

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.

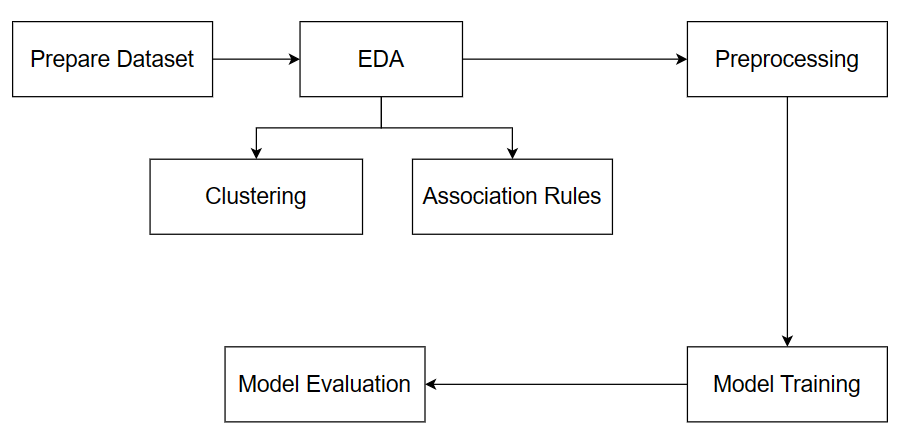
Top of Form

The Dataset used during our analysis :

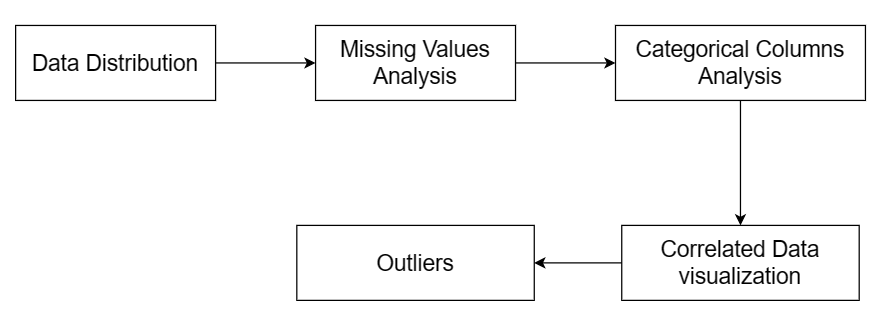
|  |  |
| --- | --- |
| Link | https://www.kaggle.com/datasets/mishra5001/credit-card |
| Number of features | 122 |
| Number of records | 307511 |

# Project Pipeline

- **Full Pipeline**



- **EDA**



# 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.

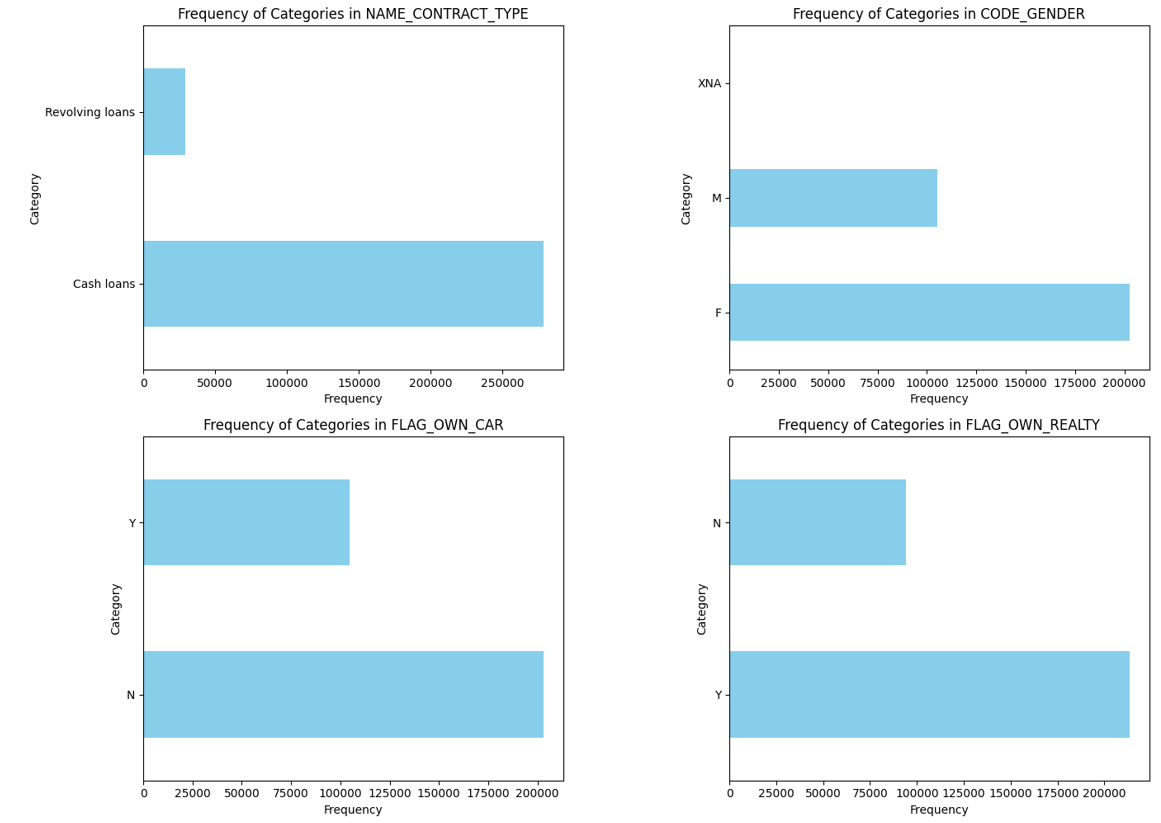
3) Some columns contain information related to the documents submitted by the customer when applying for the loan, we don’t have more details about these. columns, so we sometimes drop them (e.g. in clustering or in data mining).

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

5) After seeing the boxplots of the numerical features, we removed the outliers.

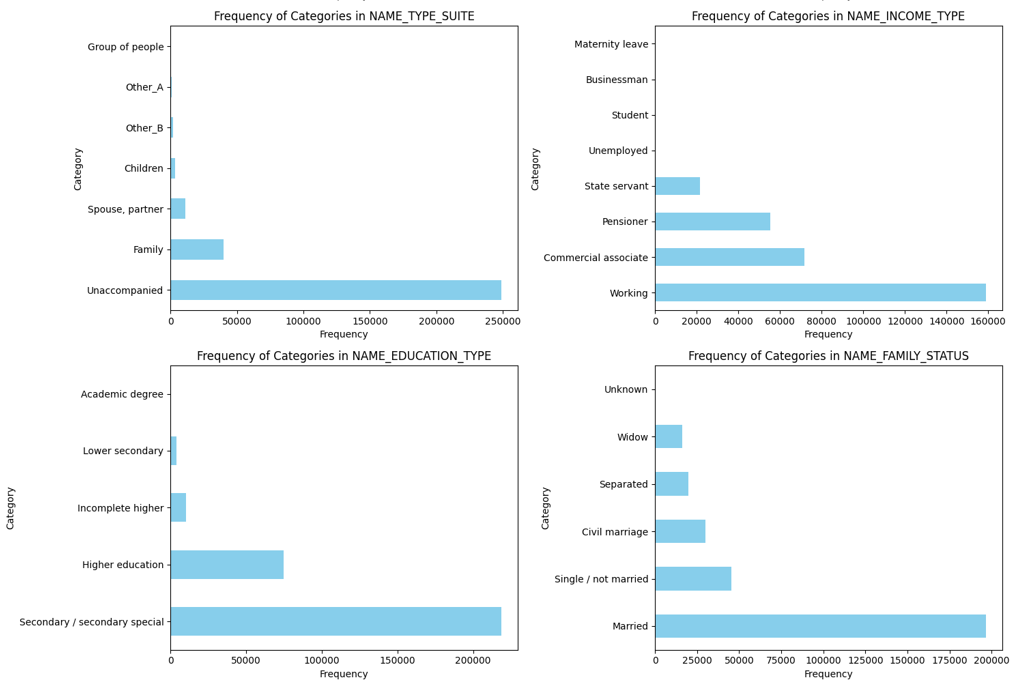
## **Data visualization**

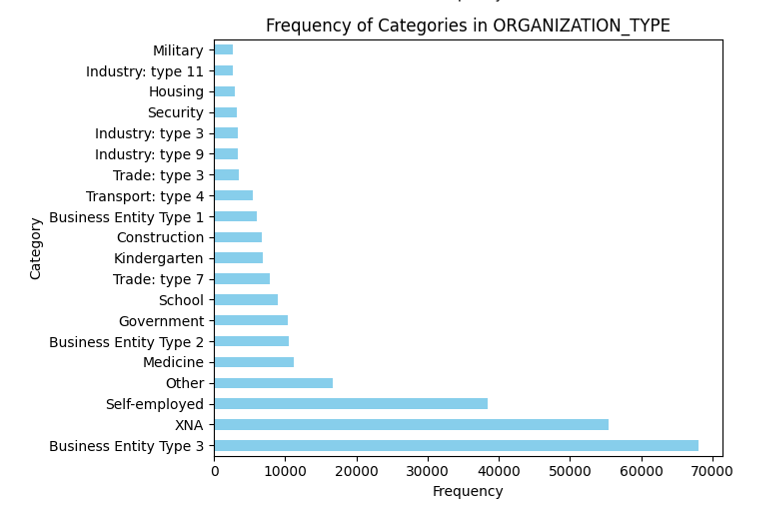
1) The distribution of categorical values



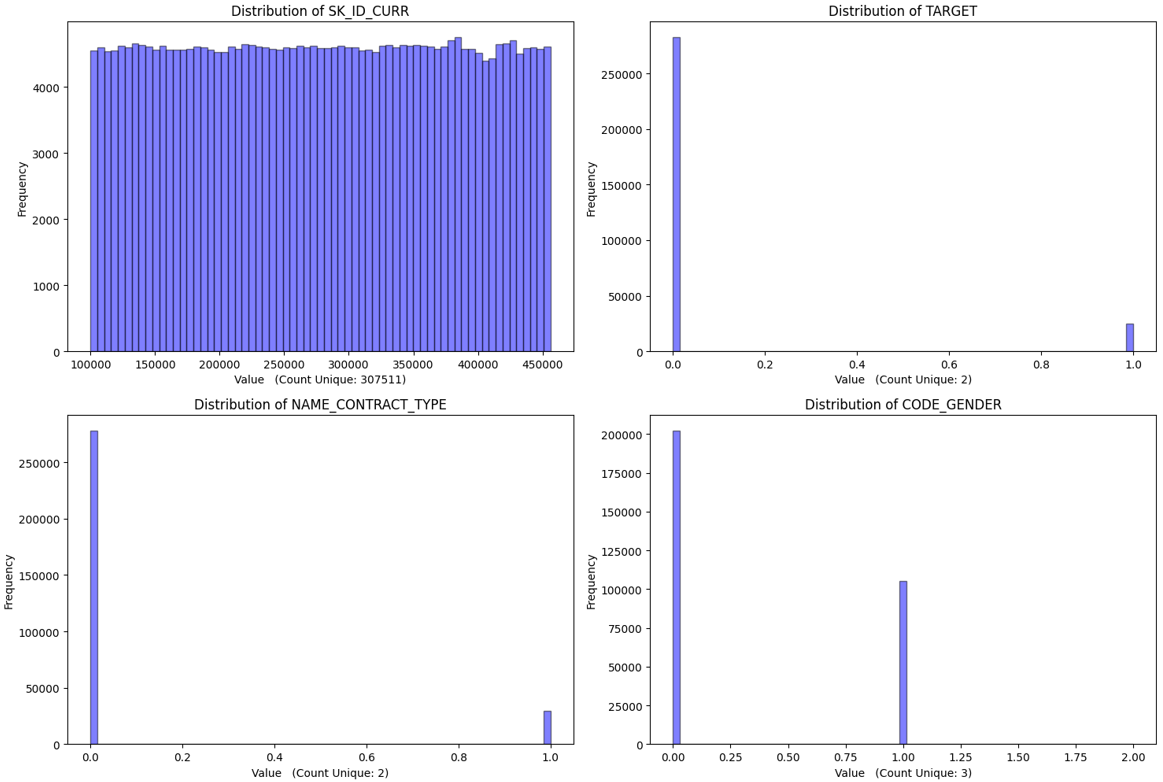
A close-up of a graph

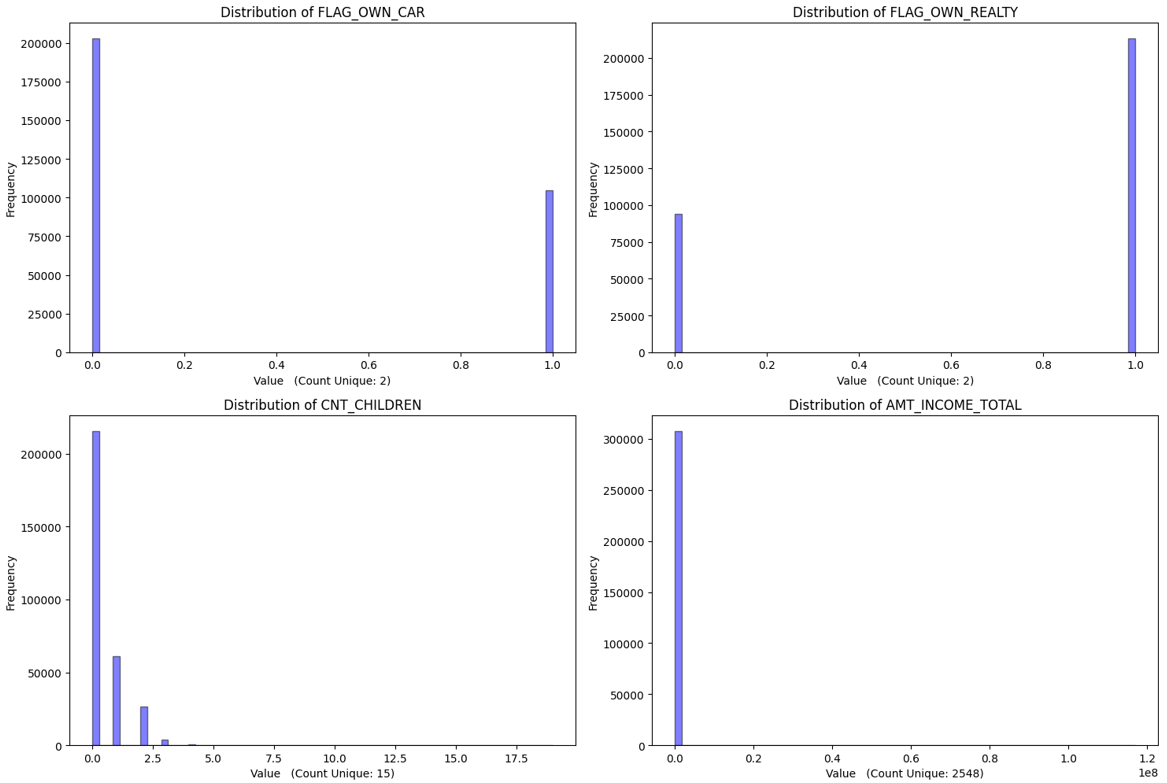
Description automatically generated

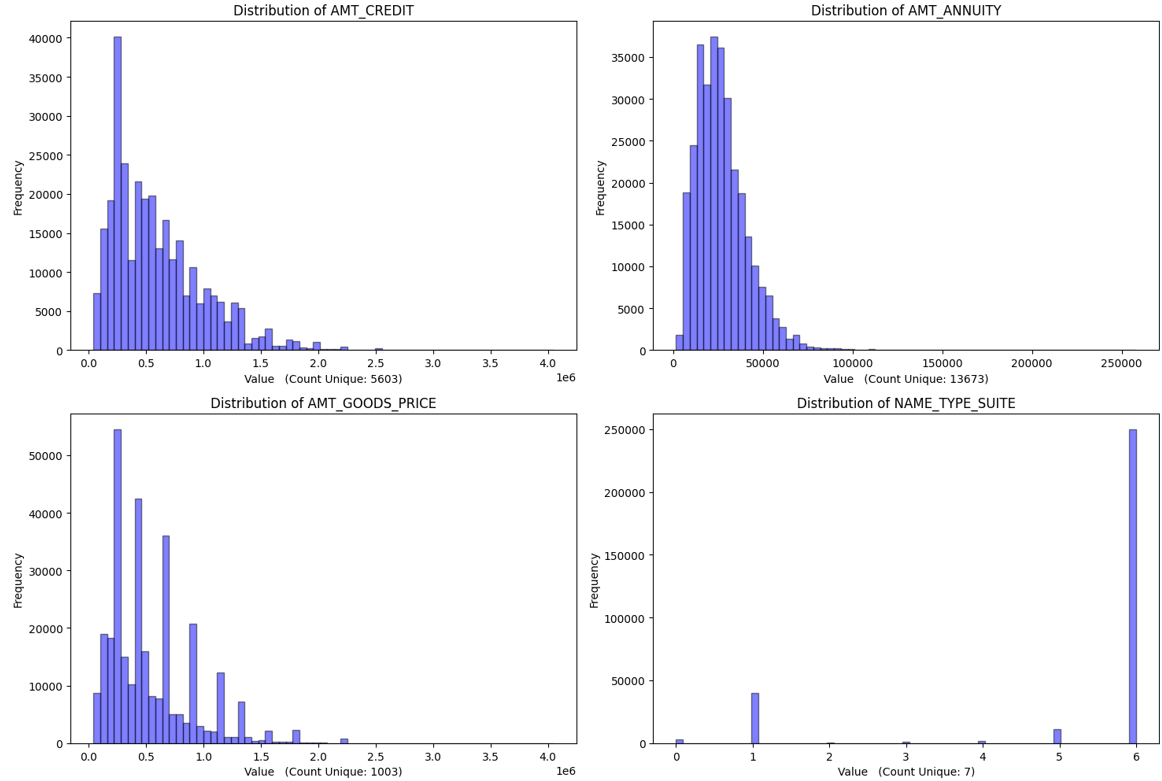


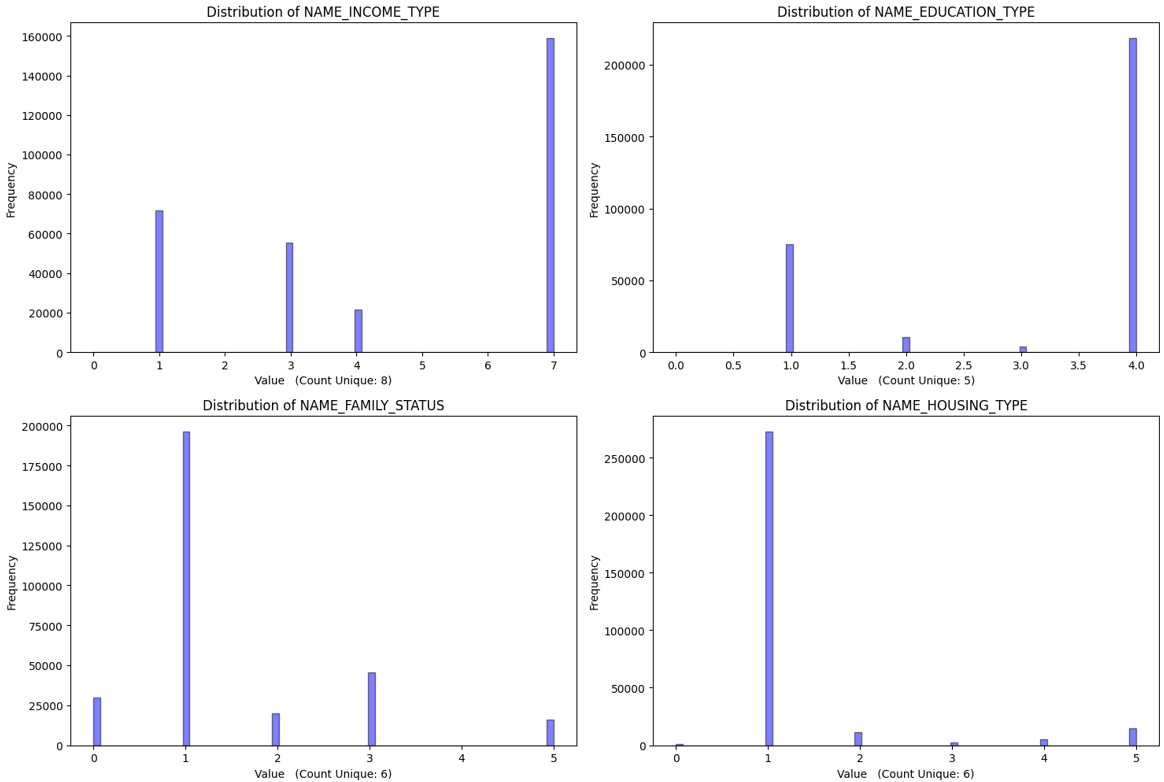


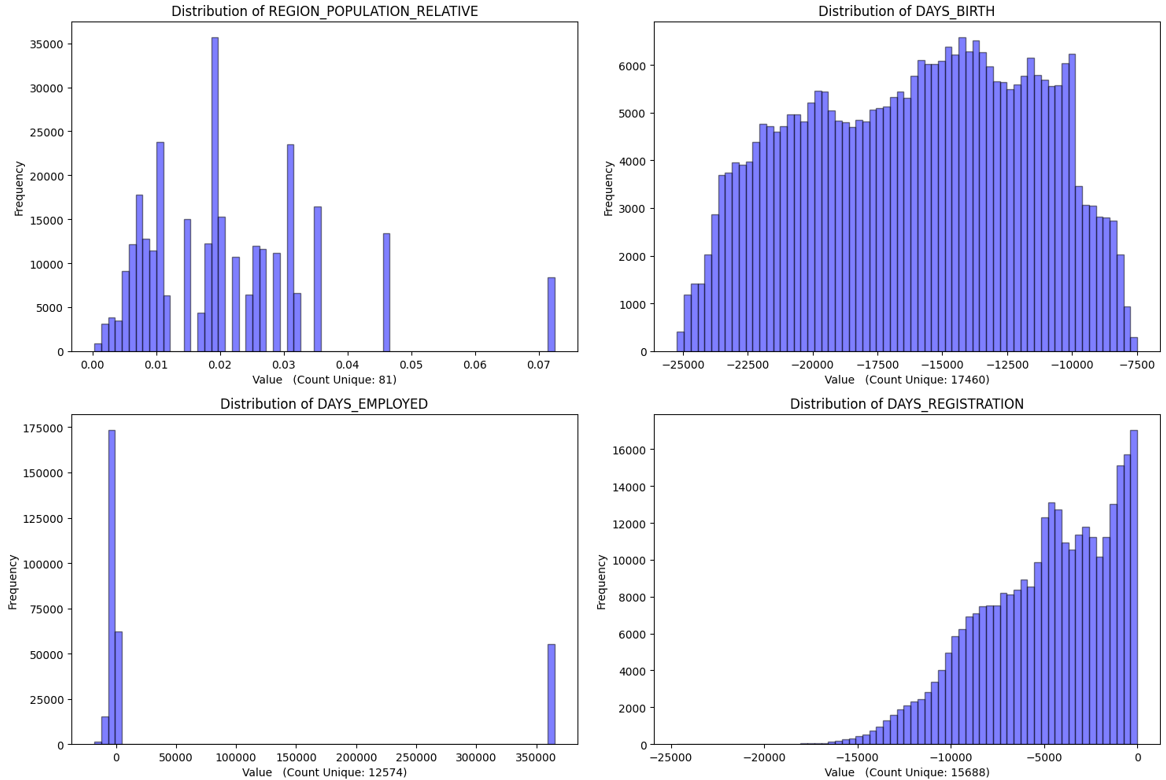
2) The Distribution of numerical data

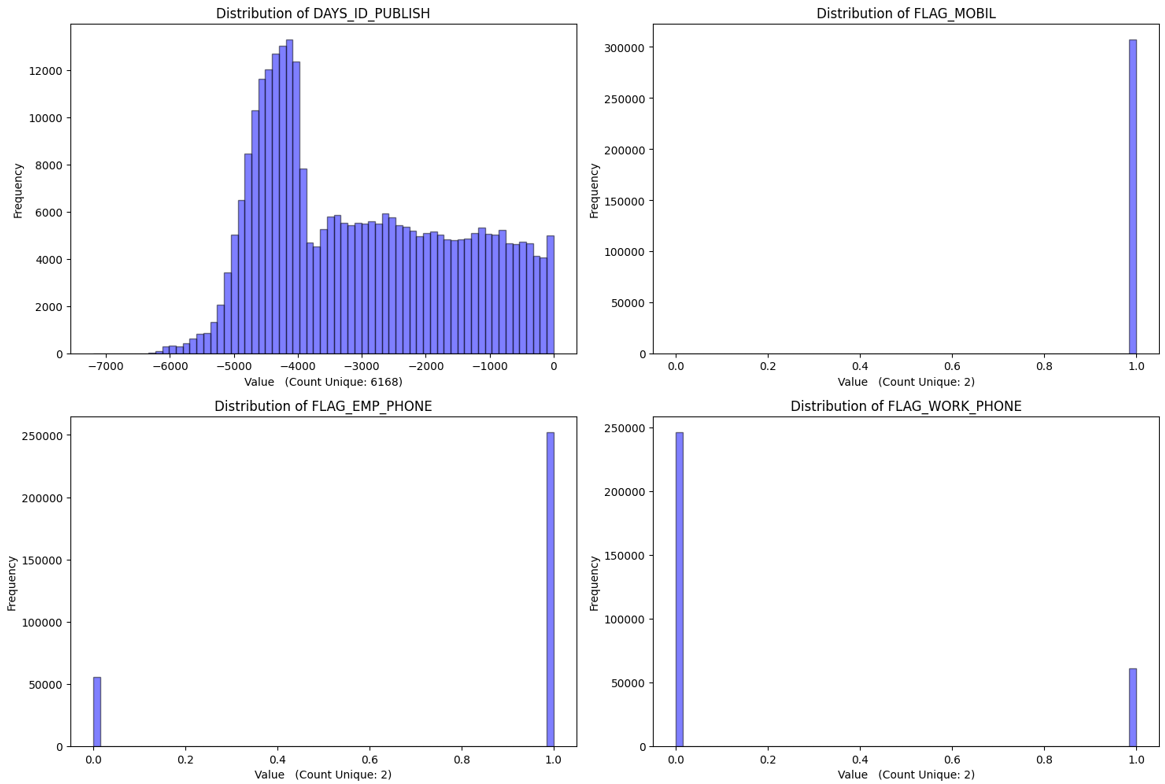






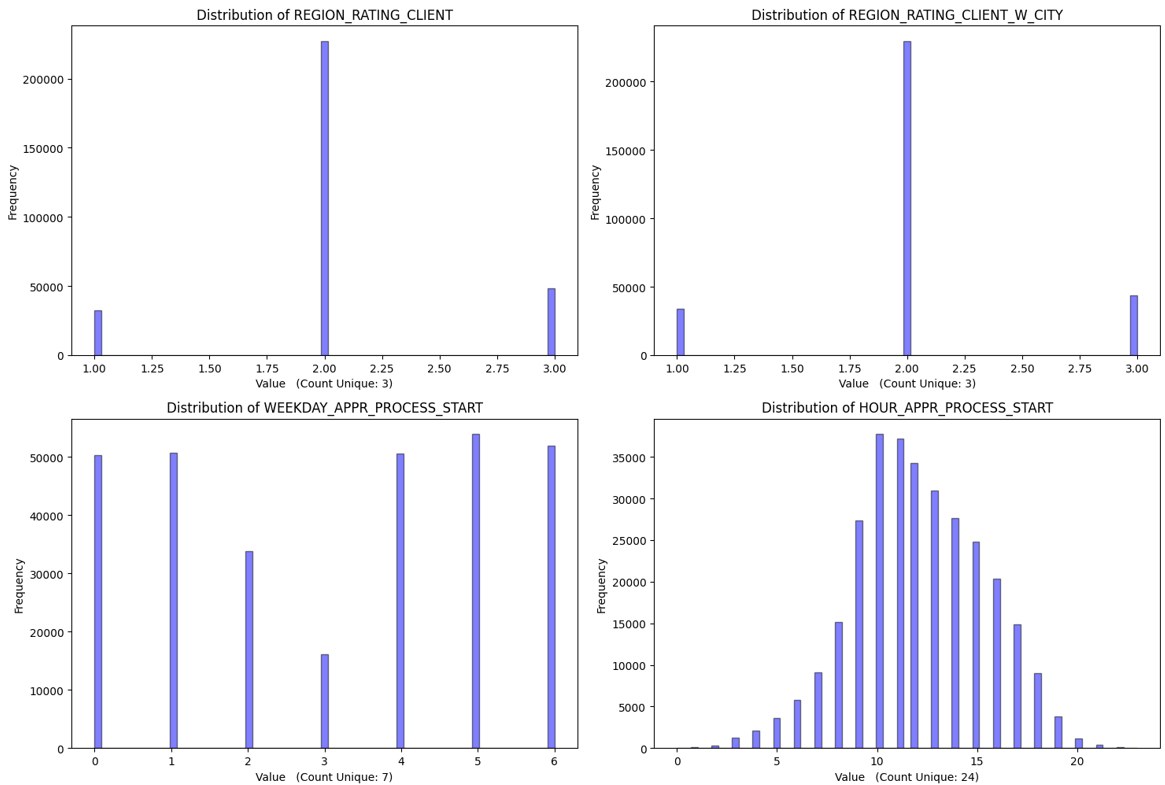


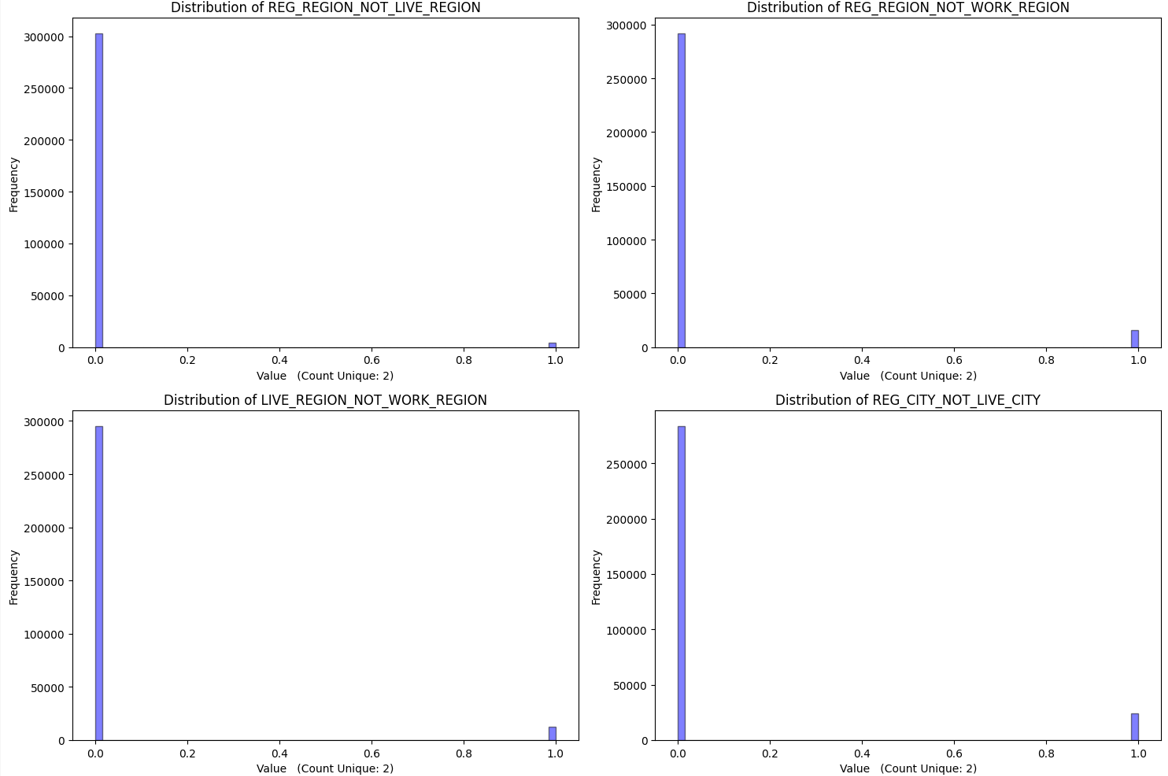


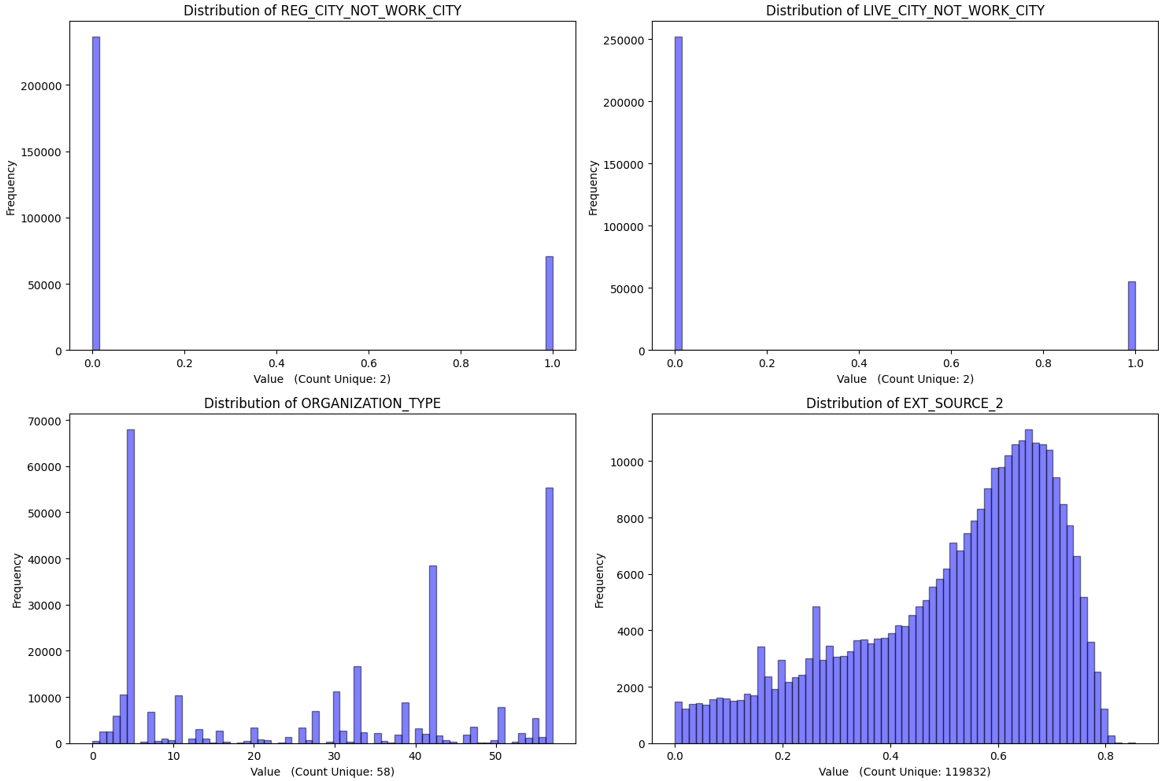


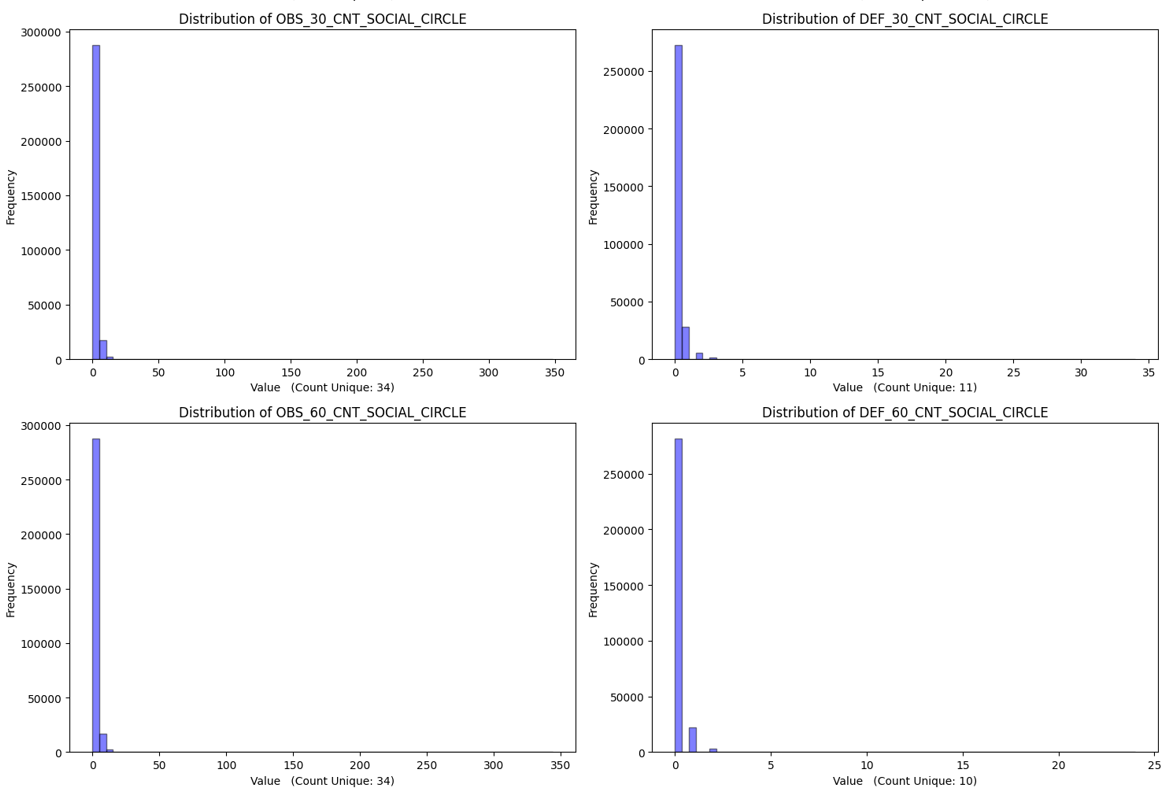
A graph of a number of data

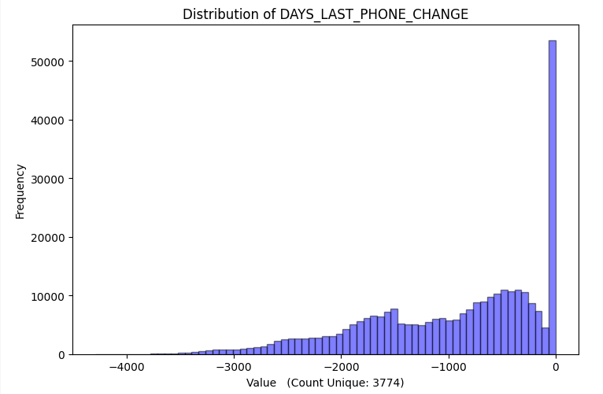
Description automatically generated with medium confidence



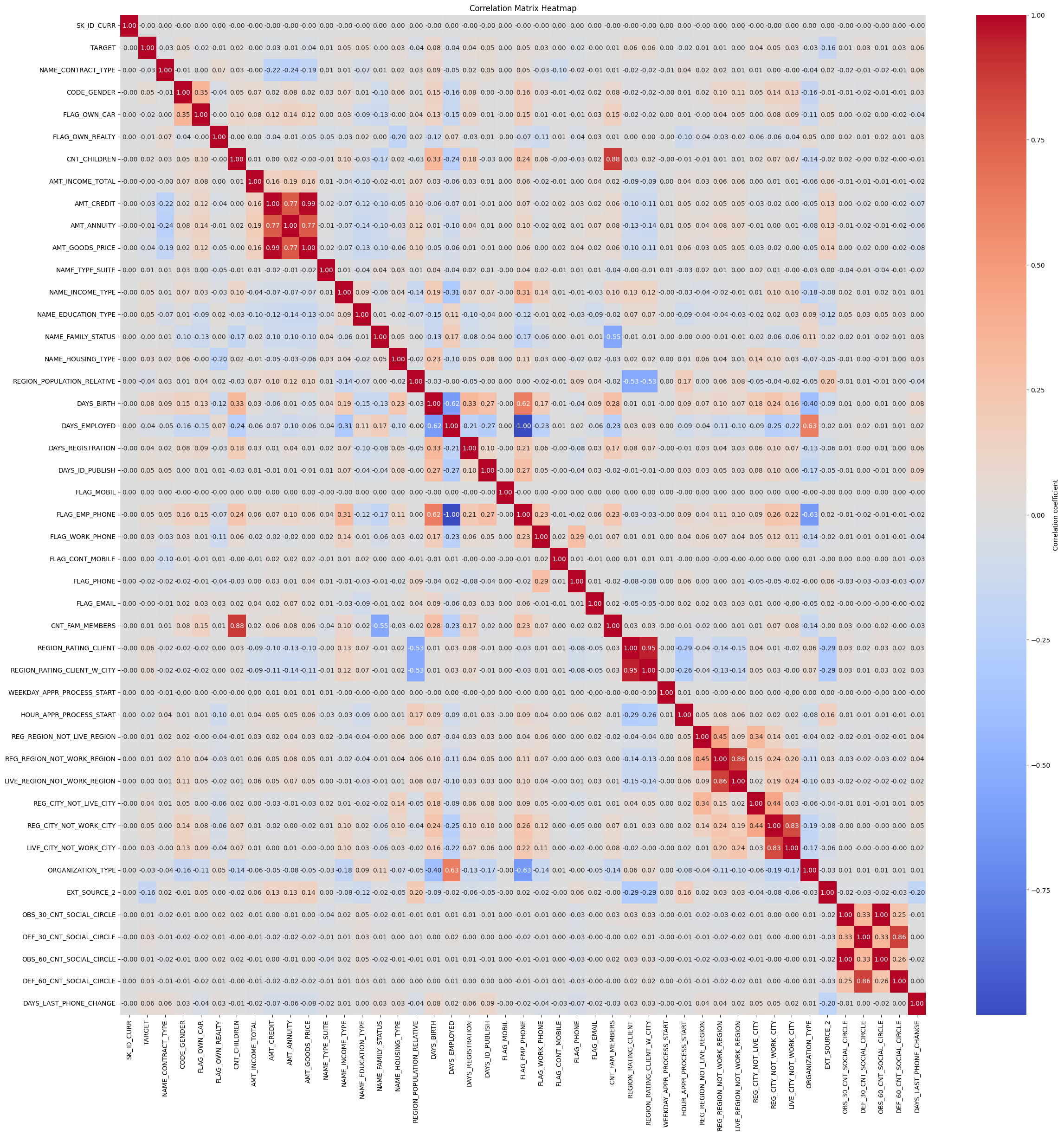








3) The correlation matrix

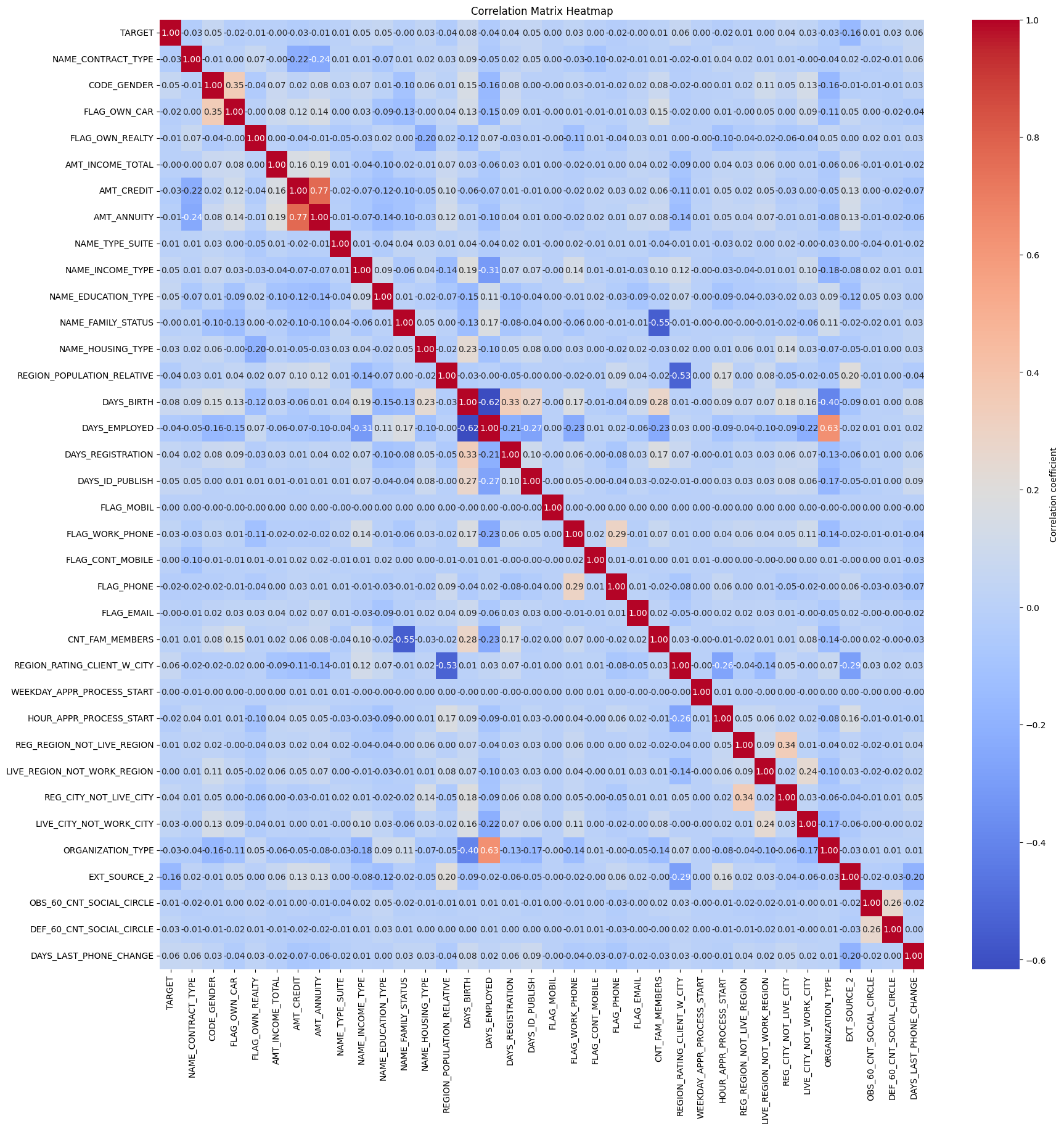


4) Drop Highly correlated data

A black screen with white text

Description automatically generated

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

6) The new correlation matrix after removing correlated data7) After investigating the boxplots of the features, it was found that some columns contain outliers as below and thus we removed them

A collage of a graph

Description automatically generated

8) After removing the outliers, the boxplots of these features become more acceptable

A collage of a graph

Description automatically generated with medium confidence

## **Extracting insights from data**

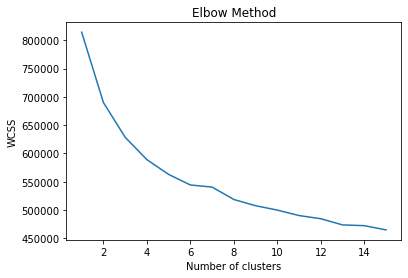
1) The data is clearly unbalanced with 91% of the records with a target value of 0 (false)

2) The data has many binary flags and categorial data with a limited amount of numerical features

3) The flag documents are not explained well and are removed in clustering

4) Female clients higher than male clients for loans

5) Most of the clients applied for Cash loans over revolving loans, Clients who take cash loans often face payments difficulties compared to those who choose revolving loans

6) Clustering the data using k-means showed the follow elbow graph  
  
so we choose 8 as the best number of clusters and the records were distributed as follows  
A graph with blue bars

Description automatically generated

7) clustering didn’t give much insight to the data because the features were distributed over the clusters

8) Association rules were used to find more insights for the relationships between the features,

Some rules are given in the table below

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Antecedents | Consequents | Support | Confidence | lift |
| Female Clients | No defaults | 0.61 | 0.93 | 1.011 |
| Married Clients | No defaults | 0.59 | 0.92 | 1.006 |
| Cash Loans | No Defaults | 0.83 | 0.92 | 0.997 |
| Clients with Realty | No defaults | 0.64 | 0.92 | 1.001 |
| Married Clients | Apply for cash loans | 0.58 | 0.91 | 1.005 |
| Income between 0 and 200000, with education ‘Secondary / secondary special’ | Apply for cash loans | 0.5 | 0.91 | 1.005 |
| Loan annuity between 0 and 30000, with loan type cash | No Defaults | 0.52 | 0.91 | 0.99 |
| Credit amount of the loan between 0 and 600000 | No Defaults | 0.55 | 0.91 | 0.99 |

A blue dots on a white background

Description automatically generated

A blue dots on a white background

Description automatically generated

A blue dots on a white background

Description automatically generated

A blurry image of red dots

Description automatically generated

## **Model/Classifier training.**

### **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 under sampling 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.

### **Map Reduce with Bayes Gaussian:**

We implemented 2 MapReduce functions one for calculating the mean of each class and another one for calculating the covariance since the mean is required in the calculations of the covariance. Then we predict the results using Gaussian Naïve Bayes.

### **Mapper 1**

Each row of the dataset is turned into a key-value pair. The key is the 'TARGET' (class label), and the value is a dictionary containing all other attributes with their values.

* **Reducer 1**

Records having the same key (class label) are combined. The values (attributes) are summed up. This step effectively computes the total sum of each attribute for each class.

* **Mapper 2**

Each row is again mapped to a key-value pair where the key is the class label, and the value is another dictionary. This dictionary computes the product of differences from the mean for this attribute and all other attributes.

* **Reducer 2**

Similar to the reduce step 1, but this time it aggregates the results of attribute pair products, summing them up for covariance calculation.

After each reduce task the values are divided by the counts of each class

# Results and Evaluation

## ResultsA screen shot of a black screen Description automatically generated

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.

## **Trying combining all Flag documents:**

Since Flag documents didn’t provide any meaning we tried to combine them into one field and trained the model but didn’t affect the performance Metrics.