CREDIT CARD FRAUD DETECTION

Saurav Prasad
Computer Science and Engineering
PES University
Bengaluru, Karnataka
sauravprasad112@gmail.com

Mrityunjay Jha
Computer Science and Engineering
PES University
Bengaluru, Karnataka
jmrityunjay1999@gmail.com

Siddhant Kumar
Computer Science and Engineering
PES University
Bengaluru, Karnataka
siddhantkumarxyz@gmail.com

***Abstract*—Recent research has shown that machine learning techniques have been applied very effectively to the problem of payments related fraud detection. Such ML based techniques have the potential to evolve and detect previously unseen patterns of fraud. We applied multiple ML techniques based on Random Forest and its variation such as Refined Weighted Random Forest and SMOTE(logistic regression) to the problem of credit fraud detection using a labelled dataset containing fraud and non-fraud transactions. We will try to make our proposed approaches to detect fraud transactions with high accuracy and reasonably low number of false positives. Random forest (RF) is widely used in many applications due to good classification performance. However, its voting mechanism assumes that all base classifiers have the same weight. We mainly focus on the weighted voting mechanism and then propose a novel weighted RF. SMOTE stands for Synthetic Minority Over-sampling Technique. In this technique we basically generate synthetic points for minority class to make the number of minority samples (fraud examples) equal to that of majority samples(non-fraud examples). Then we use logistic regression to train the model. At last we compare the results.**

Keywords—Weighted Random Forest, SMOTE, Logistic Regression

I. INTRODUCTION

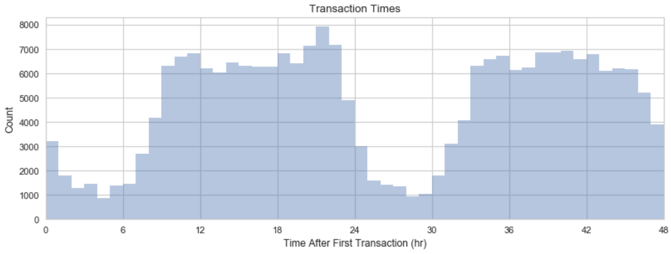
A credit card is a thin handy plastic card that contains identification information such as a signature or picture, and authorizes the person named on it to charge purchases or services to his account - charges for which he will be billed periodically. Today, the information on the card is read by automated teller machines (ATMs), store readers, banks and is also used in online internet banking systems. They have a unique card number which is of utmost importance. Its security relies on the physical security of the plastic card as well as the privacy of the credit card number. There is a rapid growth in the number of credit card transactions which has led to a substantial rise in fraudulent activities. Credit card fraud is a wide-ranging term for theft and fraud committed using a credit card as a fraudulent source of funds in a given transaction. Generally, the statistical methods and many data mining algorithms are used to solve this fraud detection problem. A high importance is given to develop an efficient and secure electronic payment system to detect whether a transaction is fraudulent or not. In this paper, we will focus on credit card fraud and its detection measures. A credit card fraud occurs when one individual uses another individuals' card for their personal use without the knowledge of its owner. When such cases take place by fraudsters, it is used until its entire available limit is depleted. Thus, we need a solution which minimizes the total available limit on the credit card which is more prominent to frauds.

DATASET

The datasets contain transactions made by credit cards in September 2013 by european cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions. It contains only numeric input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, they have not provided the original features and more background information about the data. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

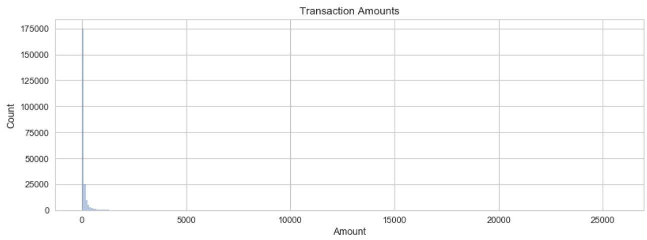
II. EXPLORATORY DATA ANALYSIS

A. Time

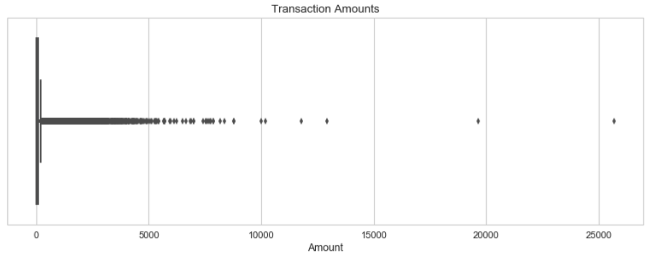


This is the histogram of transaction times with one bin per hour. We see that there are two lulls in credit card transactions during nighttime on each day.

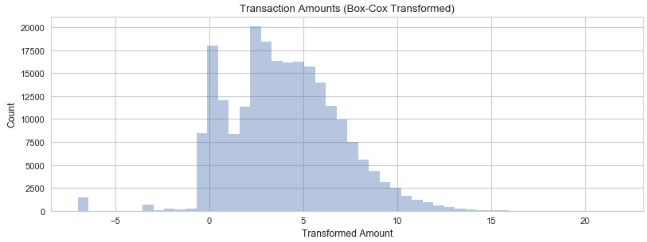
B. Amount



By looking at this histogram, it looks like that there are many outliers. We will verify it by plotting the box plot.

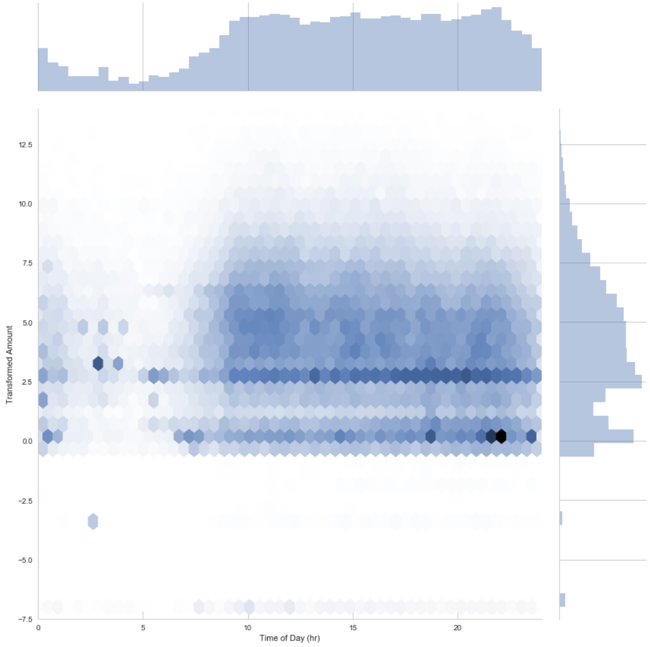


We can clearly see that there are many outliers in the right side compared to the left side which indicates that the amounts are right-skewed.



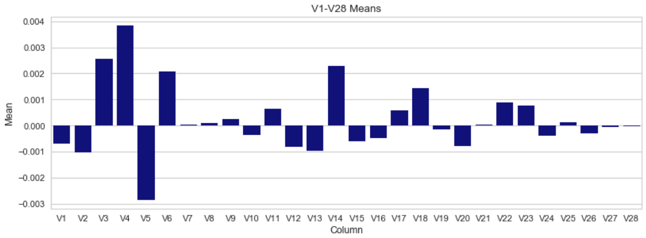
The distribution appears to be bimodal, suggesting a divide between "small" and "large" purchases.

C. Time vs. Amount



The transaction amounts appear to be similarly distributed throughout the daytime hours. However, in the earliest hours of the day, around 5-7 AM, amounts around 2.5 are the most common.

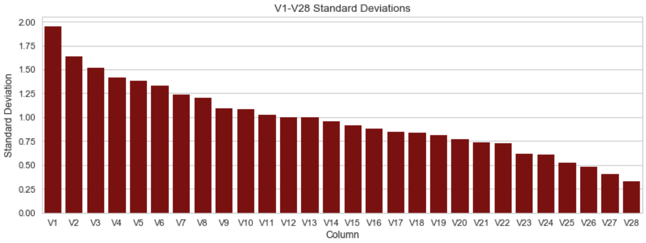
D. V1-V28



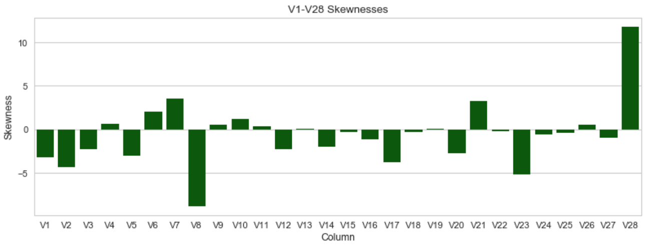
All of V1-V28 have approximately zero mean.

The PCA variables have roughly unit variance, but as low as ~0.3 and as high as ~1.9.

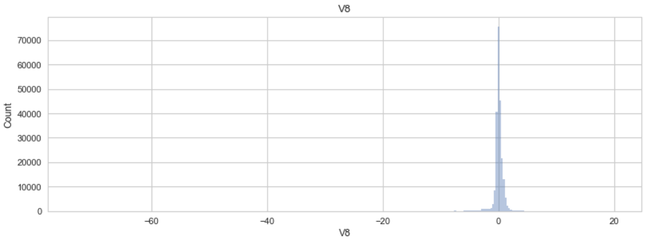
The PCA variables have roughly unit variance, but as low as ~0.3 and as high as ~1.9.



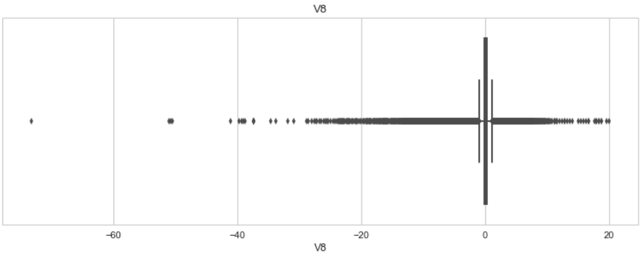
The PCA variables have roughly unit variance, but as low as ~0.3 and as high as ~1.9.



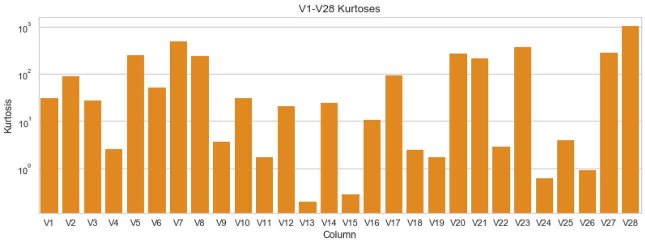
A few of the PCA variables are significantly skewed.



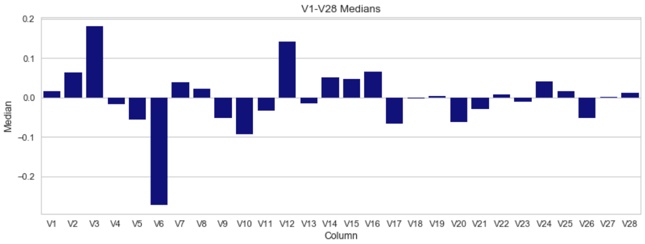
The histogram doesn't show us outliers.



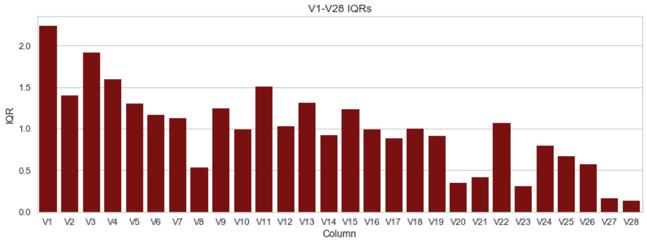
The boxplot is also hard to read due to the large number of outliers, which indicates high kurtosis in V8.



We've learned that many of the PCA variables are heavy-tailed. The large numbers of outliers in V1-V28 motivates us to consider robust descriptive statistics.



The medians are also roughly zero.



The IQRs of V1-V28 are on a similar scale as the standard deviations.

III. MODELLING

A. *Weighted Random Forest*

a. *Background*

The point is to explore how class imbalance leads to indesirable classification performance and how we can use non-unit weighting to address this. The dataset used here is a highly unbalanced dataset where the relevant class is whether the transaction is fraudulent (class 1) or not (class 0). In particular in the setting of fraud, each undetected fraudulent transaction carries an average cost, e.g., due direct monetary loss or an indirect reputation loss. On the other hand, a cost may also be associated with false positives, due to e.g., that a human must manually examine this transaction.

For this reason, it does not make sense to just accept an arbitrary tradeoff dictated by a classifier. An acceptable trade off would balance expectations about costs of both false positives and false negatives.

The point is to explore how class imbalance leads to indesirable classification performance, in particular favouritism of the majority class and whether we (indirectly) can adapt the loss function to create a more appropriate balance between the false positive rate and false negative, rather than just minimizing the total number of misclassifications.

b. *Implementation*

Since the data is so unbalanced, using a typical accuracy score to evaluate our classification algorithm would be misleading. For example, if we just used a majority class to assign values to all records, we will still be having a high accuracy, but we would be classifying all ones incorrectly.

Train/Test Split :To keep this simple, we use a traditional training/test split.(70% train and 30% test).

Normal unbalanced RandomForestClassifier approach:
This is usual Random forest classifier which is the default in scikit-learn. All transactions regardless labels are weighted the same. We still had quite the bias toward classification of the majority class.

Using balanced RandomForestClassifier approach:
Weights are inversely proportional with the frequency of class observation.

Finally we implemented custom weighting having Wneg(non-fraud weighting) = 10**-5. We ran a loop to find Wpos(fraud weighting) so that to find the tradeoff between false positive rate and false negative rate. We then calculated all the evaluation metrics.

B. *SMOTE*

SMOTE stands for Synthetic Minority Over-sampling Technique.
SMOTE creates new synthetic points in order to have an equal balance of the classes.
It will take more time to train but will be more accurate than random undersampling.

- Solving the Class Imbalance: SMOTE creates synthetic points from the minority class in order to reach an equal balance between the minority and majority class.
- Location of the synthetic points: SMOTE picks the distance between the closest neighbors of the minority class, in between these distances it creates synthetic points.
- Final Effect: More information is retained since we didn't have to delete any rows unlike in random undersampling.
- Accuracy || Time Tradeoff: Although it is likely that SMOTE will be more accurate than random under-sampling, it will take more time to train since no rows are eliminated as previously stated.

Pipeline model Implementation

A machine learning pipeline is used to help automate machine learning workflows. They operate by enabling a sequence of data to be transformed and correlated together in a model that can be tested and evaluated to achieve an outcome, whether positive or negative.
Machine learning (ML) pipelines consist of several steps to train a model. Machine learning pipelines are iterative as every step is repeated to continuously improve the accuracy of the model and achieve a successful algorithm. To build better machine learning models, and get the most value from them, accessible, scalable and durable storage solutions are imperative, paving the way for on-premises object storage.

IV. RESULT

	Weighted Random Forest	Smote (Logistic Regression)	Decision Tree(Research Paper)	Random Forest(Research Paper)
Precision	0.885	Non-Fraud : 1.0 Fraud : 0.11	*	*
Recall	0.867	Non-Fraud : 0.99 Fraud : 0.86	*	*
F1	0.876	Non-Fraud : 0.99 Fraud : 0.20	*	*
Accuracy score	*	0.99	0.94	0.95

V. CONCLUSION

In our paper, Machine learning techniques like weighted Random Forest and SMOTE(Logistic Regression) were used to detect fraud in credit card systems. So, we are making use of average precision-recall score to compare both the models.We can see that the average precision-recall score for the Smote Model is 0.76 whereas for Random Forest it is 0.73. We can clearly infer from the above data that the Smote technique works better when compared to Random Forest.

Our model(Logistic Regression using SMOTE technique) has an accuracy of 99% while the two research papers referred had the accuracies of 0.94 and 0.95.

REFERENCES

.

[1] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang and C. Jiang, "Random forest for credit card fraud detection," 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), Zhuhai, 2018, pp. 1-6, doi: 10.1109/ICNSC.2018.8361343.

[2] Lakshmi S V S S1,Selvani Deepthi Kavila21,2 Machine Learning For Credit Card Fraud Detection Department of CSE, Anil Neerukonda International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 24 (2018) pp. 16819-16824 © Research India Publications. Institute Of Technology And Sciences(A), Visakhapatnam-531162,India