**Submission Document**

**Objective-** To build up a classification model classifying muled bank(the bank account in which illegal activities are taking place as per Problem Statement) account as 1 and non-muled bank account as 0 , and to submit/predict the probability of a customer with a certain key(Primary Key) the probability that it can be a muled(class 1) bank account.

**Given Dataset Overview-** For training data a development data is shared which consists of ‘Target’ column.

For submission, validation dataset is given which do not have ‘Target’ column.

**Basic library import –** Imported all necessary libraries that were used for the code, that is **numpy, pandas, matplotlib and scikit learn, Keras** and all the other necessary modules required for training the model.

**Basic Work Flow-**  Data Analysis was done on the development data that was shared , the Excel file was converted into csv file and the software used was **Google Collab and it’s GPU training.**

**About Dataset through Analysis done till this point-** The dataset was **imbalanced** with respect to the target variable, the last 2000 rows of the dataset have ‘Target’ class as 1 and the first 98000 rows had the class 0 , so the data was highly imbalanced that should be kept in mind while validation and testing.

To facilitate a more focused and convenient analysis, the dataset was partitioned into four subsets: df\_demog(for demographic attributes), df\_tx(for transaction attributes), df\_others(for other attributes), and df\_object(for non numeric data attributes). This division allowed for separate and targeted analysis on each subset, enhancing the comprehensibility and effectiveness of the overall data examination."



* We were quite reluctant to reduce any row which has target value as 1 , because already rows having target value as 1 were minority.
* All the missing values in the rows which have target value as 1 were carefully filled , sometimes manually , and sometimes individually writing code for that segment.

**Dropping columns -** Dropped Primary Key as it was index and is hindrance in training the data.

* **Co-Relation-** The columns which have absolute no or very low co relation that is (high co relation **among themselves{to avoid multi-collinearity}** and low co relation with target variable have been removed , using simple **pandas function of corr() and manually also columns** have been removed which seemed useless).

The removed columns were = ['demog\_7', 'demog\_10', 'demog\_12', 'txn\_35', 'txn\_36', 'txn\_49', 'txn\_50', 'txn\_65', 'txn\_70', 'txn\_71', 'txn\_72', 'demog\_38','others\_42','others\_43','others\_44','others\_45','country\_code','os','demog\_22','txn\_37', 'txn\_55', 'txn\_61', 'txn\_62', 'txn\_66', 'others\_3', 'others\_17', 'others\_19', 'others\_20', 'others\_29', 'others\_30', 'others\_31', 'others\_32', 'others\_40', 'others\_41']

**Creation of helpful function that were used frequently-** functions like **analyze\_columns (df,’column\_name’**) was created which was frequently used to analyze any column by giving column name and dataset , it gives the value\_counts for the target value 1 and 0 and also tell the number of missing values separately, Like this many more functions were created , this was the most useful for the analysis of the columns.

**Addressing the missing values-** As occupation was extremely important column w.r.t ‘Target’ variable , so complete case analysis was done that is all the rows which have missing values in occupation column **,** and it was assured that none of the rows from the target value 1 is removed.

* For important numeric columns KNN imputer was used to the impute the data
* For categorical data Simple Imputer with strategy set to mode was used
* For other numeric columns Simple imputer with mean strategy was used.
* For txn and others column , as it was fully numeric so mean impute was used to fill in the missing values
* The missing values of the known columns like occupation, income etc was given special attention.

**Highly important columns-** Highly important columns were considered those which have different nature for the ones with target value as 0 and with ones.

Example Insights

* The bank customers with mused bank account have majority in the Rural Area and the one’s who have normal bank account have majority in Tier 1 cities.
* Like demog\_40 and demog\_43 column , they have different nature for the ones which have target value as 1 and 0
* Like other\_5 columns , it also had reverse nature of the entries with target 0 and target 1 etc.
* Many small insights were noted down and worked upon.

**Encoding the categorical data- One hot encoding** the occupation, email\_domain data and **ordinal encoding** the data of Income (The highest income given the largest number) and City tier (Rural given the least as 0 and Tier 1 city gets the highest number as 8 and so on).

* Encoding of demog\_4 data which consists of **mixed data type variable** which consists of alphabet(‘N’) and numbers(from 1 to 7) , demog\_4 column was break down into 2 columns of ‘is\_N’ and ‘is\_numeric’.

**Till now all the missing data was addressed and all the data was converted into numeric form**

**Train Test Split- Train test split of scikit learn** was used for splitting between train and validation data from development data given for training and validation data shared was kept for final prediction.

* All **the pipeline followed for the training data(development data**) was also simultaneously followed for the validation data also , so the validation data was also prepared for prediction(that is it’s missing values were filled and encoding was also done).

**Using SMOTE-** As data was highly imbalanced so over sampling was done for class 1 in ‘Target’ column , and synthetic dataset was generated for training so that model can understand class 1 also properly , so training on different ML models was done using **X\_resampled and y\_resampled** , using smote the dataset was increased such that ML algorithms gets enough learning for both the classes.

**Standard Scaler- Standard scaler** for the training data was applied before training into the ML algorithms for uniformity and also for better and faster results.

**ML model Training ALGORITHM – [Random Forrest Classifier, Decision Tree Model, Logistic Regression, XG BOOST, Naïve Bayes, Deep ANN]**

**Metrics for the classification model- Accuracy , F1 Score , Classification Matrix**. F1 score becomes an important matrix for determination of an imbalanced dataset , a good model with imbalanced dataset can be identified with a high F1 score , because it has to have high precision and high recall value.

* As dataset was skewed the accuracy was generally high above 97% in almost all the algorithms , **so our main matrix to decide was classification matrix and F1 score**
* Trained on selected features after doing feature selection once again to improve accuracy as , feature selection is iterative process
* Fine tuning the model to attain maximum F1 score and accuracy.

**Training Summary**

**The Best Model among these came out to be XG BOOST , it out performs the other algorithms even deep ANN so XB BOOST is use to create the CSV file for predicting the probability on the validation dataset.**

**Following is the information of the XG BOOST MODEL**

Confusion Matrix:

[[18694 38]

[ 27 391]]

True Positives (TP): 391

False Positives (FP): 38

True Negatives (TN): 18694

False Negatives (FN): 27

Accuracy: 0.9966

F1 Score: 0.9233

Precision: 0.9114

Recall: 0.9354