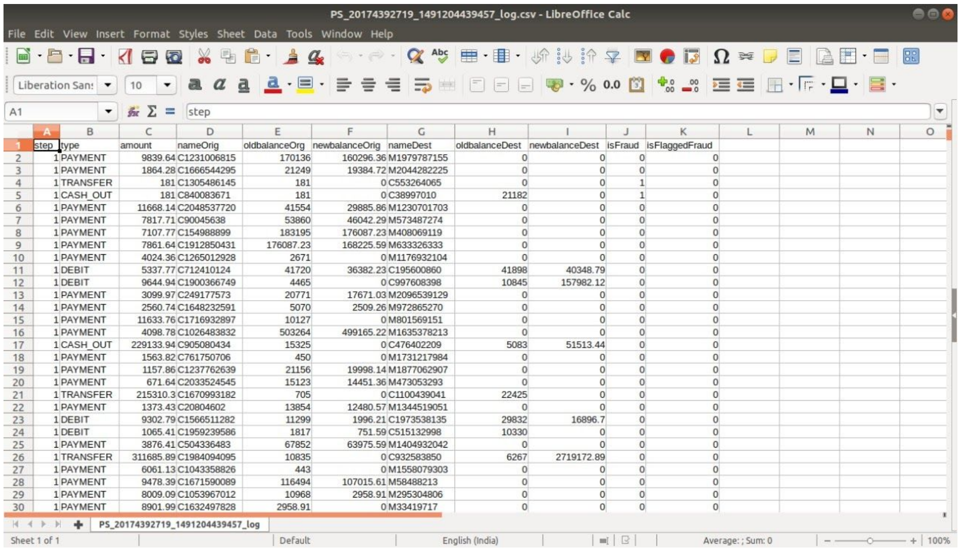
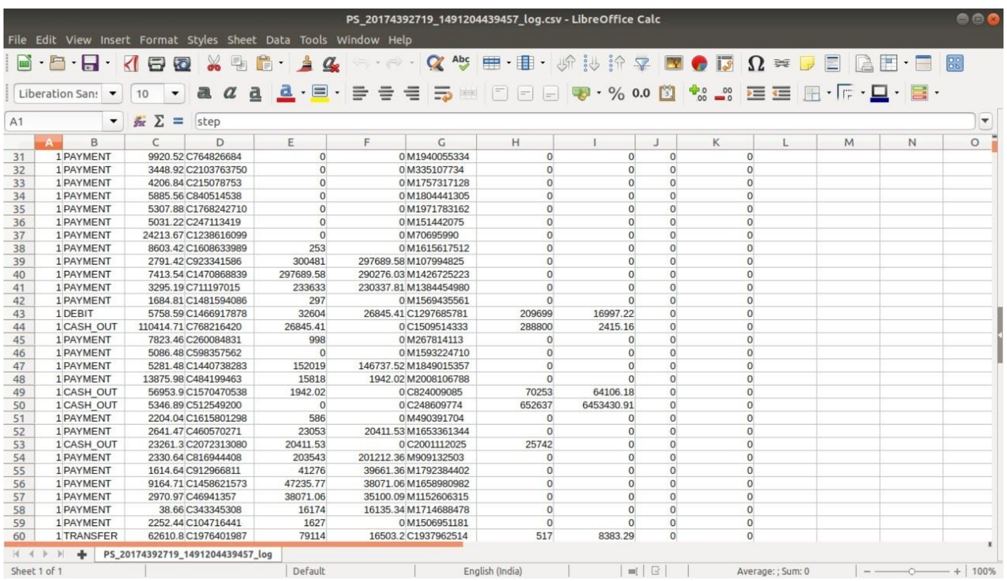
Test Data Set Availability with Test Results

1. Data Set downloaded from Kaggle :
    1.1. Link : https://www.kaggle.com/ntnu-testimon/paysim1
2. Data Set Screenshots :
    2.1.



1	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
2	1	PAYMENT	9839.64	C1231006815	170136	160296.36	M1979787155	0	0	0	0
3	1	PAYMENT	1864.28	C1666544295	21249	19384.72	M2044282225	0	0	0	0
4	1	TRANSFER	181	C1305486145	181	0	C553264065	0	0	1	0
5	1	CASH_OUT	181	C840083671	181	0	C38997010	21182	0	1	0
6	1	PAYMENT	11668.14	C2048537720	41554	29885.86	M1230701703	0	0	0	0
7	1	PAYMENT	7817.71	C90045638	53860	46042.29	M573487274	0	0	0	0
8	1	PAYMENT	7107.77	C154988899	183195	176087.23	M408069119	0	0	0	0
9	1	PAYMENT	7861.64	C1912850431	176087.23	168225.59	M633326333	0	0	0	0
10	1	PAYMENT	4024.36	C1265012928	2671	0	M1176932104	0	0	0	0
11	1	DEBIT	5337.77	C712410124	41720	36382.23	C195600860	41898	40348.79	0	0
12	1	DEBIT	9644.94	C1900366749	4465	0	C997608398	10845	157982.12	0	0
13	1	PAYMENT	3099.97	C249177573	20771	17671.03	M2096539129	0	0	0	0
14	1	PAYMENT	2560.74	C1648232591	5070	2509.26	M972865270	0	0	0	0
15	1	PAYMENT	11633.76	C1716932897	10127	0	M801569151	0	0	0	0
16	1	PAYMENT	4098.78	C1026483832	503264	499165.22	M1635378213	0	0	0	0
17	1	CASH_OUT	229133.94	C905080434	15325	0	C476402209	5083	51513.44	0	0
18	1	PAYMENT	1563.82	C761750706	450	0	M1731217984	0	0	0	0
19	1	PAYMENT	1157.86	C1237762639	21156	19998.14	M1877062907	0	0	0	0
20	1	PAYMENT	671.64	C2033524545	15123	14451.36	M473053293	0	0	0	0
21	1	TRANSFER	215310.3	C1670993182	705	0	C1100439041	22425	0	0	0
22	1	PAYMENT	1373.43	C20804602	13854	12480.57	M1344519051	0	0	0	0
23	1	DEBIT	9302.79	C1566511282	11299	1996.21	C1973538135	29832	16896.7	0	0
24	1	DEBIT	1065.41	C1959239586	1817	751.59	C515132998	10330	0	0	0
25	1	PAYMENT	3876.41	C504336483	67852	63975.59	M1404932042	0	0	0	0
26	1	TRANSFER	311685.89	C1984094095	10835	0	C932583850	6267	2719172.89	0	0
27	1	PAYMENT	6061.13	C1043358826	443	0	M1558079303	0	0	0	0
28	1	PAYMENT	9478.39	C1671590089	116494	107015.61	M58488213	0	0	0	0
29	1	PAYMENT	8009.09	C1053967012	10968	2958.91	M295304806	0	0	0	0
30	1	PAYMENT	8901.99	C1632497828	2958.91	0	M33419717	0	0	0	0

    2.2.



	A	B	C	D	E	F	G	H	I	J	K
31	1	PAYMENT	9920.52	C764826684	0	0	M1940055334	0	0	0	0
32	1	PAYMENT	3448.92	C2103763750	0	0	M335107734	0	0	0	0
33	1	PAYMENT	4206.84	C215078753	0	0	M1757317128	0	0	0	0
34	1	PAYMENT	5885.56	C840514538	0	0	M1804441305	0	0	0	0
35	1	PAYMENT	5307.88	C1768242710	0	0	M1971783162	0	0	0	0
36	1	PAYMENT	5031.22	C247113419	0	0	M151442075	0	0	0	0
37	1	PAYMENT	24213.67	C1238616099	0	0	M70695990	0	0	0	0
38	1	PAYMENT	8603.42	C1608633989	253	0	M1615617512	0	0	0	0
39	1	PAYMENT	2791.42	C923341586	300481	297689.58	M107994825	0	0	0	0
40	1	PAYMENT	7413.54	C1470868839	297689.58	290276.03	M1426725223	0	0	0	0
41	1	PAYMENT	3295.19	C711197015	233633	230337.81	M1384454980	0	0	0	0
42	1	PAYMENT	1684.81	C1481594086	297	0	M1569435561	0	0	0	0
43	1	DEBIT	5758.59	C1466917878	32604	26845.41	C1297685781	209699	16997.22	0	0
44	1	CASH_OUT	110414.71	C768216420	26845.41	0	C1509514333	288800	2415.16	0	0
45	1	PAYMENT	7823.46	C260084831	998	0	M267814113	0	0	0	0
46	1	PAYMENT	5086.48	C598357562	0	0	M1593224710	0	0	0	0
47	1	PAYMENT	5281.48	C1440738283	152019	146737.52	M1849015357	0	0	0	0
48	1	PAYMENT	13875.98	C484199463	15818	1942.02	M2008106788	0	0	0	0
49	1	CASH_OUT	56953.9	C1570470538	1942.02	0	C824009085	70253	64106.18	0	0
50	1	CASH_OUT	5346.89	C512549200	0	0	C248609774	652637	6453430.91	0	0
51	1	PAYMENT	2204.04	C1615801298	586	0	M490391704	0	0	0	0
52	1	PAYMENT	2641.47	C460570271	23053	20411.53	M1653361344	0	0	0	0
53	1	CASH_OUT	23261.3	C2072313080	20411.53	0	C2001112025	25742	0	0	0
54	1	PAYMENT	2330.64	C816944408	203543	201212.36	M909132503	0	0	0	0
55	1	PAYMENT	1614.64	C912966811	41276	39661.36	M1792384402	0	0	0	0
56	1	PAYMENT	9164.71	C1458621573	47235.77	38071.06	M1658980982	0	0	0	0
57	1	PAYMENT	2970.97	C46941357	38071.06	35100.09	M1152606315	0	0	0	0
58	1	PAYMENT	38.66	C343345308	16174	16135.34	M1714688478	0	0	0	0
59	1	PAYMENT	2252.44	C104716441	1627	0	M1506951181	0	0	0	0
60	1	TRANSFER	62610.8	C1976401987	79114	16503.2	C1937962514	517	8383.29	0	0

3. Test Results after running the Executable code :

3.1. Data Set Rows

```
   step      type    amount  ... NewBalanceDest  isFraud  isFlaggedFraud
1     1   PAYMENT   1864.28  ...            0.0      0.0             0.0
2     1  TRANSFER    181.00  ...            0.0      1.0             0.0
3     1  CASH_OUT    181.00  ...            0.0      1.0             0.0
4     1   PAYMENT  11668.14  ...            0.0      0.0             0.0
5     1   PAYMENT   7817.71  ...            0.0      0.0             0.0

[5 rows x 11 columns]
```

3.2. Data Types

```
step                int64
type               object
amount            float64
nameOrig           object
OldBalanceOrig    float64
NewBalanceOrig    float64
nameDest           object
OldBalanceDest    float64
NewBalanceDest    float64
isFraud           float64
isFlaggedFraud    float64
dtype: object
```

3.3. Fraud occurring transaction types

```
The types of fraudulent transactions are ['TRANSFER', 'CASH_OUT']

The number of fraudulent TRANSFERs = 88

The number of fraudulent CASH_OUTs = 93
```

3.4. Column datatypes after labelling the fraud variables

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 130786 entries, 2 to 301098
Data columns (total 7 columns):
step              130786 non-null int64
type              130786 non-null int64
amount            130786 non-null float64
OldBalanceOrig    130786 non-null float64
NewBalanceOrig    130786 non-null float64
OldBalanceDest    130786 non-null float64
NewBalanceDest    130786 non-null float64
dtypes: float64(5), int64(2)
memory usage: 8.0 MB
```

3.5. Data Pattern Analysis - Both the old and new balance in the recipient´s account were zero, but transferred amount was not zero

```
The percentage of 'fraudulent' transactions where both the old and new balance in the recipient´s account were zero,
but the transacted amount was not zero: 39.7790%

The percentage of 'genuine' transactions where both the old and new balance in the recipient´s account were zero,
but the transacted amount was not zero: 0.4073%
```

3.6. Data pattern analysis - Both the old and new balance in the sender´s account were zero, but transferred amount was not zero

```
The percentage of 'fraudulent' transactions where both the old and new balance in the sender´s account were zero,
but the transacted amount was not zero: 5.5249%

The percentage of 'genuine' transactions where both the old and new balance in the sender´s account were zero,
but the transacted amount was not zero: 48.6053%
```

3.7. Data after explanatory variable labelling

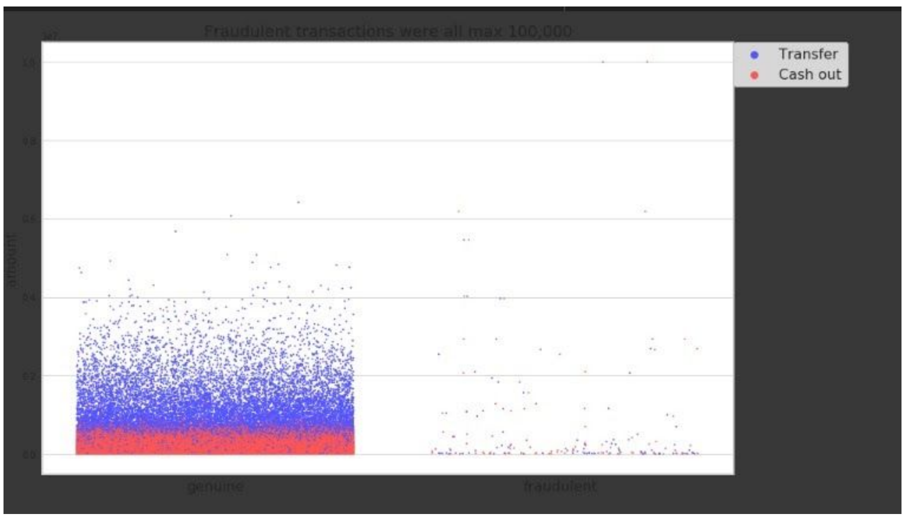
	step	type	amount	OldBalanceOrig	NewBalanceOrig	OldBalanceDest	NewBalanceDest
2	1	0	181.00	181.0	0.0	-2.0	-2.00
3	1	1	181.00	181.0	0.0	21182.0	0.00
15	1	1	229133.94	15325.0	0.0	5083.0	51513.44
19	1	0	215310.30	705.0	0.0	22425.0	0.00
24	1	0	311685.89	10835.0	0.0	6267.0	2719172.89

3.8. Feature engineering result

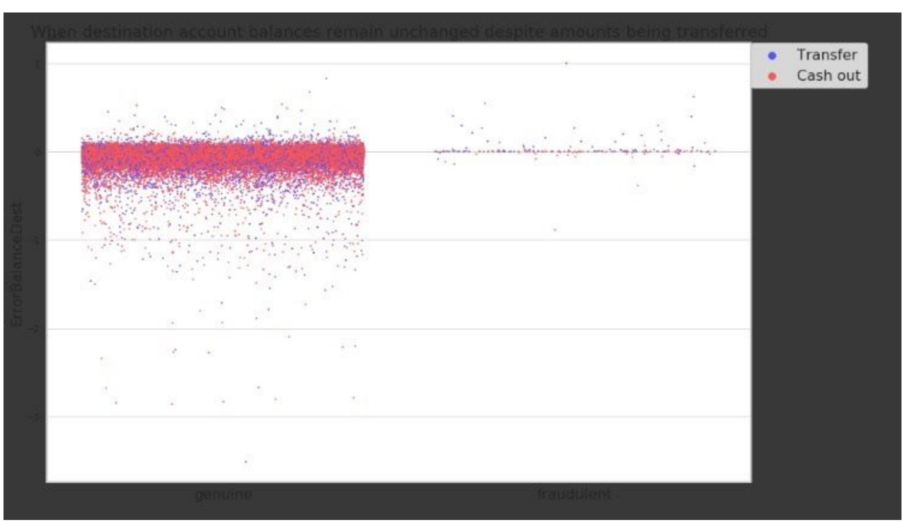
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 130786 entries, 2 to 301098
Data columns (total 9 columns):
step                130786 non-null int64
type                130786 non-null int64
amount              130786 non-null float64
OldBalanceOrig      130786 non-null float64
NewBalanceOrig      130786 non-null float64
OldBalanceDest      130786 non-null float64
NewBalanceDest      130786 non-null float64
ErrorBalanceOrig    130786 non-null float64
ErrorBalanceDest    130786 non-null float64
dtypes: float64(7), int64(2)
memory usage: 10.0 MB
```

3.9. Dispersion over time visualization

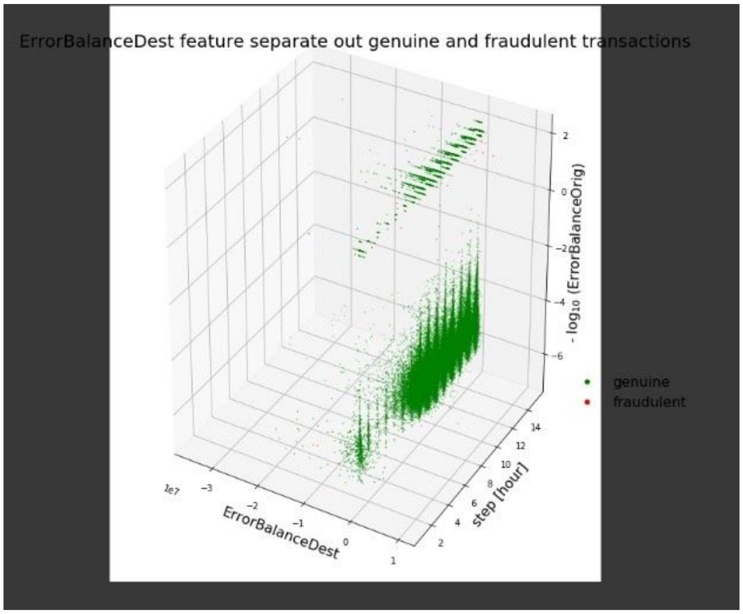
3.10. Dispersion over amount



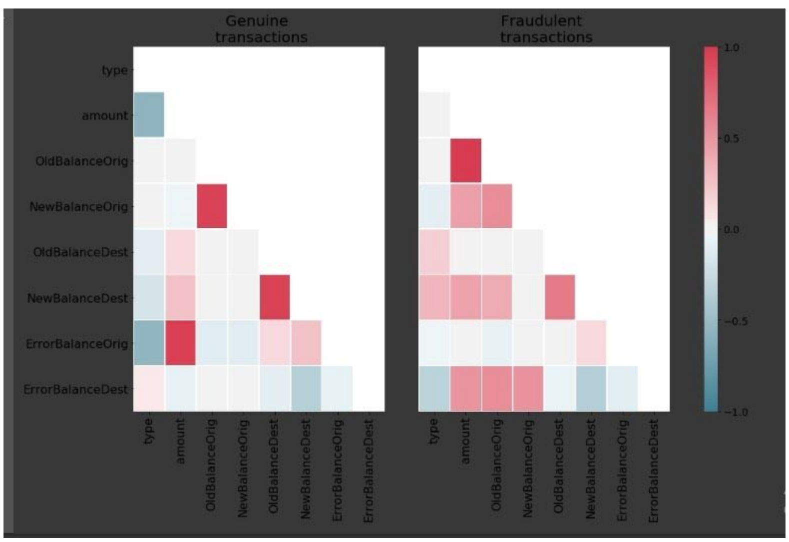
3.11. Dispersion over error in balance in destination accounts



3.12. Separating out genuine from fraudulent transactions



3.13. Genuine and fraudulent transactions



3.14. Data skewness value

```
[30]  1 print('skew = {}'.format( len(Xfraud) / float(len(X)) ))

skew = 0.0013839401770831774
```

3.15. AUPRC value

```
[32]  # Long computation in this
      weights = (Y == 0).sum() /
      clf = XGBClassifier(max_dep
      probabilities = clf.fit(tra
      print('AUPRC = {0:.6f}'.for

AUPRC = 1.000000
```

3.16. Confusion Matrix

```
array([[104316,    168],
       [    12,    132]])
```

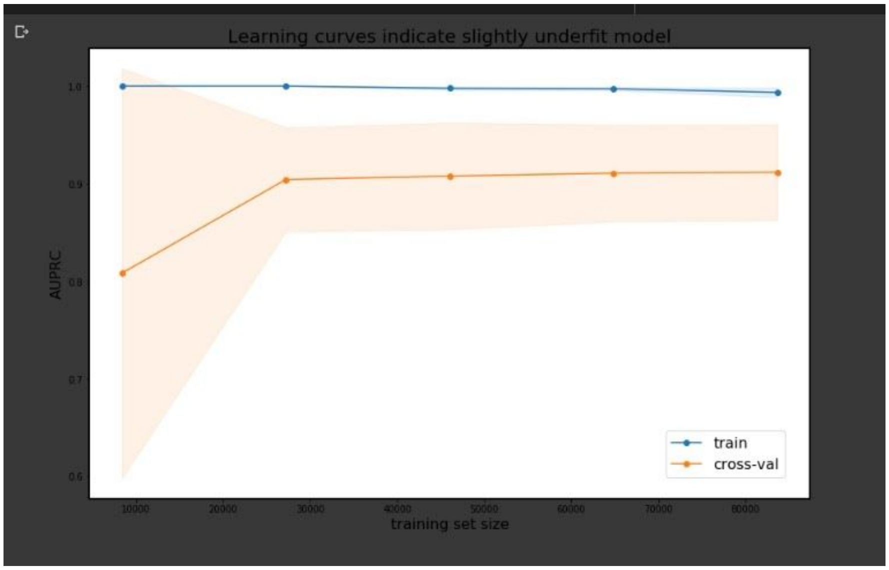
3.17. Precision-recall values

```
[35]  from sklearn.metrics import precis
      precision_score(trainY, Y_predict)

0.44

[36]  recall_score(trainY, Y_predict)

0.9166666666666666
```

3.18. Mean of Precision-recall

```
[37]  from sklearn.metrics import f1_score
      f1_score(trainY, Y_predict)

0.5945945945945945
```

3.19. Underfit model handling with Bias-variance tradeoff



4.