Course Project 2
Credit Risk Analysis

Explanation Of The Project

By Shalini
Student of DS&AI Batch
Batch ID 8208

Steps I Follow in Project

1. Import Libraries

- Numpy(#for numerical computations on dataset)
- Pandas(#for data manipulation and analysis on dataset)
- Matplotlib (matplotlib.pyplot)(#library to create plots)
- Seaborn(#for informative statistical graphics)
- Warnings(import warnings for removing the additional text from the output)
- Drive(#Connect with google drive for reading file)

2. Exploring data

- View dataframe
- View the First Few Rows
- Data Summary
- Checking Shape
- Summary Statistics
- Describe Categorical Data
- Unique Values

3. Handling Missing Data

- Step 1: Identify Missing Data
- Step 2: Visualize Missing Data
- Step 3: Decide on Handling Strategy
- Drop columns with more than 50% missing values
- Drop Rows with Missing Values in Important Columns

4. Analyzing Categorical and Numerical data

- Analyzing Categorical Data
- Visualize Numerical Data

5. Outlier Identification and Visualization

- Step 1: Using Statistical Methods (Z-Score and IQR)
- Step 2: Visualization Methods

6. Univariate and Bivariate Analysis

- Step 1:Univariate Analysis
- For Numerical Variables
- For Categorical Variables
- Step 2:Bivariate Analysis
- Numerical vs. Numerical Variables
- Categorical vs. Numerical Variables
- Categorical vs. Categorical Variables

7. Step 7:Top 10 Correlations Analysis (Segmented by Target)

- Step 1: Segment Data and Calculate Correlations
- Step 2: Visualize the Top Correlations

Step 1:Import Libraries

1st Cell

- #importing necessary libraries need for dataset
- import numpy as np #for numerical computations on dataset
- import pandas as pd #for data manipulation and analysis on dataset
- import matplotlib.pyplot as plt #library to create plots
- import seaborn as sns #for informative statistical graphics
-

2nd Cell

- # import warnings for removing the additional text from the output when
- # implementing the plots
- import warnings
- warnings.filterwarnings("ignore")

3rd Cell

- #Connect with google drive for reading file
- from google.colab import drive
- drive.mount('/content/drive')

4th Cell

- #read csv file 'application_data.csv' from drive
- app_data = pd.read_csv('/content/drive/MyDrive/Project_files/application_data.csv')

Step 2:Exploring data

- View dataframe
  app_data
- View the First Few Rows
  app_data.head()
- Data Summary
  app_data.info()
- Checking Shape
  app_data.shape
- Summary Statistics
  app_data.describe()
- Describe Categorical Data
  app_data.describe(include='object')
- Unique Values
  app_data.nunique()

As you can see in Course_Project2-Credit_Risk_Analysis.ipynb file, I use to explore the use these. In the previous slide I store the dataframe in app_data thst's why I'm use app_data here for exploring the data.

Step 3:Handling Missing Data

- Step 1: Identify Missing Data

•The displayed DataFrame provides a quick summary of which columns have missing data and how many entries are missing.
•This information is crucial for data preprocessing steps, as handling missing values is essential for ensuring the quality of data used in analysis or modeling.
•Based on the output, further actions may be taken, such as imputing missing values, removing columns, or deciding how to handle rows with missing data in the analysis.

- **Step 2: Visualize Missing Data**

•The bar plot provides a clear visual representation of the distribution of missing values across the dataset:

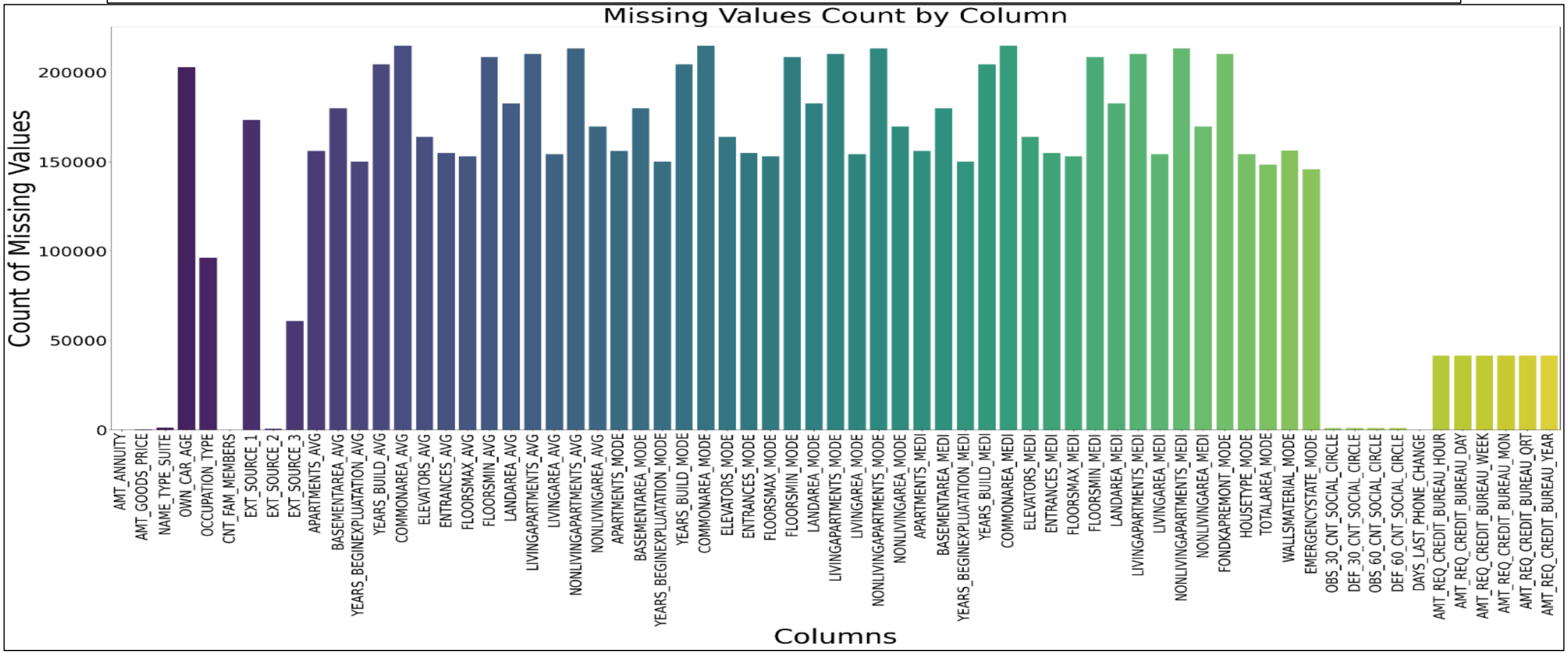
- •You can quickly identify which columns have the most missing values, helping prioritize which variables may need imputation or other forms of handling.
- •Columns with higher bars indicate a larger number of missing values, which could be a concern for data quality and may require further investigation.

•Overall, this visualization helps in assessing the completeness of the dataset and aids in planning the next steps for data cleaning and preprocessing.

Bars:

- •Each bar represents a column's missing values, with the height of the bar corresponding to the number of missing entries.
- •The bars will be colored according to the Viridis palette, providing a gradient effect.

Bar Plot For Missing Values (explained this plot in previous slide)



Step 3: Decide on Handling Strategy

1.Drop columns with more than 50% missing values

- The primary effect of this operation is to reduce the dimensionality of the dataset by removing less reliable columns that may not contribute meaningfully to analysis or modeling due to a high proportion of missing values.
- The resulting DataFrame, app_data_1, will be cleaner, potentially more manageable, and likely to provide better results in subsequent analyses or machine learning tasks, as it removes columns that could introduce noise or bias due to missing data.
- After executing this code, you can check the shape or info of app_data_1 to confirm how many columns remain and how many were dropped based on the missing values criterion.

2.Drop Rows with Missing Values in Important Columns

- The primary effect of this operation is to filter the dataset to include only rows that have complete data for the specified columns.
- Rows where either 'AMT_REQ_CREDIT_BUREAU_HOUR' 'AMT_REQ_CREDIT_BUREAU_DAY' is missing will be removed, which can help ensure that subsequent analyses or modeling processes are not affected by incomplete information.
- This is particularly important in scenarios where these columns are critical for your analysis or model, as missing values in these features could lead to errors or biased results.
- After executing this code, you can check the shape or info of app_data_2 to confirm how many rows remain and how many were dropped based on the missing values in the specified columns.

Step 4:Analyzing Categorical and Numerical data

1.Analyzing Categorical Data

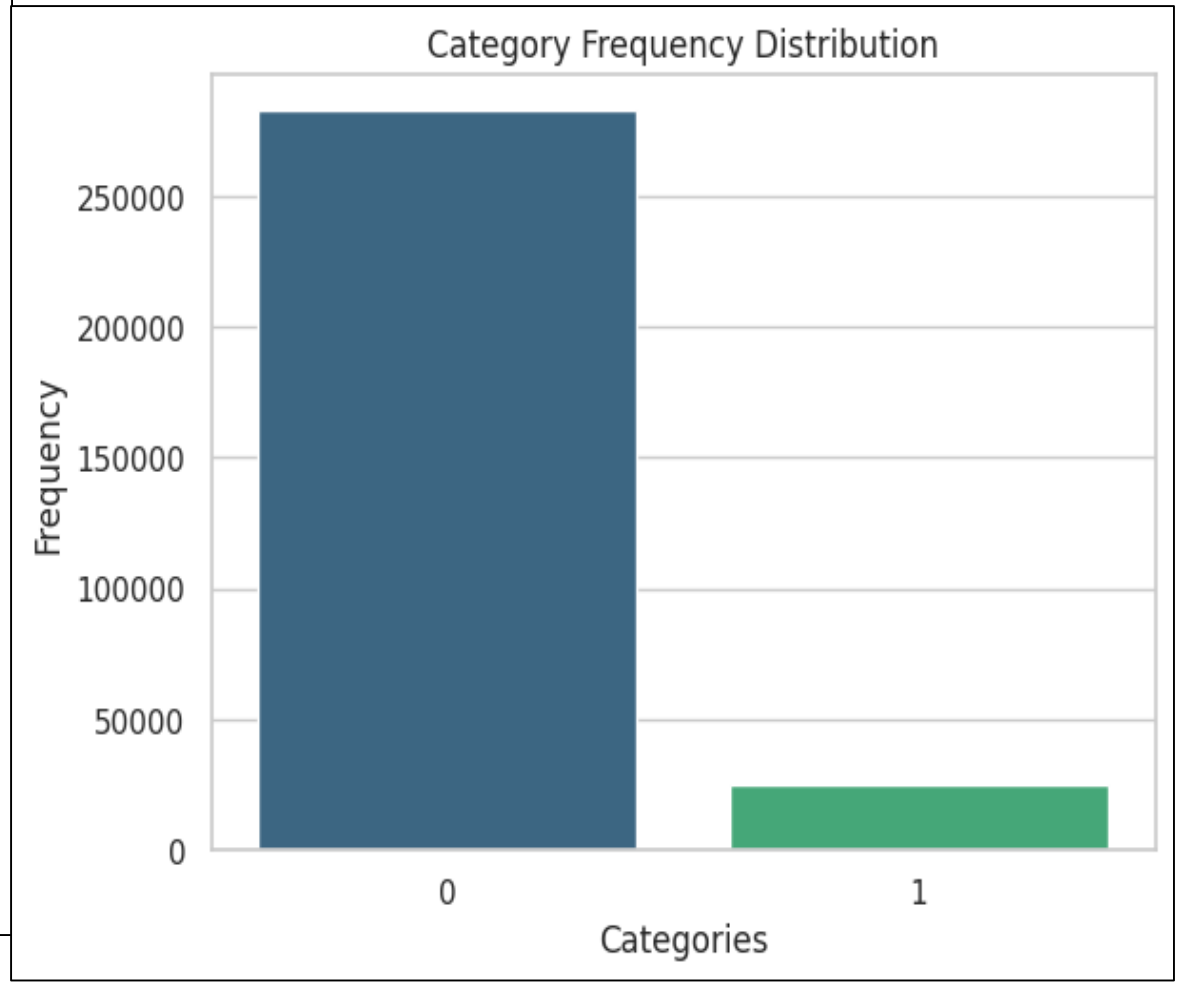
- **cat_data**:A DataFrame summarizing the categorical variables in app_data. It will show counts, unique values, the most frequent values, and their frequencies for each categorical column. This helps identify the distribution of categorical data and can guide further analysis.
- **target_0**: A DataFrame containing only the records of clients without payment difficulties. This DataFrame will have the same structure as app_data, but the number of rows will depend on how many clients fall into this category. This subset can be useful for analyses focused specifically on clients who do not face payment issues.
- **target_1**: A DataFrame containing only the records of clients with payment difficulties. Similar to target_0, this DataFrame will also retain the same structure but will only include rows for clients who have difficulties with payments. This subset is valuable for understanding the characteristics and behaviors of clients who might be at risk of defaulting.

1.Analyzing Categorical Data

Bar Plot:

•A bar plot showing each unique category in TARGET on the x-axis and their respective counts on the y-axis.

•This plot helps in understanding the distribution of the categories in TARGET , such as detecting class imbalance if one category has a significantly higher count than the other(s).



1.Analyzing Categorical Data

Doughnut Chart

Doughnut chart helps to understand the Categorical data. It's a part of pie chart and code is written in the python file.

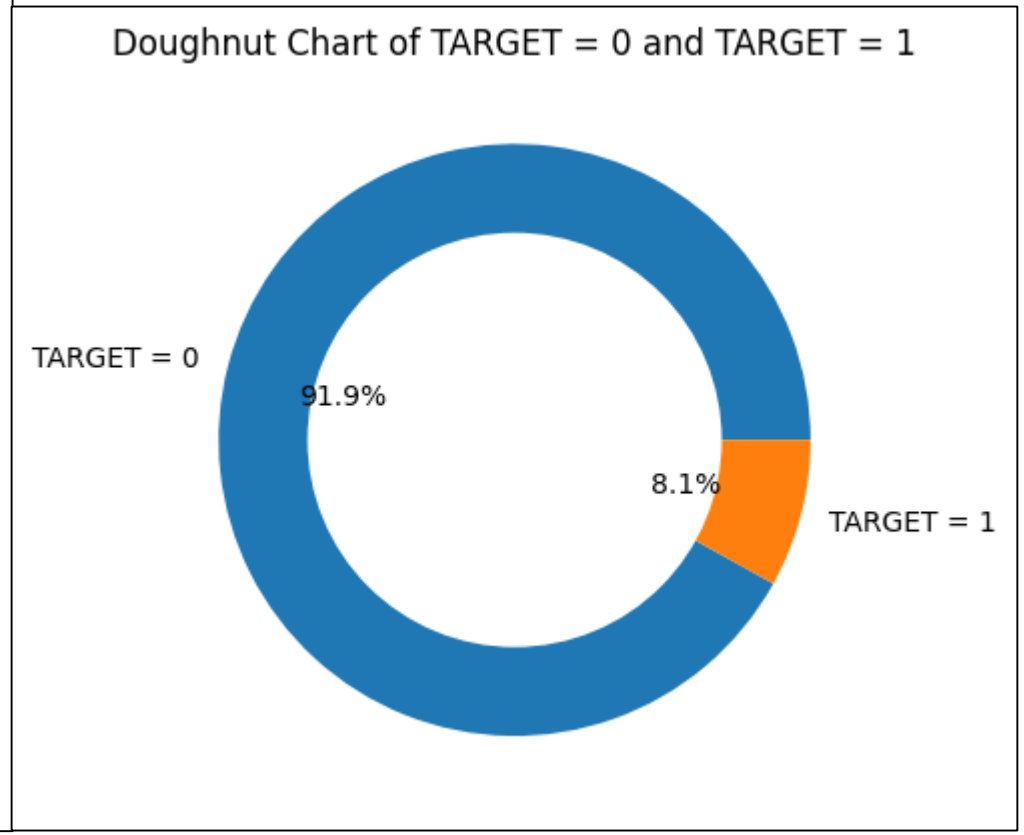
The resulting doughnut chart shows:

•Each section of the doughnut represents a category of **TARGET (0 or 1).**

•The size of each section corresponds to the proportion of records in each category.

•Percentage labels inside each section display the proportion of each category in the dataset, making it easy to understand the distribution between **TARGET = 0 and TARGET = 1**.

This visual helps in identifying any imbalance in the distribution of the **TARGET categories**.

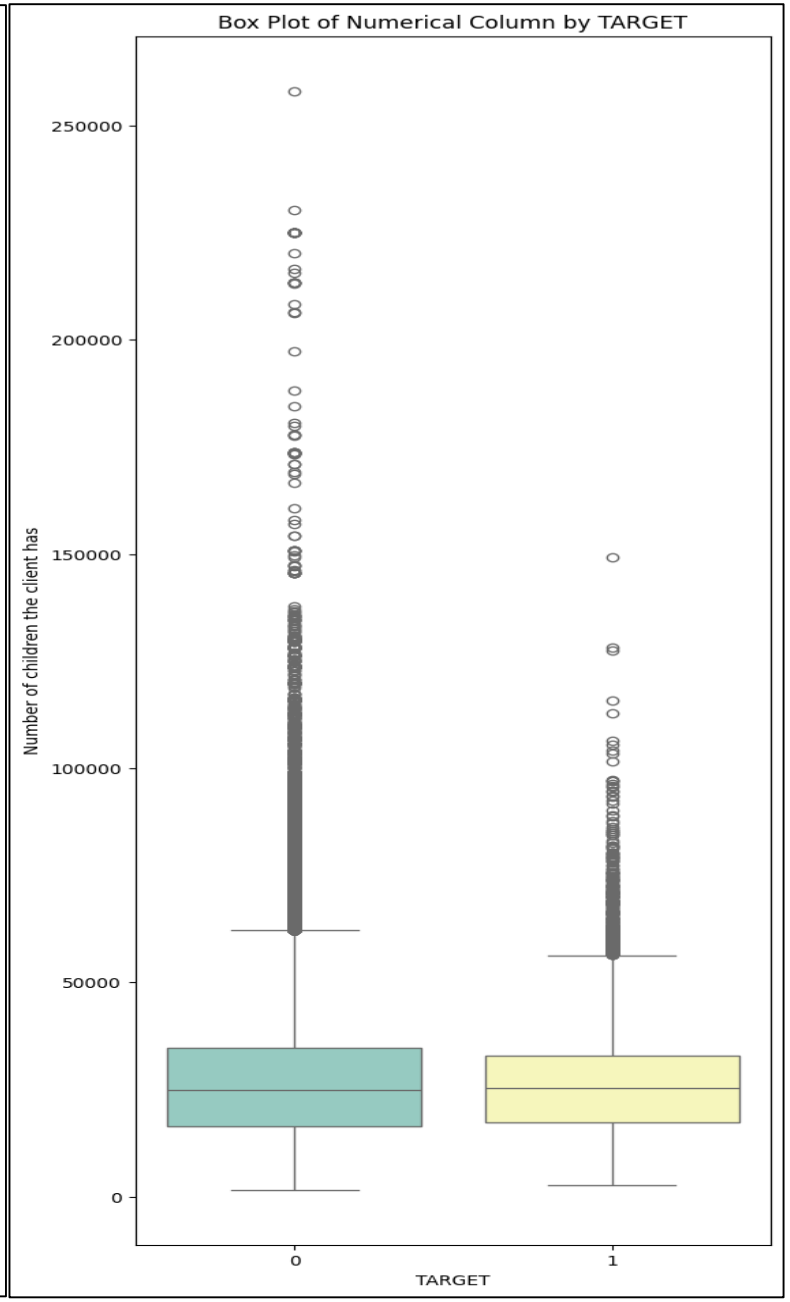


2.Visualize Numerical Data

The resulting box plot displays the distribution of AMT_ANNUITY for each category in TARGET:

- **Each box** represents the interquartile range (IQR) of AMT_ANNUITY for TARGET = 0 and TARGET = 1, showing the middle 50% of values.
- **The line within each box** (the median) represents the median AMT_ANNUITY value for that category.
- **Whiskers** extend to the minimum and maximum values within a defined range, while outliers (if any) appear as individual points beyond the whiskers.

This visualization allows us to compare the spread, central tendency, and outliers of AMT_ANNUITY for each TARGET category, highlighting differences in annuity distribution between the two groups.

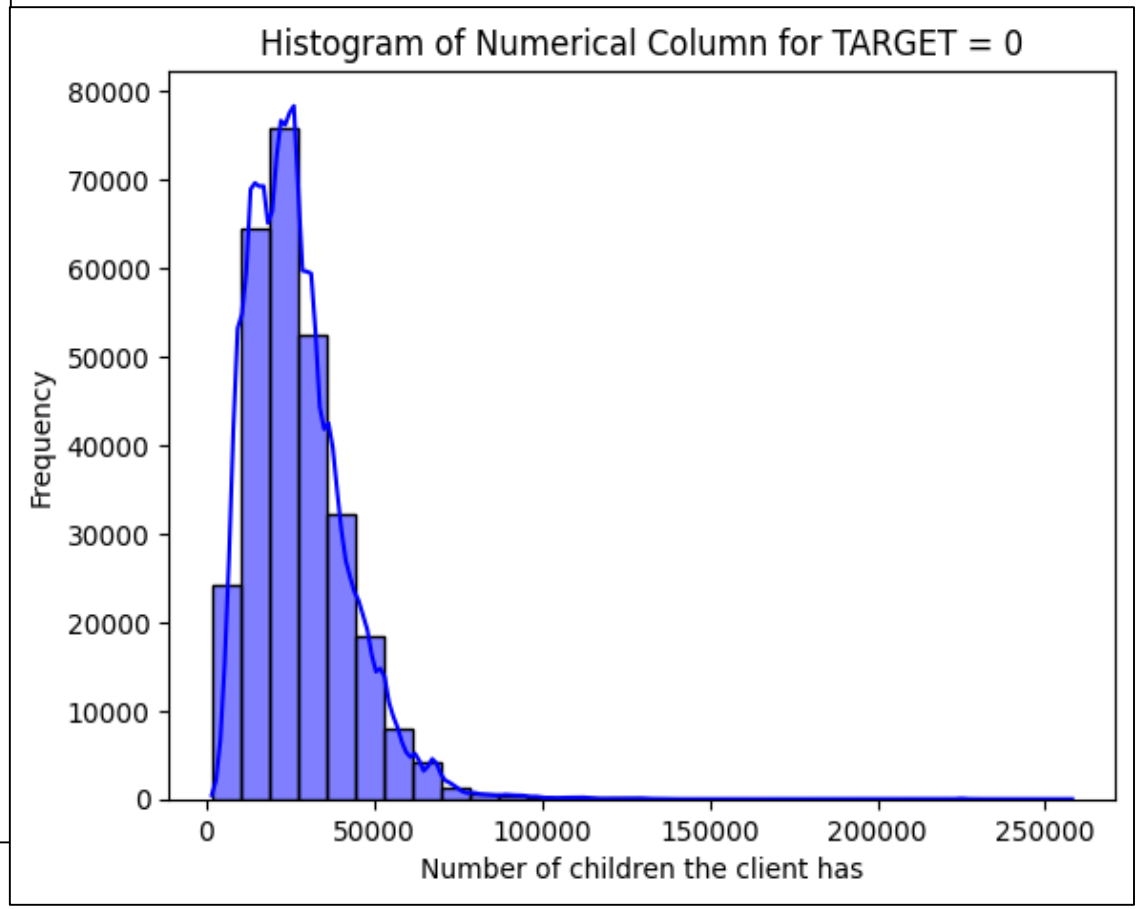


2.Visualize Numerical Data

The resulting histogram shows:

- **Bars**: Represent the frequency of AMT_ANNUITY values in each bin for TARGET = 0, with taller bars indicating more values in that range.
- **KDE Curve**: The smooth line over the histogram indicates the estimated probability density, helping to visualize the shape of the distribution.
- This plot allows us to see the distribution pattern of AMT_ANNUITY values for clients with TARGET = 0, such as if it's skewed (e.g., to the left or right) or has multiple peaks.

Overall, this histogram provides insights into how AMT_ANNUITY values are spread for clients in the TARGET = 0 category.

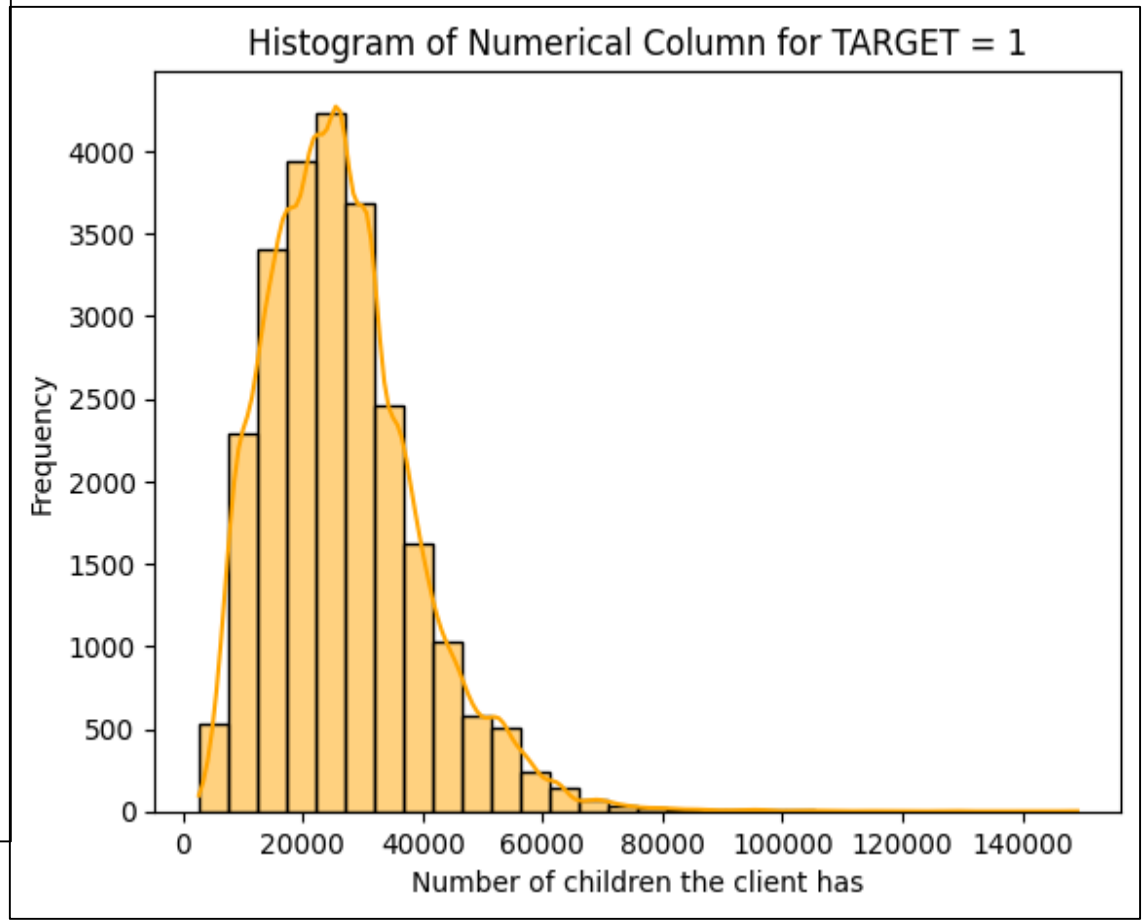


2.Visualize Numerical Data

The resulting histogram shows:

- **Orange Bars**: Represent the frequency distribution of AMT_ANNUITY values specifically for records where TARGET = 1. Taller bars indicate a higher count of AMT_ANNUITY values within that range.
- **KDE Curve**: A smooth orange line that overlays the histogram, illustrating the general shape of the data distribution for AMT_ANNUITY when TARGET = 1.

This plot helps visualize the pattern and spread of AMT_ANNUITY for clients in the TARGET = 1 category. Comparing this distribution to that of TARGET = 0 can provide insights into whether AMT_ANNUITY differs significantly between the two categories. For example, it could reveal if clients with TARGET = 1 tend to have higher or lower annuity amounts.



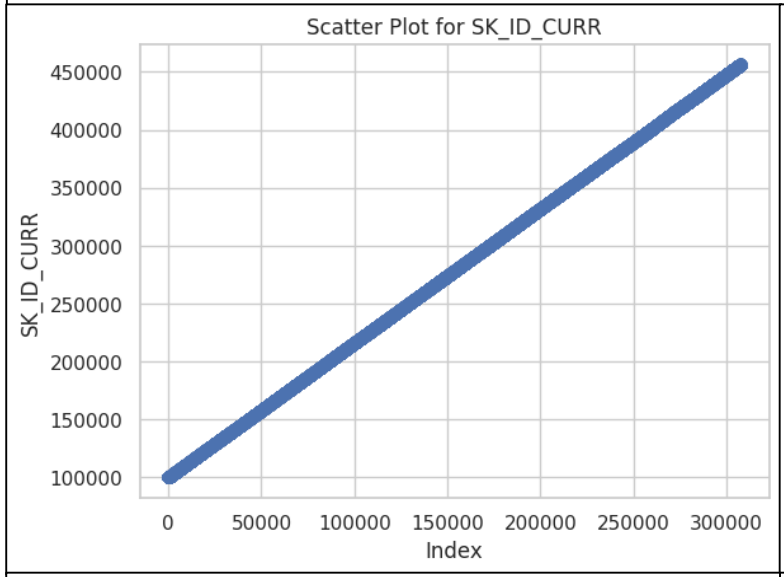
Step 5:Outlier Identification and Visualization

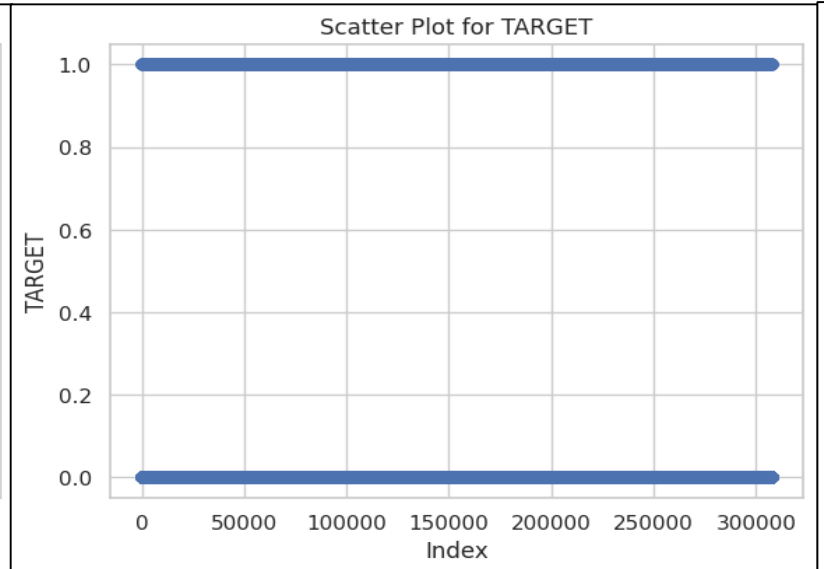
1. Using Statistical Methods (Z-Score and IQR)

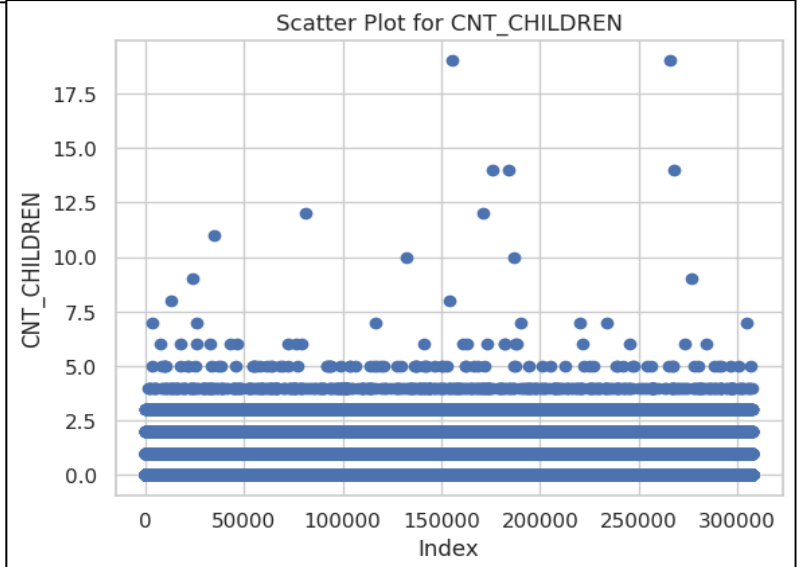
- **Outliers identified by Z-Score:** A subset of app_data that includes rows where the Z-score of the specified column exceeds ±3. These are data points that are unusually high or low relative to the mean.
- **Outliers identified by IQR**: A subset of app_data containing rows where the values in the specified column fall outside the lower or upper bounds defined by the IQR. These rows indicate values that are either significantly below or above the typical range of the column.

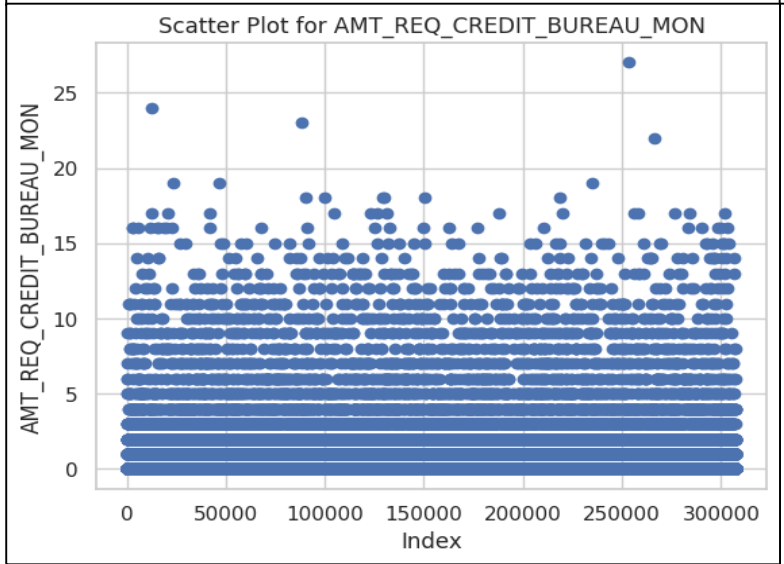
The Z-Score and IQR helps identify unusually high or low values in each numeric column using two different statistical methods, which can assist in understanding the distribution of data and in deciding how to handle outliers. For the output run the given in Python Code file.

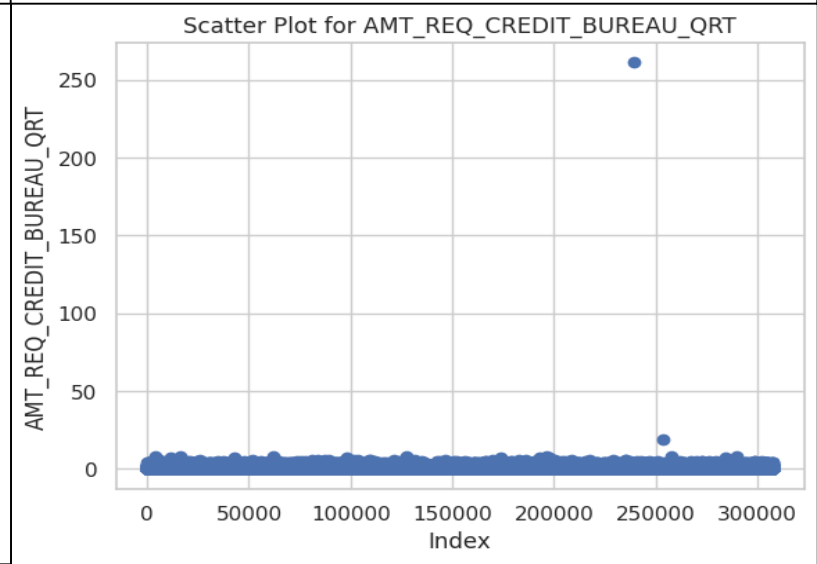
2. Visualization Methods(1st three and last three plot)

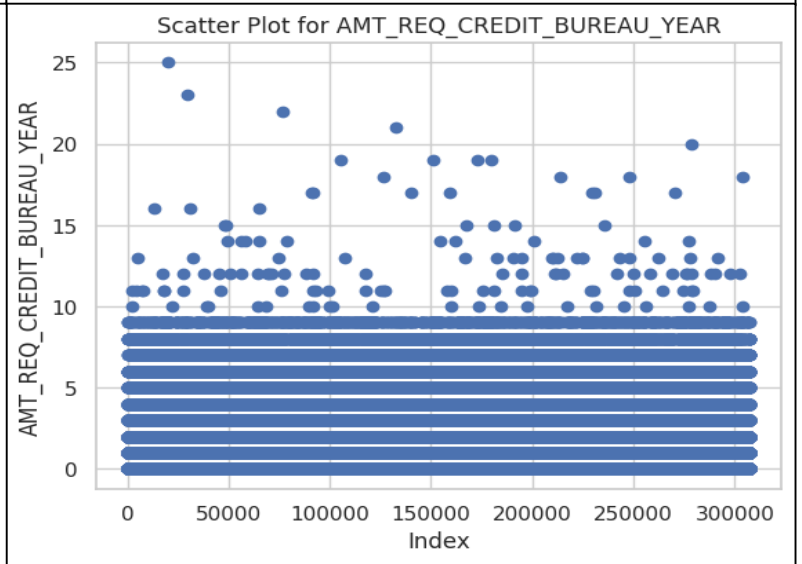












The explanation for the previous slide graphs

The code generates a separate scatter plot for each numerical column in app_data. Each scatter plot shows:

•**X-Axis (Index)**: Represents the position of each record in the DataFrame, ranging from 0 up to the last row.

•**Y-Axis (Column Values)**: Shows the actual values in the current column for each record.

The scatter plot helps visualize the spread and trends in each column:

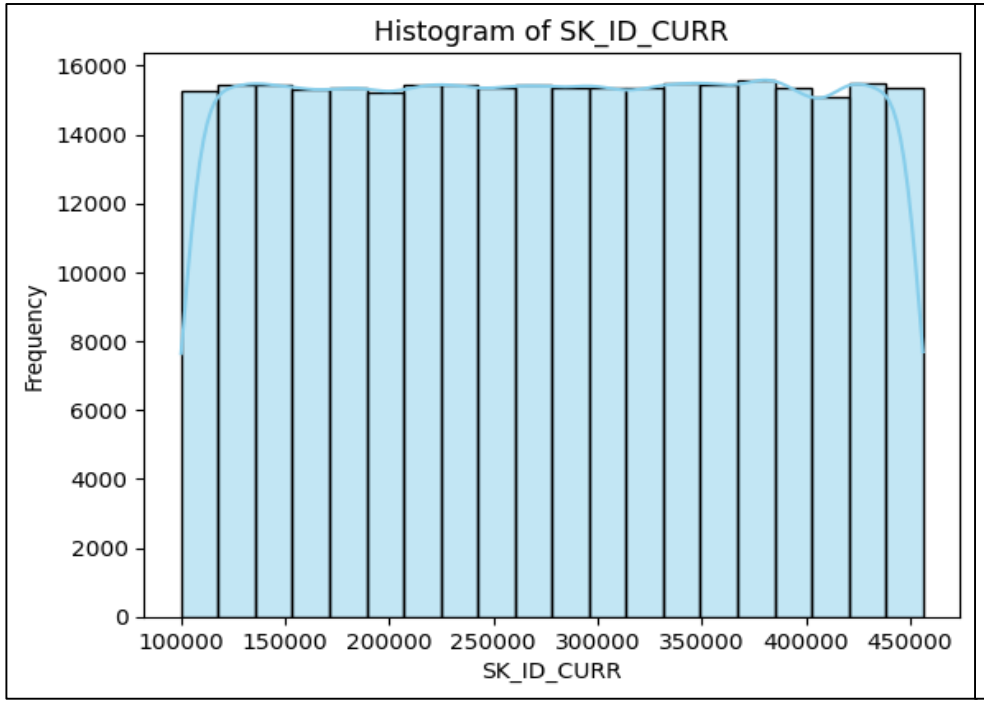
- •**Trends**: Detect patterns across records, such as gradual increases, decreases, or clusters of similar values.
- •**Outliers**: Identify any points that appear significantly higher or lower than the rest of the data, which could represent potential outliers.
- •**Clustering**: Determine if values in a column tend to cluster around specific ranges.

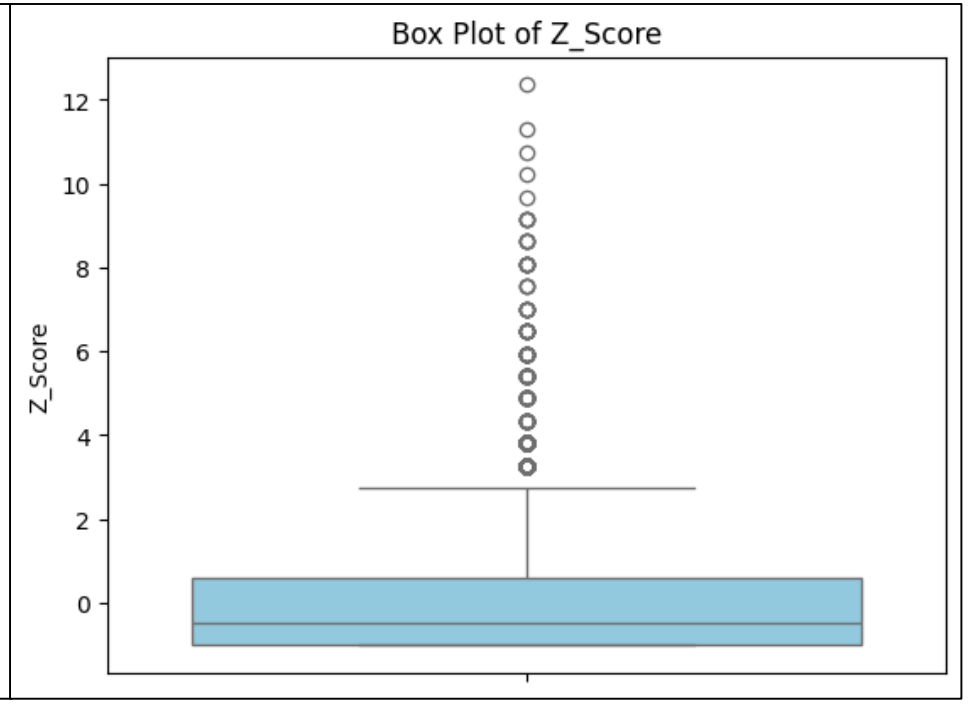
This visualization aids in exploring the distribution and spotting potential anomalies in each numeric column.

Step 6:Univariate and Bivariate Analysis

- Step 1:Univariate Analysis

For Numerical Variables(1st and last plot) for more plot run the code



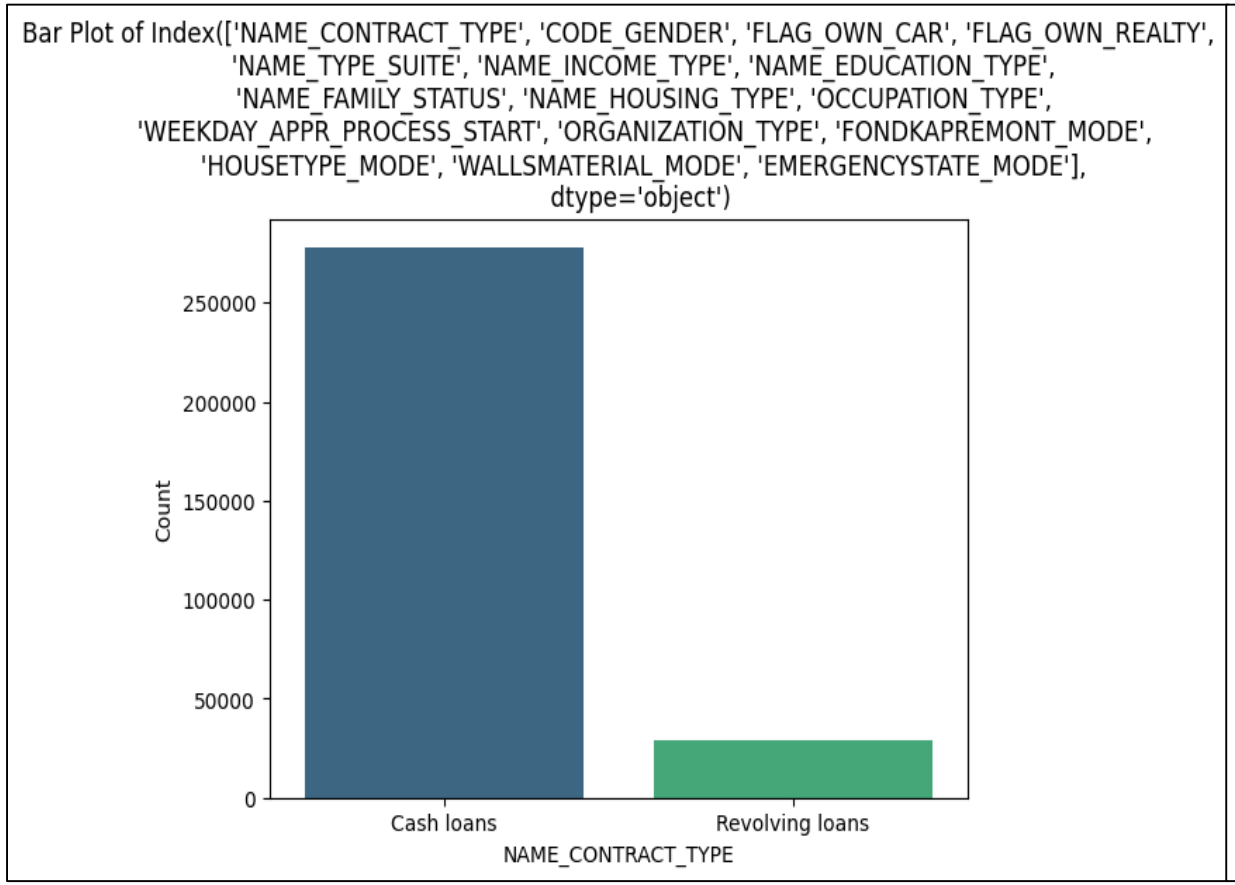


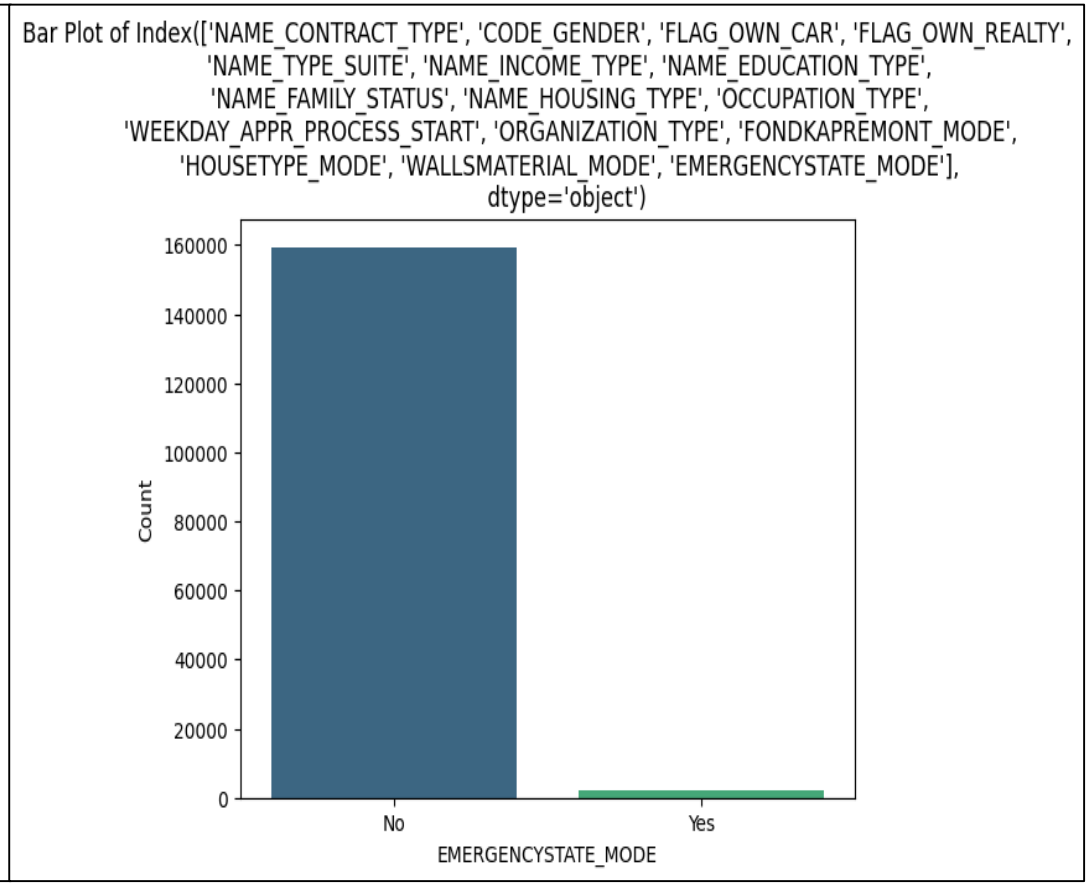
Explanation for previous slide plots

•**Summary Statistics Output**:
•This output provides a concise statistical overview of each numerical column, helping you understand the data's central tendency, variability, and potential outliers.
•**Histograms**:
•Each histogram shows the distribution of the values in the respective column.
•The bars indicate the frequency of values within specified intervals (bins), and the KDE curve helps visualize the underlying distribution shape.
•You can identify the distribution type (e.g., normal, skewed) and spot potential outliers or clusters in the data.
•**Box Plots**:
•Each box plot visually represents the distribution of the values in the respective column.
•The box shows the interquartile range (IQR) with a line for the median, while the "whiskers" extend to the minimum and maximum values within 1.5 times the IQR.
•Any points outside the whiskers are plotted individually, indicating potential outliers.
•This plot helps you quickly see the central tendency, spread, and any outliers in the data.

Step 1:Univariate Analysis

For Categorical Variables(1st and last plot) for more plot run code





Explanation for previous slide plots

•**Frequency Counts Output(for this see code output)**:
•If the code is corrected as mentioned, you will see frequency counts for each categorical column printed in the console.
•Each count shows how many times each unique value appears in that column. This provides insight into the distribution of categories within each categorical variable, helping you understand the dataset's composition.
•**Bar Plots**:
•Each bar plot visualizes the frequency of each category within the respective categorical column.
•The x-axis represents the unique categories, while the y-axis shows the count (frequency) of occurrences for each category.
•The bars are colored according to the 'viridis' palette, enhancing visual appeal and clarity.
•These plots allow for easy comparison between categories, helping to identify dominant categories or any imbalances in the distribution.

Step 2:Bivariate Analysis

- Numerical vs. Numerical Variables

•**Correlation Coefficient Output**:

•The printed correlation coefficient value indicates how closely related the two columns are:

- •A value close to **1** would indicate a strong positive correlation (as one variable increases, the other also increases).
- •A value close to **-1** would indicate a strong negative correlation (as one variable increases, the other decreases).
- •A value close to **0** suggests little to no linear relationship.

•For example, if the output was Correlation between AMT_REQ_CREDIT_BUREAU_DAY and AMT_REQ_CREDIT_BUREAU_WEEK: 0.85, it would imply a strong positive correlation, meaning as the number of days increases, the number of weeks also tends to increase.

•**Scatter Plot**:

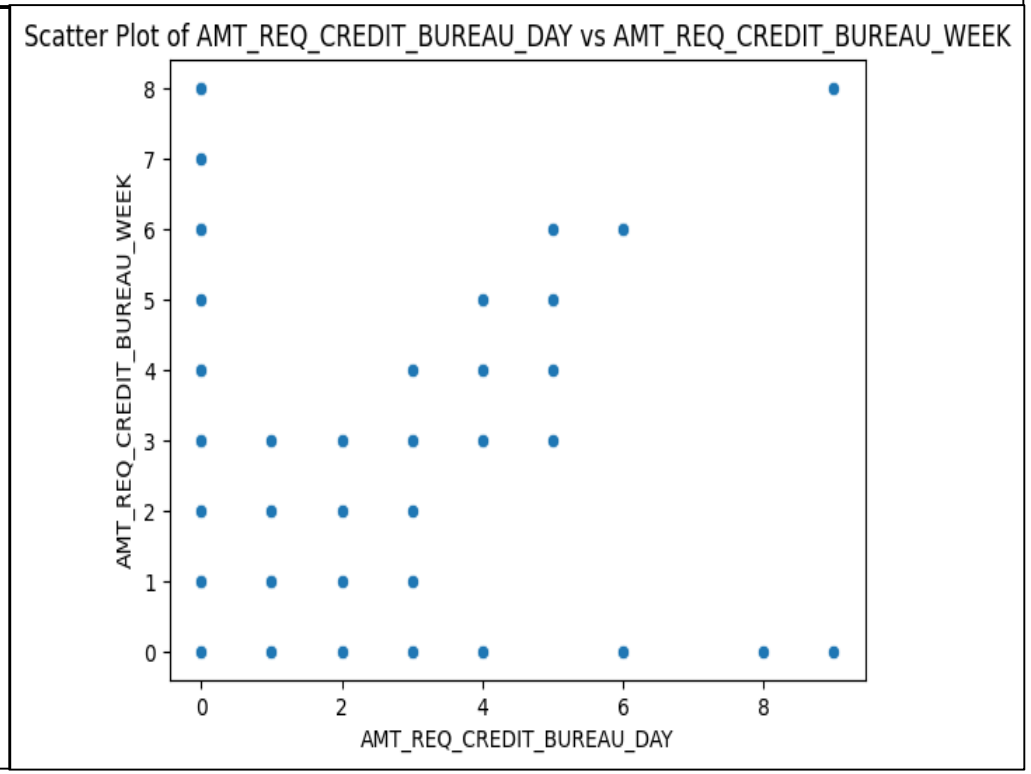
•The scatter plot visualizes the relationship between AMT_REQ_CREDIT_BUREAU_DAY and AMT_REQ_CREDIT_BUREAU_WEEK.

•Each point represents a data point in the app_data DataFrame.

•The pattern of the points can reveal:

- •A positive correlation (points cluster in an upward trend),
- •A negative correlation (points cluster in a downward trend),
- •No correlation (points are dispersed without any apparent trend).

•If the points appear to form a distinct linear pattern, this would suggest a strong correlation.



Step 2:Bivariate Analysis

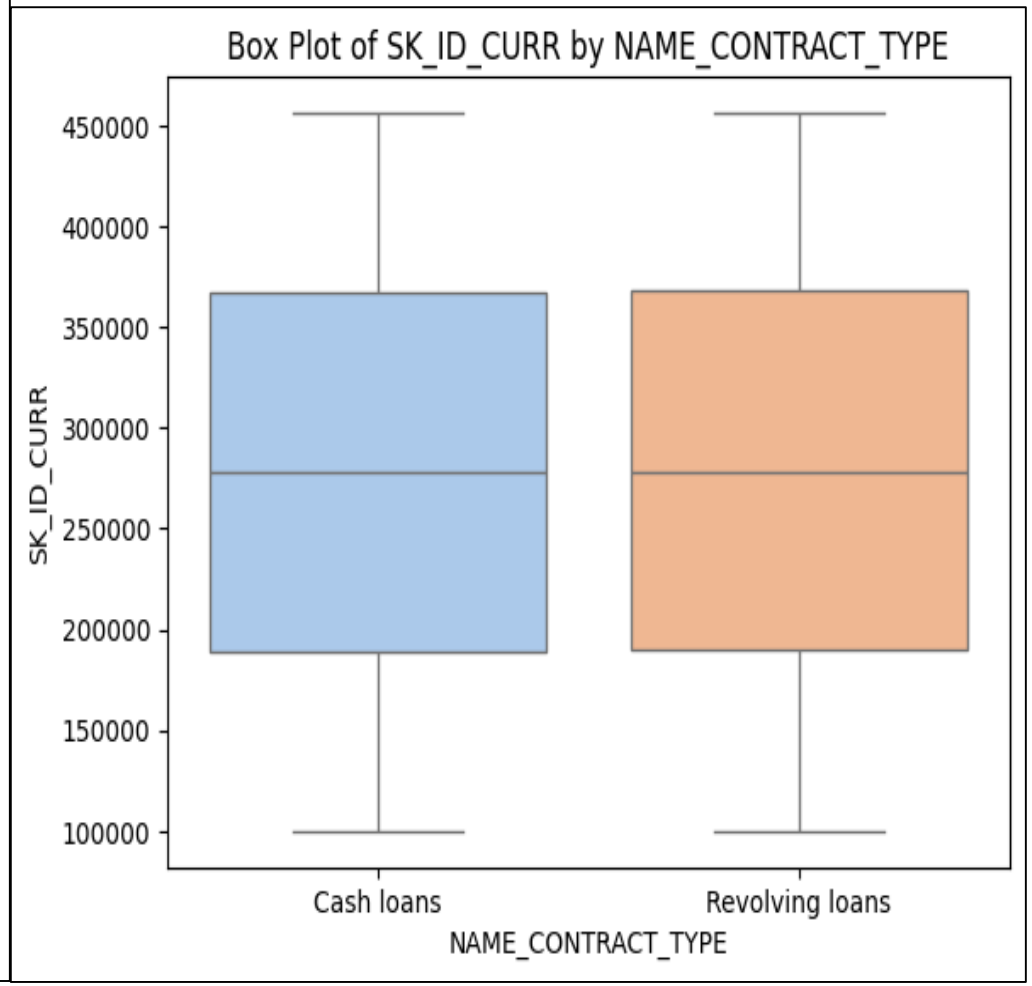
- Categorical vs. Numerical Variables

Box Plot Visualization:

•The box plot visualizes the distribution of the numerical values (num_col) across the different categories defined by the categorical column (cat_col).

•Each box represents the interquartile range (IQR) of the numerical values for each category:

- •**Box**: The box spans from the 25th percentile (Q1) to the 75th percentile (Q3), showing where the central 50% of the data lies.
- •**Line Inside Box**: The line inside the box represents the median (50th percentile) of the numerical values.
- •**Whiskers**: The lines extending from the top and bottom of the boxes (whiskers) typically extend to the maximum and minimum values within 1.5 times the IQR.
- •**Outliers**: Any points beyond the whiskers are plotted individually and represent outliers.



Step 2:Bivariate Analysis

- Categorical vs. Categorical Variables

•**Crosstab Output**:

•The printed crosstab table shows the counts of each combination of NAME_INCOME_TYPE and NAME_EDUCATION_TYPE.

•For example, if there are categories like "Salary," "Business," and "Pension" for income type and "High School," "Bachelor's," and "Master's" for education type, the crosstab would show how many people fall into each income-education category pair.

•This table helps identify patterns in the data, such as which income types are most common among specific education types.

•**Heatmap Visualization**:

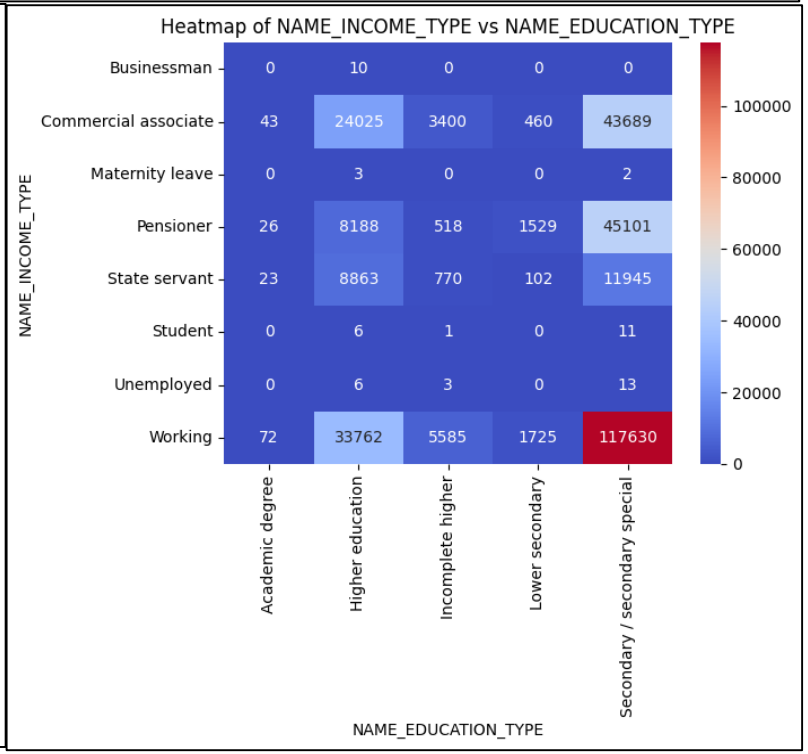
•The heatmap visually represents the same information as the crosstab but in a more intuitive format.

•Each cell's color intensity indicates the count of occurrences for that specific income and education type combination:

- •Darker shades (in the color scheme) represent higher counts, while lighter shades represent lower counts.

•The annotations inside the cells provide exact counts for clarity.

•This heatmap allows for quick visual analysis, making it easy to identify trends and relationships between income and education levels.



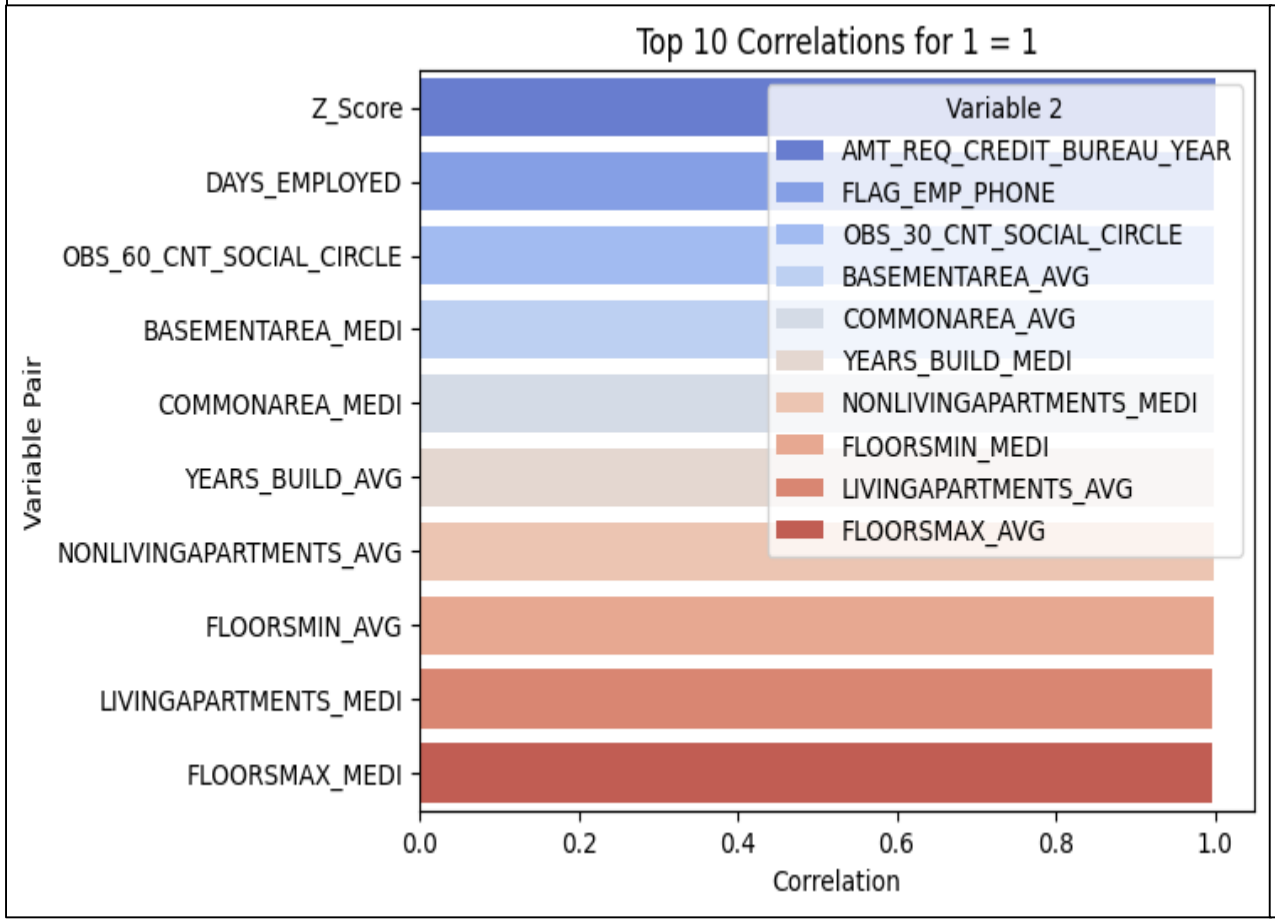
Step 7:Top 10 Correlations Analysis (Segmented by Target)

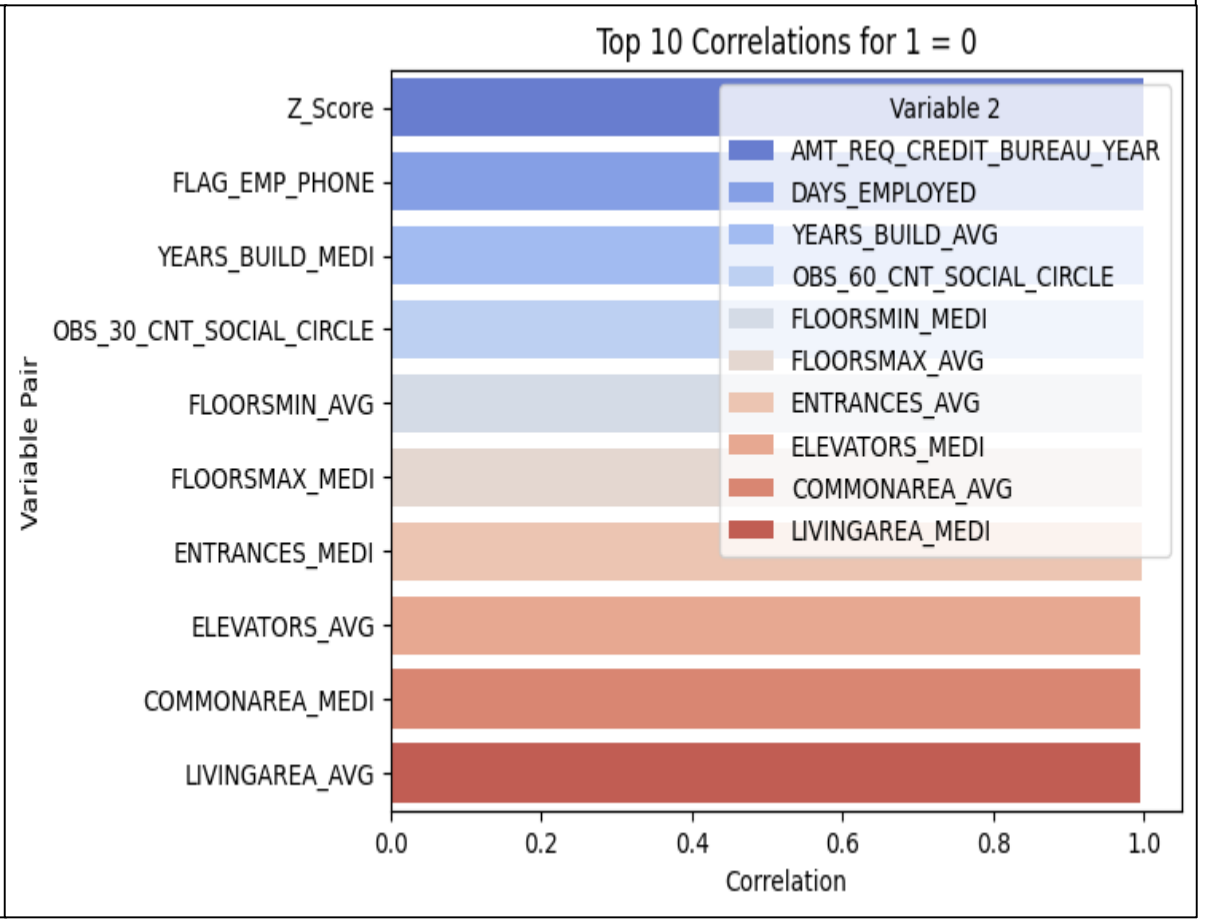
Step 1: Segment Data and Calculate Correlations

Top 10 Correlations for Each Segment:

- The output will display two separate lists of top correlations:
  - One for clients **without** payment difficulties (TARGET = 0).
  - One for clients **with** payment difficulties (TARGET = 1).
- Each list contains pairs of features along with their correlation coefficients, showing how strongly they are related within each segment.
- By comparing the correlations for both segments, you can identify whether certain relationships are stronger or weaker for clients with payment difficulties compared to those without. This could help in understanding different financial behaviors or risk factors associated with clients.

Step 2: Visualize the Top Correlations (Plots for this)





Explanation for previous slide plots

Bar Plots for Each Target Value:

- The output will consist of two bar plots, one for each unique target value (0 and 1):
  - The first plot will show the top correlations for clients **without** payment difficulties (TARGET = 0).
  - The second plot will show the top correlations for clients **with** payment difficulties (TARGET = 1).
- Each plot will display the top 10 pairs of correlated variables along the y-axis, with the length of the bars representing the strength of the correlations on the x-axis.
- By comparing the two plots, you can identify whether the relationships between variables differ significantly based on the presence of payment difficulties.