IDENTIFY FINANCIAL FRAUDS WITH MACHINE LEARNING MODELS

USING SYNTHETIC FINANCIAL DATA FROM PAYSIM

IMAGE SOURCE CREDIT

HTTP://100.24.114.164/FILES/PDF/RP151_FRAUD_IN_MOBILE_FINANCIAL_SERVICES_JMUDIRI.PDF

BY – JAY GUPTA

WWW.LINKEDIN.COM/IN/JAYGUPTANETWORK

Problem Definition

Identify Financial Frauds with Machine Learning Models Using synthetic financial data from paysim

RECOMMENDATION

From 2 Models (XGBoost & Random Forest), Random Forest Machine Learning Model is Recommended

WHY RANDOM FOREST?

While Both Models (XGBoost & Random Forest) Provided Equally Good Results, Random Forest was Better in terms of predicting only 1 false positive (FP) and 11 false negatives (FN)

HOW RESULTS WERE ACHIEVED?

DATA SCIENCE METHODOLOGY USED

INSIGHTS GAINED DURING THE MODELING

DATA STRUCTURE AND FEATURES

	index	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud
0	0	1	PAYMENT	9839.64	C1231006815	170136.0	160296.36	M1979787155	0.0	0.0	0
1	1	1	PAYMENT	1864.28	C1666544295	21249.0	19384.72	M2044282225	0.0	0.0	0
2	2	1	TRANSFER	181.00	C1305486145	181.0	0.00	C553264065	0.0	0.0	1
3	3	1	CASH_OUT	181.00	C840083671	181.0	0.00	C38997010	21182.0	0.0	1
4	4	1	PAYMENT	11668.14	C2048537720	41554.0	29885.86	M1230701703	0.0	0.0	0

INSIGHTS GAINED DURING THE MODELING

FRAUDS IDENTIFIED DURING CASH-OUT AND TRANSFERS

```
# Checking which transaction type has the fradulent transactions

print("List of transaction types with fraud values: ",list(data.loc[data.isFraud == 1].type.drop_duplicates().values))
```

```
List of transaction types with fraud values:  ['TRANSFER', 'CASH_OUT']
```

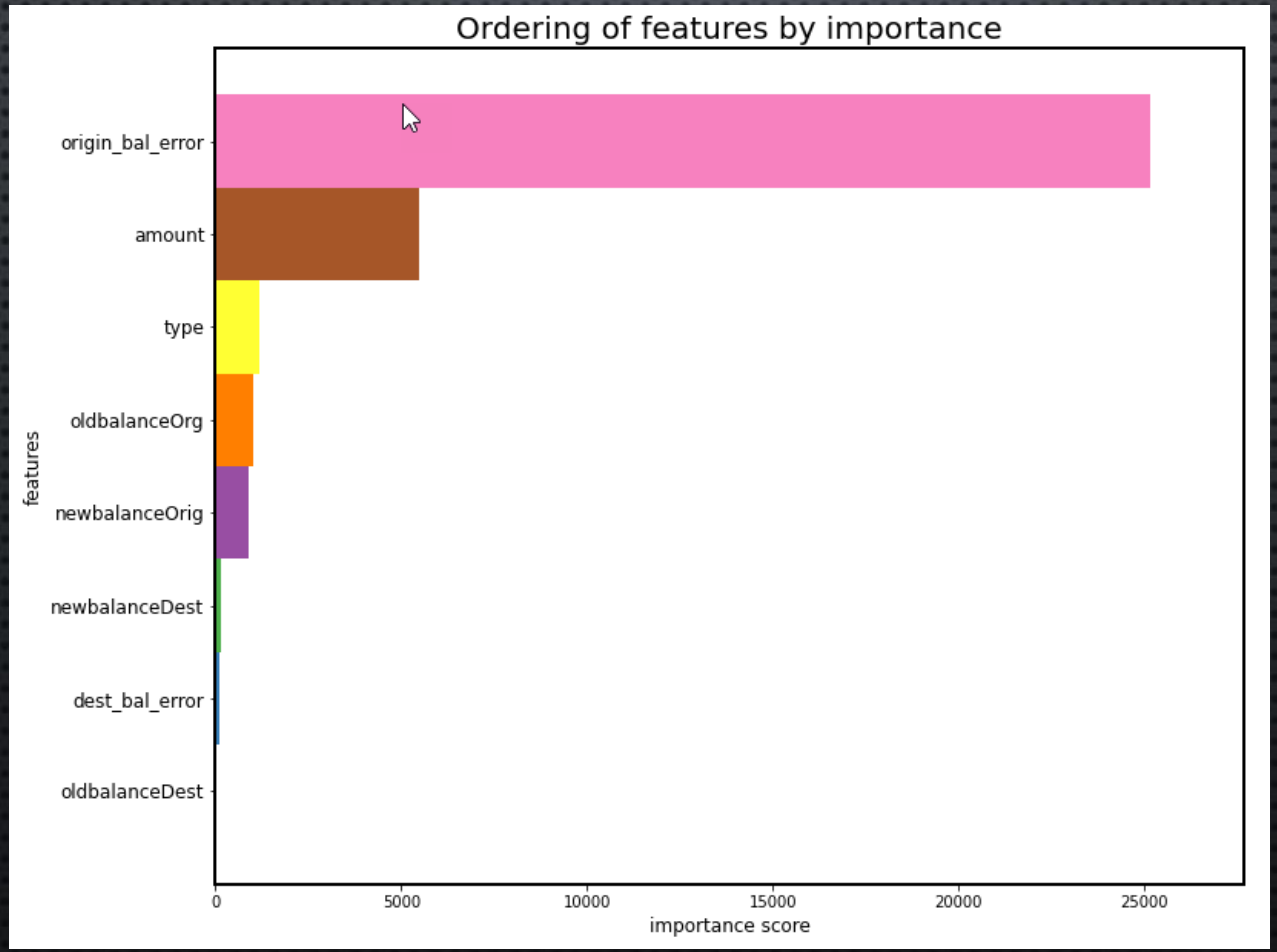
```
# Finding number of potential fraud transactions for 'CASH_OUT' and 'TRANSFER' types
transfer_frauds = data.loc[(data.isFraud==1) & (data.type=='TRANSFER')]
cashout_frauds = data.loc[(data.isFraud==1) & (data.type=='CASH_OUT')]
print("Total number of transfer frauds are {}".format(len(transfer_frauds)))
print("Total number of cash-out frauds are {}".format(len(cashout_frauds)))
```

```
Total number of transfer frauds are 4097
Total number of cash-out frauds are 4116
```

- The 'CASH_OUT' potential frauds are those frauds where money is paid to a merchant where there may be some arrangement between the fraudster and the merchant for siphoning the money.
- The 'TRANSFER' potential frauds are those frauds where money is sent to another customer

INSIGHTS GAINED DURING THE MODELING

RANKING OF THE IMPORTANCE OF FEATURES



INSIGHTS GAINED DURING THE MODELING

MODEL PERFORMANCE EVALUATION

Model Performance Evaluation

For the model performance, there is more focus on the recall rate because it is all about avoiding the frauds to manage the impact on the company's reputation as well as minimize the losses. Hence confusion matrix became an important parameter for the model performance evaluation.

For the easy reference, difinitions are presented below:

```
Accuracy = (TP + TN) / Total
Presicion = TP / (TP + FP)
Recall = TP / (TP + FN)
```

Here is the summary of the models along with its' False Positives and False Negatives:

a. XGBoost - 2 False Positives(FP) and 11 False Negatives(FN)

b. Random Forest - 1 False Positives(FP) and 11 False Negatives(FN)

While most of the other metrics are closer to each other, accuracy score for Random Forest is slightly better along with its less False Positives. Hence it is recommended that Random Forest model should be used for this application.

MACHINE LEARNING MODELING

FUTURE CONSIDERATIONS

- In the future, I would like to use the **SMOTE (Synthetic Minority Oversampling Technique)** as data is imbalanced. I would like to use the Imbalanced-learn Library.
- For this project I did not use other machine learning models such as Logistic Regression and Decision Tree Classifier algorithms which I will use in future for a similar problem.
- There is a very detailed report titled "Mobile Money for the Unbanked" which states many techniques to mitigate these frauds. In future if there is more information available, I would like to focus on the mitigations too.

```
# Finding the percentage of frauds
data.isFraud.value_counts(normalize=True)*100
```

```
0    99.870918
1     0.129082
Name: isFraud, dtype: float64
```

This means for every 1000 genuine transactions, there may be 1.29 potential fraud transactions.

THANK YOU

BY – JAY GUPTA

LINKEDIN: HTTPS://WWW.LINKEDIN.COM/IN/JAYGUPTANETWORK/

GITHUB: HTTPS://JAYGUPTACAL.GITHUB.IO/PORTFOLIO/