Liability account monitoring for Money Mules

1. Problem Description----------

2. Data Understanding ----------

3. Data Preparation----------

4. Feature Engineering----------

5. Model Selection, Training and Evaluation----------

6. Feature Importance----------

7. Test Result ----------

8. Conclusion----------

Team Name: **EkyamAI**

Abhishek Gantait

Abhinav Kumar Jha

Rohit Singh Rathore

Shriram Pradeep

1. Problem Description

To design and implement an to develop an automated transaction monitoring framework using Machine Learning algorithms, enabling a organization like Bank to effectively identify and mitigate fraudulent transactions in real-time.

The above task is a highly imbalanced binary classification, with the objective to distinguish between legitimate and potentially fraudulent transactions conducted through savings accounts.

2. Data Understanding

2. Overview of Datasets:

Dataset statistics

Number of variables	178
Number of observations	2000
Missing cells	21265
Missing cells (%)	6.0%
Total size in memory	2.7 MiB
Average record size in memory	1.4 KiB

Variable types

Numeric	165
DateTime	1
Categorical	10
Unsupported	2

```
df['Target'].value_counts()
```

```
Target
0    98000
1     2000
Name: count, dtype: int64
```

It has been observed the significant class imbalance in dataset.

The features can be grouped in 4 categories transaction, demographic, others, miscellaneous.

```
Number of demographic columns  43
Number of transaction columns  81
Number of others columns  45
Number of miscellaneous columns 7
 which are ['account_opening_date', 'country_code', 'income', 'city_tier', 'occupation', 'os', 'email_domain']
```

There are high percentage of NaN i.e missing values present in data.

```
Columns with NaN values and their percentage:
others_45: 99.31%
others_44: 99.13%
others_42: 97.9%
others_43: 95.42%
txn_80: 57.48%
txn_78: 56.73%
txn_81: 51.46%
txn_77: 45.42%
txn_79: 44.96%
others_17: 41.36%
others_16: 41.36%
others_15: 41.36%
txn_53: 25.79%
txn_50: 25.79%
txn_51: 25.79%
txn_52: 25.79%
```

There is very high correlation among the highly correlated features

```
Top Correlation pairs
demog_11   demog_29     1.000000
demog_13   demog_17     1.000000
demog_25   demog_35     1.000000
txn_78     txn_79       1.000000
demog_24   demog_26     0.999593
demog_19   demog_34     0.999286
           demog_41     0.994755
demog_34   demog_41     0.994329
txn_11     txn_75       0.992327
demog_3    demog_29     0.991297
           demog_11     0.991297
txn_12     txn_76       0.988867
txn_3      txn_4        0.984382
```

3. Data Preparation

3.1 Dropping quasi-constant features:

Removing the columns where a single value covers 99% (or any high percentage) of the data. These columns, often referred to as constant or quasi-constant features, may not contribute much information to the model and can potentially lead to overfitting. Only excluding **'txn_' features** which will be used in feature engineering part.

Features	Percentage Of Quasi Constant
country code	0.995005189
demog_6	0.990099876
demog_7	1
demog_10	1
demog_11	0.997737472
demog_12	1

demog_14	0.998337907
demog_16	0.993029518
demog_18	0.99761247
demog_22	0.996124952
others_3	0.993575
others_17	0.992142249
others_19	0.995475
others_20	0.997975
others_27	0.994462015
others_29	0.993225
others_31	0.993425
demog_25	0.994262428
demog_28	0.999924999
demog_29	0.997737472
demog_31	0.996824881
demog_35	0.994262285
demog_36	0.994499794
demog_37	0.9999875
demog_38	1
others_33	0.99585
others_34	0.9909375
others_38	0.9995875
others_40	0.9951875
others_41	0.99775
demog_39	0.9999375

3.2 Drop High Proportion Missing Value Columns:

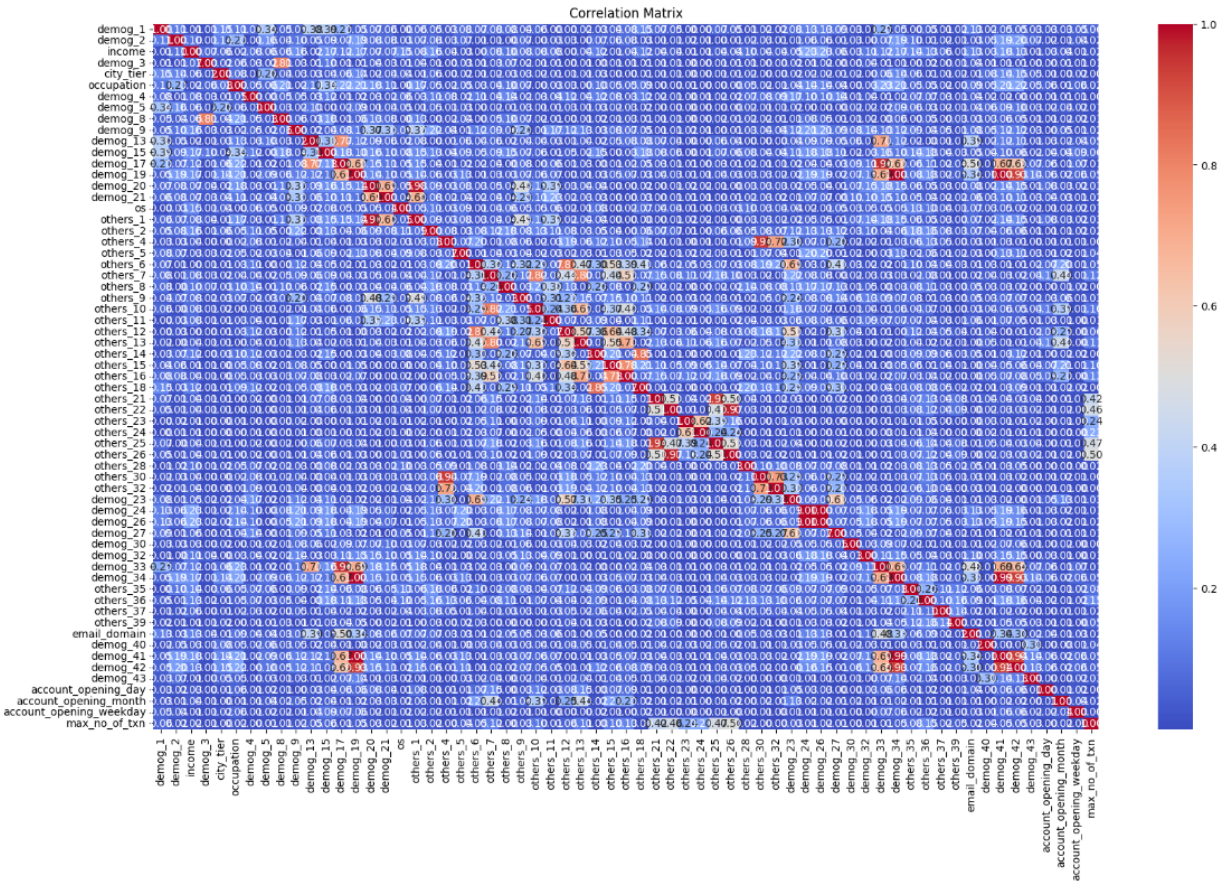
Dropping columns with a high proportion of missing values is a critical step in the data preprocessing pipeline. It ensures the integrity and reliability of the dataset, leading to more robust and accurate machine learning models. In this case study, threshold value of 95% and above has been considered for dropping the relevant features.

Features	Percentage Of Missing Values
others_42	0.9789
others_43	0.9544
others_44	0.9912625
others_45	0.9930625

3.3 Remove Highly Correlated Features:

To mitigate multicollinearity and improve model generalization, highly correlated features were identified and subsequently dropped from the feature set.

Features exhibiting correlation coefficients above a significant threshold of **.95** were identified as highly correlated and removed.



Below are the features to be removed due to high correlation.

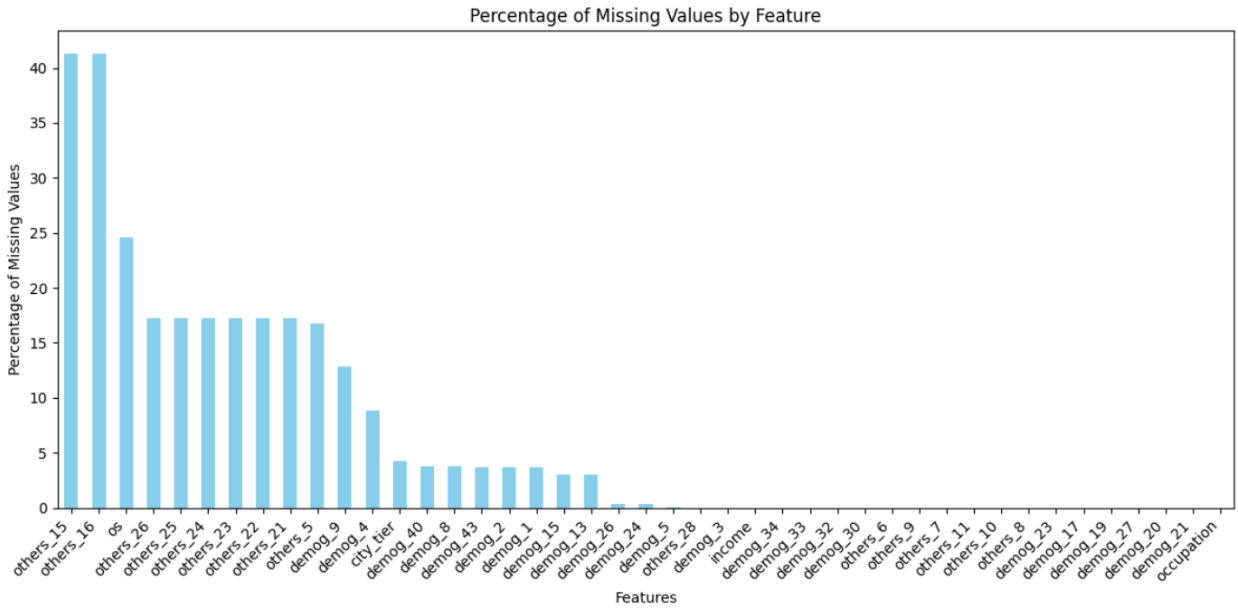
- **others_1**
- **others_26**
- **demog_26**
- **demog_33**
- **demog_34**
- **demog_41**

3.4 Missing Value Imputation:

Missing values in a dataset can hinder model performance and lead to biased results. Therefore, missing values imputation aims to handle missing data by replacing or filling in with estimated or calculated values.

1. Identifying Missing Values:

Use of descriptive statistics or visualization techniques to identify missing values in the dataset.



2. Choosing Imputation Strategies:

Mode imputation technique is used to impute the missing values ensuring consistency in imputation across training and test datasets to avoid data leakage and maintain model integrity.

3.5 Encoding Categorical Features:

Categorical features represent qualitative variables with a finite number of distinct categories or levels. Encoding Categorical Features transforms categorical variables into numerical representations suitable for machine learning algorithms.

Below are the features have been converted to numerical representation using label encoding, ensuring to avoid dimensionality issues.

- **demog_2**
- **income**
- **city_tier**
- **occupation**
- **demog_4**
- **os**
- **email_domain**
- **demog_40**
- **demog_43**

4. Feature Engineering

4.1 Feature Extraction from **Account Opening Date:**

The 'account_opening_date' column in the dataset contains information about the date when user accounts were opened. To enhance the dataset and capture temporal patterns, we extracted three key features from this date:

1. **Day of the Month (account_opening_day):**
   This feature represents the numeric day of the month when the account was opened.
2. **Month of the Year (account_opening_month):**
   This feature denotes the numeric month in which the account was opened.

3. **Day of the Week (account_opening_weekday):**
   This feature provides the numeric representation of the day of the week (0 for Monday, 1 for Tuesday, and so on).
4. **Quarter of the Year (account_opening_quarter)**
   This feature denotes the numeric quarter in which the account was opened.

Finally, we dropped 'account_opening_date' column from dataset.

4.2 Feature extraction from **Transaction Data:**

It includes generating features by doing following.

1. Calculating MEAN, MAX, MIN, MEDIAN, STANDARD DEVIATION, SKEWNESS
2. Calculating TOTAL, NON-ZERO (COUNT, MEAN), ZERO (COUNT)
3. Calculating MAXIMUM NON-ZERO AND ZERO TRANSACTIONS
4. Standardize the transactional data using StandardScaler and compute the mean of absolute values.
5. Perform DBSCAN clustering on transactional data and compute the mean of cluster labels.
6. Perform KMeans clustering on transactional data and compute the mean of cluster labels.
7. Use Isolation Forest to detect anomalies in transactional data and compute the total anomaly score.

for each user's transaction columns.

5. Model Selection/Training/Evaluation

The selection of models was based on model's performance. Models were evaluated based on their ability to accurately classify each class of transactions as legitimate or fraudulent. Advance tree-based ML algorithms have been used to train on the data.

8.1 Models Considered:

i. XGBoostClassifier including All Features (Baseline)
ii. XGBoostClassifier including Selected Features
iii. Stacking Classifier (**Base Learner:** RandomForestClassifier, XGBoostClassifier, CatBoostingClassifier, **Meta Learner:** XGBoostClassifier)

8.2 Training Methodology

Model evaluation was conducted using the following techniques:

Cross-Validation: To ensure unbiased performance estimates and due to imbalance class data, **Stratified k-fold cross-validation** with **k=10** was employed. Performance achieved by the most performant model on test data was .**9657(F1-Macro)** with Standard Deviation of **.005** across the cross-validation process.

Evaluation Metrics: Considering Imbalance class distribution, models were assessed based on F1-macro. However, analysis was also done based on key metrics including accuracy, F1-score(weighted), and auc-roc score.

Feature Selection:

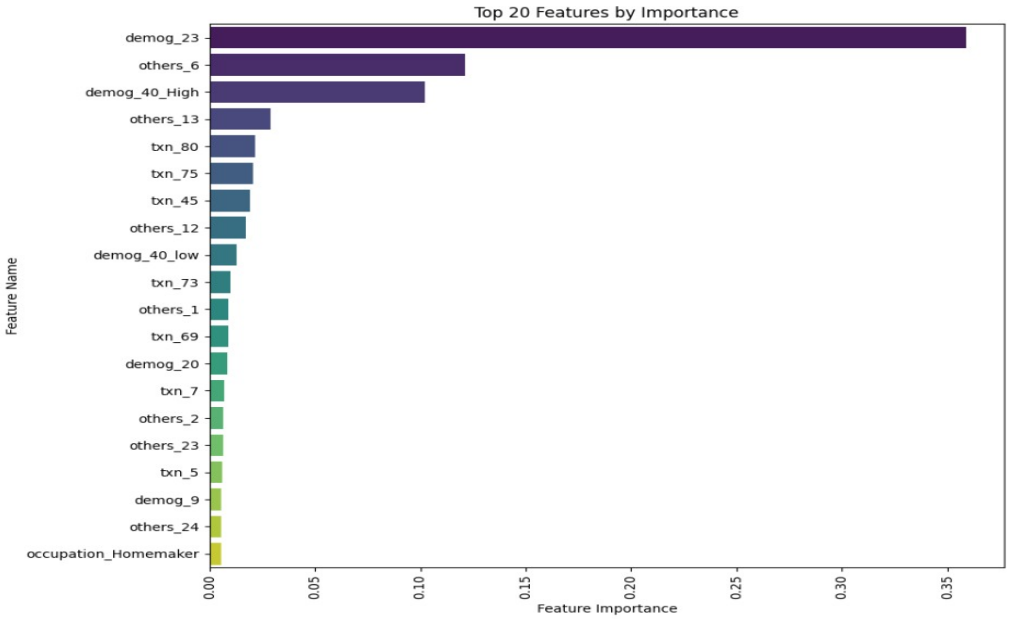
Based on the performance of baseline model, features have been selected with importance greater than **zero**.

Below are the top 116 important features have been selected:

demog_23	mean	demog_2_6	txn_51
others_6	max_consecutive_zeros	others_26	demog_2_1
demog_40_High	city_tier_Tier 2	txn_12	txn_34
txn_75	others_9	non_zero_std	email_domain_gmail
others_13	txn_73	demog_2_7	txn_13
txn_80	txn_79	txn_20	txn_29
txn_45	others_36	txn_42	city_tier_Tier 4
others_12	total_transactions	non_zero_count	city_tier_Tier 5
demog_20	txn_18	city_tier_Tier 7	txn_60
demog_40_low	std_dev	txn_1	txn_3
demog_43_High	demog_32	txn_2	income_10L to 25L
os_ios	others_16	city_tier_Rural	txn_51
others_23	demog_26	others_21	demog_2_1
txn_5	txn_81	demog_21	txn_34
others_1	total_anomaly_score	demog_43_medium	email_domain_gmail
txn_69	others_10	others_22	txn_13
others_2	others_35	demog_24	txn_29
others_24	txn_31	income_100001 to 5L	city_tier_Tier 4
txn_74	txn_58	city_tier_Tier 3	city_tier_Tier 5
txn_7	txn_41	max	txn_60
non_zero_avg	occupation_Homemaker	skewness	txn_3
demog_40_medium	others_7	demog_2_2	income_10L to 25L
city_tier_Tier 1	others_17	income_5L to 10L	txn_51
demog_9	others_25	txn_17	
txn_26	others_8	occupation_Self_Employed	
kmeans_cluster	txn_77	demog_2_3	
txn_11	demog_1	max_consecutive_non_zeros	
occupation_Student	z_score_mean	city_tier_Tier 6	
txn_6	occupation_Salaried	demog_43_low	
txn_25	txn_78	txn_4	
txn_53	zero_count	demog_2_4	
others_11	demog_17	email_domain_Others	
txn_8	city_tier_Tier 8	occupation_Other	
os_and	txn_9	income_0 to 1L	
txn_76	others_15	txn_19	

6. Feature Importance

Feature importance analysis can inform feature selection strategies by identifying the most relevant features for model training. Stacking classifier has been used to select important features with Base Learner: RandomForest, XGBoost and CatBoosting and Meta Learner: XGBoost.



Below are the top 116 important features have been selected:

demog_23	mean	demog_2_6	txn_51
others_6	max_consecutive_zeros	others_26	demog_2_1
demog_40_High	city_tier_Tier 2	txn_12	txn_34
txn_75	others_9	non_zero_std	email_domain_gmail
others_13	txn_73	demog_2_7	txn_13
txn_80	txn_79	txn_20	txn_29
txn_45	others_36	txn_42	city_tier_Tier 4
others_12	total_transactions	non_zero_count	city_tier_Tier 5
demog_20	txn_18	city_tier_Tier 7	txn_60
demog_40_low	std_dev	txn_1	txn_3
demog_43_High	demog_32	txn_2	income_10L to 25L
os_ios	others_16	city_tier_Rural	txn_51
others_23	demog_26	others_21	demog_2_1
txn_5	txn_81	demog_21	txn_34
others_1	total_anomaly_score	demog_43_medium	email_domain_gmail
txn_69	others_10	others_22	txn_13
others_2	others_35	demog_24	txn_29
others_24	txn_31	income_100001 to 5L	city_tier_Tier 4
txn_74	txn_58	city_tier_Tier 3	city_tier_Tier 5
txn_7	txn_41	max	txn_60
non_zero_avg	occupation_Homemaker	skewness	txn_3
demog_40_medium	others_7	demog_2_2	income_10L to 25L
city_tier_Tier 1	others_17	income_5L to 10L	txn_51
demog_9	others_25	txn_17	
txn_26	others_8	occupation_Self_Employed	
kmeans_cluster	txn_77	demog_2_3	
txn_11	demog_1	max_consecutive_non_zeros	
occupation_Student	z_score_mean	city_tier_Tier 6	
txn_6	occupation_Salaried	demog_43_low	
txn_25	txn_78	txn_4	
txn_53	zero_count	demog_2_4	
others_11	demog_17	email_domain_Others	
txn_8	city_tier_Tier 8	occupation_Other	
os_and	txn_9	income_0 to 1L	
txn_76	others_15	txn_19	

7. Test Result:

Model	Accuracy	F1-score(weighted)	F1-macro	AUC-ROC
BaseLine XGBoost	.9972	.9971	.9636	.9557
XGBoost	.9973	.9973	.9657	.9594
Stacking Classifier	.9971	.9970	.9625	.9556

- The high accuracy and High F1-score (Macro) indicate that the model effectively distinguishes between different classes within the test dataset.
- The weighted F1 score addresses class imbalance issue by calculating the F1 score for each class independently and then averaging them, weighted by the number of instances for each class.
- The ROC AUC score provides a comprehensive summary of the model's discriminative power across different classes.

8. Conclusion

1. The test results of the ML model shows its potential to distinguish between legitimate and potentially fraudulent transactions.
2. The selected XGBoost Classifier demonstrated superior performance in accurately classifying transactions as legitimate or fraudulent.
3. Other metrices like Accuracy, F1-score(Weighed) and ROC AUC metrics further validate the effectiveness of the model in identifying fraudulent transactions.
4. The trained model is stable which is evident from its low standard deviation value of **0.005**