Fraud Transaction Detection

Alumna :Lucia Correa
Docentes : Yonatan Alvarez, Enrique Revuelta

General Context

Fraud is an unauthorized activity taking place in electronic payments systems, but these are treated as illegal activities. Fraud detection methods are continuously developed to defend criminals in adapting to their strategies.

In recent years we have seen a huge increase in Fraud attempts, making fraud detection important as well as challenging. Despite countless efforts and human supervision, **hundreds of millions** are lost due to **fraud**. Fraud can happen using various methods ie, stolen credit cards, misleading accounting, phishing emails, etc.

Goals

- **Payments fraud Dataset**
  - Data input
  - Cleaning
  - Analysis
  - Metrics
  - Dashboard

- **Machine Learning as potential solution**
  - LogReg
  - XGBC

- **Conclusions**

Data Input

Type: CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER.

isFraud: This is the transactions made by the fraudulent agents inside the simulation.

Step: maps a unit of time in the real world. In this case 1 step is 1 hour of time.

isFlaggedFraud: The business model aims to control massive transfers and flags illegal attempts.

OldbalanceDest: initial balance recipient before the transaction.

NewbalanceDest: new balance recipient after the transaction.

NameDest: customer who receives the transaction.

Amount: amount of the transaction in local currency

NameOrig: customer who started the transaction.

NewbalanceOrig: new balance after the transaction.

OldbalanceOrg: initial balance before the transaction.

step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest
1	PAYMENT	9839.64	C1231006815	170136.00	160296.36	M1979787155
1	PAYMENT	1864.28	C1666544295	21249.00	19384.72	M2044282225
1	TRANSFER	181.00	C1305486145	181.00	0.00	C553264065
1	CASH_OUT	181.00	C840083671	181.00	0.00	C38997010
1	PAYMENT	11668.14	C2048537720	41554.00	29885.86	M1230701703
1	PAYMENT	7817.71	C90045638	53860.00	46042.29	M573487274
1	PAYMENT	7107.77	C154988899	183195.00	176087.23	M408069119
1	PAYMENT	7861.64	C1912850431	176087.23	168225.59	M633326333
1	PAYMENT	4024.36	C1265012928	2671.00	0.00	M1176932104
1	DEBIT	5337.77	C712410124	41720.00	36382.23	C195600860

Data Cleaning & Transforming

- **Modified columns**

'nameOrig': 'name_Orig','
'newbalanceOrig': 'new_balance_Orig',
'oldbalanceOrg': 'old_balance_Org',
'oldbalanceDest': 'old_balance_Dest',
'newbalanceDest': 'new_balance_Dest',
'isFlaggedFraud': 'is_Flagged_Fraud',
'isFraud': 'is_Fraud'

- **Modified categorical values**

'CASH_IN' : 'CASH-IN'
'CASH_OUT': 'CASH-OUT'

- **Modified numerical values**

TRANSFER : 0
CASH_OUT: 1

- **Some more transforming**

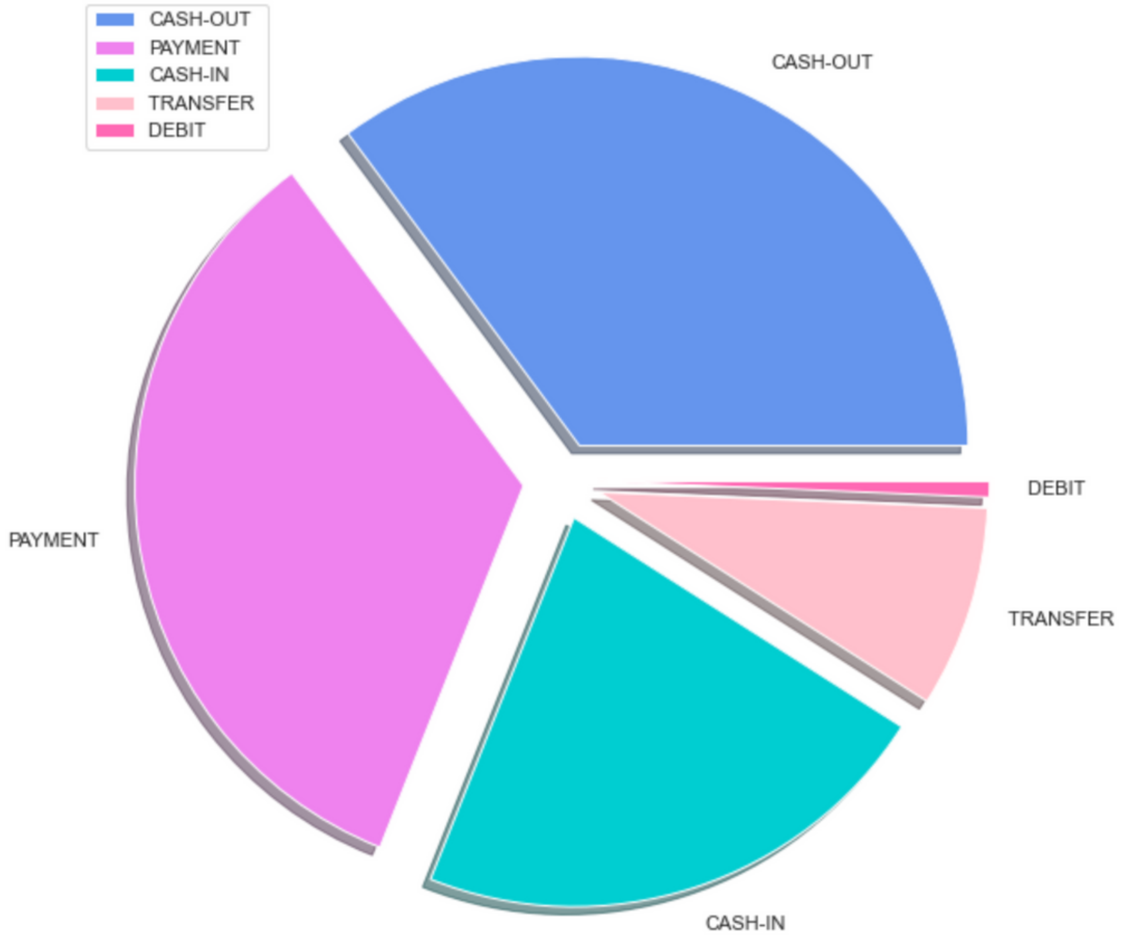
Filtered dataframe on type = Transfer and Cash-Out
Created new dataframe called = fraud_payments
Dropped columns = is_Fraud, name_Dest, name_Orig, step

Metrics

- Types of money transactions
- Descriptive Statistics
- Potential amount sensible to fraud
- Total number of frauds
- Number of transactions which are actual fraud
- Total amount lost due to frauds: Fraud vs Flagged fraud.

Analysis

Types of money transactions



Analysis

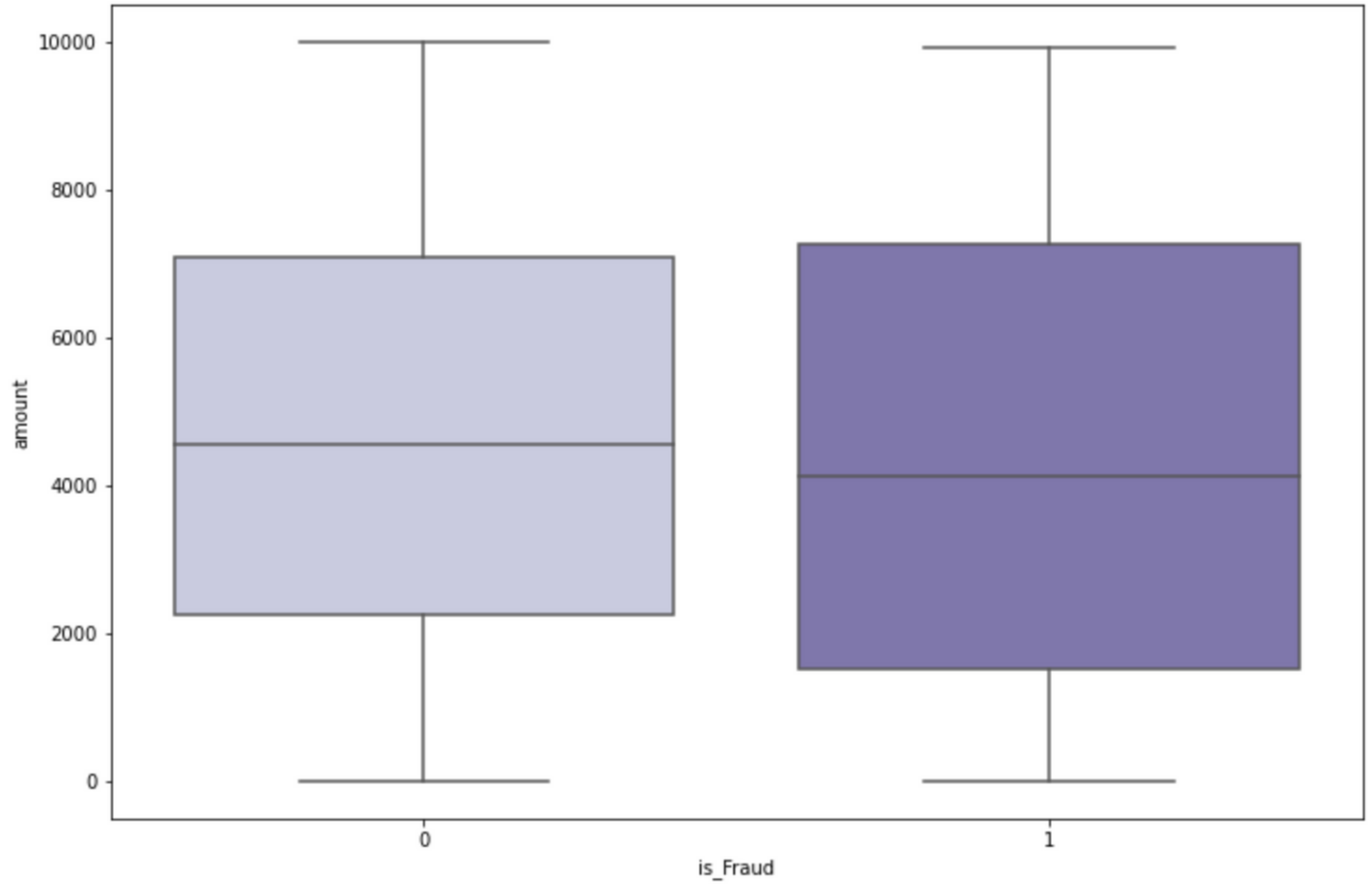
Descriptive Statistics

	step	amount	old_balance_Org	new_balance_Orig	old_balance_Dest	new_balance_Dest	is_Fraud	is_Flagged_Fraud
count	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06
mean	2.433972e+02	1.798619e+05	8.338831e+05	8.551137e+05	1.100702e+06	1.224996e+06	1.290820e-03	2.514687e-06
std	1.423320e+02	6.038582e+05	2.888243e+06	2.924049e+06	3.399180e+06	3.674129e+06	3.590480e-02	1.585775e-03
min	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	1.560000e+02	1.338957e+04	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
50%	2.390000e+02	7.487194e+04	1.420800e+04	0.000000e+00	1.327057e+05	2.146614e+05	0.000000e+00	0.000000e+00
75%	3.350000e+02	2.087215e+05	1.073152e+05	1.442584e+05	9.430367e+05	1.111909e+06	0.000000e+00	0.000000e+00
max	7.430000e+02	9.244552e+07	5.958504e+07	4.958504e+07	3.560159e+08	3.561793e+08	1.000000e+00	1.000000e+00
range	7.420000e+02	9.244552e+07	5.958504e+07	4.958504e+07	3.560159e+08	3.561793e+08	1.000000e+00	1.000000e+00
variation coefficient	5.847723e-01	3.357344e+00	3.463606e+00	3.419485e+00	3.088194e+00	2.999298e+00	2.781548e+01	6.306051e+02
skew	3.751769e-01	3.099395e+01	5.249136e+00	5.176884e+00	1.992176e+01	1.935230e+01	2.777954e+01	6.306036e+02
kurtosis	3.290706e-01	1.797957e+03	3.296488e+01	3.206698e+01	9.486741e+02	8.621565e+02	7.697030e+02	3.976591e+05

- All data in general has high kurtosis it is telling us that the dataset tend to have heavy tails .
- 25% of the new_balance_orig is 0.
- All data in general also have high skew which means is not symmetric.

Analysis

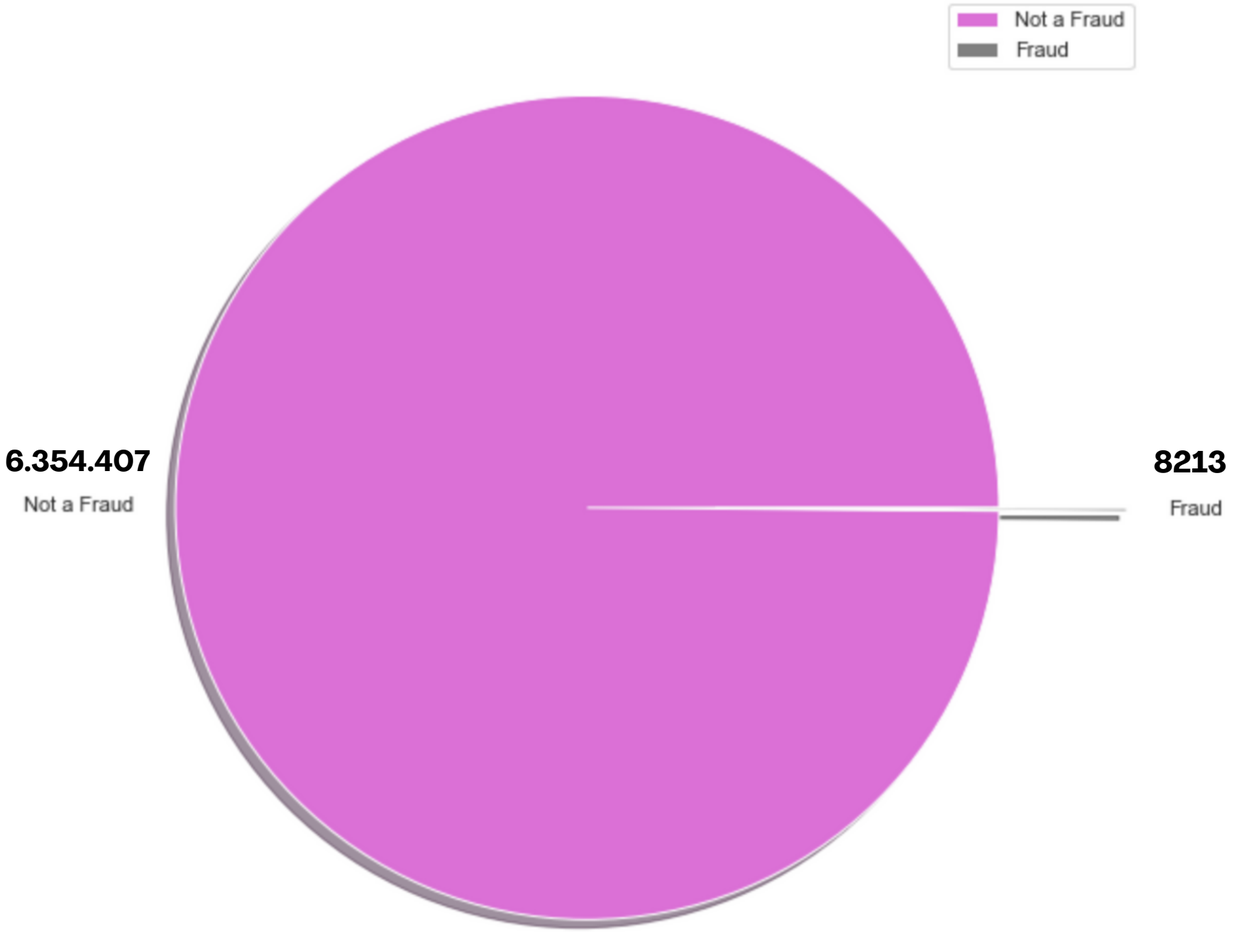
Potential amount sensible to fraud



- For payments lower than 1e5 we can observe that the data is asimetric for amounts where no fraud exist and the median is closer to low value amounts which is saying that fraud tend to ocurr quite often in smalls amount

Analysis

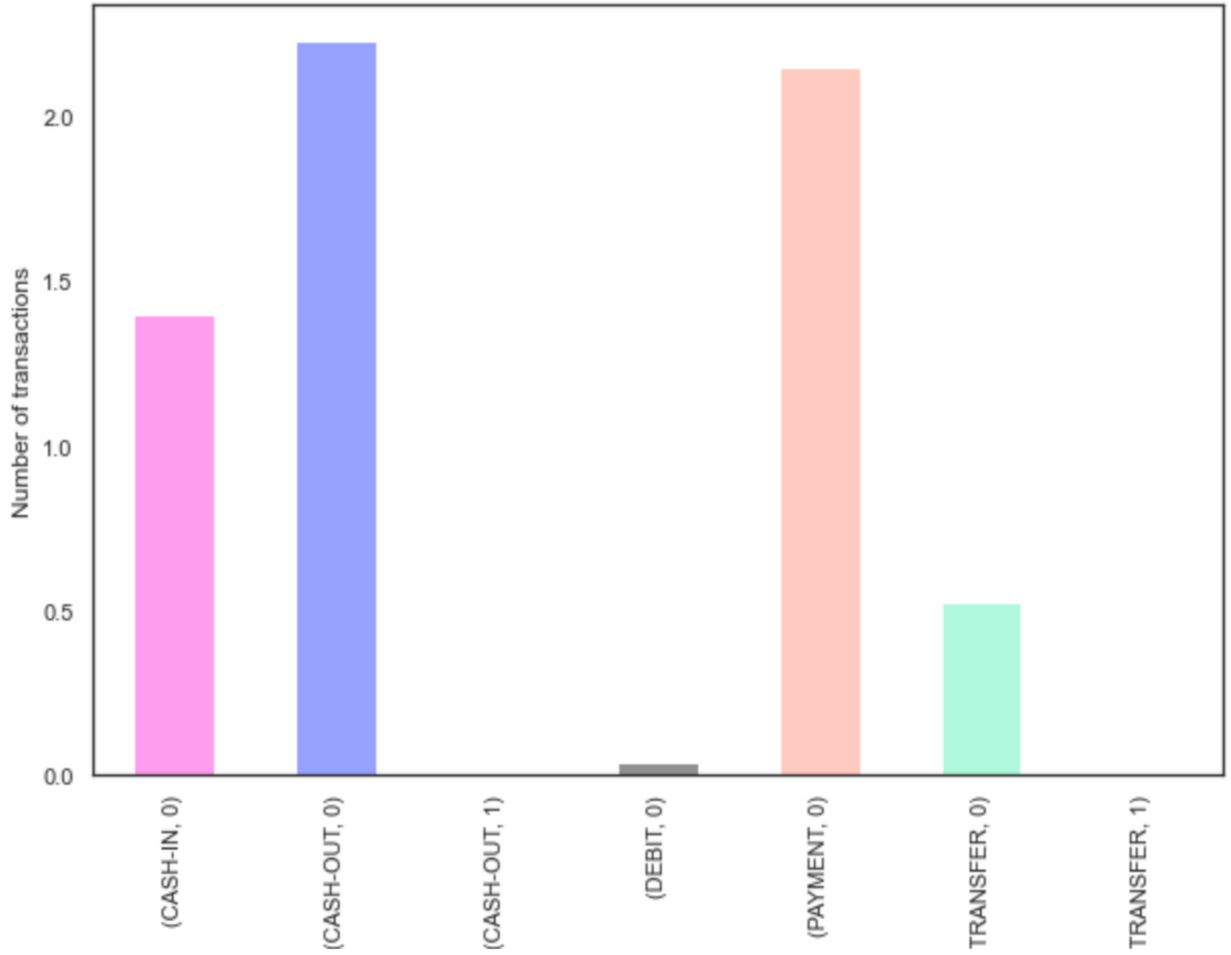
Total number of frauds



Analysis

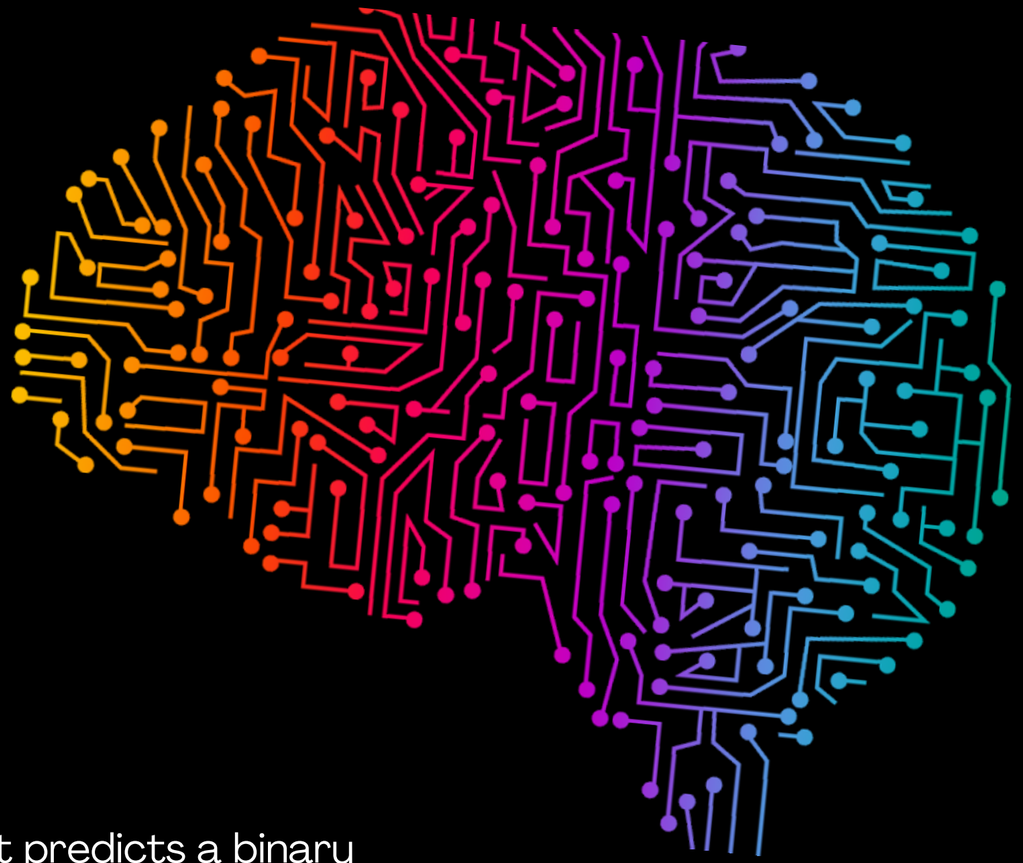
Number of transactions which are actual fraud

- 1 indicates fraud.
- 0 indicates no fraud.



Can anyone think how to solve this problem?

Machine Learning as potential solution



Two machine learning models where applied to try to predict fraud so fraudulent transactions can be detected in the future.

LogReg: Logistic regression is a classification algorithm that predicts a binary outcome based on a series of independent variables

XGBClassifier: is a tree based ensemble machine learning algorithm which is a scalable machine learning system for tree boosting.

LogReg and XGBC in action

In order for this models to work first I need to prepare de data. To do so, I split the data between:

- X = fraud_payments_copy.drop(columns=['is_Fraud', 'is_Flagged_Fraud', 'name_Orig', 'name_Dest', 'step'], axis=1)
- y = fraud_payments_copy['is_Fraud']
- data = pd.concat([X, y], axis=1)
- **Test Train Split**
- X_train, X_test, y_train, y_test=train_test_split(X, y, test_size=0.2)
- **Training and Predicting**

Results

	modelos	acc_score	prec_score	rec_score	f1_score
0	LogReg	0.998465	0.673786	0.7139[illegible]2	0.693307
1	xgbc	0.999345	0.965879	0.757202	0.848904

Conclusions

- Existing rule-based system is **not capable** of detection of all the fraud transaction.
- **Machine learning** can be used for the detection of fraud transaction.
- Predictive models produce **good precision score** and are capable of detection of fraud transaction.

Thanks for your attention!