Synthetic Financial Datasets For Fraud Detection

Data Science Student
Colorado University Boulder

ABSTRACT

From traditional to emerging sectors, there is not one single business that is fully immune from fraud. Some studies show that fraud of various kinds could cost businesses 1%-1.75% of their annual sales, this translates to around $200 USD billion a year worldwide!

As one of the most common fraudulent activities, digital transaction fraud impacts around 127 million people or approximately $8 billion in attempted fraudulent charges on Americans alone. Thus imperative for financial companies to understand the characteristics of fraudulent transactions and develop predictive models accordingly to flag down potentially risky activities for fraud prevention.

INTRODUCTION

Fraud detection is an important problem that affects many different industries, including the financial and banking sectors, as well as government organizations, insurance companies, and law enforcement. Recent years have seen a sharp increase in fraud attempts, making this subject more important than ever. Despite efforts on the part of the problematic organizations, fraud costs hundreds of millions of dollars every year. Locating these can be challenging because just a small number of samples from a large community can confirm fraud. Statistics and data mining make it possible to anticipate fraud, identify it right away, and respond quickly to minimize the negative impacts of fraud.

Due to the sensitive nature of financial data, there is a huge lack of publicly available datasets on financial services. Financial datasets are important for fraud prevention research since they can allow for further analysis, and make organizations more agile in order to adjust to new threats or trends. However, the private nature of financial transactions leads to no publicly available financial datasets.

In this study, I analyze a synthetic dataset[1] generated using a simulator called PaySim, the dataset is available on Kaggle. The data was generated using aggregated data from private sources to generate a synthetic dataset that resembles very closely normal transactions but abstracts away personal details due to privacy concerns. This could be a good approach to increase the availability of financial data so that more research can be done in this domain, and fraud prevention strategies can make more progress in the race against malicious actors.

In this study, I train a classifier to effectively flag fraudulent transactions.

RELATED WORK

The use of synthetic financial data to train models for fraud detection is relatively new. The idea of this work is that synthetic data can resemble original data well enough that companies can make their original data available as synthetic data so that further research can be performed and new methods for detection and prevention can be developed, as new trends arise.

(Lopez-Rojas and Axelsson, 2012a) offered the first implementation of a simulator for financial transactions with a mobile money transactions simulator. The challenges of implementing an appropriate fraud detection control on a mobile money system that was being developed led to the implementation of this simulator. The problem of the dearth of actual data was first addressed in this work. The performance of various machine learning algorithms in identifying money laundering trends was tested using the fictitious dataset produced by the simulator.

In their 2013 paper, Gaber et al. proposed a further method for creating synthetic logs for fraud detection. The significant distinction in this instance was the availability of real data to calibrate and compare the simulator's outputs' level of quality. This study's goal was to produce test results that researchers could use to compare various strategies. This work considerably differs from the synthetic data we will be using, since the PaySim simulator proposes a new approach to data analysis and pays close attention to assessing the quality of the final synthetic data set.

Relevant related work is as follows:

- Synthetic Logs Generator for Fraud Detection in Mobile Transfer Services[2]
- PaySim: A Financial Mobile Money Simulator For Fraud Detection[3]
- Analysis of fraud controls using the PaySim financial simulator[4]
- Advantages of the PaySim Simulator for Improving Financial Fraud Controls[5]
- Fraud Detection in Mobile Payment Utilizing Process Behavior Analysis[6]

PROPOSED WORK

The goal of this project is to develop a model that can detect and classify fraudulent transactions effectively. Since the dataset closely resembles real-life financial transactions, if deployed to a production environment, it could help mitigate fraud by detecting and classifying fraudulent activity in real-time. Helping to reduce the impact of fraud on businesses and account users, as well as helping to preserve the trust of all actors.

1. Analysis of the Synthetic Financial Dataset For Fraud Detection

We begin by doing some exploratory data analysis to build some intuition around our data.

```
# Read the data
df=pd.read_csv("Fraud.csv")
df.shape
```

```
Out[92]:

(6362620, 11)
```

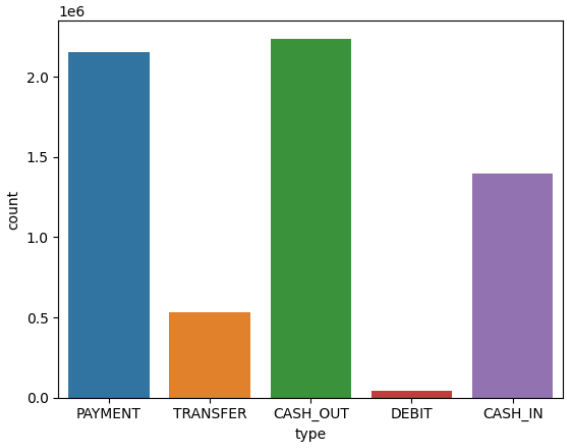
After importing the dataset we see that we have approximately 6.3 million observations with 11 variables.

```
# Getting information about data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6362620 entries, 0 to 6362619
Data columns (total 11 columns):
 #   Column          Dtype
---  ------          -----
 0   step            int64
 1   type            object
 2   amount          float64
 3   nameOrig        object
 4   oldbalanceOrg   float64
 5   newbalanceOrig  float64
 6   nameDest        object
 7   oldbalanceDest  float64
 8   newbalanceDest  float64
 9   isFraud         int64
 10  isFlaggedFraud  int64
dtypes: float64(5), int64(3), object(3)
memory usage: 534.0+ MB
```

Here we see the different types of variables our dataset contains. The target variable will be the “isFraud” column, which is a boolean class represented in a binary format by 0 or 1.

We see that one of the columns' names is type, and there are five categories corresponding to the different types of transactions of observations found in the dataset. The types are “Payment, Transfer, Cash_Out, Debit, and Cash_in”, here is a bar plot showing the distribution of different types of observations.

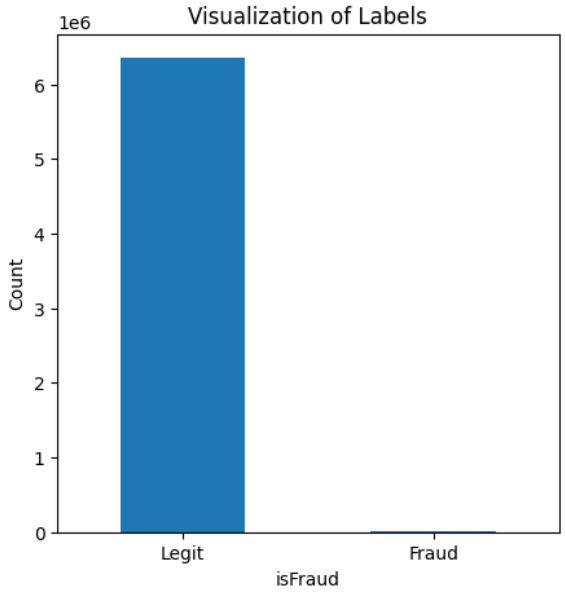


After some exploration, we realize that this is a highly imbalanced dataset where the target variable “isFraud” possess two classes, 0 and 1. Where the former is the majority class, and the latter the minority class.

```
legit = len(df[df.isFraud == 0])
fraud = len(df[df.isFraud == 1])
legit_percent = (legit / (fraud + legit)) * 100
fraud_percent = (fraud / (fraud + legit)) * 100

print("Number of Legit transactions: ", legit)
print("Number of Fraud transactions: ", fraud)
print("Percentage of Legit transactions: {:.4f} %".format(legit_percent))
print("Percentage of Fraud transactions: {:.4f} %".format(fraud_percent))
```

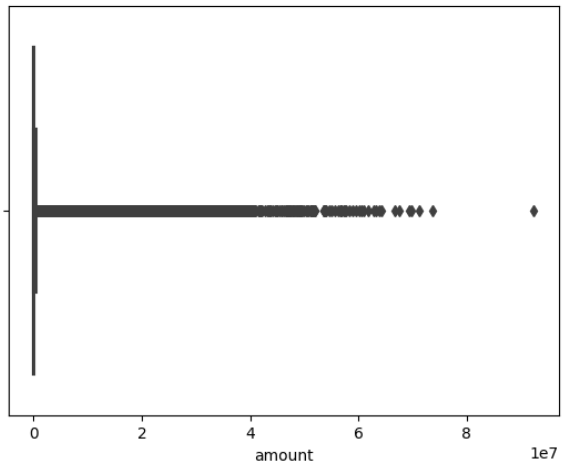
```
Number of Legit transactions:  6354407
Number of Fraud transactions:  8213
Percentage of Legit transactions: 99.8709 %
Percentage of Fraud transactions: 0.1291 %
```

The histogram below shows the counts for both classes, the column where isFraud is nearly invisible because it only represents 0.1291% of the data.

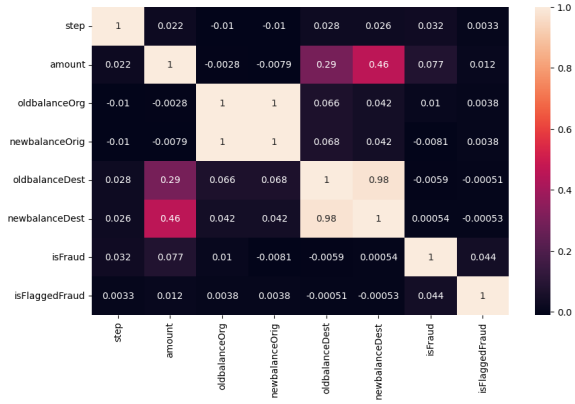


To work around this imbalance, we will use an oversampling strategy. Where we will resample the minority class randomly with replacement, to match the size of the majority class, so that we have a 50/50% ratio among both classes.

The amount variable is of particular interest since it represents the funds involved in a financial transaction. The boxplot below shows the amounts found in the dataset, and we can see we have a quite large number of outliers.



Below we show the correlation matrix to analyze which features have stronger linear relationships with the target variable.



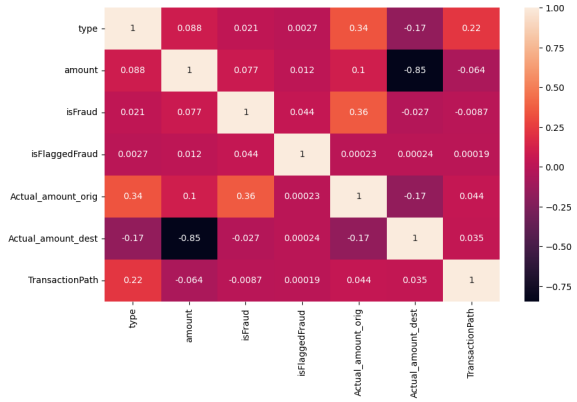
2. Data Preprocessing

After further analysis I found that the variables “oldbalanceOrigin” and “newbalanceOrigin”, as well as “oldBalanceDest” and “newBalanceDest” have very high Variance Inflation Factors. To overcome this limitation I created two new features, “actualBalanceOrigin”, and “actualBalanceDest”. After this process of feature engineering[7] I proceed to drop the following columns 'oldbalanceOrg', 'newbalanceOrig', 'oldbalanceDest', 'newbalanceDest', 'step','nameOrig', and 'nameDest'.

I reduced the number of columns from 11 to 7, after this, I computed the VIF scores shown below.

	variables	VIF
0	type	2.687803
1	amount	3.818902
2	isFraud	1.184479
3	isFlaggedFraud	1.002546
4	Actual_amount_orig	1.307910
5	Actual_amount_dest	3.754335
6	TransactionPath	2.677167

Here is the new correlation matrix, with the resulting dataset.



3. Data Splitting

After normalizing the data, I split the data 70/30, the former being the training set. Here is the resulting shape of the train and test sets.

```
Shape of X_train:  (4453834, 6)
Shape of X_test:  (1908786, 6)
```

Since it is known that this dataset is highly imbalanced, below I print the counts of legit and fraud transactions in the training set, which when summed are equivalent to 4,453,834 which represents 70% of the data.

```
Counts before oversampling minority class:

0    4448056
1       5778
Name: isFraud, dtype: int64
```

Under this condition, even a dummy classifier would perform with very high accuracy. So to overcome the imbalance problem, I proceeded to use an oversampling technique as shown below.

```
Counts after oversampling:

0    4448056
1    4448056
Name: isFraud, dtype: int64
```

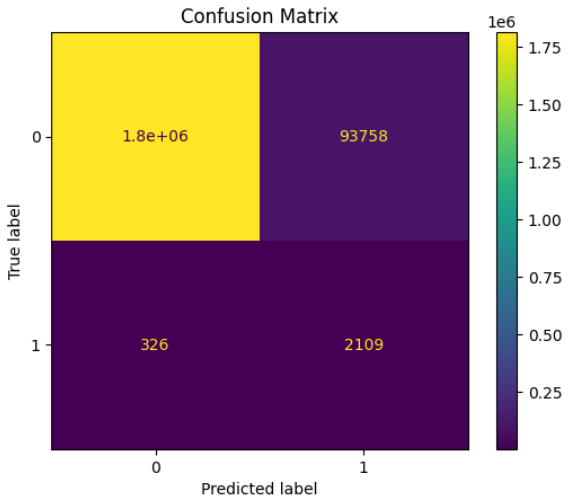
3. Data Modeling

After processing and splitting the data into train and test sets, I proceeded to train a logistic regression model.

```
# LOGISTIC REGRESSOR
y_train = upsampled.isFraud
X_train = upsampled.drop('isFraud', axis=1)

logistic_regressor = LogisticRegression(solver='liblinear', random_state=0)
logistic_regressor.fit(X_train, y_train)

y_pred_lr = logistic_regressor.predict(X_test)
logistic_regressor_score = logistic_regressor.score(X_test, y_test) * 100
```

The resulting classification report is as follows:

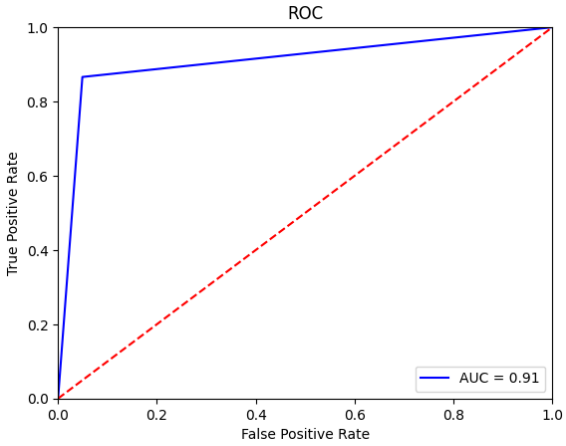
```
Classification Report              precision    recall  f1-score   support

           0       1.00      0.95      0.97   1906351
           1       0.02      0.87      0.04      2435

    accuracy                           0.95   1908786
   macro avg       0.51      0.91      0.51   1908786
weighted avg       1.00      0.95      0.97   1908786
```



EVALUATION

The AUC refers to the area under the curve, where an excellent model has an AUC near 1 which means it has a good measure of separability. The closer a model's AUC comes to 0 the worse measure of separability between classes.

Our model scores an AUC of 0.91 which is considered to be a great score.



A multiclass classification problem in machine learning can typically be broken down into numerous binary classification tasks. Each binary classification issue divides the target objects into two groups, each of which is classified into a different class (or category). The binary classification and multiclass classification prediction results and actual outcomes are compiled in a table called the confusion matrix. Accuracy, precision, recall, and F1 score are four widely used metrics that are typically used to assess the success of machine learning. These measures are based on confusion matrices. One of these, accuracy, can intuitively indicate the outcome of the forecast. When accuracy is insufficient to adequately reflect the specifics of the evaluation outcomes, precision and recall are useful complementing metrics.

These are the metrics we will be using to assess the success rate of this project.

DISCUSSION

The project is expected to be completed in approximately 4 weeks' time. I plan on getting familiar with the dataset and performing exploratory analysis, then selecting the features to be used for training the model. After tidying the dataset, and trying out different classifiers and architectures, I will select the best-performing model. After this, I will write the final report stating all my findings.

In regards to the potential challenges we will be facing, one of the biggest issues is that the dataset I will be using is very large, the .csv file is 493.53 MB in size. The scale is a challenge since processing and training can take quite a lot of time to complete, also we can see that this dataset is highly imbalanced. The dataset contains approximately 6.3 million observations with 11 columns, but approximately 0.998709 are legitimate transitions, and fraudulent transactions represent only 0.001291 of the data.

I will follow an oversampling strategy to overcome this imbalance in the data, and oversample the minority class to match the count of the majority class, so that we have a 50/50 ratio of both classes.

CONCLUSION

Fraud is a big problem in today's world, and machine learning can help mitigate the negative impact it creates on businesses.

Here we have implemented a model that performs really well, further exploration with other models such as random forests, boosting machines, or neural networks could be interesting, to compare the performance of those models against our regressor.

One of the greatest advantages of these models is that, once a final architecture has been chosen, the weights of the models can be saved, and the model can be deployed to any financial or e-commerce back-end, and it can help classify legitimate and fraudulent transactions. What's more, the data produced on these platforms can be collected, processed, and tidied to feed it back to the model periodically, so it can continue learning.

Fraudsters are always finding innovative ways to abuse and brake services for their own interests. And Machine learning models can identify and learn these patterns.

REFERENCES

[1]Lopez Rojas, E. (2022, August 10). *Synthetic Financial Datasets For Fraud Detection*. Kaggle. Retrieved February 6, 2023, from https://www.kaggle.com/datasets/ealaxi/paysim1?datasetId=1069&sortBy=voteCount

[2]Hemery, B., & Pasquet, M. (2013, May). *Synthetic Logs Generator for Fraud Detection in Mobile Transfer Services*. RESEARCH GATE. https://www.researchgate.net/publication/236156069_Synthetic_Logs_Generator_for_Fraud_Detection_in_Mobile_Transfer_Services

[3]Lopez Rojas, E. A. (September 2016). *PAYSIM: A FINANCIAL MOBILE MONEY SIMULATOR FOR FRAUD DETECTION*. PAYSIM: A FINANCIAL MOBILE MONEY SIMULATOR FOR FRAUD DETECTION. https://www.researchgate.net/publication/313138956_PAYSIM_A_FINANCIAL_MOBILE_MONEY_SIMULATOR_FOR_FRAUD_DETECTION

[4]Lopez Rojas, E. A., & Axelsson, S. (2018, January). *Analysis of fraud controls using the PaySim financial simulator*. Research Gate. https://www.researchgate.net/publication/326615166_Analysis_of_fraud_controls_using_the_PaySim_financial_simulator

[5]Lopez Rojas, E. A., & Berneaud, C. (2019, July). *Advantages of the PaySim Simulator for Improving Financial Fraud Controls*. Research Gate. https://www.researchgate.net/publication/334301755_Advantages_of_the_PaySim_Simulator_for_Improving_Financial_Fraud_Controls

[6]Rieke, R., & Giot, R. (2013, September). *Fraud Detection in Mobile Payment Utilizing Process Behavior Analysis*. Research Gate. https://www.researchgate.net/publication/