Financial Data Analysis and Visualization of Loan Applications

23.12.2023

Our Group

Agourram Sami
El Amrani Soufiane
Idrissi Amine
Omar Bouhadi Farouq

Supervised by:

Pr. ASSKLOU Abdelouahed
Pr. BOUISSALI Yousra

Outline

INTRODUCTION

In the dynamic landscape of financial institutions, making informed decisions is crucial to ensure the sustainable growth and stability of the organization. "Bank X" has entrusted us with a compelling project that involves delving into a comprehensive dataset encompassing various facets of their operations. This dataset is a treasure trove, containing intricate details about loan applications, credit bureau data, previous loan history, and repayment behaviors.

Our primary objective is to harness the power of data analytics to extract meaningful insights from this intricate dataset. By scrutinizing the diverse information at our disposal, our aim is to uncover patterns, trends, and relationships that can empower "Bank X" to make strategic decisions in refining their loan approval processes. The significance of our role lies in the potential to optimize the decision-making workflow, enhance risk assessment, and ultimately contribute to the bank's overarching goal of providing efficient and responsible financial services.

1 . Data Loading

We started by loading a diverse array of datasets, each offering unique insights into the financial interactions and histories associated with "Bank X." The datasets serve as the foundational bedrock for our subsequent data analytics endeavors. Below is a detailed breakdown of the loaded datasets:

2.1 loan_applications.csv

- Provides essential details about each loan application, forming the basis for understanding the applicant profile.

2.2 previous_credits.csv

- Offers insights into the credit history of applicants outside the realm of "Bank X," contributing to a comprehensive credit assessment.

2.3 credit_bureau_balance.csv

- Enables the tracking of credit behaviors over time, aiding in the assessment of long-term financial stability.

2.4 previous_POS_cash_loans.csv

- Illuminates spending patterns and repayment behaviors associated with Point of Sale (POS) and cash loans.

2.5 previous_credit_cards.csv

- Explores the credit card usage history within "Bank X," informing decisions related to creditworthiness.

2.6 previous_loan_applications.csv

- Offers insights into historical loan applications, facilitating trend analysis and risk assessment.

2.7 repayment_history.csv

- Provides a granular view of how borrowers have historically met their repayment obligations, aiding in risk evaluation.

2 . Data Downcasting

Motivation

In the realm of data analytics, optimizing computational resources is paramount. Large datasets, while rich in information, can pose challenges in terms of memory utilization and processing speed. To address these challenges, we employed a technique known as data downcasting.

Methodology

Our downcasting process involved utilizing the downcast function from the pandas_downcast (pdc) library. This function intelligently converts the numeric data types in all our dataframes to their smallest possible representation while ensuring that no data loss occurs.

One-Hot Encoding

Our primary objective of one-hot encoding is to convert categorical variables into a binary matrix format, where each category becomes a separate column. This binary representation allows us to incorporate categorical information seamlessly into our analyses, unlocking the latent insights embedded within these variables.

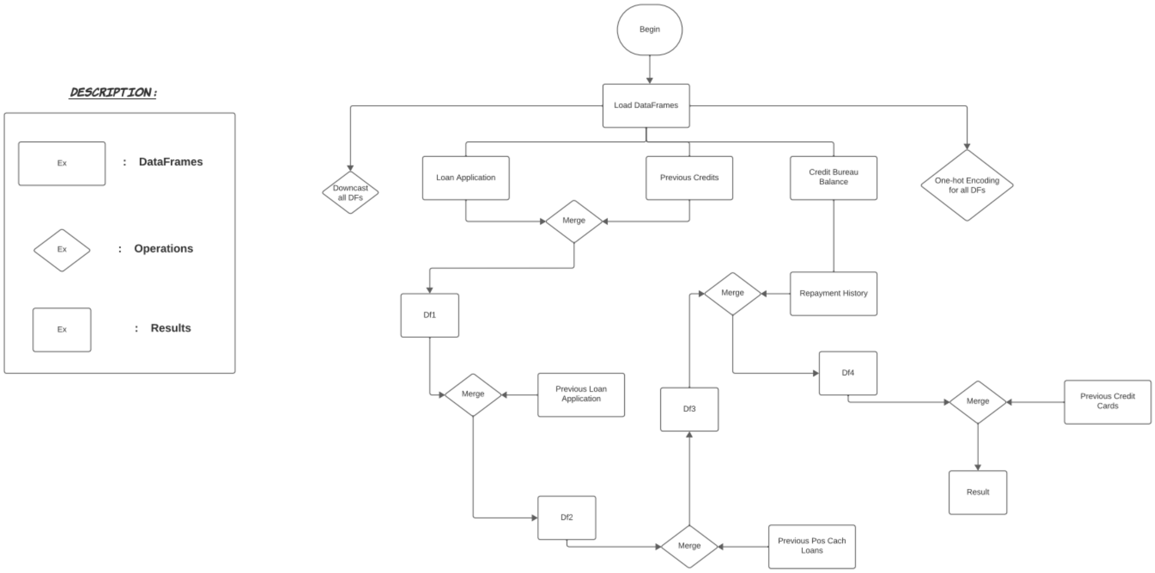
The get_dummies function from the pandas library serves as our tool of choice for one-hot encoding.

Merge

In order to construct a comprehensive and informative dataset for our analysis, we engaged in a meticulous process of merging and aggregating all our dataframes.

- Each block of our merge code follows a similar pattern: group by a unique identifier (sk_id_curr), perform aggregation functions (e.g., mean, count) on relevant columns, and merge the aggregated results back into the main DataFrame (df).
- The merging process ensures that information from various datasets is consolidated under a common identifier (sk_id_curr), facilitating a unified and holistic view of the applicant's financial profile.

This merging and aggregation process lies in the creation of a comprehensive dataset that encapsulates diverse aspects of an applicant's financial history.



Data Description

We created a convenient and comprehensive function to obtain a quick overview of the merged dataframe. It covers various aspects, including the size of the dataframe, missing values, duplicates, and the distribution of data types across columns.

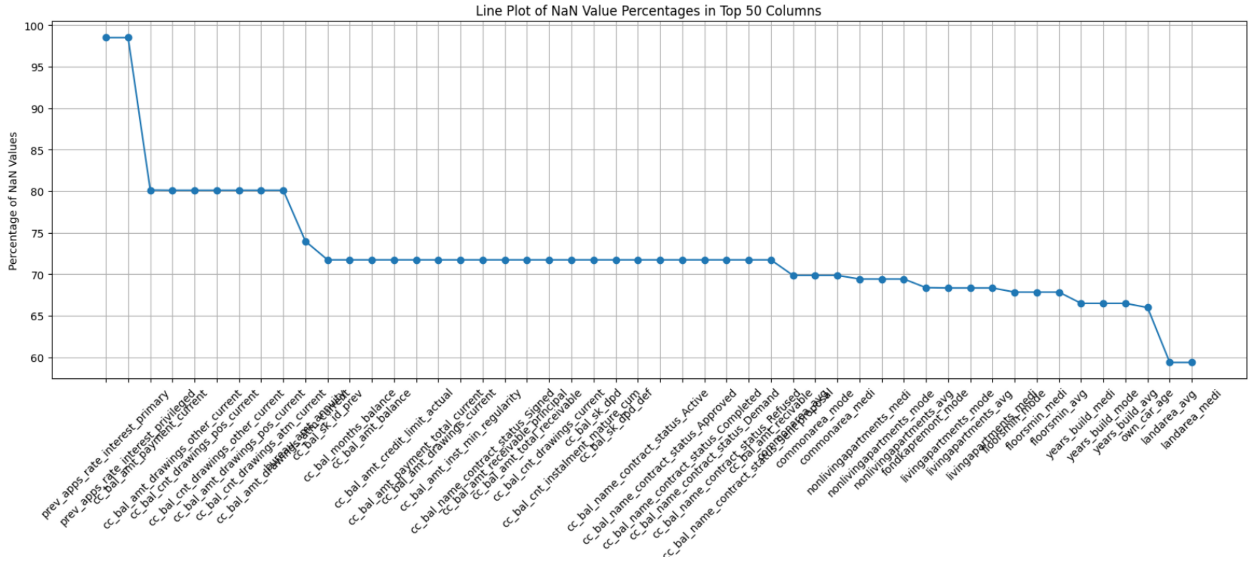
- Our function “data_describe” utilizes pandas functions to compute various statistics about the input dataframe, such as the number of rows and columns, the percentage of missing values and duplicates, the counts of different data types (object, float, int, bool), and the memory usage in megabytes.
- The information is then stored in a dictionary (data_dict) and converted into a pandas DataFrame (comparative_table) for clear presentation.

- The function prints a summary message and returns the comparative table.

Missing values

First, we checked for columns in our data that were missing a lot of information. We found several columns missing more than 60% of their data, so we removed them. Then, for the remaining data with missing bits (NaN values), we decided to fill in these gaps. We thought about using a complex method called spline interpolation, because, according to information found on the internet, this method is good for capturing non-linearity of data, but we ended up using a simpler method: filling in missing values with the median. We'll need to check later if this was a good choice when we test our models.

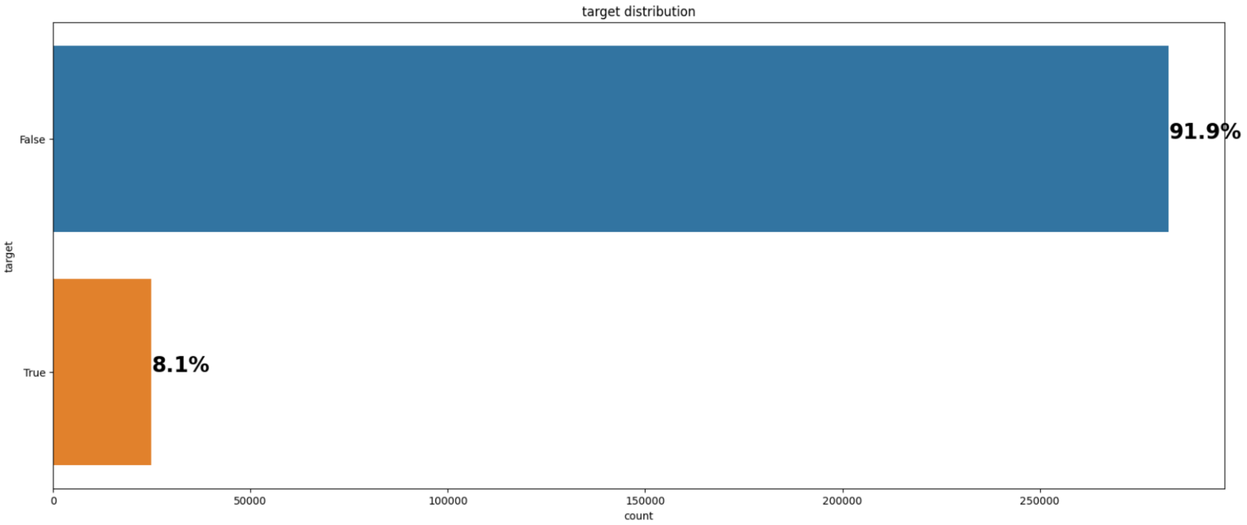
We also plotted the columns and NaN value percentages.



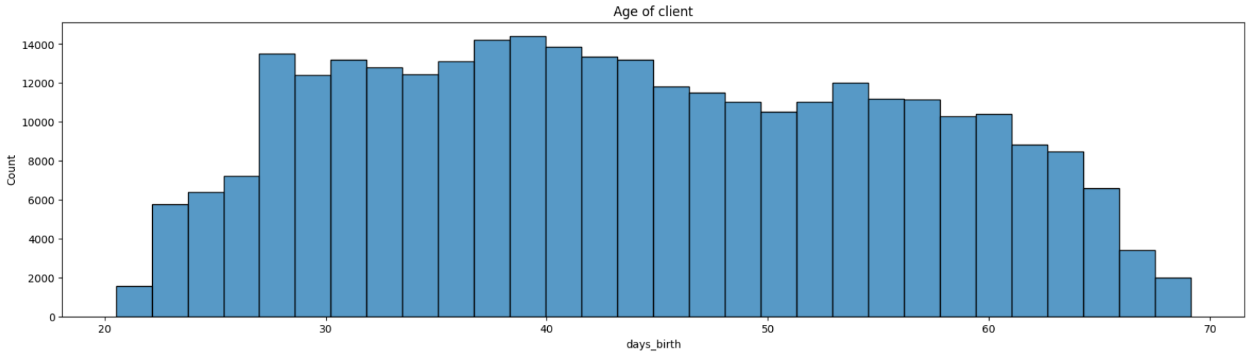
Data Visualization

For the visualization part, we chose some key features to highlight, for example : _age_, _contract types_, _gender_, _car ownership_, _family status_ ... etc. This visualization serves two purposes: first, to see how our data is spread out, and second, to find connections between specific features and the target value we want to predict. An interesting example of this is the Income type column. We observe that **51%** of the loan applicants are working. However **40%** of applicants that are on maternity leave don't refund their loan. This could be relevant to our future model, as banks should definitely consider the income type of their applicants.

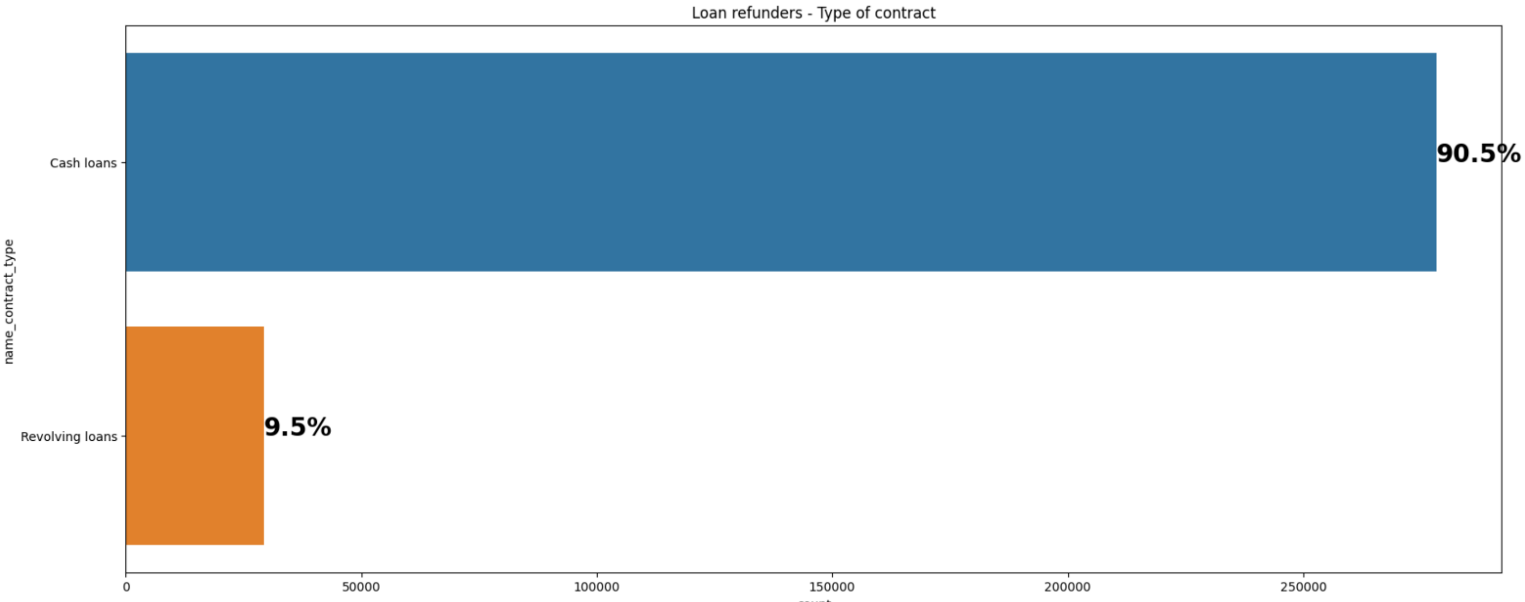
- Target Distribution :

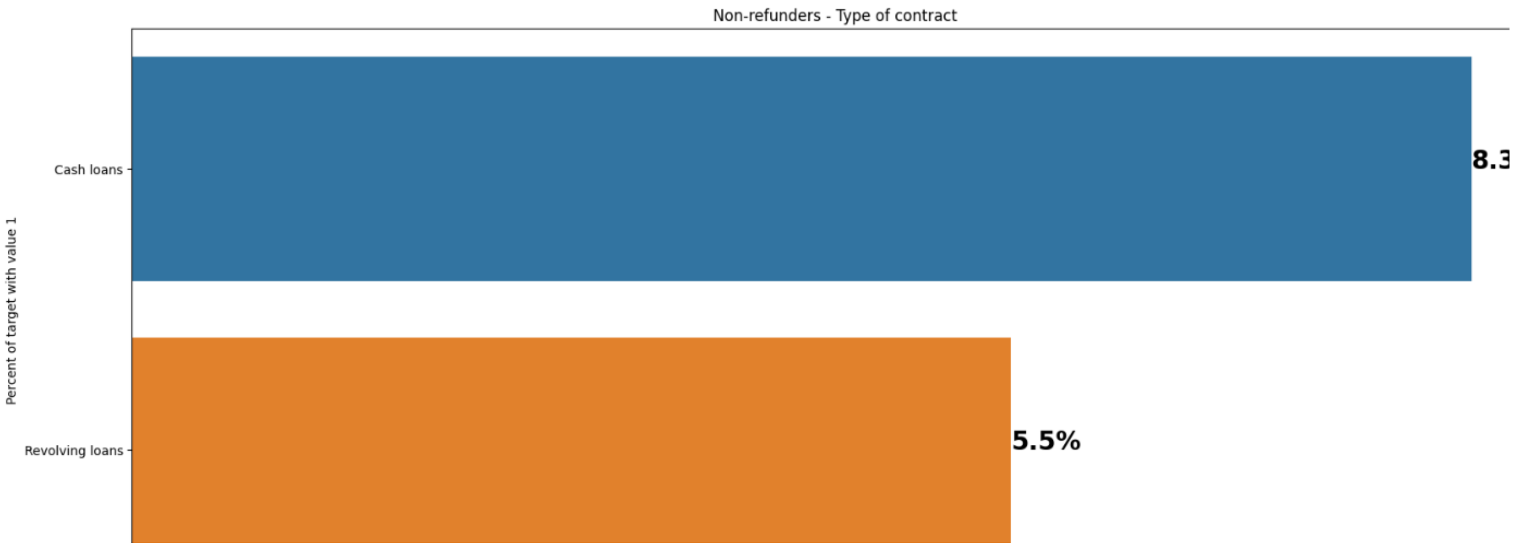


- Age Distribution :

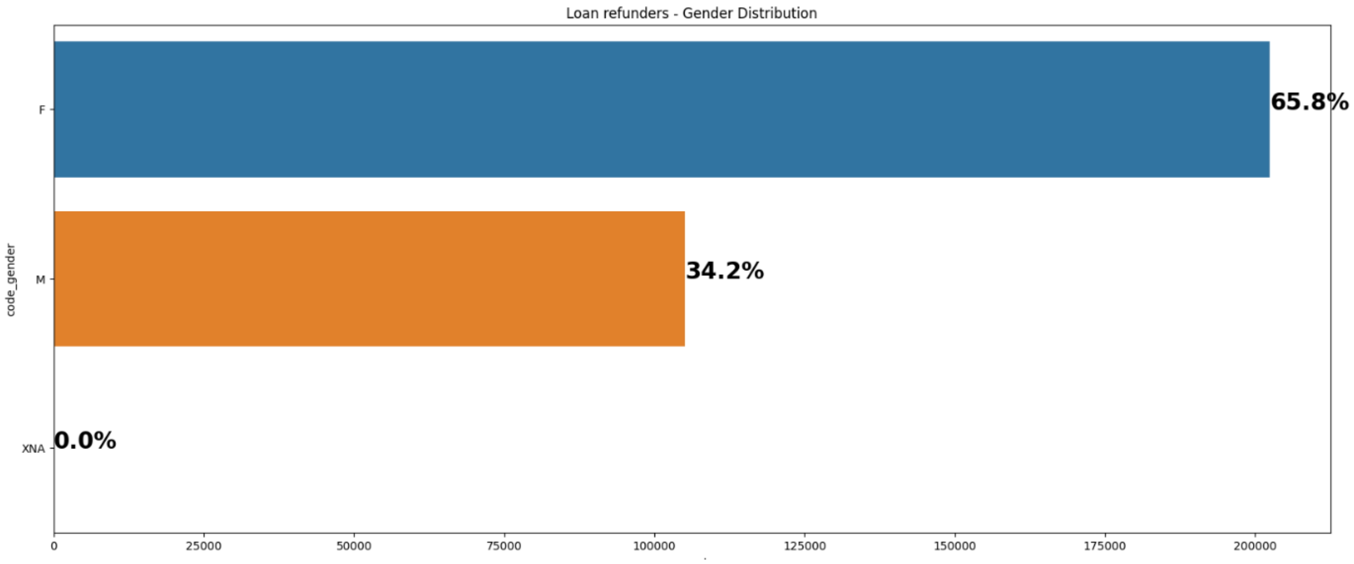


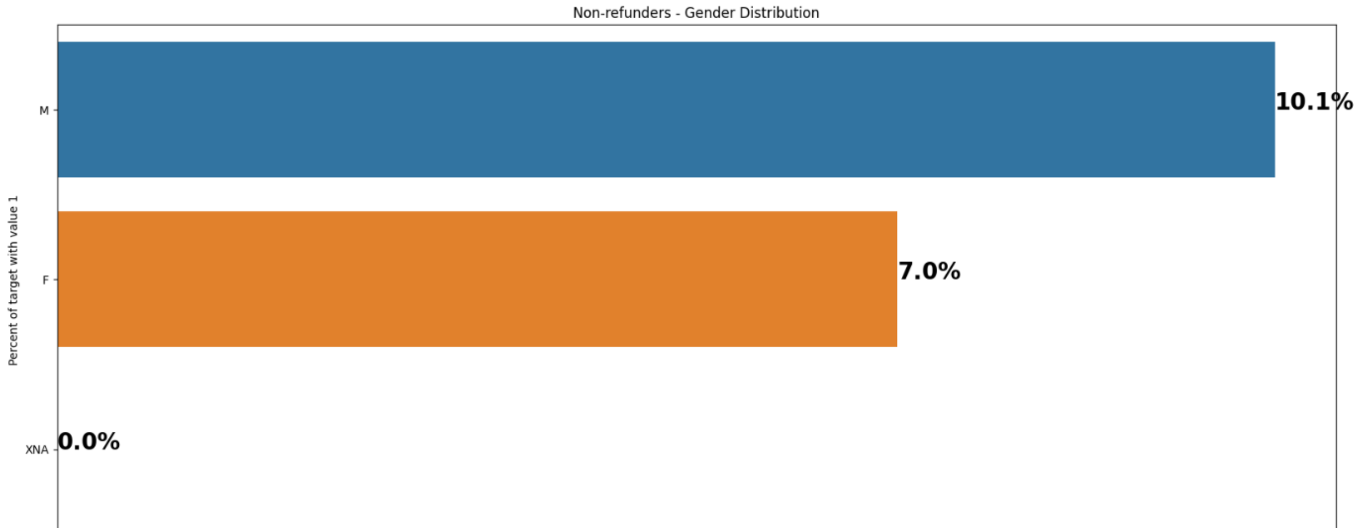
- Type of contract for loan refunders and also non-refunders :



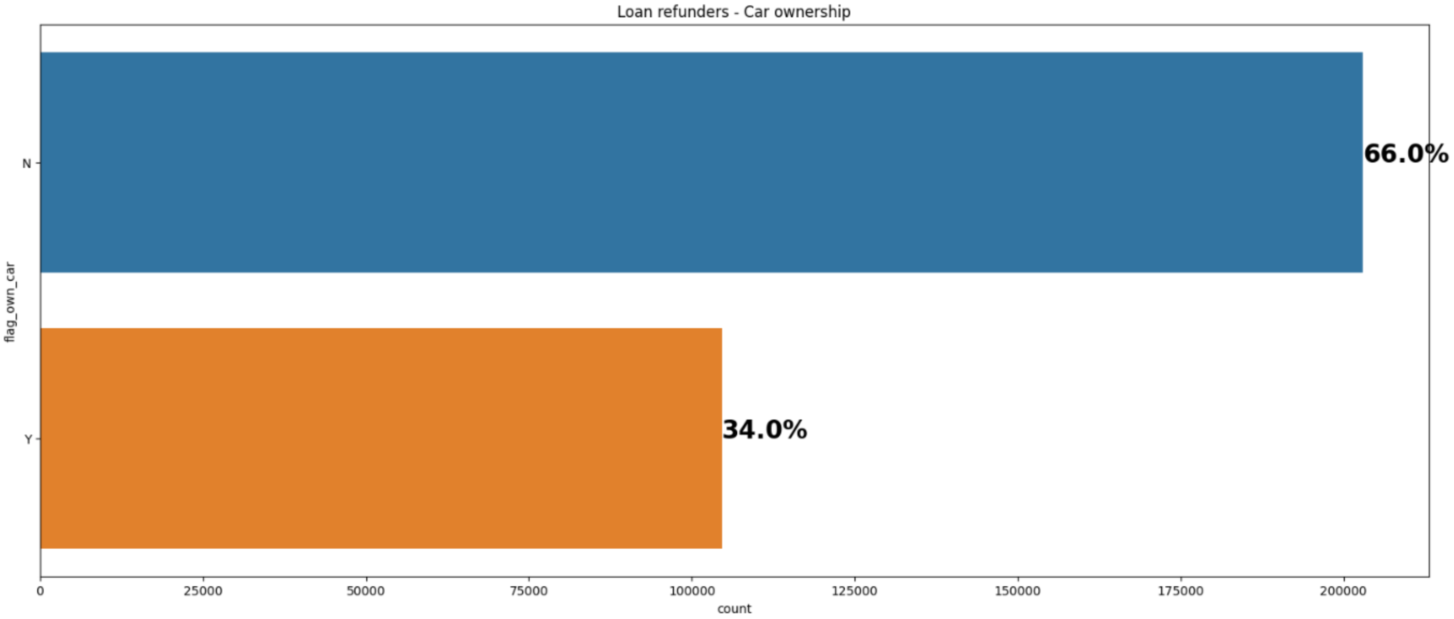


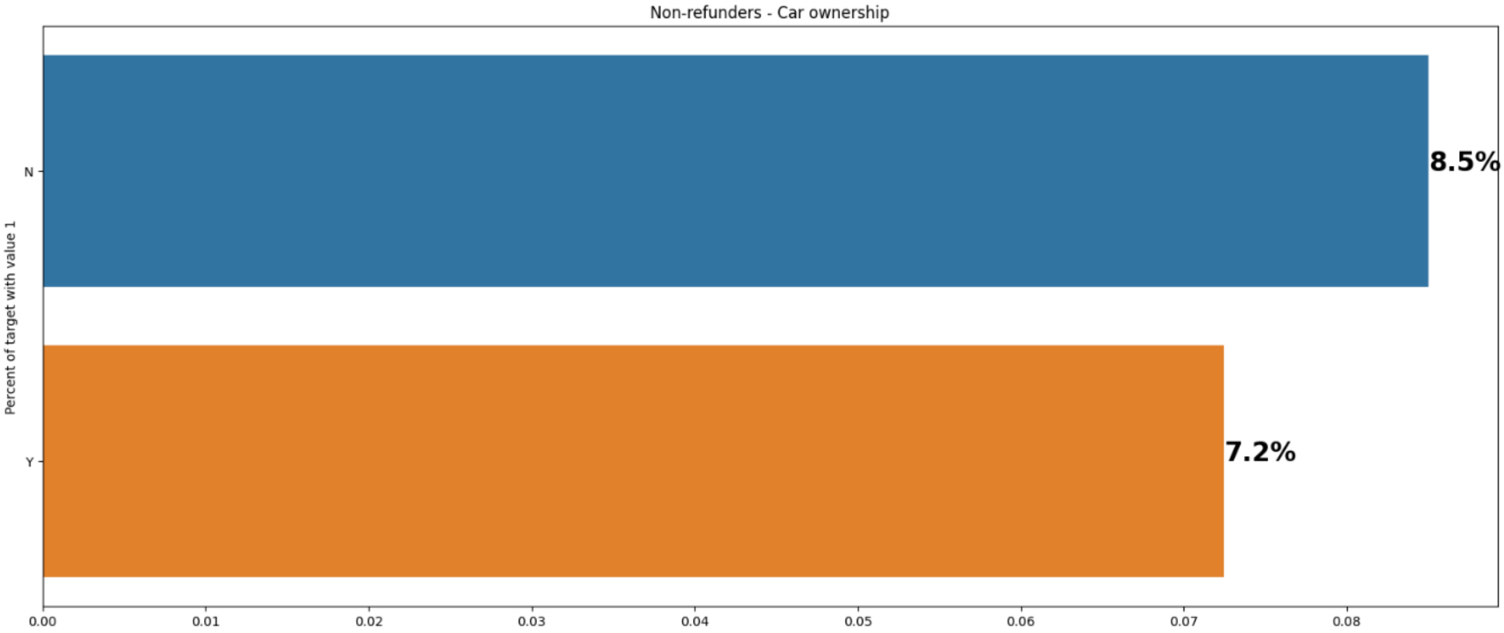
- Gender Distribution for loan refunders and non-refunders:



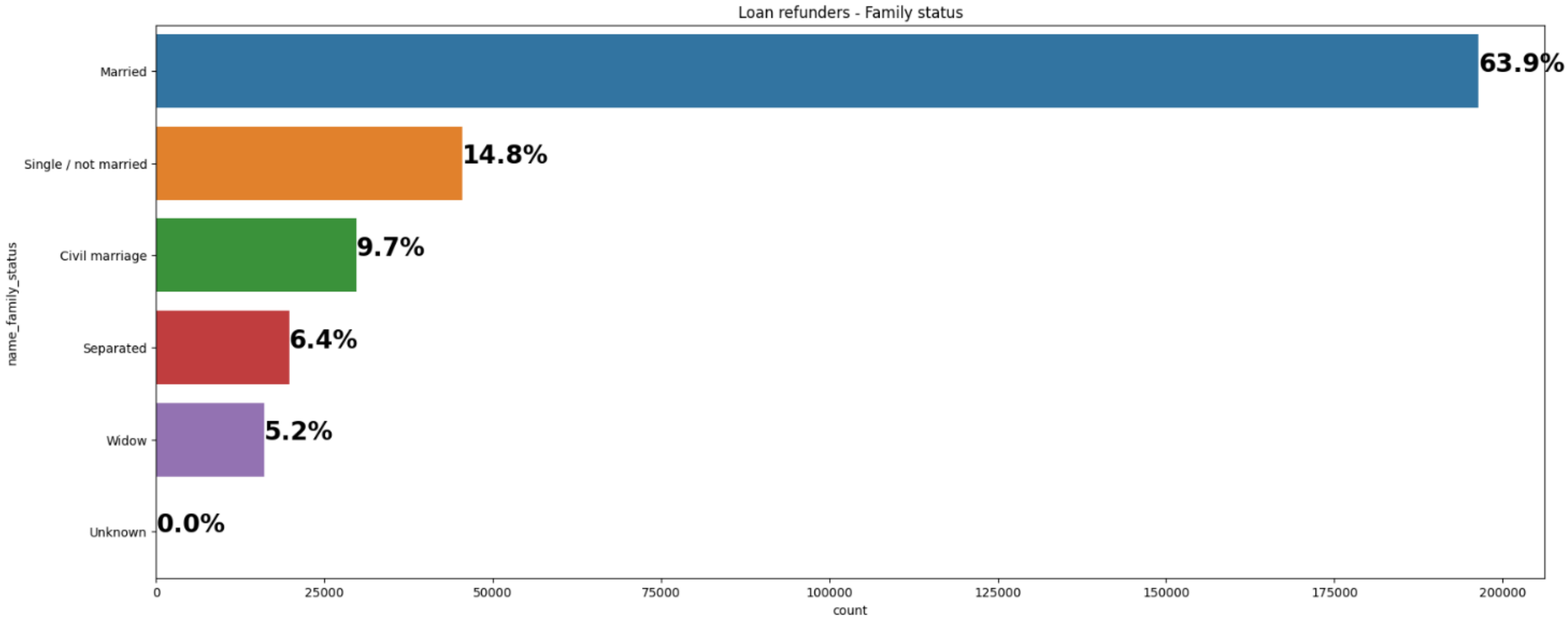


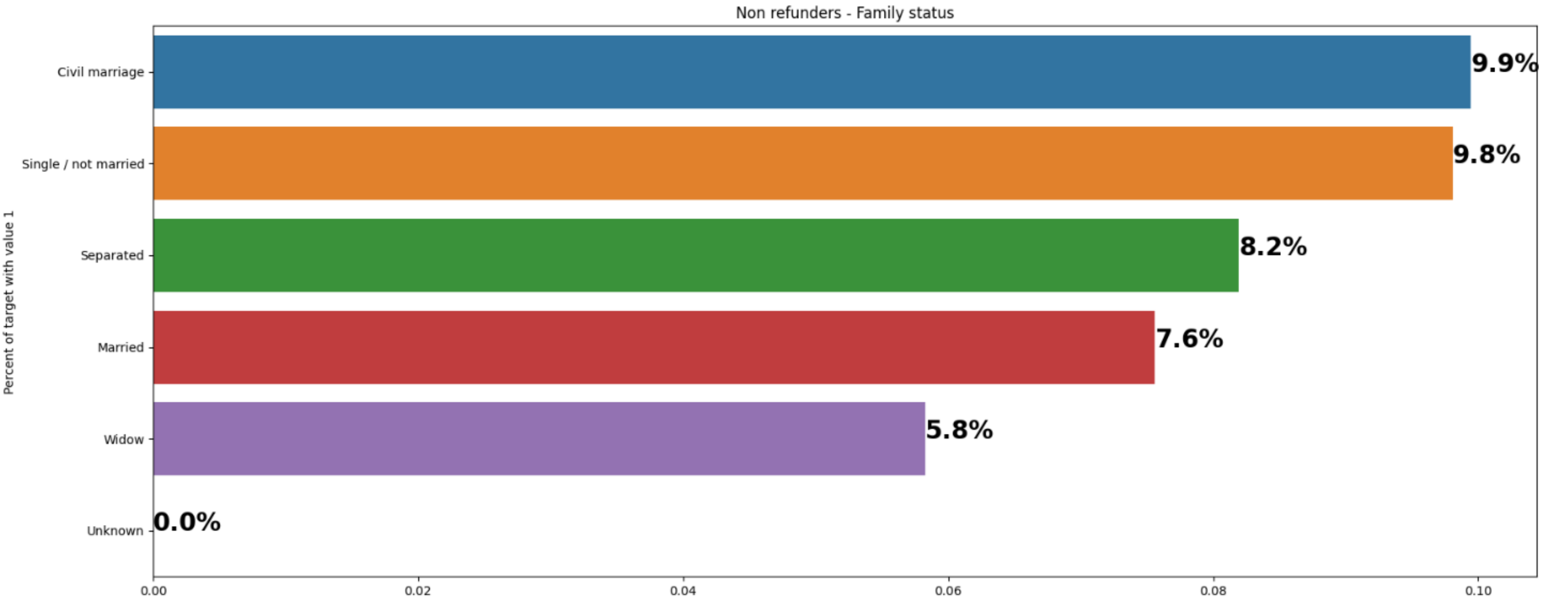
- Car Ownership for loan refunders and non-refunders :



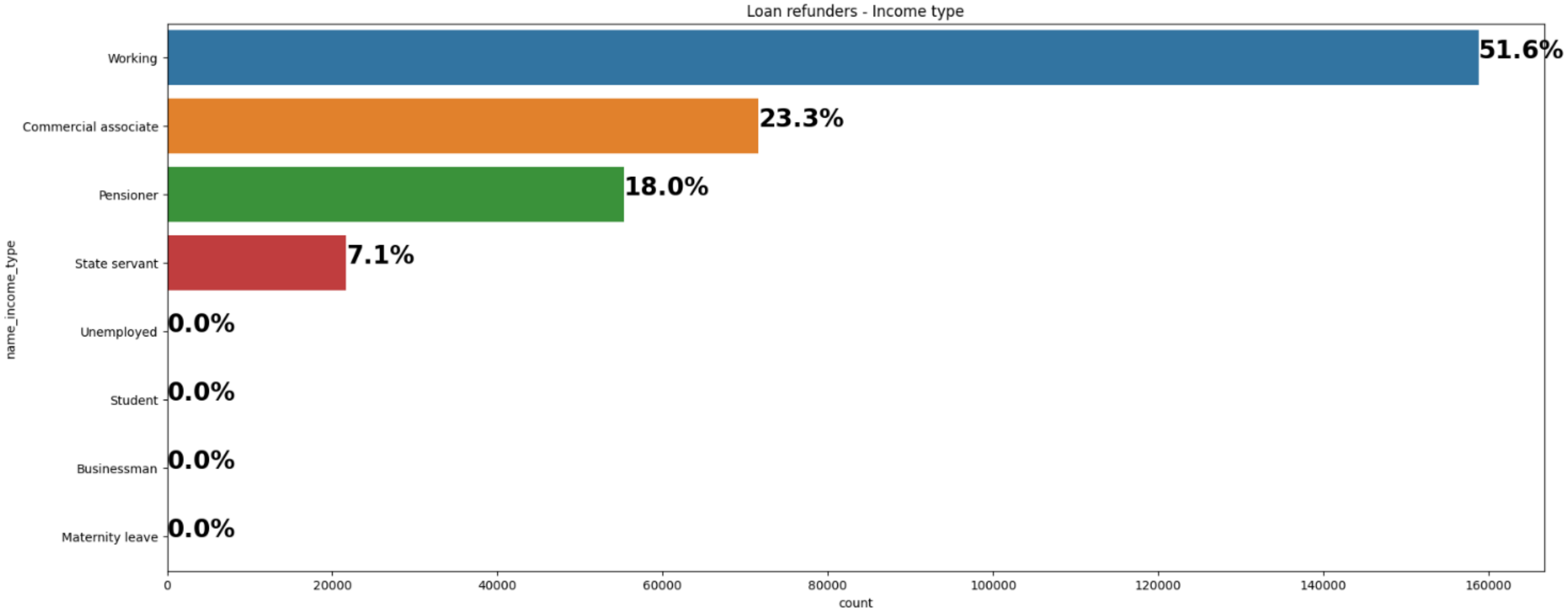


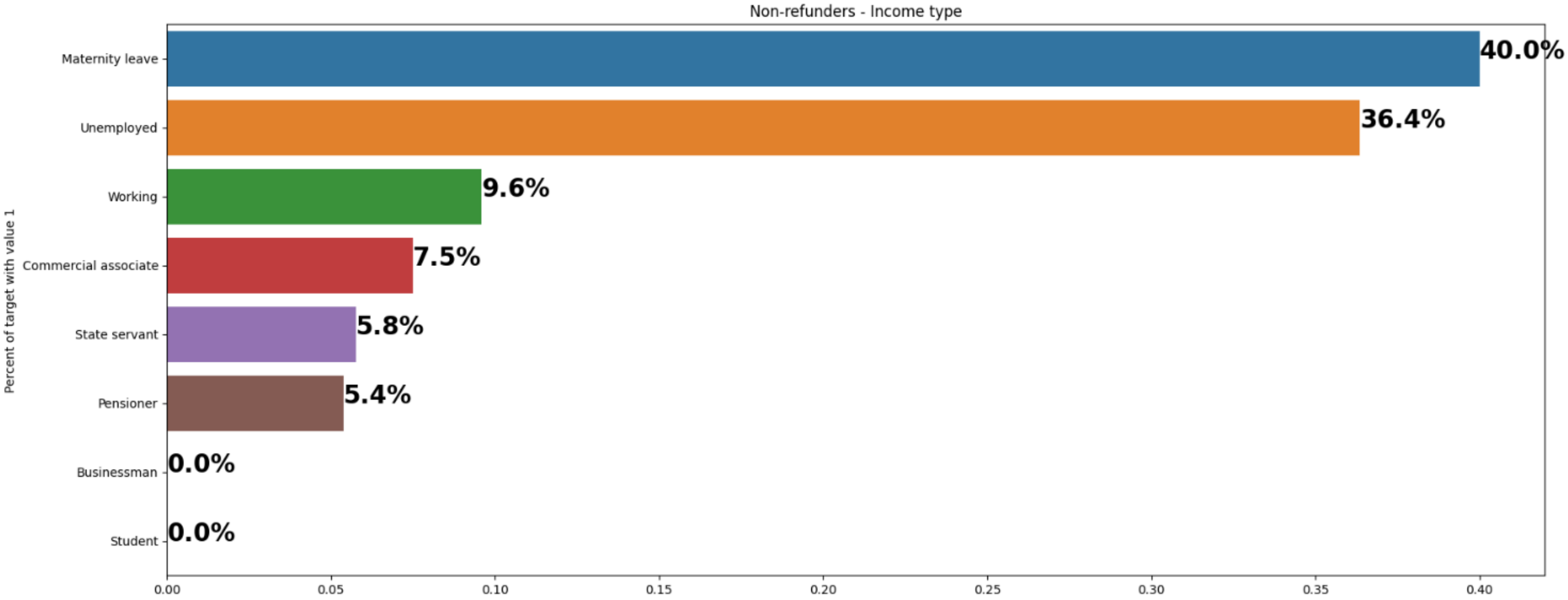
- Family Status for loan refunders and non-refunders :



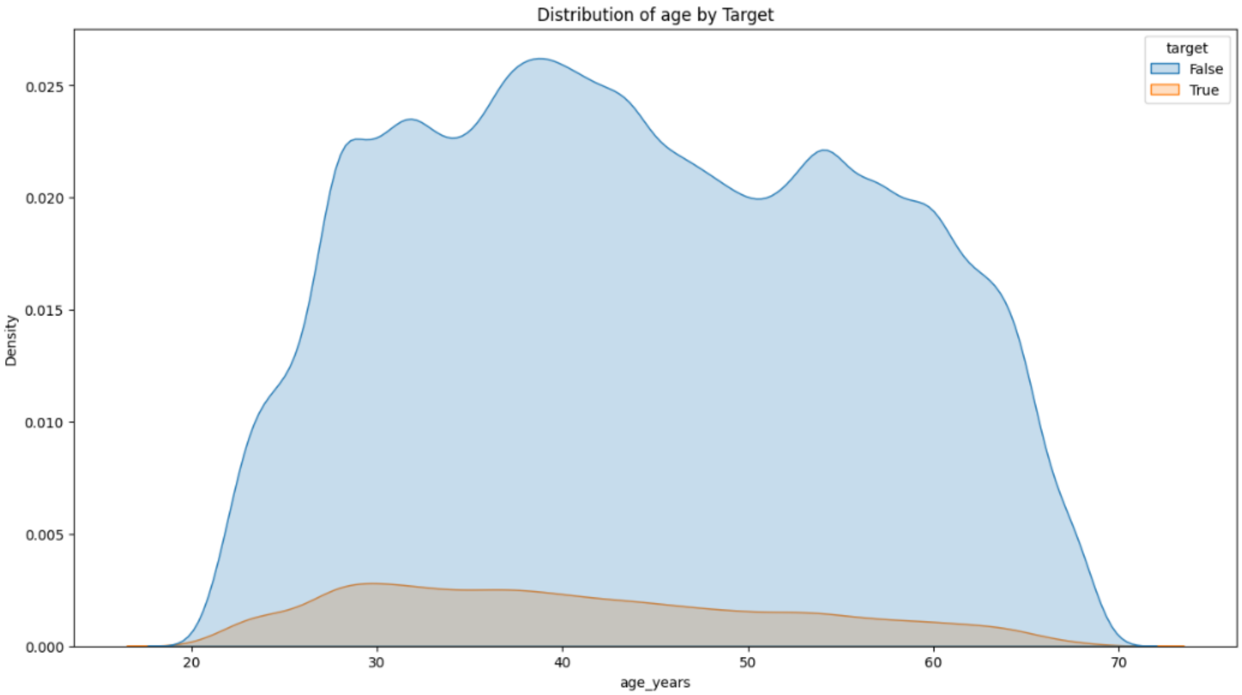


- Income Type for loan refunders and non refunders :

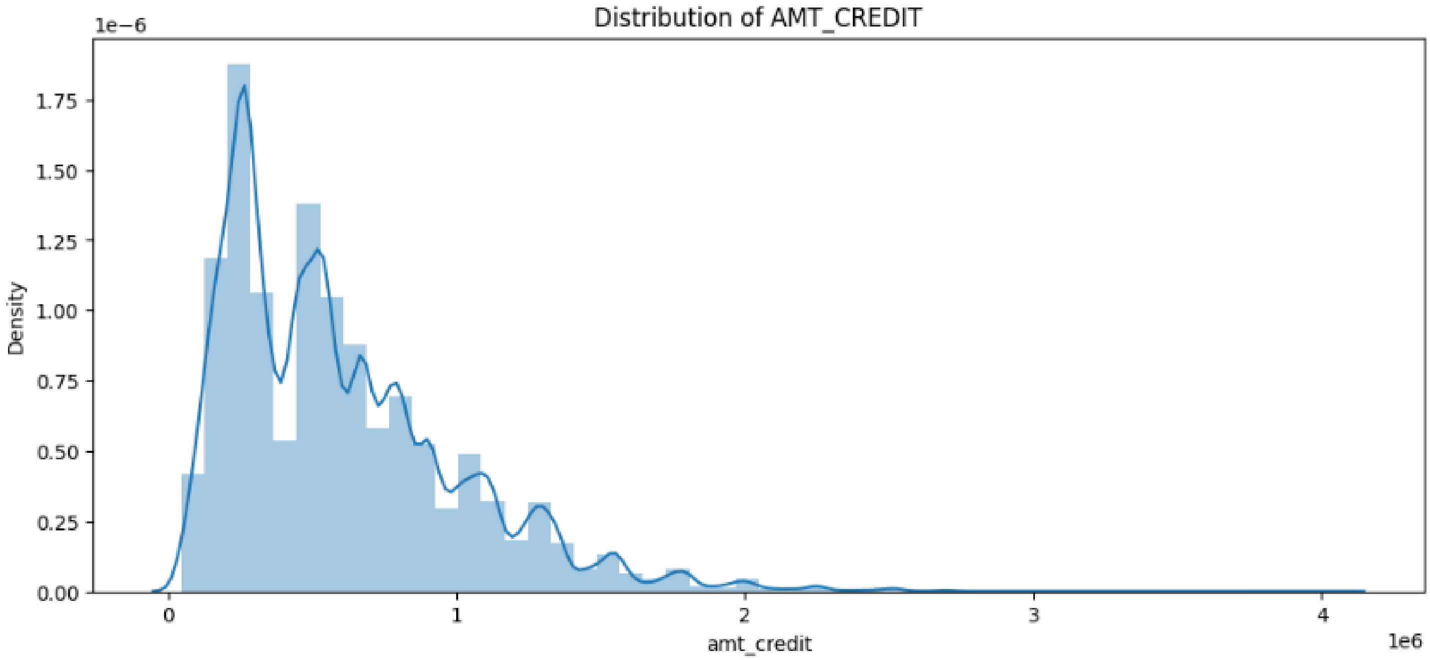




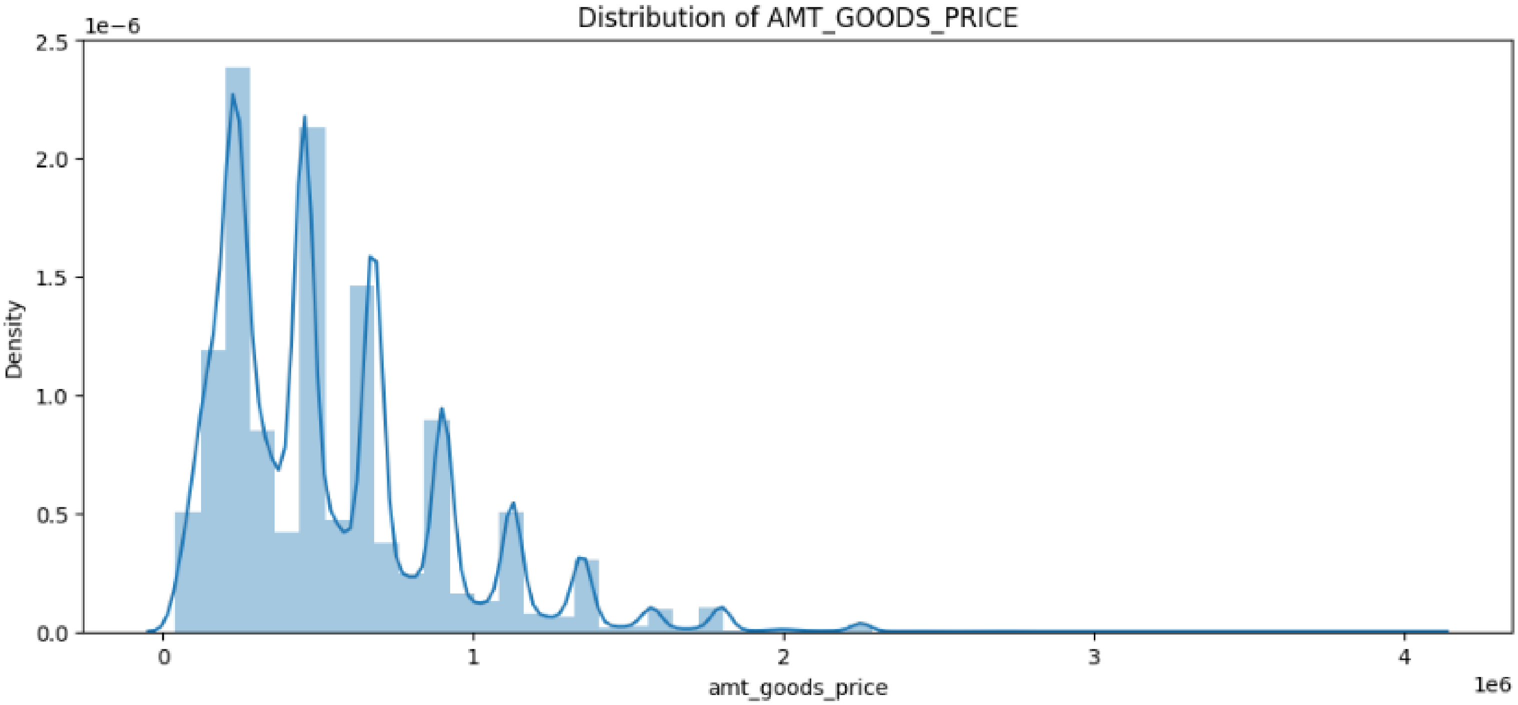
- Distribution of age by target :



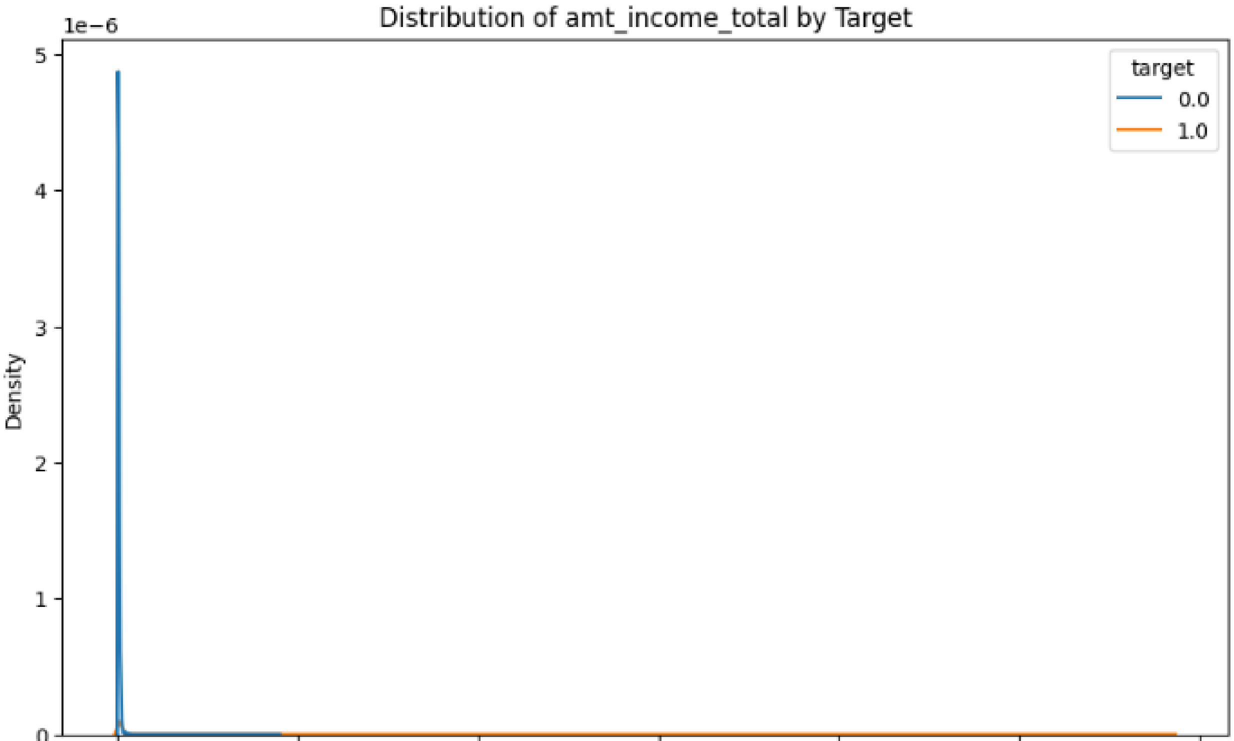
- Distribution of AMT_Credit :



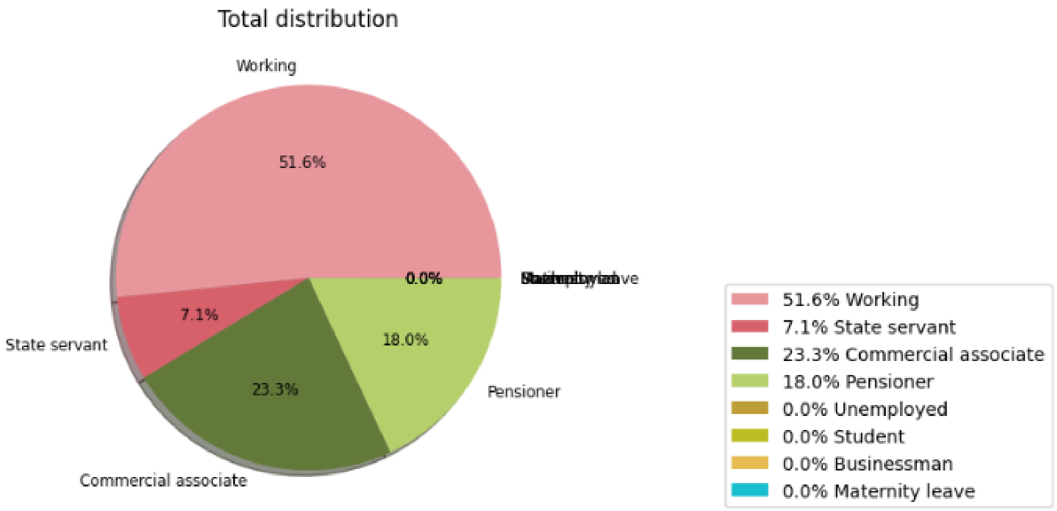
- Distribution of AMT_GOODS_PRICE :

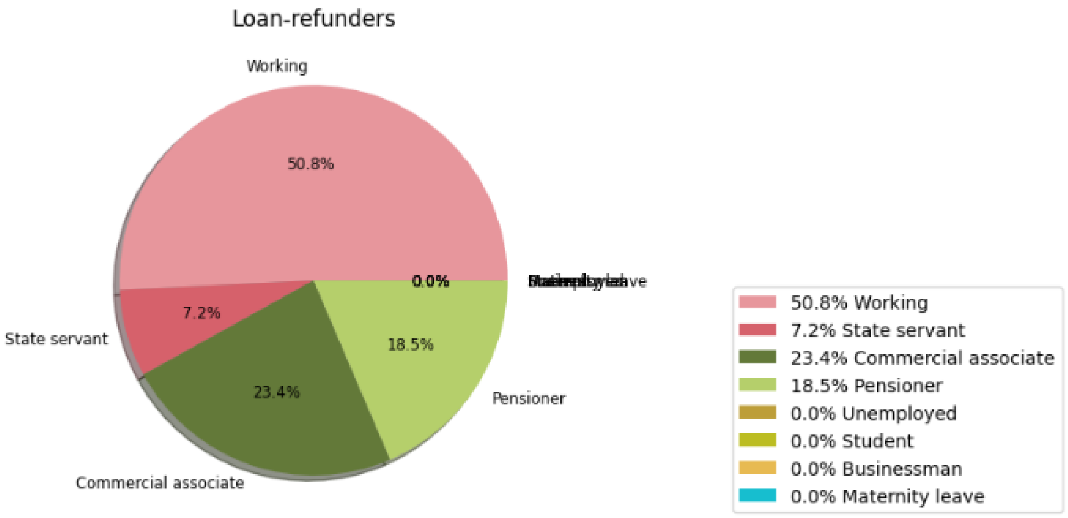


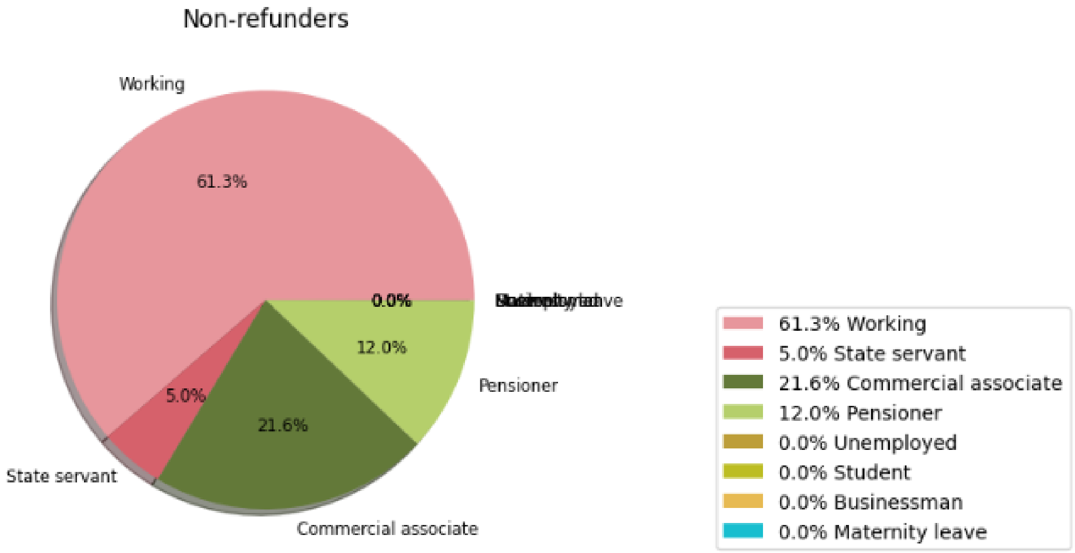
- Distribution of amt_income_total by target :



- Distribution for status of loan refunders and non-refunders :


Total distribution


Loan-refunders


Non-refunders

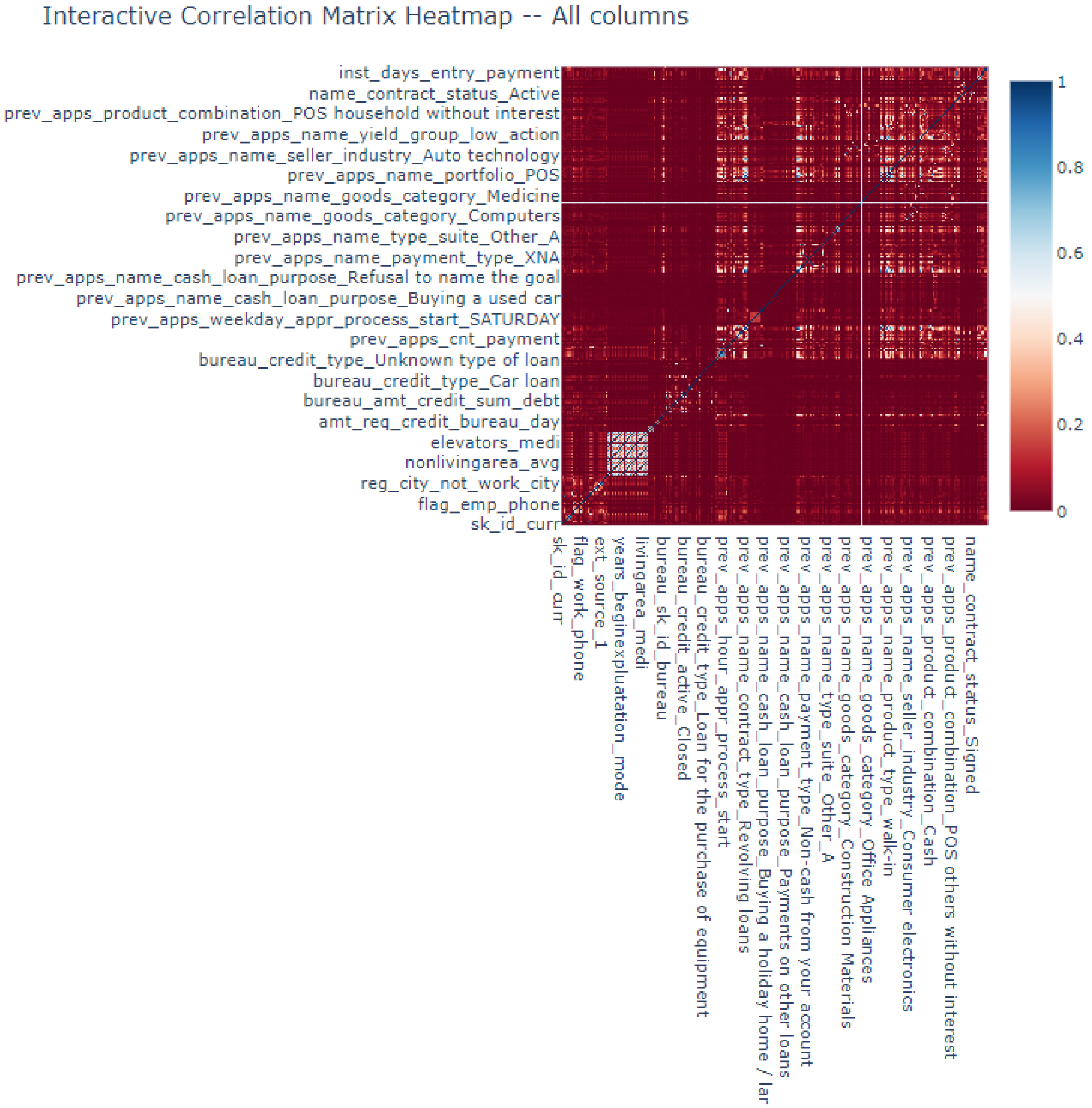
Dimension Reduction

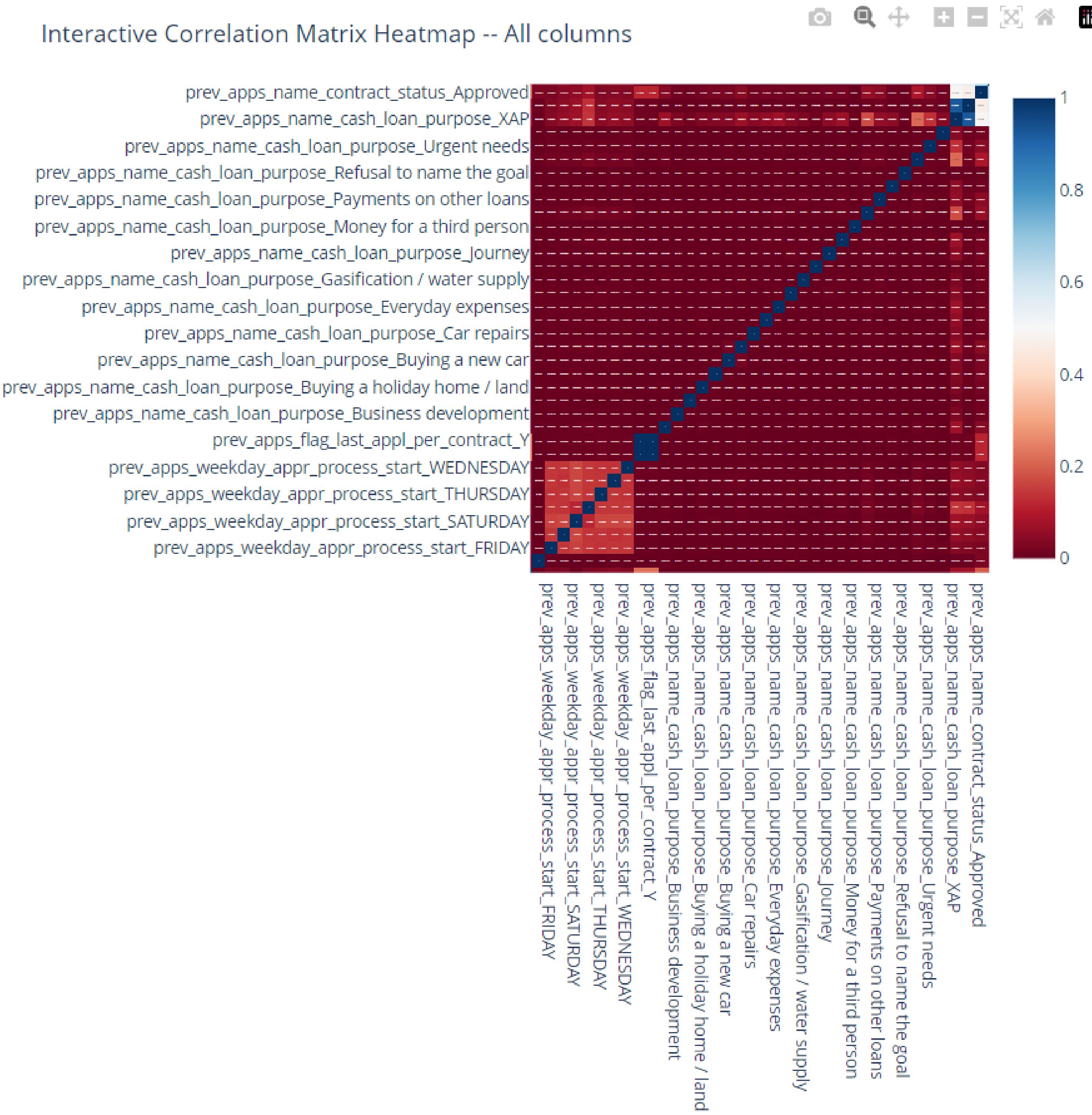
Columns with high Nan value percentage:

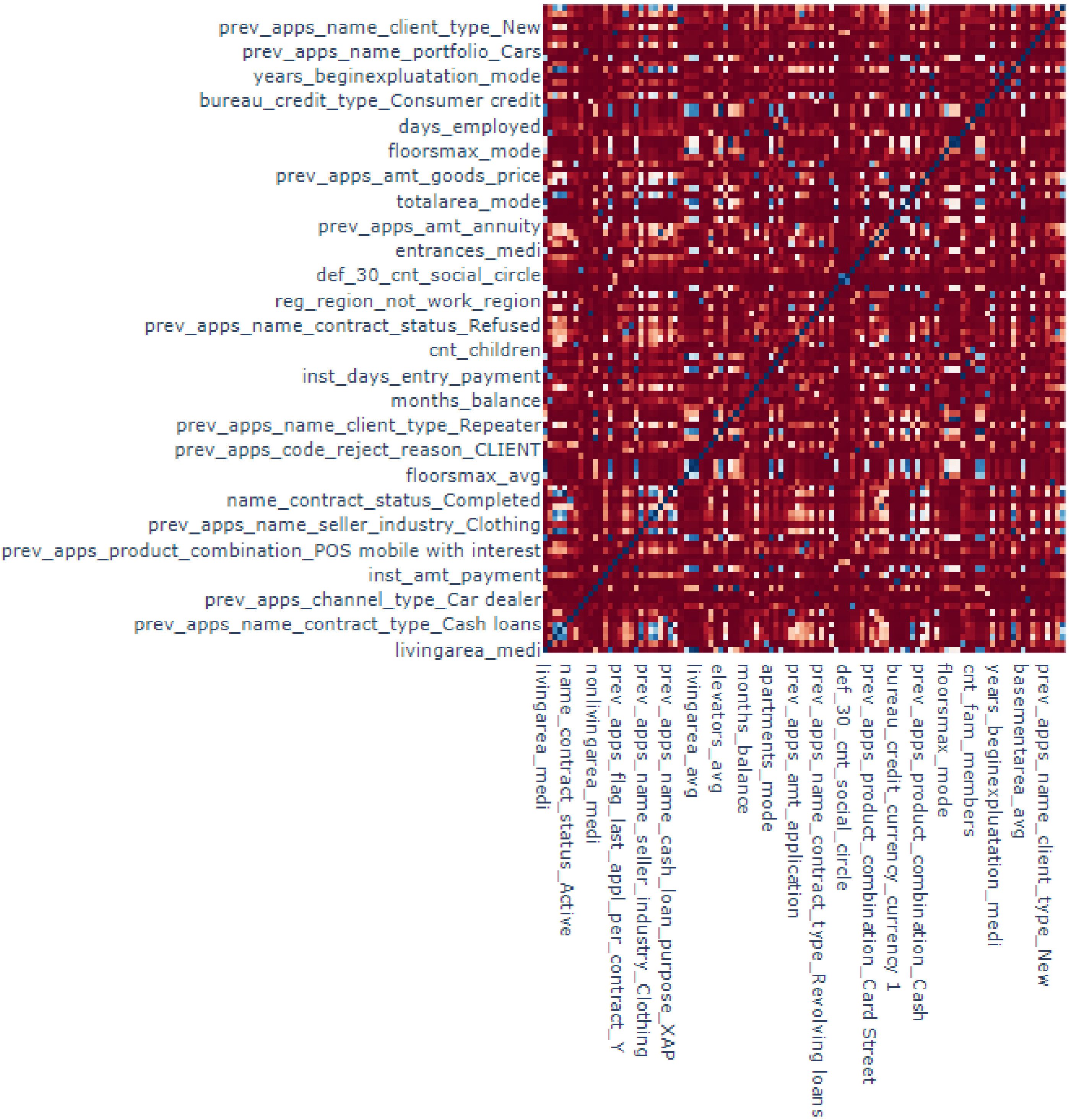
As mentioned before, a lot of columns had a relatively high percentage of nan values, so we decided to drop those columns.

Columns with high correlation factors:

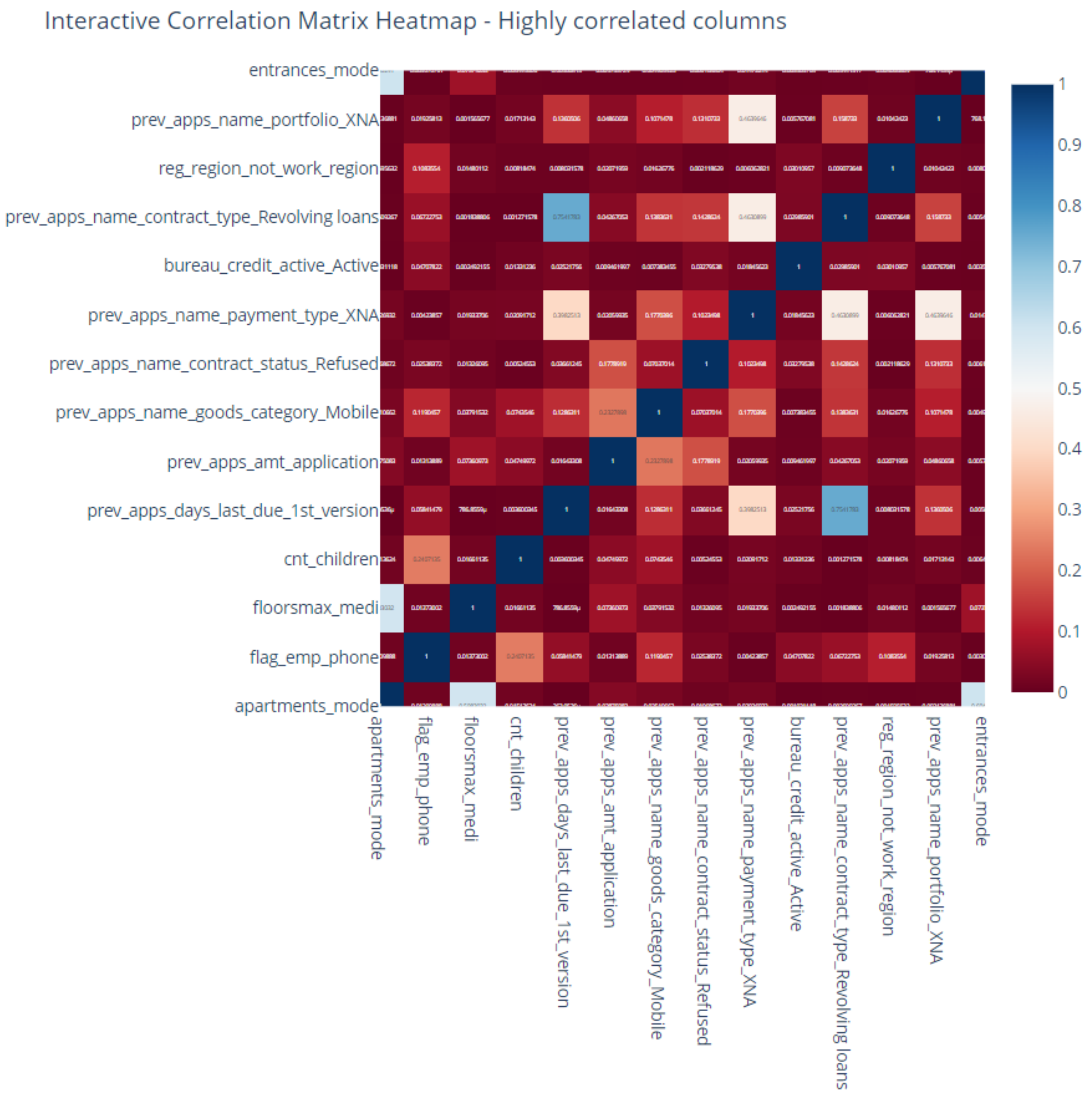
Reducing the dimensionality of a DataFrame using a correlation matrix is a strategy often employed in feature selection during data preprocessing. This approach focuses on identifying and removing highly correlated features to reduce the number of dimensions (i.e., the number of features) in the dataset. Once we identified pairs of highly correlated features (> 80%), we dropped a column from each pair.




Interactive Correlation Matrix Heatmap -- All columns
prev_apps_name_contract_status_Approved
prev_apps_name_cash_loan_purpose_XAP
prev_apps_name_cash_loan_purpose_Urgent needs
prev_apps_name_cash_loan_purpose_Refusal to name the goal
prev_apps_name_cash_loan_purpose_Payments on other loans
prev_apps_name_cash_loan_purpose_Money for a third person
prev_apps_name_cash_loan_purpose_Journey
prev_apps_name_cash_loan_purpose_Gasification / water supply
prev_apps_name_cash_loan_purpose_Everyday expenses
prev_apps_name_cash_loan_purpose_Car repairs
prev_apps_name_cash_loan_purpose_Buying a new car
prev_apps_name_cash_loan_purpose_Buying a holiday home / land
prev_apps_name_cash_loan_purpose_Business development
prev_apps_flag_last_appl_per_contract_Y
prev_apps_weekday_appr_process_start_WEDNESDAY
prev_apps_weekday_appr_process_start_THURSDAY
prev_apps_weekday_appr_process_start_SATURDAY
prev_apps_weekday_appr_process_start_FRIDAY
prev_apps_weekday_appr_process_start_FRIDAY
prev_apps_weekday_appr_process_start_SATURDAY
prev_apps_weekday_appr_process_start_THURSDAY
prev_apps_weekday_appr_process_start_WEDNESDAY
prev_apps_flag_last_appl_per_contract_Y
prev_apps_name_cash_loan_purpose_Business development
prev_apps_name_cash_loan_purpose_Buying a holiday home / land
prev_apps_name_cash_loan_purpose_Buying a new car
prev_apps_name_cash_loan_purpose_Car repairs
prev_apps_name_cash_loan_purpose_Everyday expenses
prev_apps_name_cash_loan_purpose_Gasification / water supply
prev_apps_name_cash_loan_purpose_Journey
prev_apps_name_cash_loan_purpose_Money for a third person
prev_apps_name_cash_loan_purpose_Payments on other loans
prev_apps_name_cash_loan_purpose_Refusal to name the goal
prev_apps_name_cash_loan_purpose_Urgent needs
prev_apps_name_cash_loan_purpose_XAP
prev_apps_name_contract_status_Approved
1
0.8
0.6
0.4
0.2
0



Interactive Correlation Matrix Heatmap - Highly correlated columns

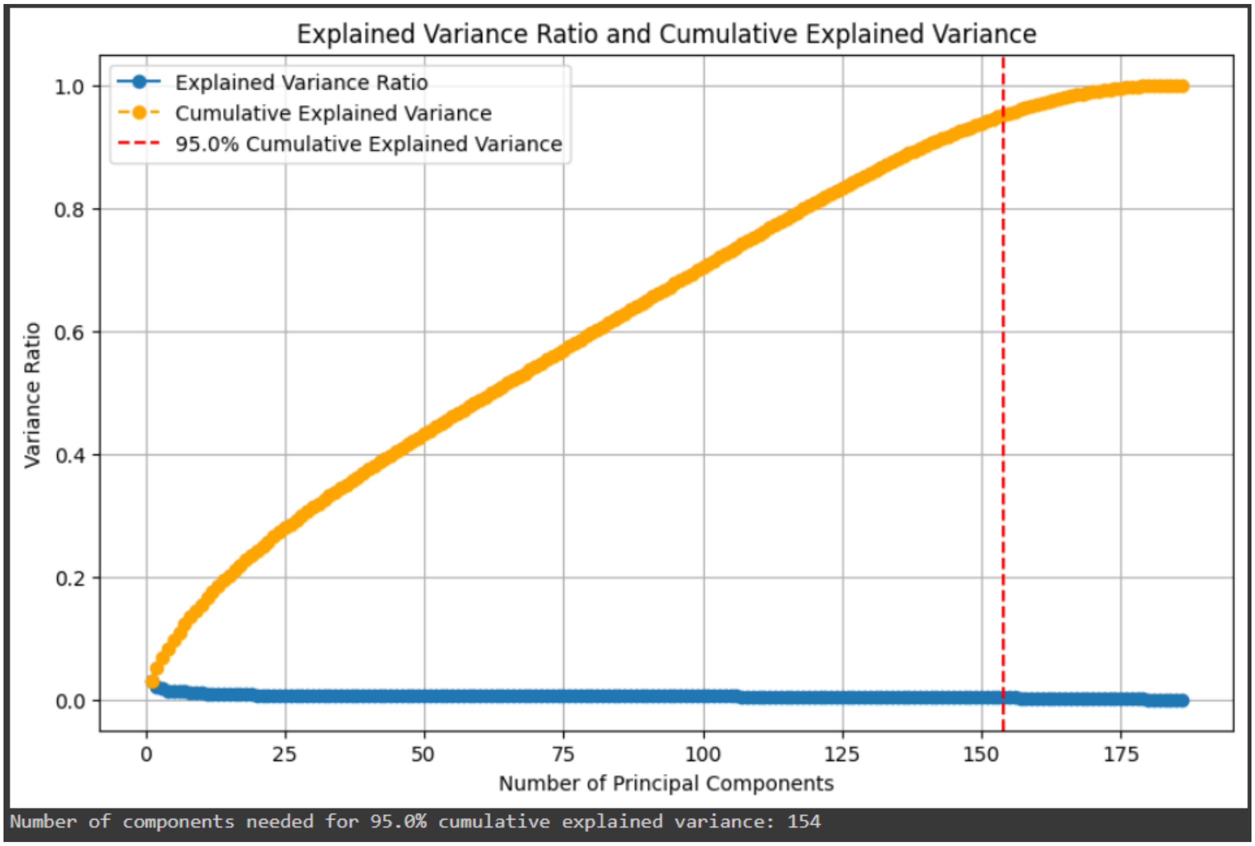


PCA for dimension reduction

Finally, we explored an alternative approach aimed at diminishing the dimensionality of our dataframe as we progress with our project. The objective is to determine the optimal number of columns for training our future model.

One method for reducing our dataframe involves employing PCA (Principal Components Analysis), a statistical technique utilized in data analysis and dimensionality reduction. Its primary objective is to streamline intricate datasets by transforming them into a new set of variables referred to as principal components. These components, which are linear combinations of the original variables, are crafted to capture as much of the data's variability as possible.

Upon completing the analysis, we obtained an interesting result, illustrated in the graph below:



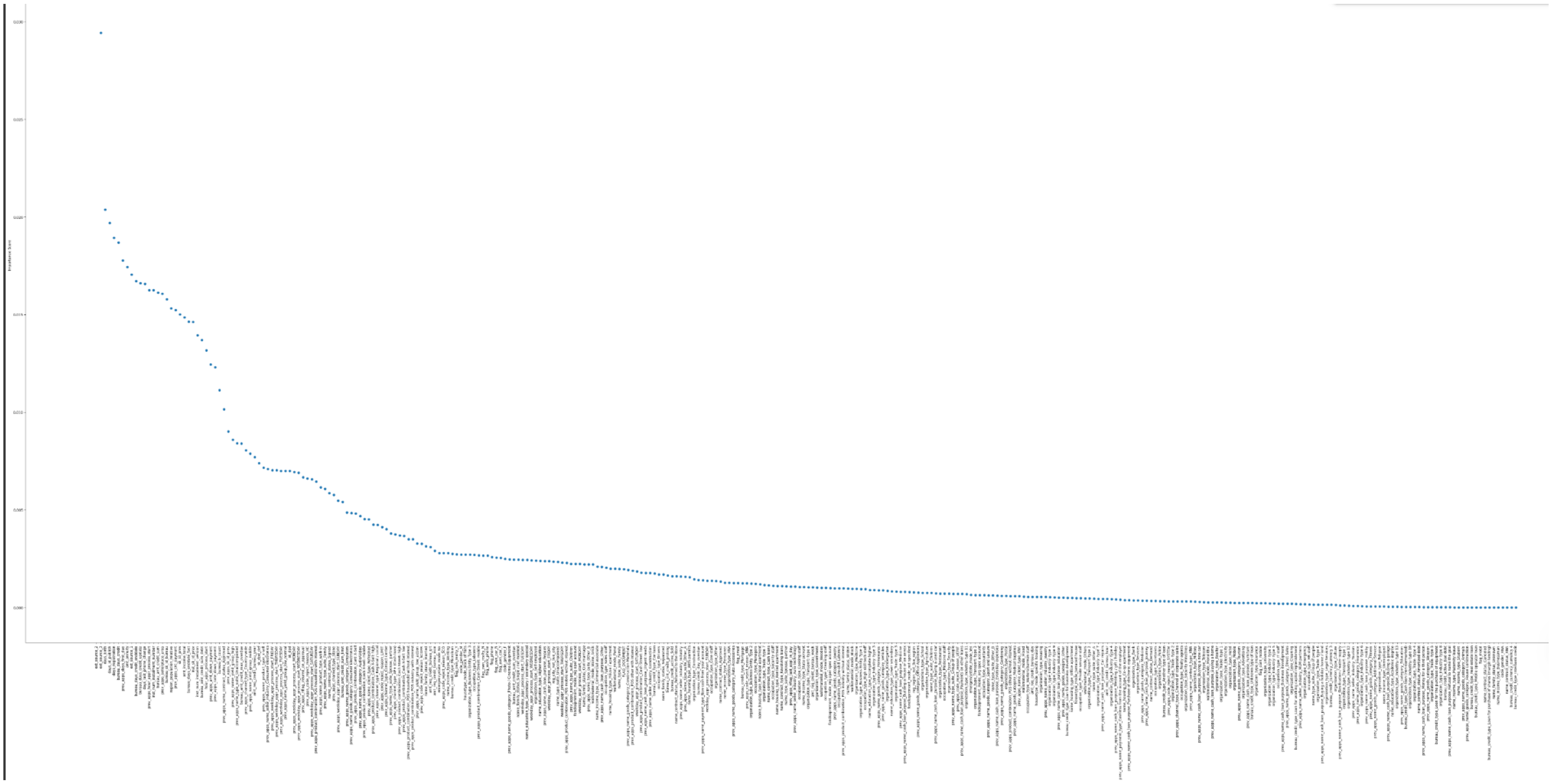
```
Number of components needed for 95.0% cumulative explained variance: 154
```

Conclusion: We can use 154 principal components instead of 186, and still retain 95% of the numeric information.

Additional analysis : (Feature Importance)

To evaluate the significance of each feature (column) in our dataset, we employed a predictive model to calculate the importance of each column. This involved analyzing how each column influenced the final predicted target. In our study, we utilized a tree-based model, namely **Random Forest**, which provided a ranking of the importance of each column, as illustrated in the accompanying graphic.

Columns and their importance:



This chart illustrates the impact of each column on our predictive model's results. We can observe that some columns have almost negligible importance coefficients, suggesting the possibility of dropping them if required. The analysis serves various purposes, primarily focusing on **Feature Selection**. Additionally, it contributes to enhancing **Model Interpretability** and guiding decisions in **Feature Engineering**.