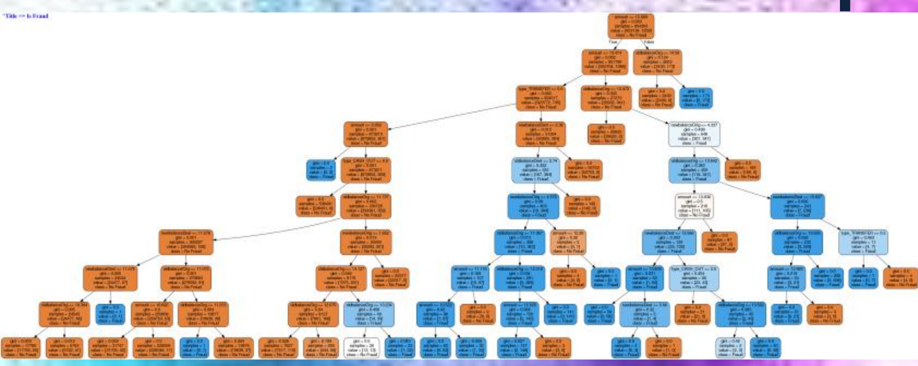
Predict a fraud using data of mobile money transactions:

Using Machine Learning algorithms.

Full code in: https://github.com/Thaleia18



The prediction task is to predict if the transaction is a fraud using the transaction information.

We will create our models using a synthetic dataset of mobile money transactions.

This dataset is scaled down 1/4 of the original dataset which is presented in "PaySim: A financial mobile money simulator for fraud detection". https://www.kaggle.com/ntnu-testimon/paysim1

The machine learning algorithms that I used are:

- Decision tree.
- K Neighbors.
- Random Forest.
- Logistic regression.

THE DATA

This data was extracted from: https://www.kaggle.com/ntnu-testimon/paysim1

Attributes:

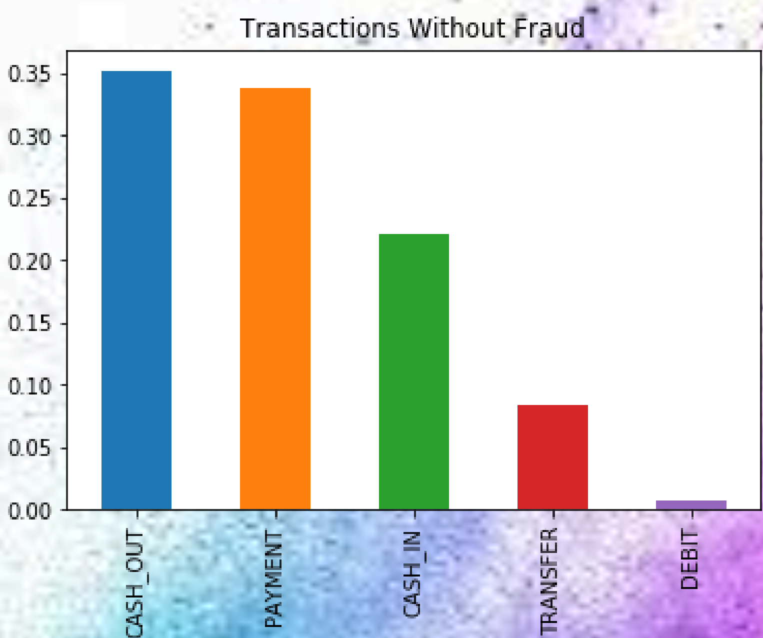
- step (numerical): Unit of time in the real world. 1 step is 1 hour.
- type (categorical): CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER
- amount (numerical): amount of the transaction
- nameOrig: customer who started the transaction
- oldbalanceOrg (numerical): initial balance before the transaction
- newbalanceOrig (numerical): customer's balance after
- nameDest: recipient ID of the transaction.
- oldbalanceDest (numerical): initial recipient balance before the transaction.
- newbalanceDest (numerical): recipient's balance after
- isFraud (boolean): identifies a fraudulent transaction (1) and non fraudulent (0)
- isFlaggedFraud (boolean): flags illegal attempts to transfer more than 200.000 in a single transaction.

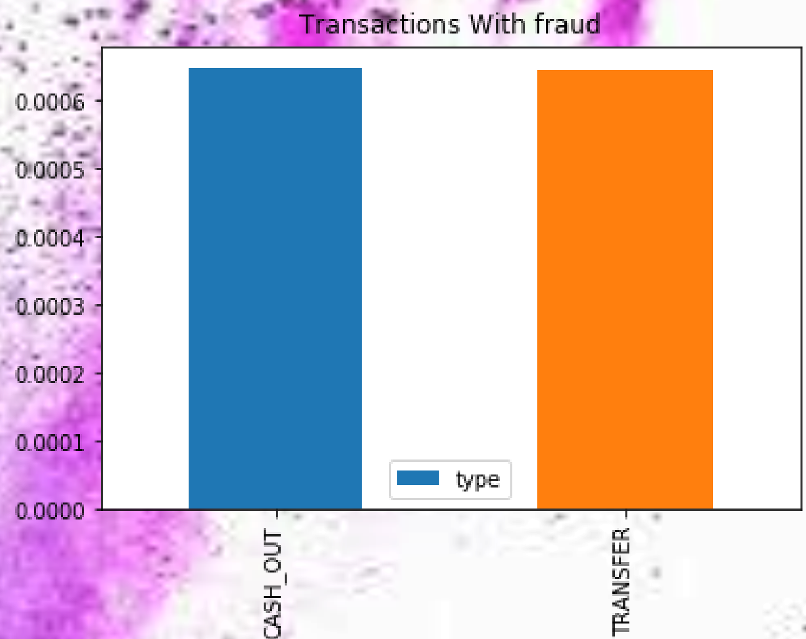
Number of rows: 6.362620e+06

SOME DATA VISUALIZATIONS.

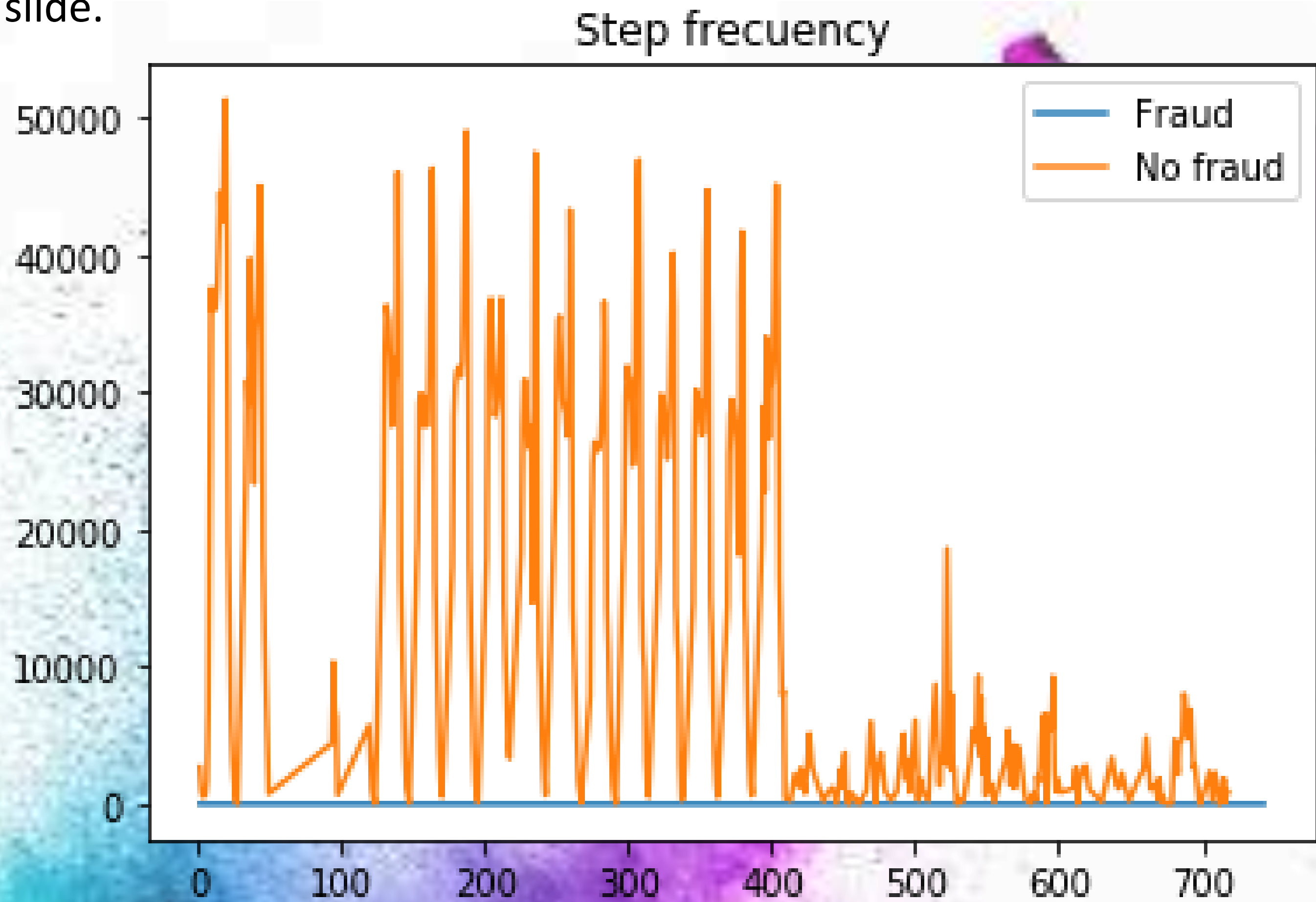
Plot of % of type of transactions with and without fraud.

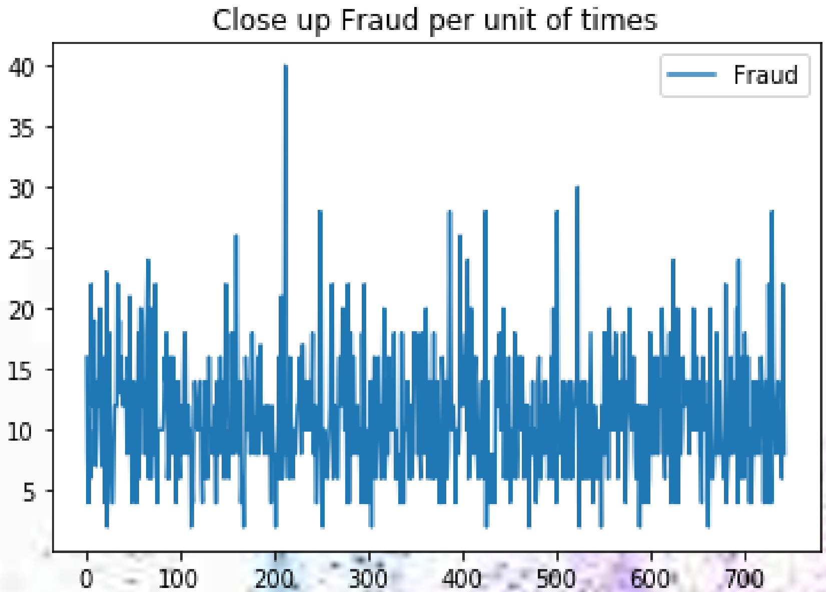
From the total transactions, just 0.12% were Fraud. This 0.12% is divided in 0.0647% in Cash out and 0.0644% Transfer.





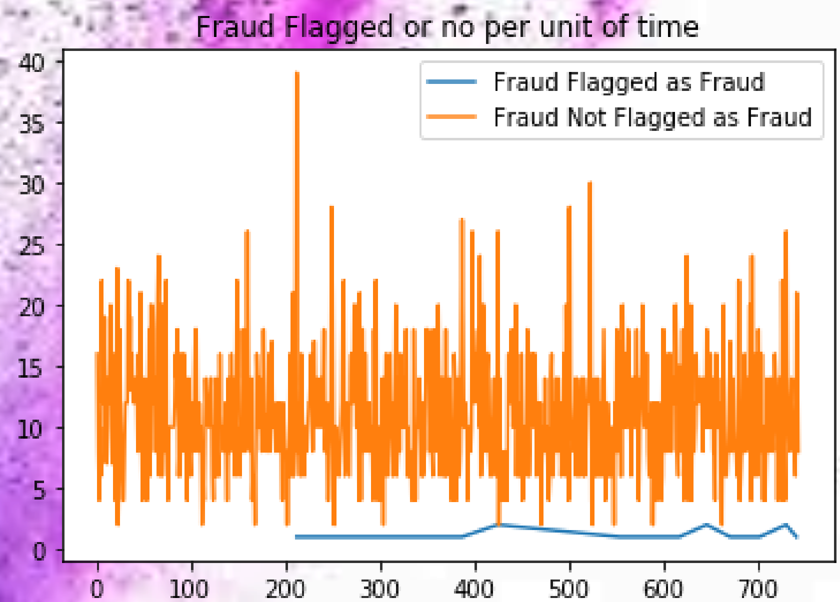
Amount of transactions with and without per unit of time. We can observe that the amount of transactions with fraud was minimal. I will do a close up to these transactions in the next slide.



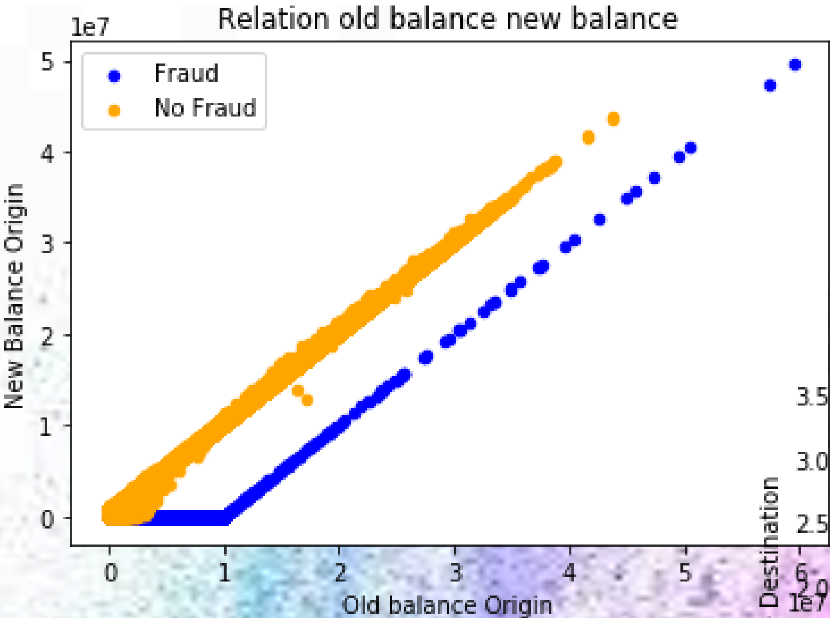


These are the transactions with fraud per unit of time. There is a max of 40 frauds at the time unit 220.

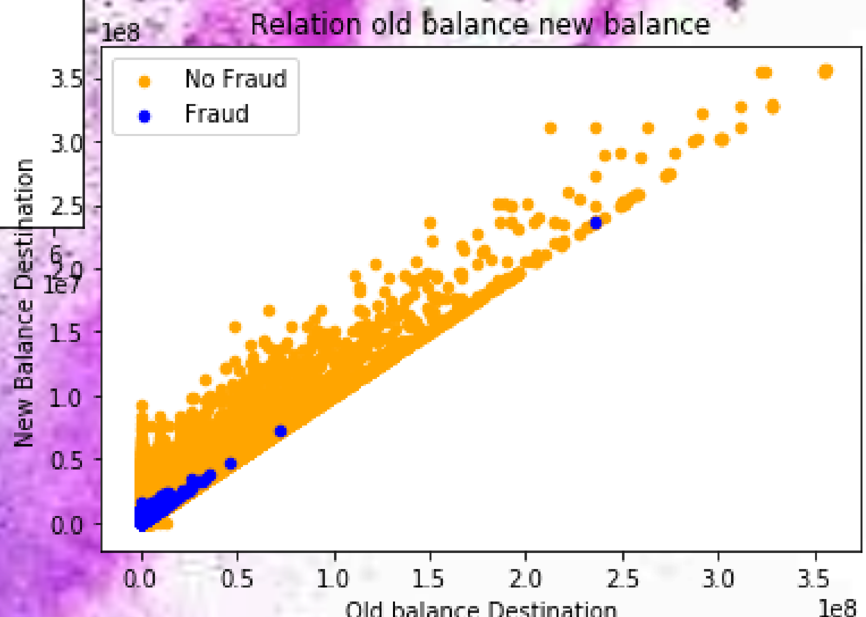
From the transactions with fraud, these are the amount of flagged fraud per unit of time.

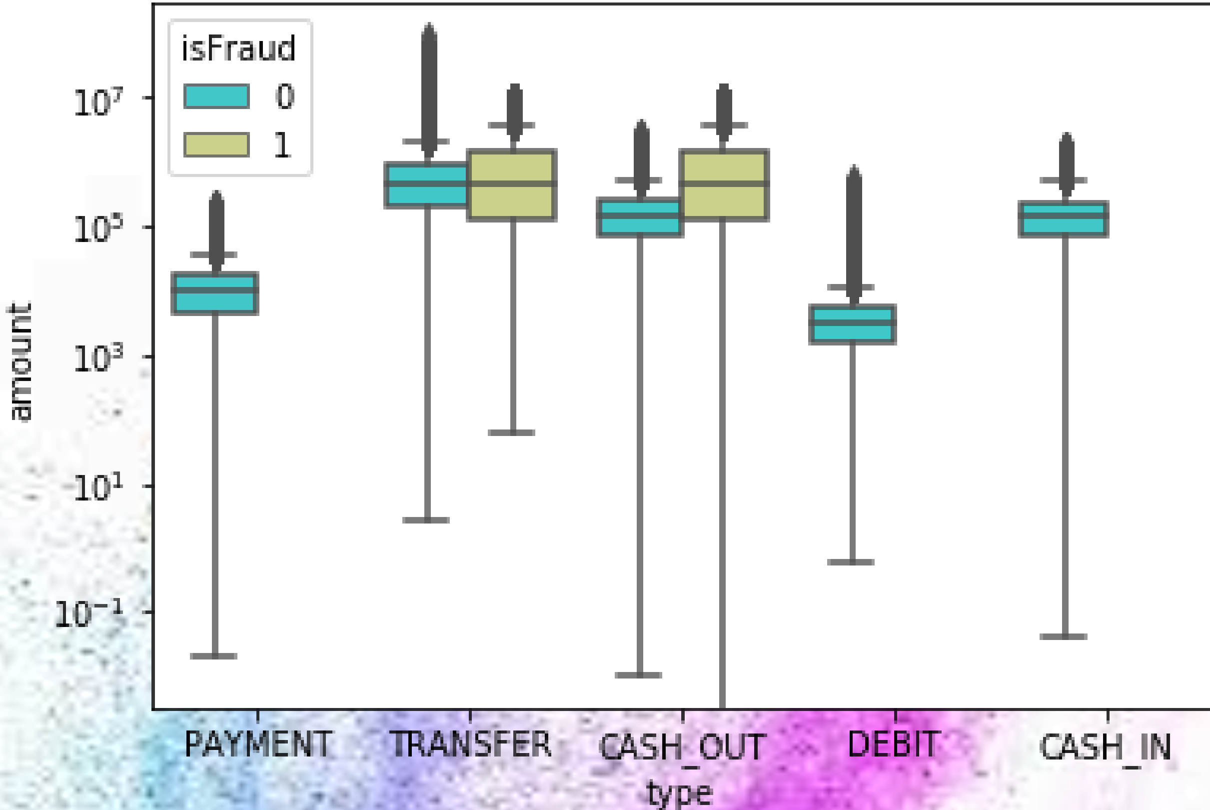


These are the plots of Old balance vs New balance for Origin and Destin accounts.



We can observe that a lot of the Origin accounts have New Balance zero.





Boxplot for the amount transaction for different transactions. Green boxes for transactions with fraud.

I decided work with the features:
features = ['amount','oldbalanceOrg', 'newbalanceOrig', 'type, 'oldbalanceDest', 'newbalanceDest', 'isFraud']

After transform the categorical features in dummy variables, my data looks like:

	amount	oldbalanceOrg	newbalanceOrig	type_CASH_IN	type_CASH_OUT	type_DEBIT	type_PAYMENT	type_TI
count	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.3626
mean	1.798619e+05	8.338831e+05	8.551137e+05	2.199226e-01	3.516633e-01	6.511783e-03	3.381461e-01	8.3756
std	6.038582e+05	2.888243e+06	2.924049e+06	4.141940e-01	4.774895e-01	8.043246e-02	4.730786e-01	2.7702
min	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.0000
25%	1.338957e+04	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.0000
50%	7.487194e+04	1.420800e+04	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.0000
75%	2.087215e+05	1.073152e+05	1.442584e+05	0.000000e+00	1.000000e+00	0.000000e+00	1.000000e+00	0.0000
max	9.244552e+07	5.958504e+07	4.[illegible]					

pe_CASH_IN	type_CASH_OUT	type_DEBIT	type_PAYMENT	type_TRANSFER	oldbalanceDest	newbalanceDest	isFraud
362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06
199226e-01	3.516633e-01	6.511783e-03	3.381461e-01	8.375622e-02	1.100702e+06	1.224996e+06	1.290820e-03
141940e-01	4.774895e-01	8.043246e-02	4.730786e-01	2.770219e-01	3.399180e+06	3.674129e+06	3.590480e-02
000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	1.327057e+05	2.146614e+05	0.000000e+00
000000e+00	1.000000e+00	0.000000e+00	1.000000e+00	0.000000e+00	9.430367e+05	1.111909e+06	0.000000e+00
000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	3.560159e+08	3.561793e+08	1.000000e+00

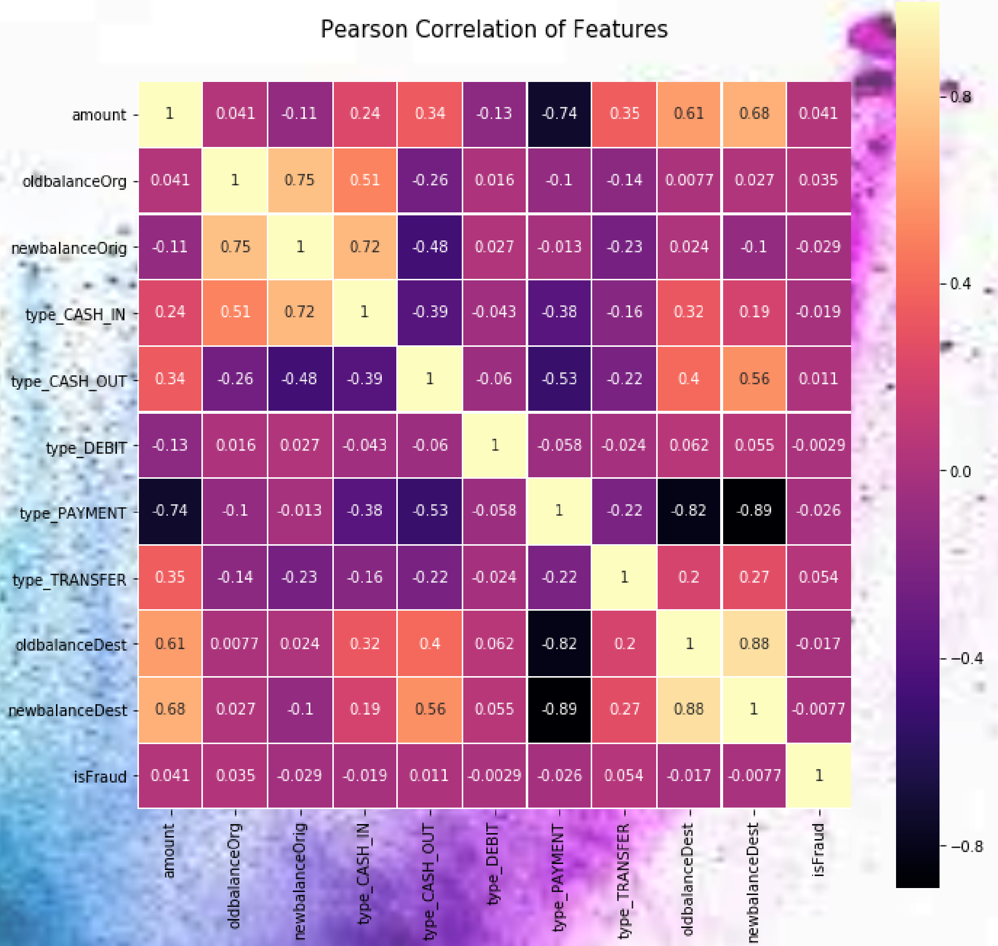
I don’t have the computer power to work with the full data set of rows: 6.362620e+06.

I decided to study what is the minimum sample size that reflects the same correlation of the features with the fraud column than using the total data.

Here is a table for the correlations using different percentages of the total data. @0% and 50% got similar values than using the 100%.

	0.01	0.05	0.1	0.2	0.5	1.0
amount	0.043840	0.037422	0.040506	0.039556	0.041360	0.040640
oldbalanceOrg	0.035987	0.033658	0.034307	0.033930	0.034810	0.034560
newbalanceOrig	-0.030817	-0.029145	-0.028601	-0.028499	-0.028762	-0.028760
type_CASH_IN	-0.019222	-0.019071	-0.019038	-0.018941	-0.019147	-0.019089
type_CASH_OUT	0.014733	0.008941	0.011984	0.010499	0.010927	0.011256
type_DEBIT	-0.002750	-0.002845	-0.002902	-0.002893	-0.002928	-0.002911
type_PAYMENT	-0.026060	-0.025793	-0.025638	-0.025465	-0.025766	-0.025697
type_TRANSFER	0.048832	0.058050	0.052264	0.054542	0.054651	0.053869
oldbalanceDest	-0.014593	-0.018829	-0.018106	-0.018672	-0.017640	-0.017281
newbalanceDest	-0.003317	-0.010464	-0.007173	-0.008556	-0.007982	-0.007659
isFraud	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

The correlation of the features with the isFraud column using 20% of the data:

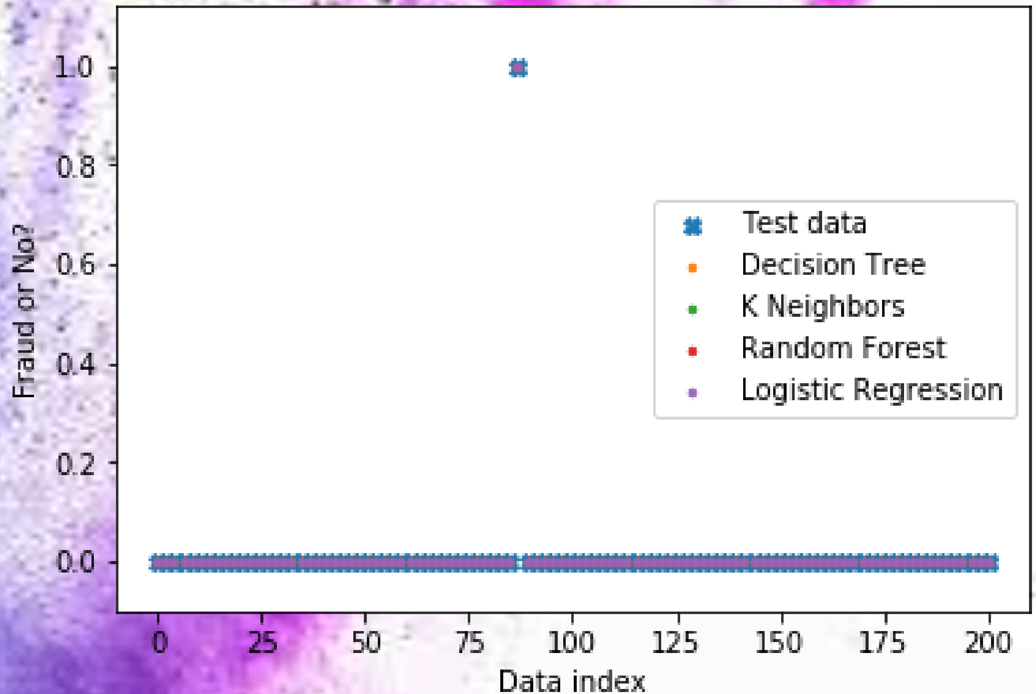


MODEL RESULTS

	DecisionTree	LogisticRegression	RandomForest	KNeighbors
Accuracy:	0.999639	0.999330	0.999667	0.999632
Precision:	0.956113	0.881423	0.977707	0.950156
Recall:	0.751232	0.549261	0.756158	0.751232
F1:	0.841379	0.676783	0.852778	0.839065

This a plot of a sample of 200 predictions .

I am comparing the results of different methods with the validation data.



Visualization of the decision tree ☺