

Big Data and Cloud Computing Project

Phase 2

|  |  |  |
| --- | --- | --- |
| **Team 5** | | |
| **Name** | **Section** | **BN** |
| **Ahmed Mostafa Mohamed** | **1** | **9** |
| **Mohamed Kotb** | **2** | **17** |
| **Mohamed Kamal** | **2** | **18** |
| **Mohamed AbdElhady** | **2** | **16** |

# Problem Description

This project aims to address the common problem of loan defaults in finance. Our objective is to highlight the key factors influencing this issue and create a predictive model that can identify individuals likely to default on loans. By understanding these factors, banks can make better lending decisions, reducing the risk of financial losses and ensuring a safer lending environment.

To achieve this goal, we will perform feature engineering to select the most important features from our dataset. These features may include socio-economic factors like income, employment status, and credit history, as well as demographic variables such as age and marital status.

Top of Form

The Dataset used during our analysis :

|  |  |
| --- | --- |
| Link | https://www.kaggle.com/datasets/mishra5001/credit-card |
| Number of features | 122 |
| Number of records | 307511 |

# Project Pipeline

# Analysis and solution of the problem

## **Data preprocessing**

1) Get features with null values greater than 5% of total count of rows and drop them

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

2) For features having less than 5% of their values as nulls where filled the missing values with the mean for numerical columns and mode for categorical columns

3) Some columns contain information related to the documents submitted by the customer when applying for the loan, we don’t have more details about these columns, so we sometimes drop them (e.g. in clustering or in data mining)

4) One of each pair of the highly correlated features was removed

5) After applying association rules on the data, we found that some information is redundant and thus only one copy

## **Data visualization**

## **Extracting insights from data**

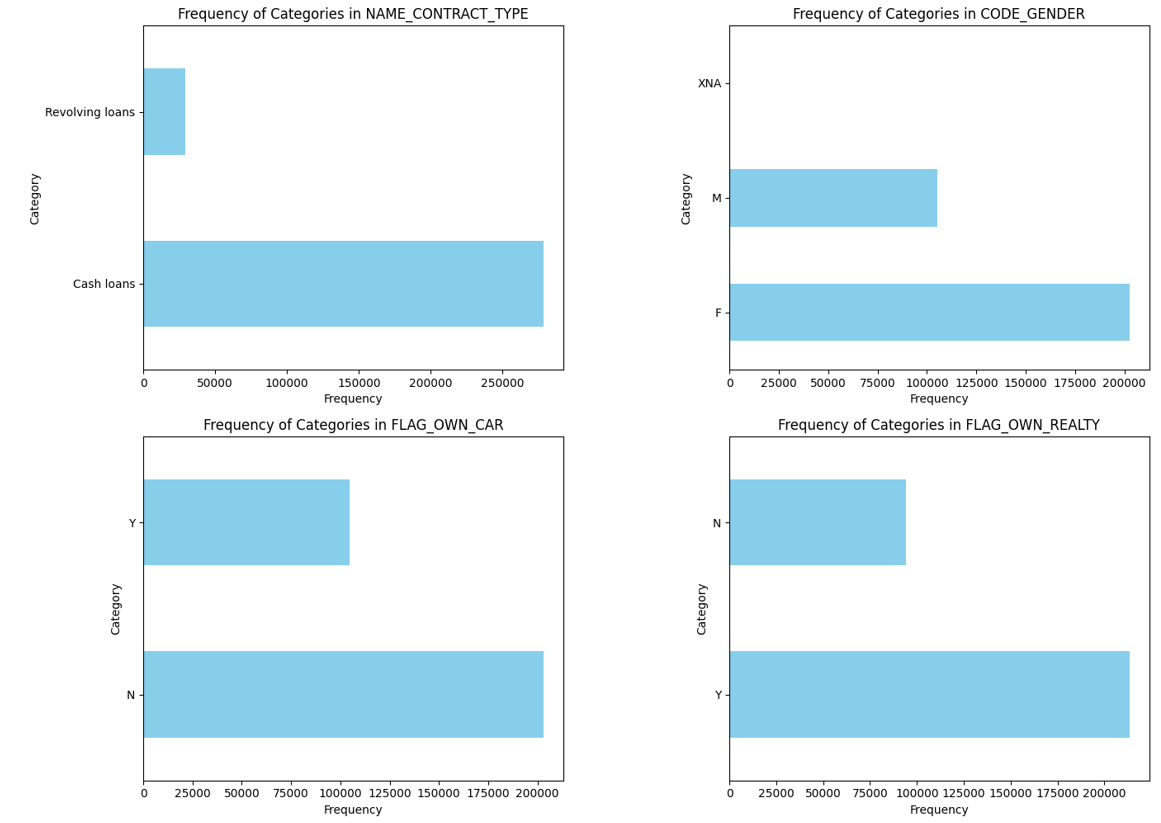
## **Model/Classifier training.**

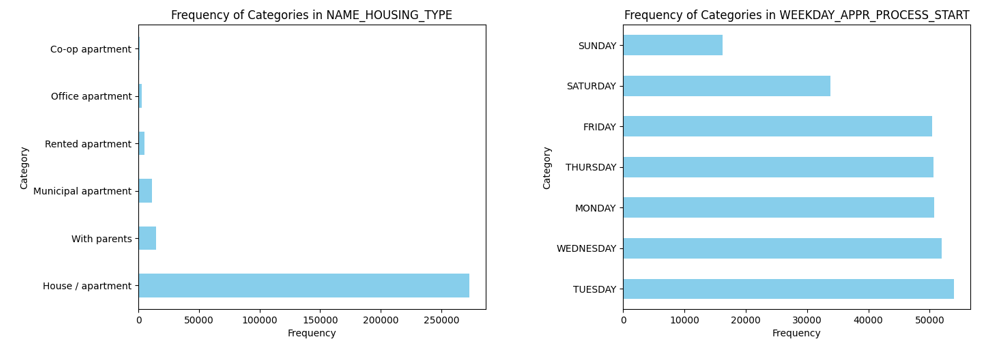
# Results and Evaluation

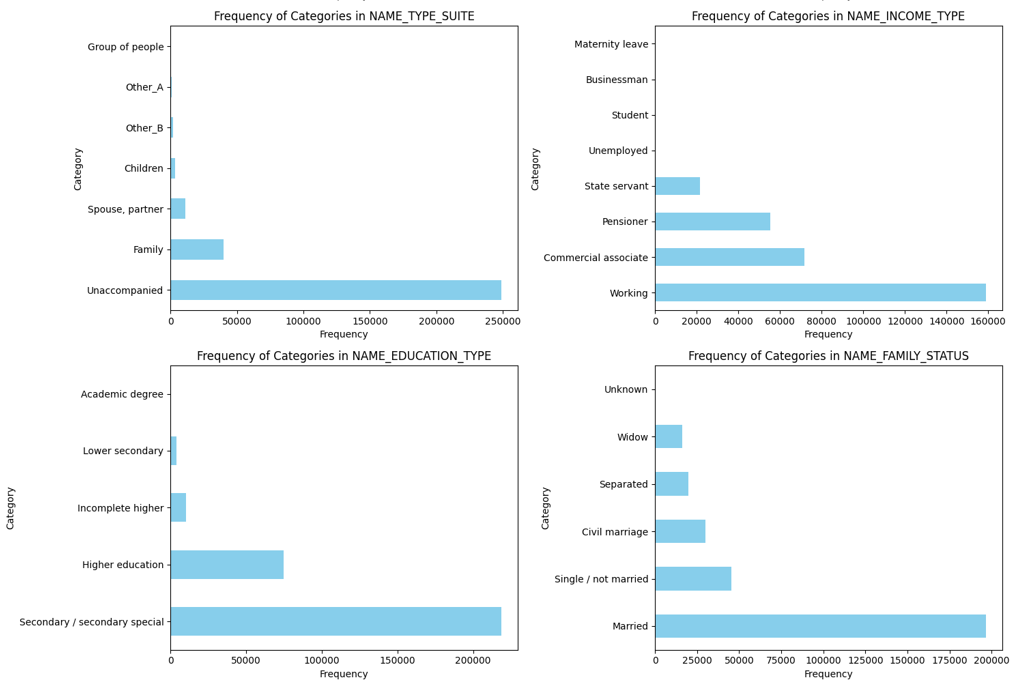
# Trials

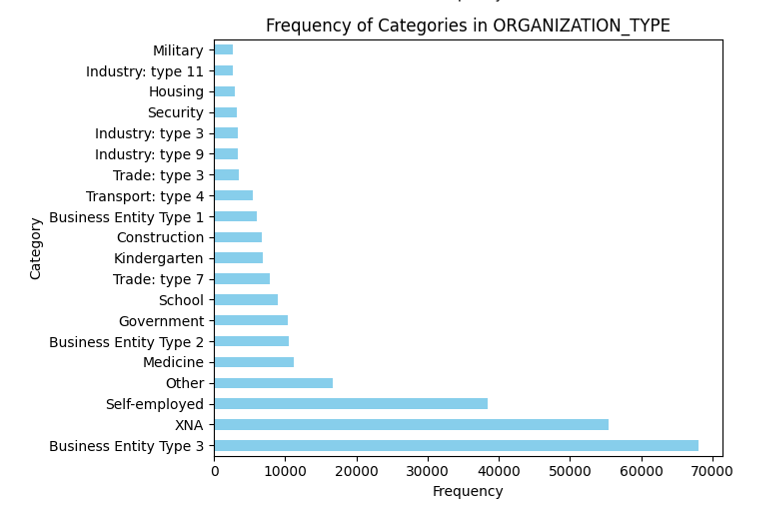
# Future work

- Draw distribution of categorical values



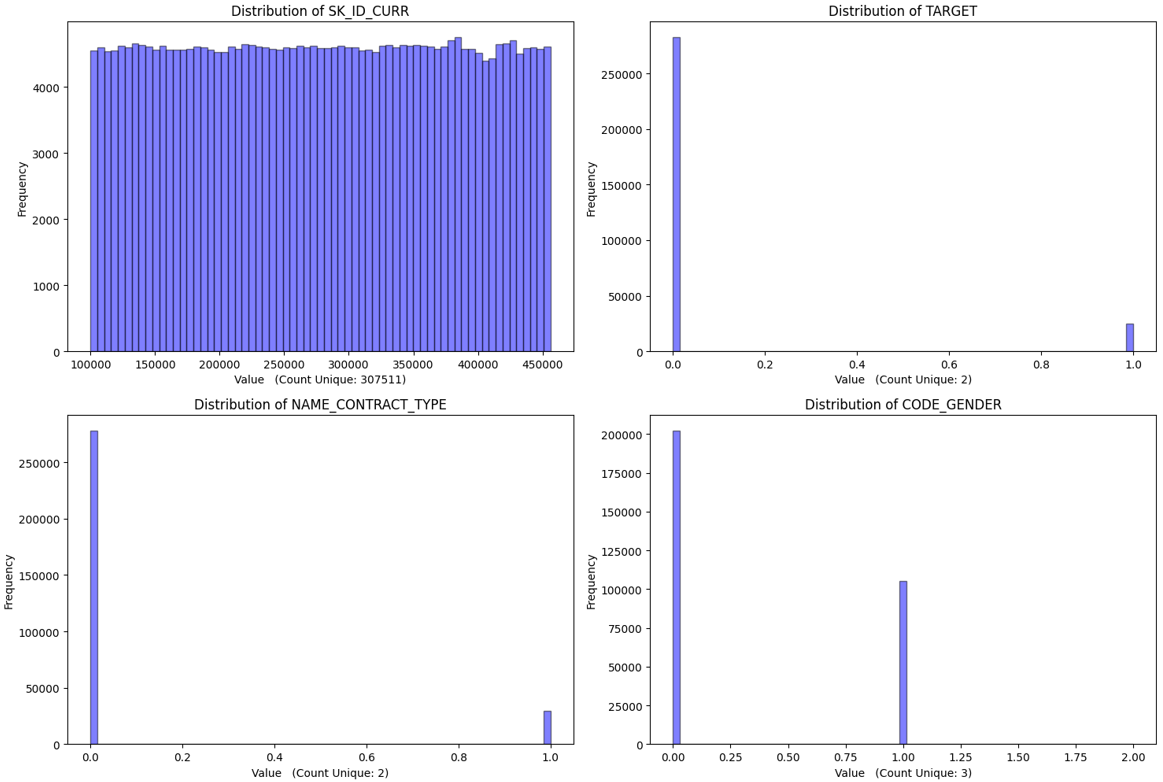


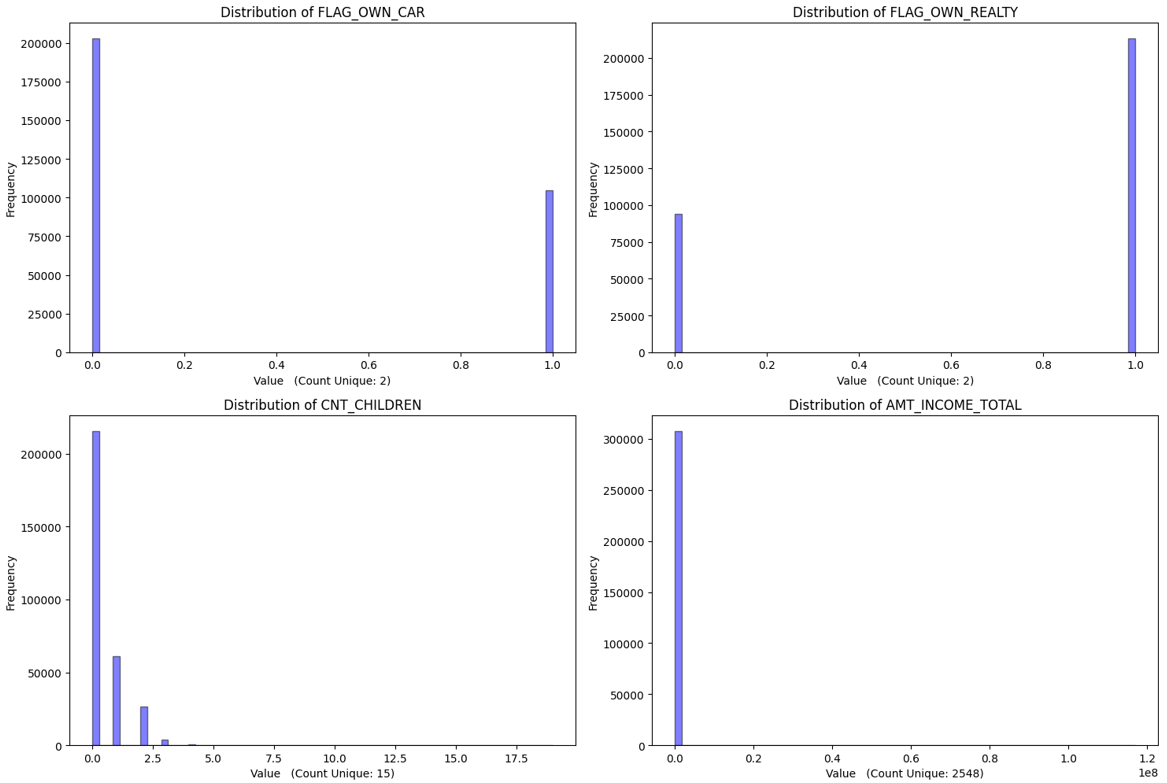


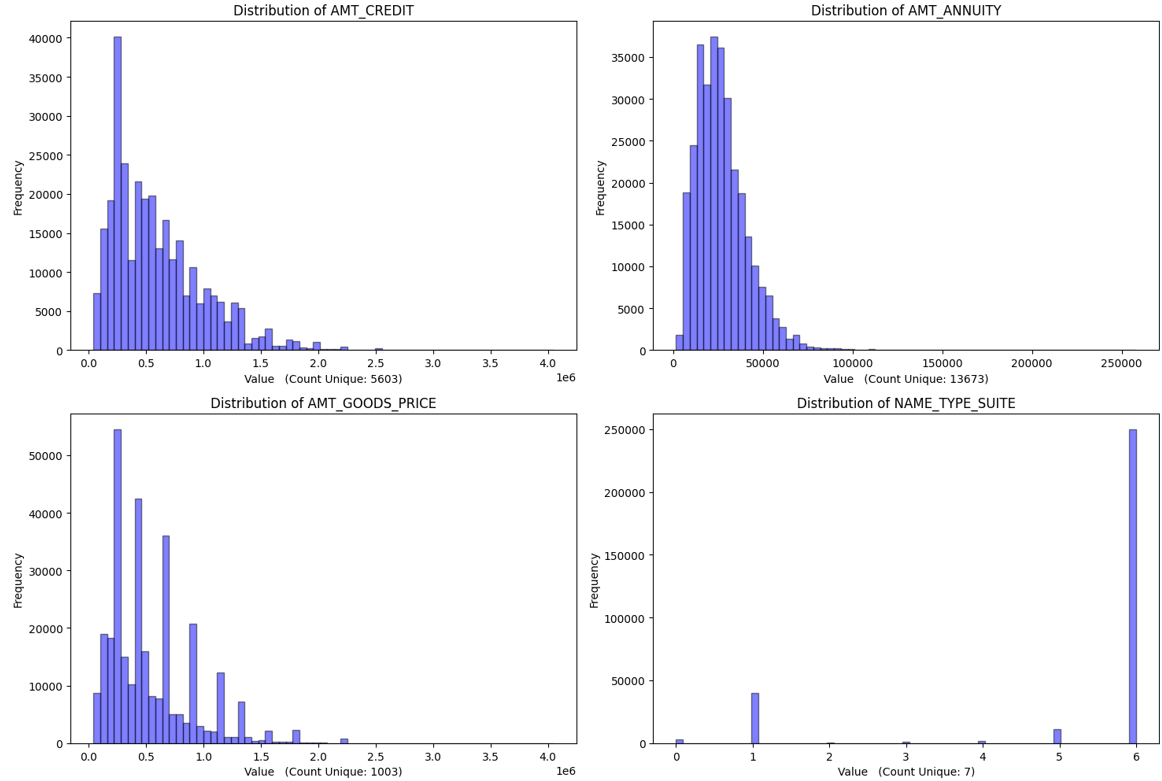


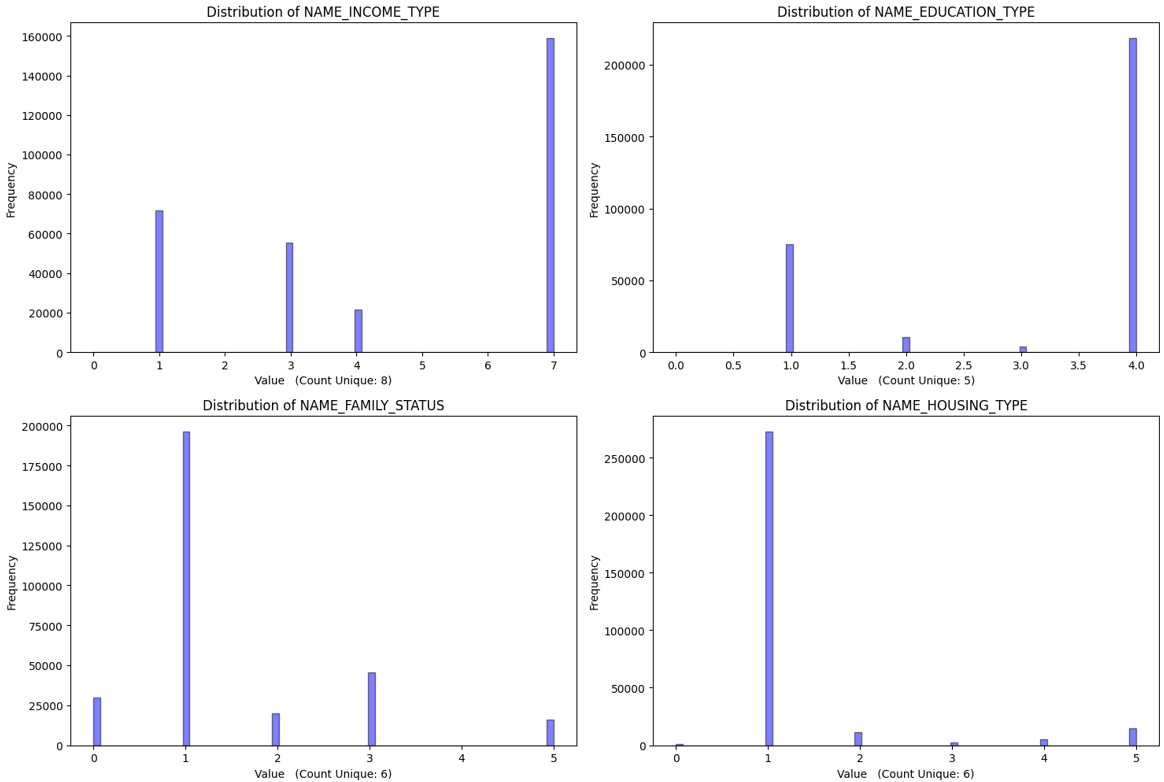
- Convert Categorical data to numerical values

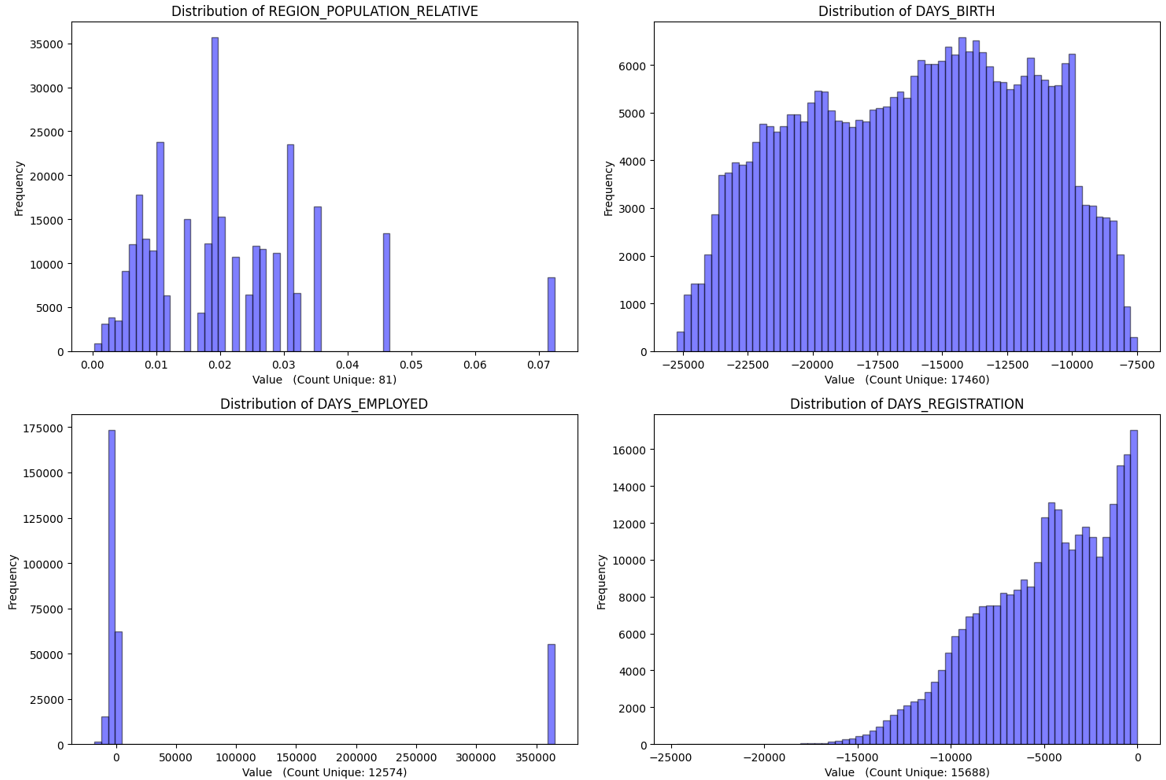
- Show Distribution of data

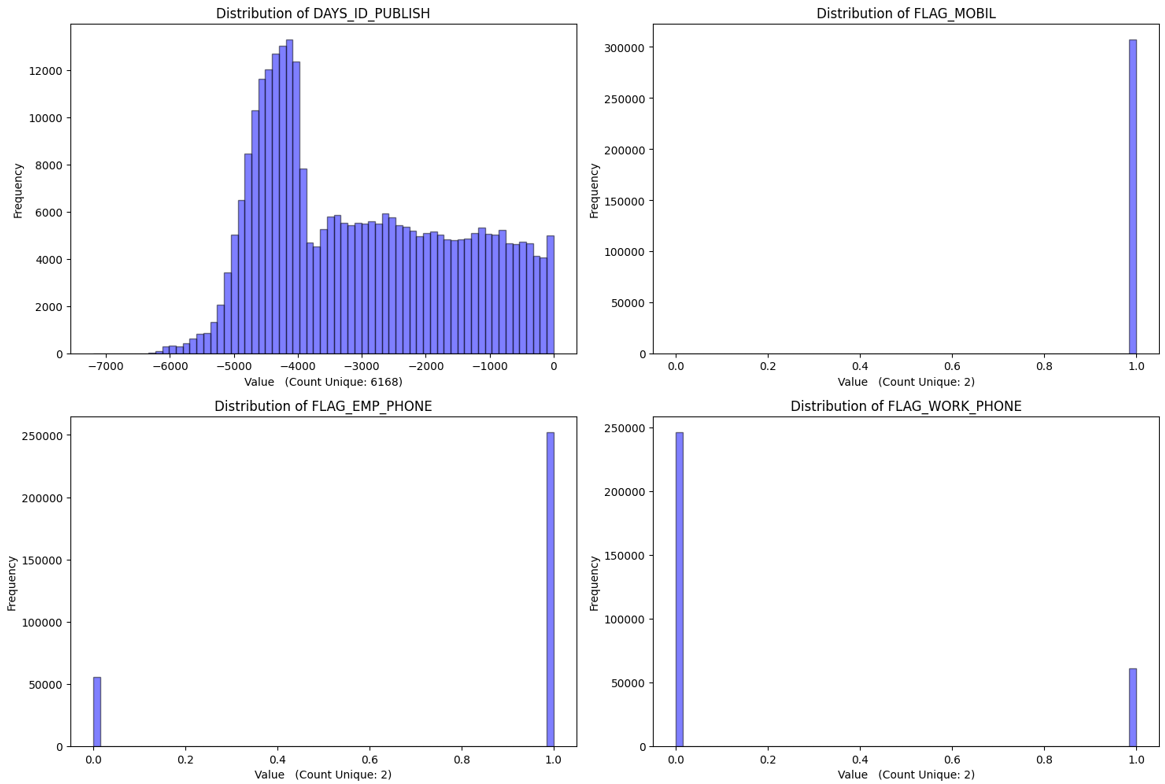


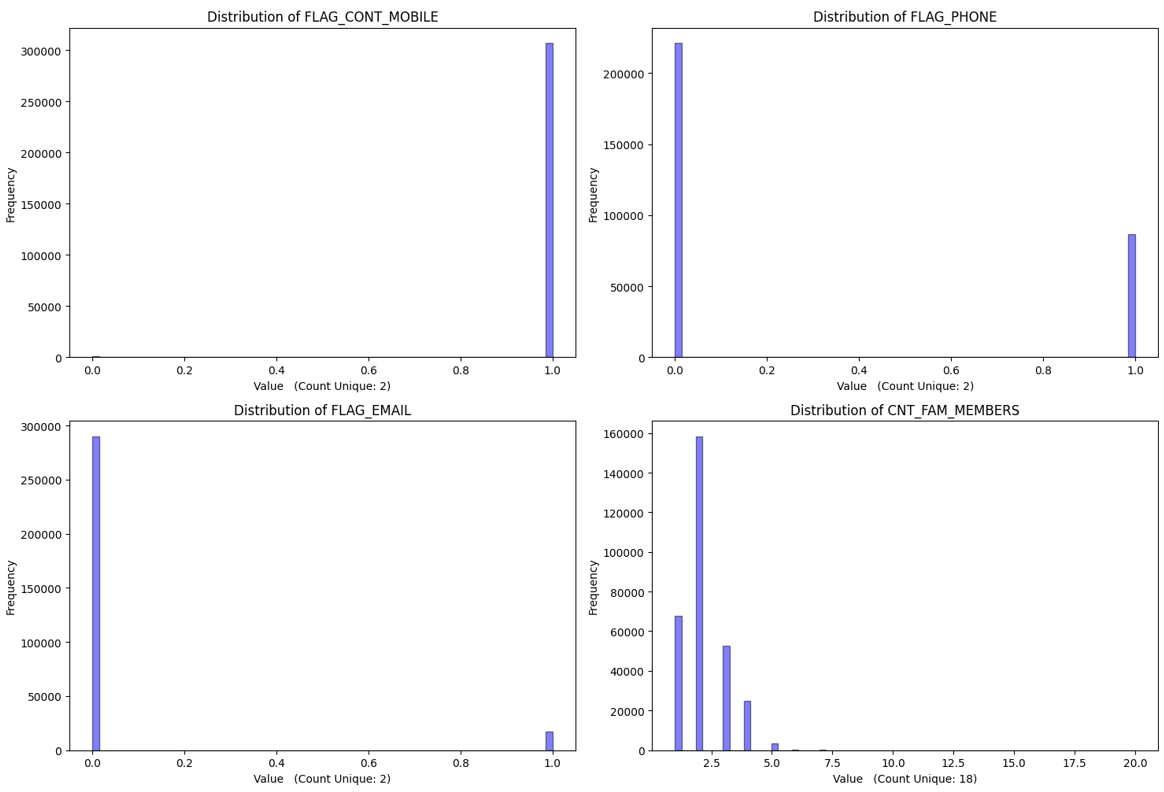


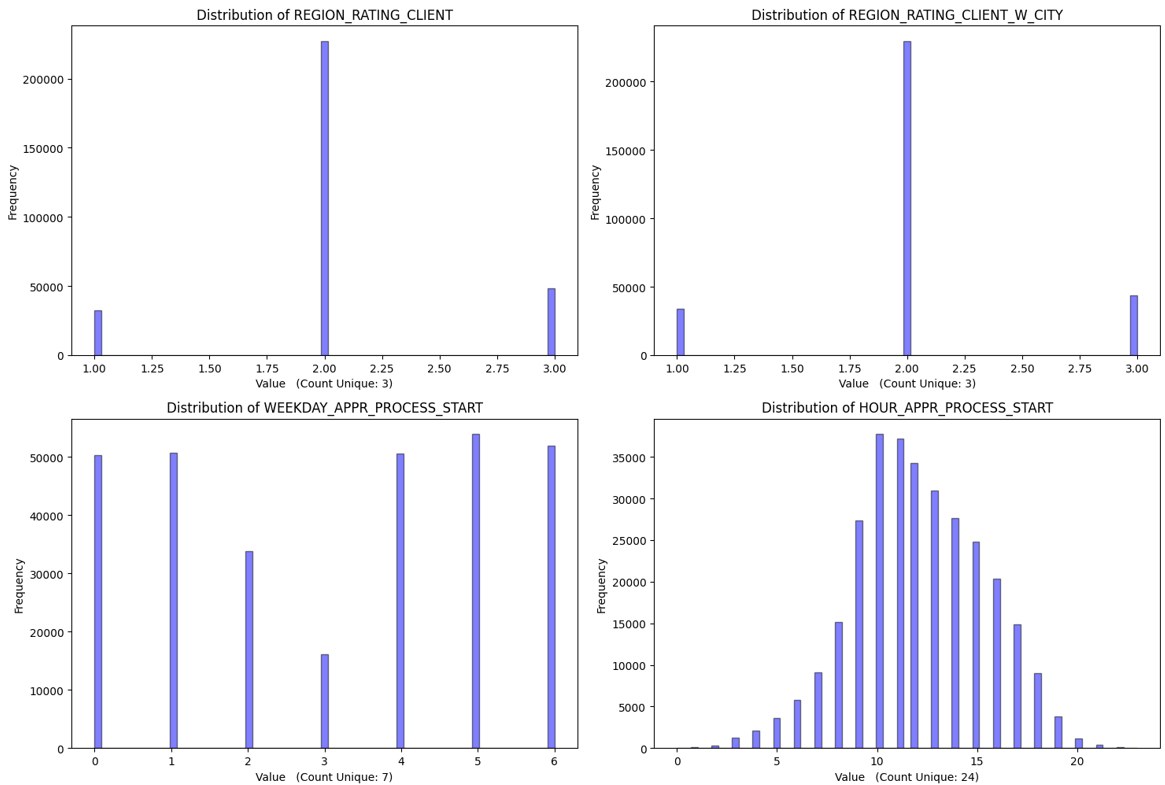


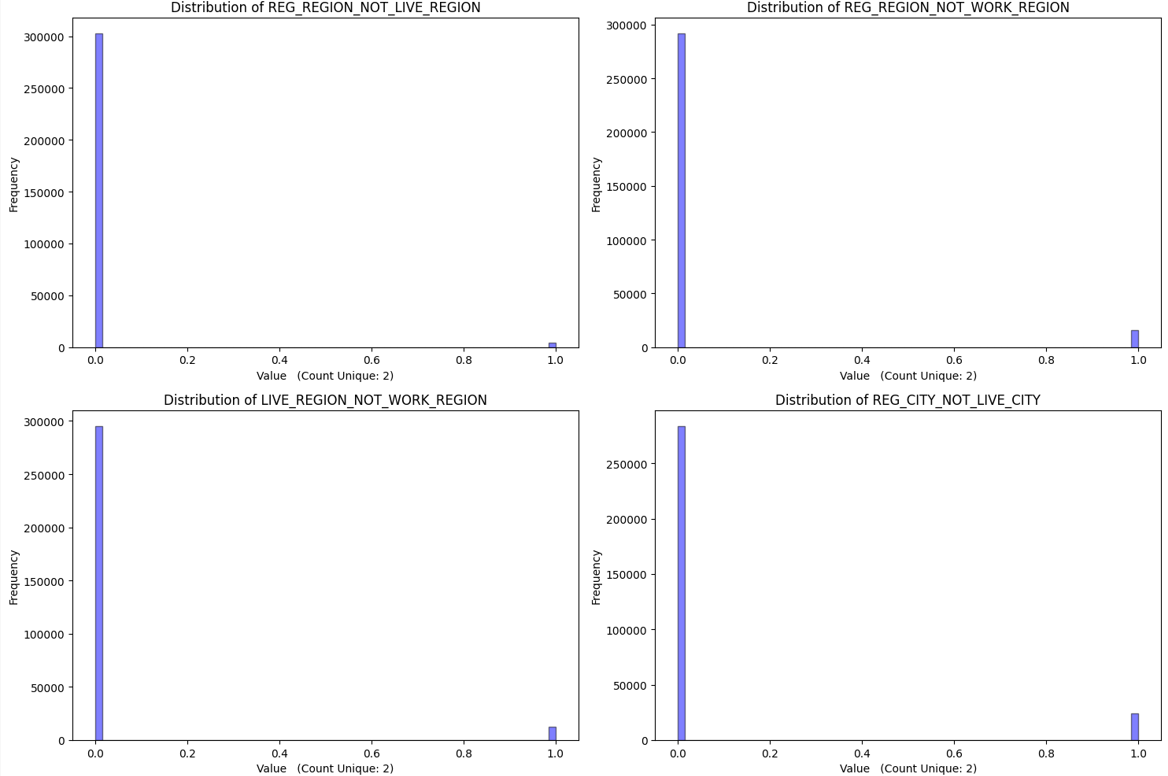


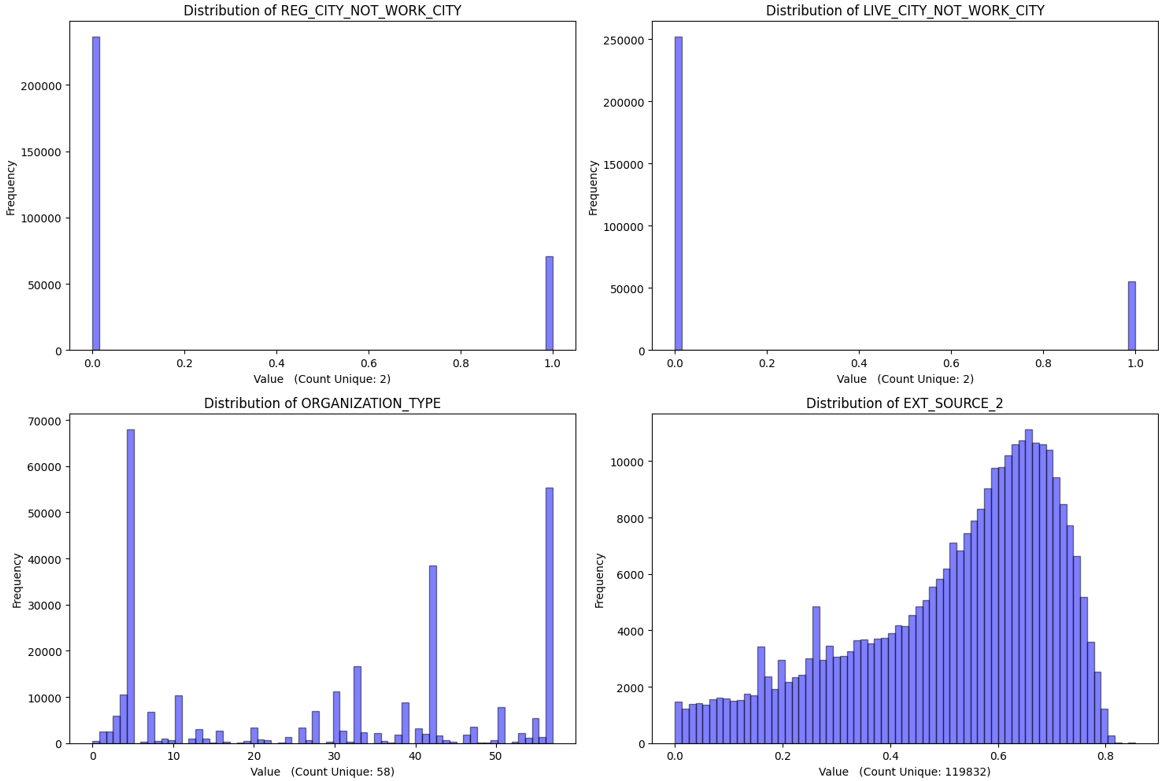


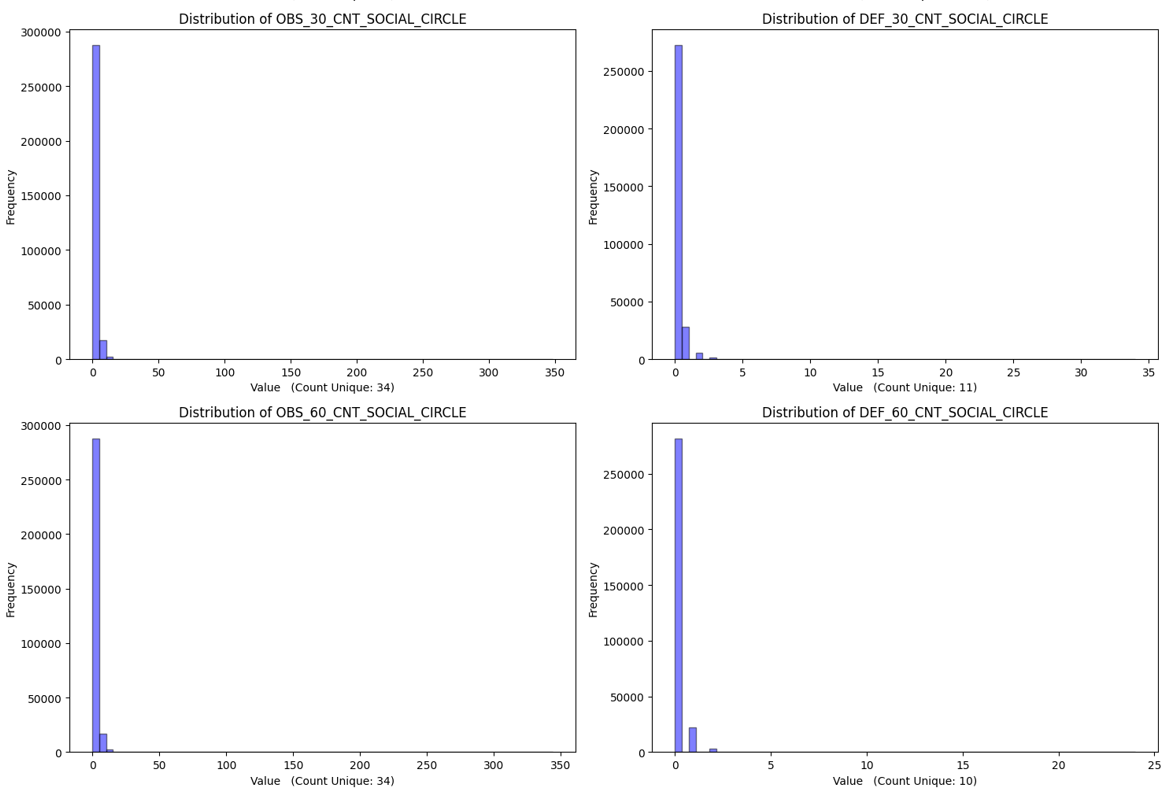


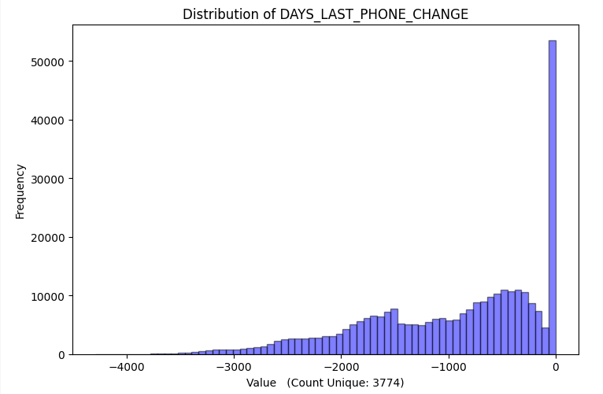




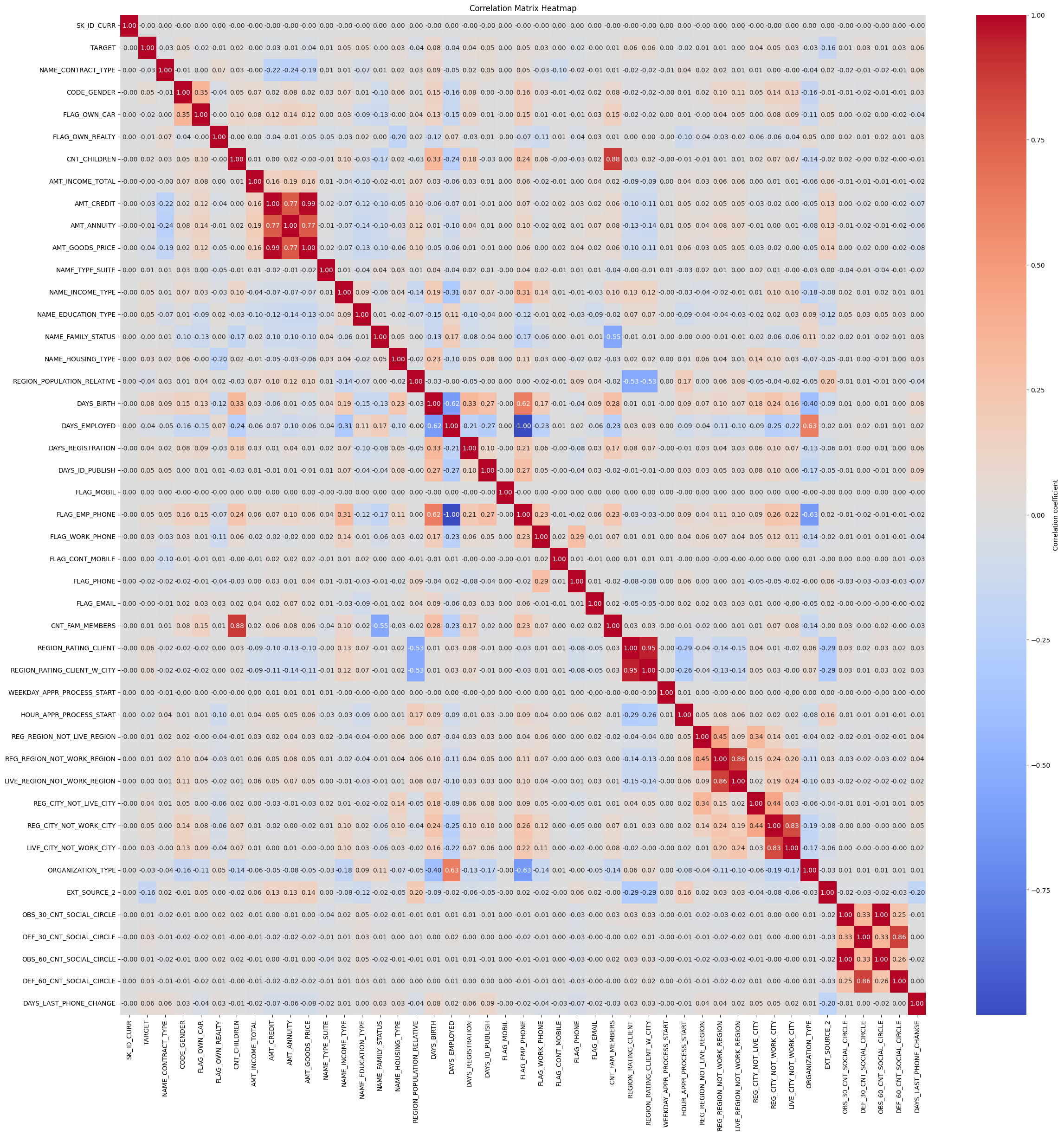




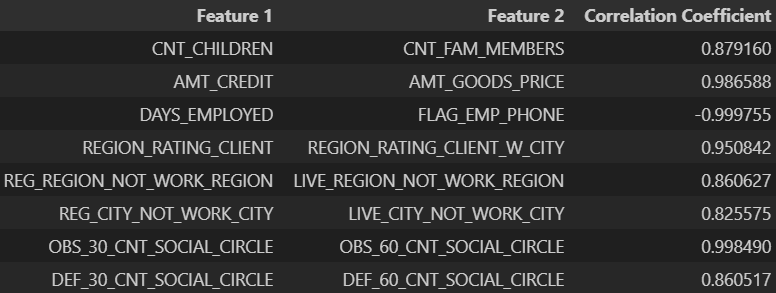




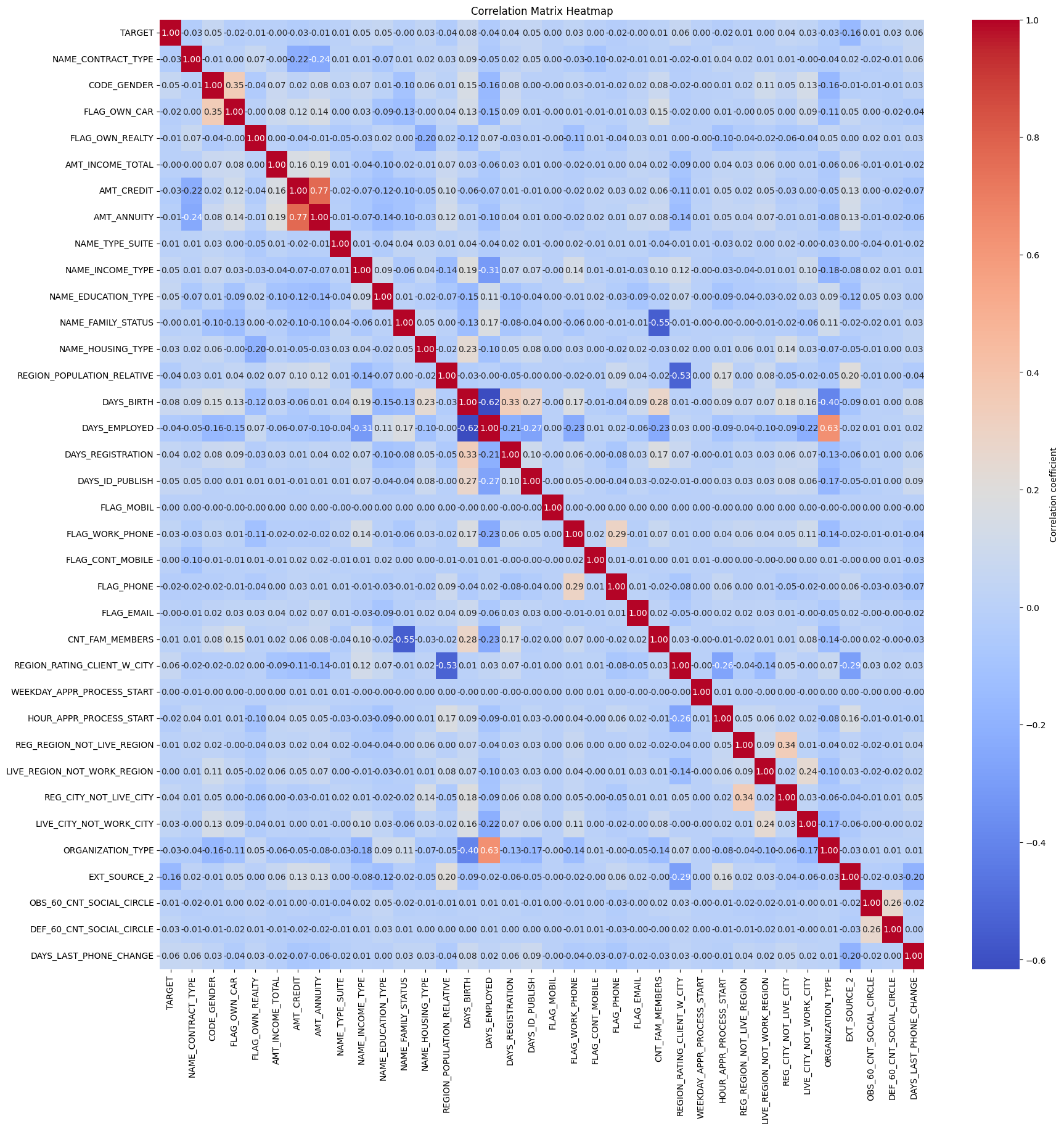
- Show the correlation matrix



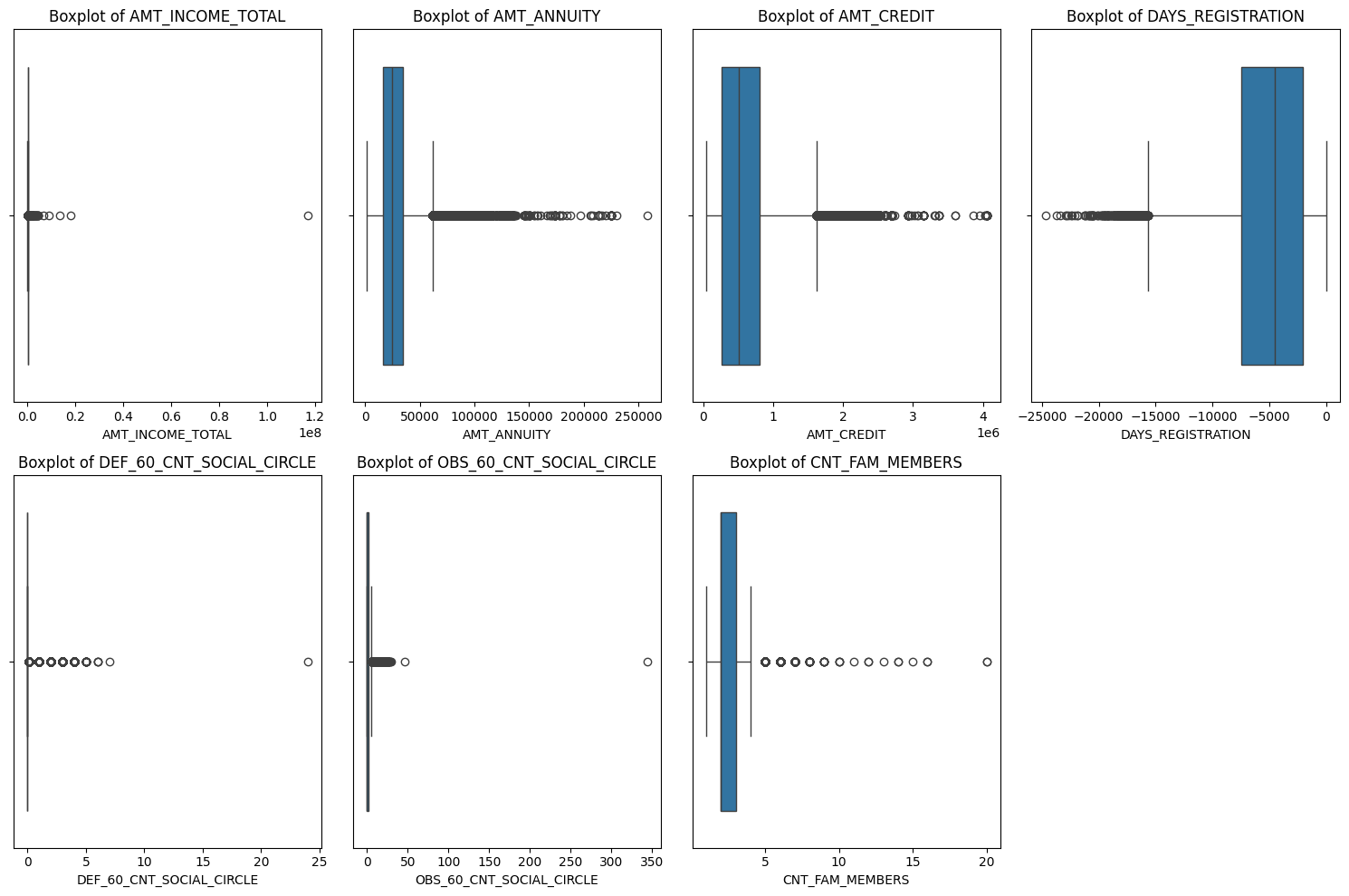
- Drop Highly correlated data



- Drop SK\_ID\_CURR since it represents the ID of loan

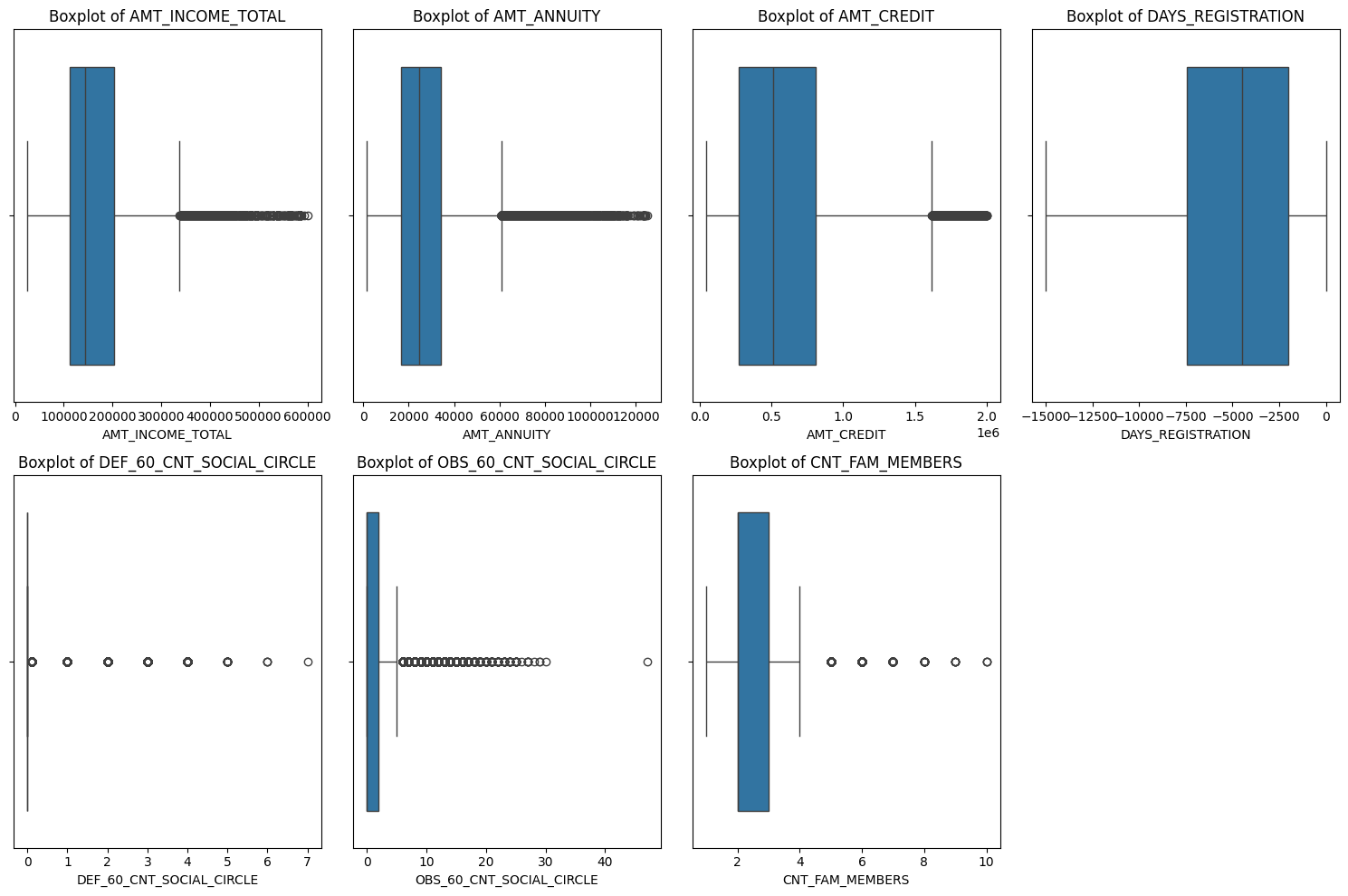
- Show new correlation matrix 

- Show Columns with high outlier



- Remove outliers

- Show new distribution of outliers

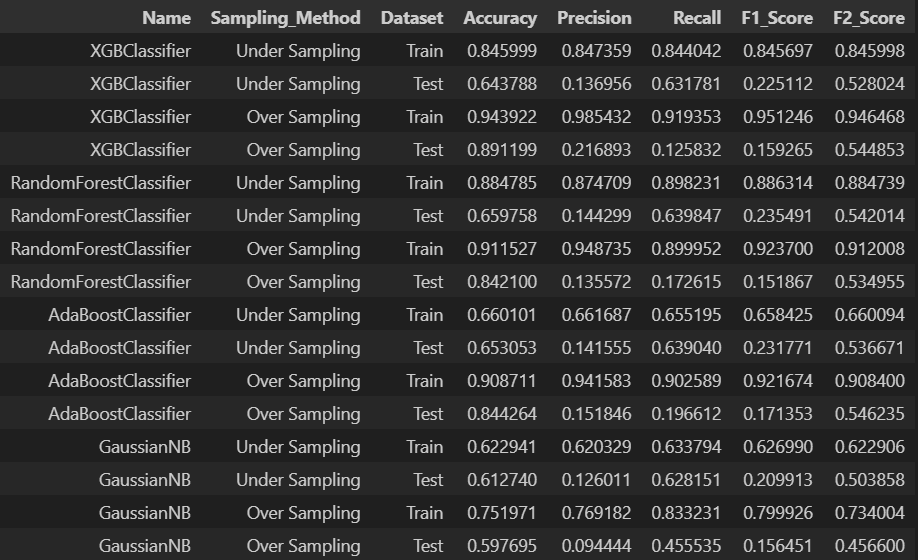


**Model Training and Evaluation**

- **Data Preparation**: we implemented a series of data preprocessing steps as identified during the exploratory data analysis (EDA). This included handling missing values, encoding categorical variables, addressing issues with highly correlated variables, and managing outliers.

- **Handling Class Imbalance**: To tackle the issue of class imbalance, we employed both undersampling and oversampling techniques. These methods helped to equalize the distribution of classes, thereby preventing model bias towards the majority class and improving the generalizability of the predictions.

- **Feature Scaling with Robust Scaler**: we used the **RobustScaler** from scikit-learn, which is particularly effective when the dataset is imbalanced, the Robust Scaler uses the median and the interquartile range for scaling. This makes it less sensitive to outlier values, ensuring that the scaling does not distort the actual data distribution.



Best Precision Model Name: **XGBClassifier**

Best Recall Model Name: **RandomForestClassifier**

Best F1\_Score Model Name: **RandomForestClassifier**

Best F2\_Score Model Name: **AdaBoostClassifier**

**Trials**

**- No Sampling Techniques**: Initially, models trained without employing any sampling techniques exhibited high accuracy. However, this approach resulted in significantly lower precision, recall, F1, and F2 scores, indicating a lack of model reliability in predicting minority class instances effectively.

**- SMOTE vs. SMOTEENN**: While experimenting with oversampling techniques, SMOTE produced marginally lower performance metrics compared to SMOTEENN. This slight underperformance could be due to SMOTE's inability to address the issues of class imbalance and overlapping class distributions that SMOTEENN targets.

**- Exclusion of Robust Scaler**: Omitting the Robust Scaler from the preprocessing pipeline led to modestly lower values across various performance metrics. So while the Robust Scaler enhances model robustness against outliers, its impact was not dramatically significant in this problem.