Money Laundering Prevention

Written By	Vaishali Parashar
Document Version	1.0
Last Revised Date	17-March-2022

Objective:

“To identify the patterns for every consumer that may lead to money laundering like transferring money to foreign banks, big deposits, transaction patterns etc. , using machine learning.

To develop a method that minimizes false negatives when evaluating new data points."

Benefits:

- Reduction of false positives in the AML Process.
- Detecting the change in customer behavior.
- Analysis of unstructured data and external data.

Data Description

Dataset from kaggle : PaySim

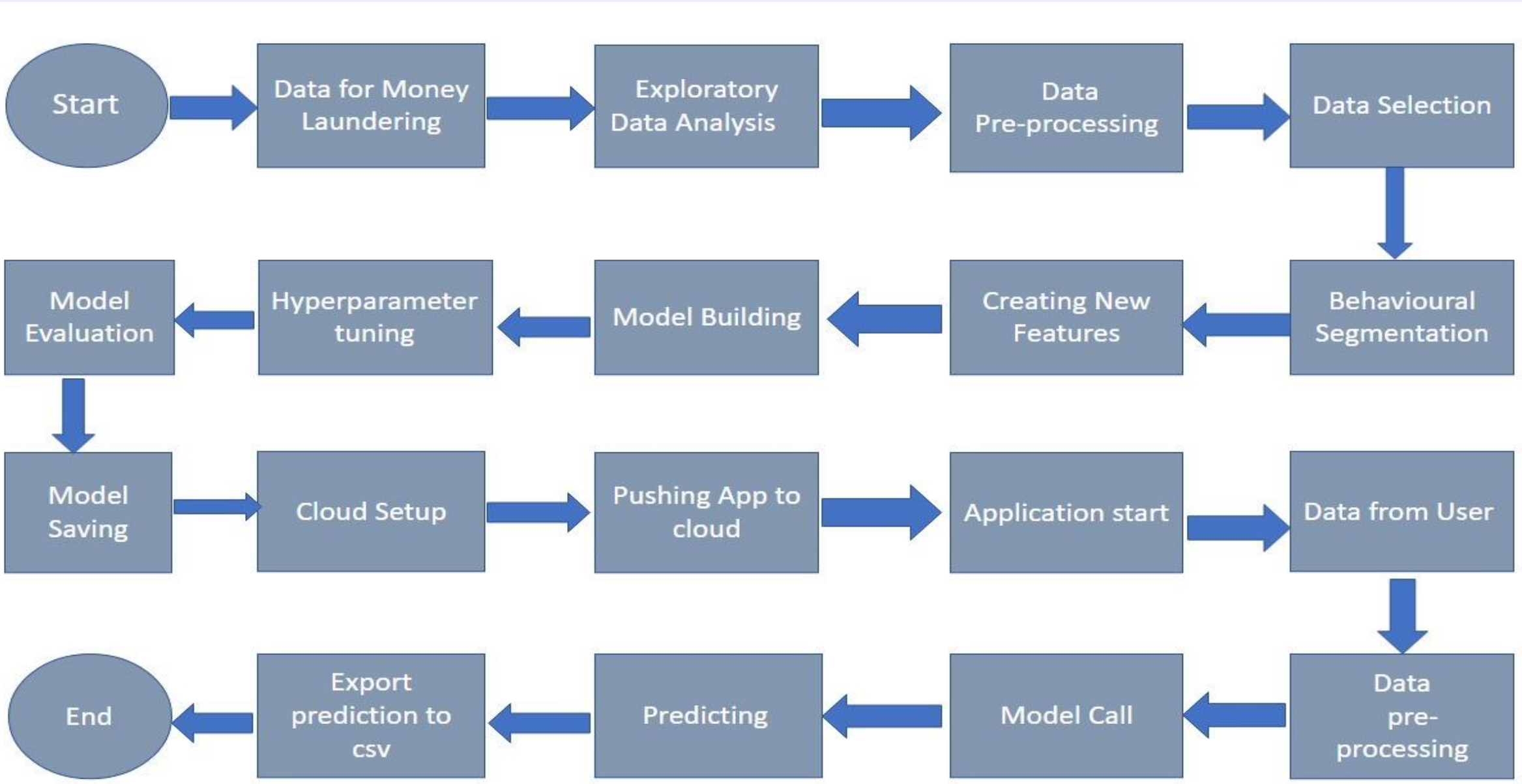
For this we are using a synthetic dataset generated using the simulator called PaySim.

PaySim simulates mobile money transactions based on a sample of real transactions extracted from one month of financial logs from a mobile money service implemented in an African country. This dataset contains 6362620 Transaction records with 11 features.

Data Sharing Agreement

- File name can be anything.
- Columns inside the file will be 11 .
- Columns are
- [step, type, amount, nameOrig, oldbalanceOrg, newbalanceOrig, nameDest, oldbalanceDest, newbalanceDest , isFraud, isFlaggedFraud]

Architecture



Architecture Description

1. Data gathering:

Data source: https://www.kaggle.com/ealaxi/paysim1

Train and Test data are stored in .csv format.

2.Exploratory Data Analysis :

Exploratory data analysis or EDA is an important step in any data analysis or data science project . EDA is the process of investigating the dataset to discover patterns and anomalies (outlier) and form hypothesis based on our understanding of the dataset.

3. Data Pre-processing

Data Pre-processing is the process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data.

3.4. Feature Engineering :

Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work. If feature engineering is done correctly, it increases the predictive power of the machine learning algorithms by creating features from raw data that facilitate the machine learning process.

3.5. Feature Selection :

A feature selection algorithm can be seen as the combination of a search technique for proposing new features subset, along with an evaluation measure which scores the different feature subsets. The simplest algorithm is to test each possible subset of features finding the one which minimizes the error rate . This is an exhaustive search of the space and is computationally intractable for all. But the smallest of feature sets.

3.6. Model Building :

A machine learning model is built by learning and generalizing from training data, then applying that acquired knowledge to new data it has never seen before to make predictions and fulfil its purpose. Lack of data will prevent you from building the model, and access to data isn't enough.

3.7.Decision Tree Classifier :

Decision tree is the most powerful and popular tool for classification and prediction . A decision tree is a flow chart like tree structure, where each internal load denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal load) holds a class label.

3.8. XGBoost Classifier :

XGBoost Model for Classification XGBoost is short for Extreme Gradient Boosting and is an efficient implementation of the scholastic gradient boosting machine learning algorithm. The stochastic gradient boosting algorithm, also called gradient boosting machines or tree boosting , is a powerful machine learning technique that performs well or even best on a wide range of challenging machine learning problems.

3.9. Hyperparameter Tuning:

While model parameters are learning during training – such as the slope and intercept in a linear regression – hyperparameters must be set by the data Scientist before training . In the case of a random forest, hyperparameters include the number of decision trees in the forest and the number of features considered by each tree when splitting a node.

3.10. Model Validation :

For machine learning systems, we should be running model evaluation and model tests in parallel. Model evaluation covers metrics and plots which summarize performance on a validation or test dataset . Model testing involves explicit cheeks for behaviors that we expect our model to follow.

3.11. Deployment :

We will be deploying the model in Heroku Cloud Platform.

3.11.01. Heroku:

Heroku is a cloud platform as a service supporting several programming languages. One of the first cloud platforms, Heroku has been in development since June 2007 , when it supported only the Ruby programming language, but now supports Java , Node.js , Scala, Clojure , Python , PHP, and Go.

3.11.02. Flask:

Flask is a micro web framework written in python. Here we use flask to create an API that allows to send data, and receive a prediction as a response. Flask supports extensions that can add application features as if they were implemented in flask itself .

Data set description

Synthetic dataset generated using the simulator called PaySim.

PaySim simulates mobile money transactions based on a sample of real transactions extracted from one month of financial logs from a mobile money service implemented in an African country. This dataset contains 6362620 Transaction records with 11 features.

The dataset looks like as follow:

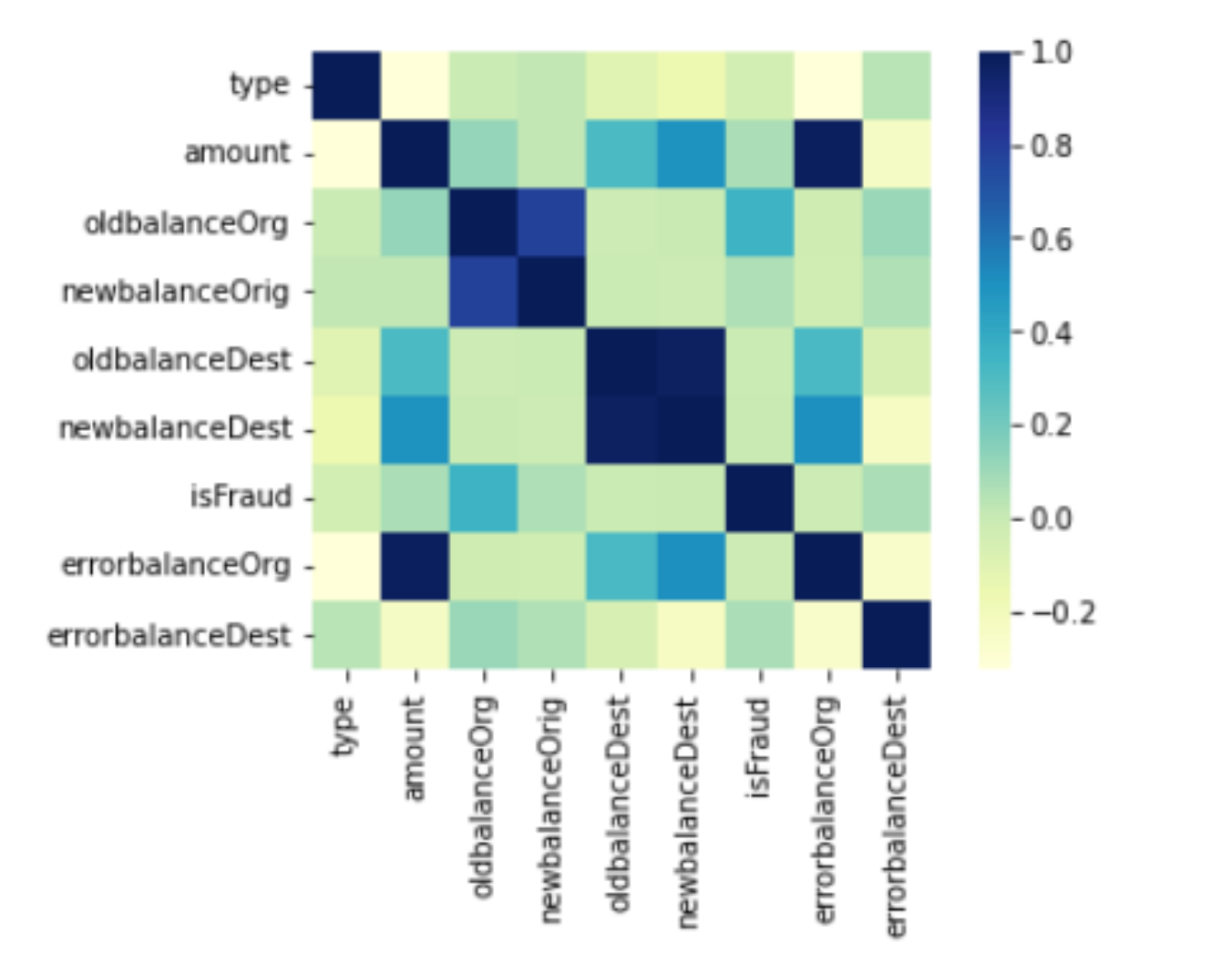
	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	1	PAYMENT	9839.64	C1231006815	170136.0	160296.36	M1979787155	0.0	0.0	0	0
1	1	PAYMENT	1864.28	C1666544295	21249.0	19384.72	M2044282225	0.0	0.0	0	0
2	1	TRANSFER	181.00	C1305486145	181.0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

The data set consists of various data types from integer to float to object as shown in Fig.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6362620 entries, 0 to 6362619
Data columns (total 11 columns):
 #   Column          Dtype
---  ------          -----
 0   step            int64
 1   type            object
 2   amount          float64
 3   nameOrig        object
 4   oldbalanceOrg   float64
 5   newbalanceOrig  float64
 6   nameDest        object
 7   oldbalanceDest  float64
 8   newbalanceDest  float64
 9   isFraud         int64
 10  isFlaggedFraud  int64
dtypes: float64(5), int64(3), object(3)
memory usage: 534.0+ MB
```

Observations

Correlation is used to understand the relation between a target variable and predictors. In this work, Item-Sales is the target variable and its correlation with other variables is observed.



Q & A:

Q1) What's the source of data?

Ans. The data for training is provided by the client from:

https://www.kaggle.com/ealaxi/paysim1

Q 2) What was the type of data?

Ans. The data was the combination of numerical and Categorical values.

Q 3) What's the complete flow you followed in this Project?

Ans. Refer the Architecture section for this.

Q 4) After the File validation what you do with incompatible file or files which didn't pass the validation?

Ans. Files like these are moved to the Achieve Folder and a list of these files has been shared with the client and we removed the bad data folder.

Q 5) How logs are managed?

Ans. We are using different logs as per the steps that we follow in validation and modeling like validation log, database operation log, preprocessing log, model training log, etc..

Q 6) What techniques were you using for data pre-processing?

- Removing unwanted attributes
- Visualizing relation of independent variables with each other and output variables
- Removing outliers
- Cleaning data and imputing if null values are present.
- Converting categorical data into numeric values.
- Scaling the data

Q 7) How training was done or what models were used?

Before diving the data in training and validation set we performed clustering over fit to divide the data into clusters. As per cluster the training and validation data were divided. The scaling was performed over training and validation data

Algorithms like Linear regression, Gradient boost, Random forest and XGBoost were used .

Q 8) How Prediction was done?

Ans. The testing files are shared by the client. We pass its data to the best model which we have saved in pickle format and get the prediction.

Q 9) Where the model was deployed?

Ans. When the model is ready, we deploy it in Heroku platform. This model is an web application where user can enter the data and these data gets extracted in the backend and user gets the prediction result.