	account balance of recipient before the transaction.
newbalanceDest	DOUBLE	account balance of recipient after the transaction.
isFraud	Binary	if the transaction is fraudulent or not

Table 2. Frequency and average amount of Valid and Fraudulent Transactions

Category	Number of Transactions	Average Amount
Valid	3,220,396	$ 156,675.40
Fraudulent	2,827	$ 1,309,250.00

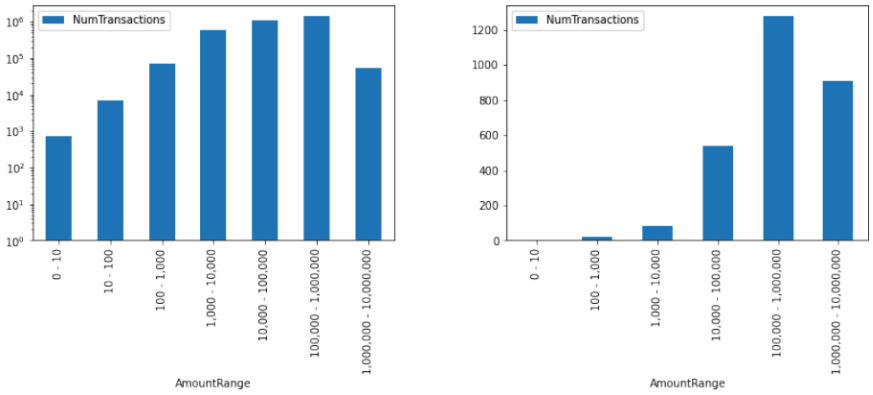


(a) Proportions of transaction by type

(b) Proportions of fraudulent transaction by type

Fig. 1. Breakdown of proportion of all and fraudulent transactions by type

Fig. 1. Breakdown of proportion of all and fraudulent transactions by type



(a) Number of transactions by amount (b) Fraudulent transactions by amount

Fig. 2. Number of all and fraudulent transactions transactions by amount

2.1.1 Exploratory Data Analysis.
From table 2 we can see the imbalanced nature of data-set and how on average fraudulent transactions amount is more than 8 times the amount of valid transactions. Figure 1 also demonstrates that how fraudulent transactions are only of type 'TRANSFER' and 'CASH OUT'. Finally, Figure 2 illustrates how the distribution of fraudulent transactions tend to have amount higher than those of valid transactions. 2.2 Pre-processing The data needs to be pre-processed because it contains categorical variables and Machine Learning algorithms use quantitative variables to discriminate between classes. The categorical variable in this data-set that will be used is type. The first transformation applied to the field type is the StringIndexer. The StringIndexer maps the string variable to indices belonging to the set $\{0, \ldots, numberOfUniqueValues - 1\}$. The order of the index assignment is ordered by frequency of that value in the data-set. Furthermore, the string-indexed variable is now transformed into a one-hot vector using the OneHotEncoder. This creates a vector of length $n - 1$ (where n is the number of unique categorical values) where all values are 0.0 except for at the index location which has a value of 1.0. Finally, all the feature variables are placed into one vector which represents the features for the respective transaction.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org. © 2021 Association for Computing Machinery. Manuscript submitted to ACM