Synthetic Fraud Modeling

Problem Statement

There is a lack of public financial data for professionals to collaborate and share fraud insights amongst each other. The purpose of this experiment is to see whether or not modeling around synthetically derived data is viable in the industry and can potentially lead to more public collaboration to combat fraud. Additionally, we want to show that we can achieve a precision of >90% and a recall of >80% with this specific dataset.

Dataset

The dataset is a simulated set of mobile money transactions from real transactions from Africa. The company is a mobile financial service company that is operating in more than 14 countries around the world. The data can be found here.

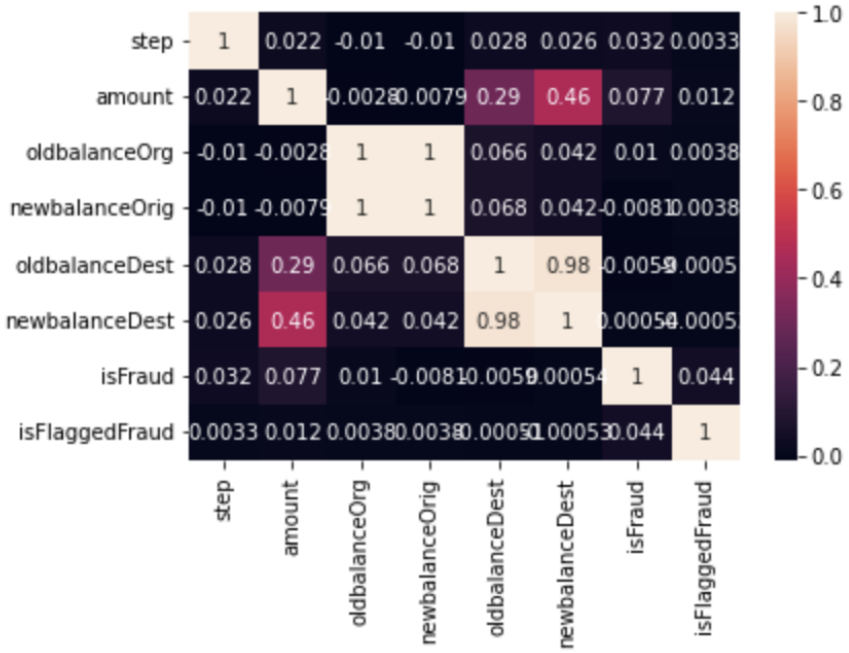
Data Cleansing and Feature Engineering

The dataset contained the following columns below:

Feature	Description
step	Unit of time in the real world. 1 step is 1 hour
amount	Dollar amount of the transaction
type	Type of transaction
nameOrig	ID for originating account
nameDest	ID for destination account
oldBalanceOrig	Originating account balance pre transaction
newBalaneOrig	Originating account balance post transaction
oldBalanceDest	Destination account balance pre transaction
newBalanceDest	Destination account balance post transaction

isFraud	Fraud flag for account takeover and/or emptying funds
isFlaggedFraud	Fraud flag for transactions for attempting to transfer more than $200,000

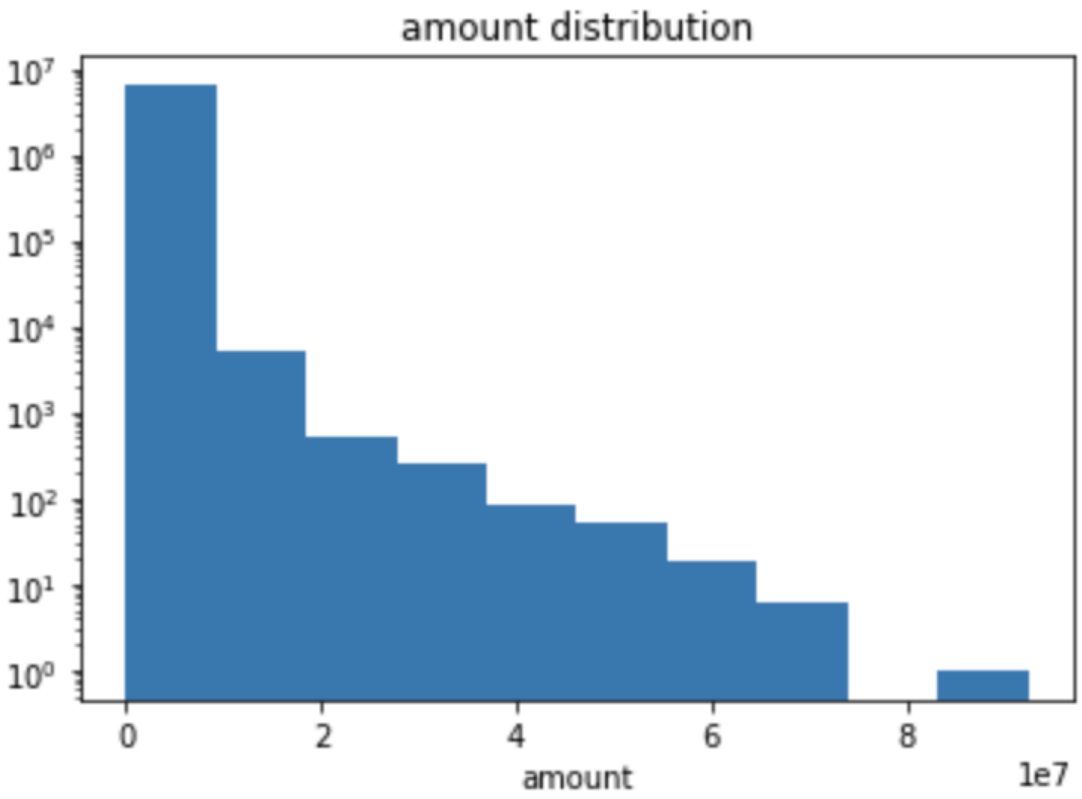
There was not much data cleansing or wrangling needed as the dataset contained no missing values.



After doing some initial analysis, I decided that the features to model around for the isFraud flag would be the amount, old and new balance origination/destination as well as the transaction type.

Exploratory Data Analysis

As previously mentioned, the preferred target variable for this dataset is isFraud. The reason why it is not isFlaggedFraud is because there are only 14 transactions that are classified with that label and they also have the isFraud flag of 1. The amount feature is the highest performing feature with a correlation of 0.077 to the isFraud flag.



When investigating the distribution of the amount feature, we can see the distribution is decently right-skewed. One thing that stood out the most was analyzing the mean and standard deviation of the amount field.

isFraud	count	mean	std
0	6354407.00	178197.04	596236.98
1	8213.00	1467967.30	2404252.95

As you can see above, the mean of the amount for the isFraud label is actually $1,467,967 and the standard deviation being $2,404,252. Additionally, the fraud rate for this simulated dataset is only 0.1%

which means that this is a heavily imbalanced dataset which will require some form of oversampling or undersampling to compensate.

When investigating the amount field even more, I was able to discover that 3.5% of fraudulent transactions have an amount of greater than 10 million dollars, which is the reason for such an elevated mean value. We will eventually have to scale and normalize this field so that it doesn't affect the modeling too much.

Modeling

There are four approaches I took to modeling around this dataset - Logistic Regression, Random Forest, Random Forest RandomSearchCV, and Oversampling/Undersampling.

The worst performer of the four approaches was logistic regression as seen below.

OLS Regression Results

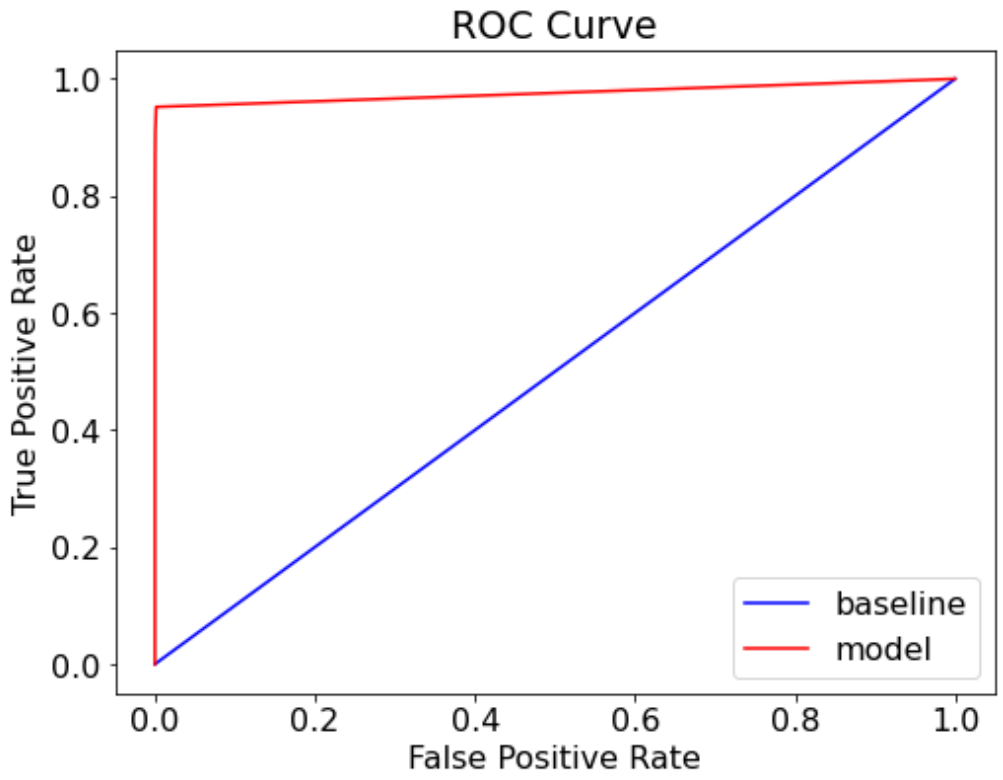
Dep. Variable:	isFraud	**R-squared:**	0.181
Model:	OLS	**Adj. R-squared:**	0.181
Method:	Least Squares	**F-statistic:**	1.169e+05
Date:	Sat, 04 Feb 2023	**Prob (F-statistic):**	0.00
Time:	00:00:45	**Log-Likelihood:**	9.5851e+06
No. Observations:	4771965	**AIC:**	-1.917e+07
Df Residuals:	4771955	**BIC:**	-1.917e+07
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.0027	3.39e-05	80.817	0.000	0.003	0.003
amount	9.452e-09	4.92e-11	192.153	0.000	9.36e-09	9.55e-09
oldbalanceOrg	1.202e-07	1.22e-10	985.088	0.000	1.2e-07	1.2e-07
newbalanceOrig	-1.203e-07	1.22e-10	-983.622	0.000	-1.21e-07	-1.2e-07
oldbalanceDest	6.8e-09	3.59e-11	189.540	0.000	6.73e-09	6.87e-09
newbalanceDest	-6.908e-09	3.56e-11	-193.937	0.000	-6.98e-09	-6.84e-09
CASH_IN	0.0157	5.03e-05	311.807	0.000	0.016	0.016
CASH_OUT	-0.0045	3.93e-05	-113.932	0.000	-0.005	-0.004
DEBIT	-0.0029	0.000	-18.880	0.000	-0.003	-0.003
PAYMENT	-0.0036	3.97e-05	-91.110	0.000	-0.004	-0.004
TRANSFER	-0.0019	5.68e-05	-34.087	0.000	-0.002	-0.002

Omnibus:	11692452.746	**Durbin-Watson:**	1.999
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	116920220497.490
Skew:	26.476	**Prob(JB):**	0.00
Kurtosis:	768.004	**Cond. No.**	2.45e+19

As you can see the overall R-Squared was only 0.181, which indicates that a logistic regression model is not the best model for this dataset.

The next model I wanted to experiment was with a random forest classifier. After utilizing the bare minimum parameters, I plotted the ROC AUC curve and you can see that I ended up with a 0.99. After seeing these numbers, it's clear that utilizing the ROC AUC as a performance metric did not make sense.



So instead of using the ROC AUC, let's take a look at the classification matrix and see what numbers we find there.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1588610
1	0.96	0.78	0.86	2045
accuracy			1.00	1590655
macro avg	0.98	0.89	0.93	1590655
weighted avg	1.00	1.00	1.00	1590655

(Random Forest)

After investigating these numbers, we can see that the recall and f1-score are numbers that we want to improve moving forward. Since we know that this is an imbalanced dataset, we should experiment with oversampling and undersampling the dataset to improve the recall and f1-score.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1588353
1	1.00	1.00	1.00	1588851
accuracy			1.00	3177204
macro avg	1.00	1.00	1.00	3177204
weighted avg	1.00	1.00	1.00	3177204

(SMOTE)

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1587074
1	0.98	0.80	0.88	2058
accuracy			1.00	1589132
macro avg	0.99	0.90	0.94	1589132
weighted avg	1.00	1.00	1.00	1589132

(NeighbourhoodCleaningRule)

The first classification matrix is a result of SMOTE oversampling. We can see that oversampling the file to a 50/50 label distribution does not produce a good result. The second classification matrix is an undersampling technique called Neighbourhood Cleaning Rule. This technique essentially removes noisy samples from the dataset through the nearest neighbors method.

Final Thoughts

Based on the model results, under-sampling the dataset through the Neighbourhood Cleaning Rule method produced the most favorable results of 0.98, 0.90, 0.88 precision, recall and f1-scores, respectively.

I do think that if more modeling were to be done, a SMOTE Random Forest where the distribution between good and fraud transactions is 70/30 or 80/20 would produce the highest precision, recall and f1-score metrics.