Project 6

Bank Loan - Case Study

Submitted By

Swathy Mugundan

Table of Contents

Problem Statement

- Imagine you're a data analyst at a finance company that specializes in lending various types of loans to urban customers.
- Your company faces a challenge: Some customers who don't have a sufficient credit history take advantage of this and default on their loans.
- Your task is to use **Exploratory Data Analysis (EDA)** to analyze patterns in the data and ensure that capable applicants are not rejected.
- When a customer applies for a loan, your company faces two risks:
    - If the applicant can repay the loan but is not approved, the company loses business.
    - If the applicant cannot repay the loan and is approved, the company faces a financial loss.

Bank Loan : Dataset Information

- The dataset contains information about loan applications & it includes two types of scenarios:
  - **Customers with payment difficulties:** These are customers who had a late payment of more than X days on at least one of the first Y installments of the loan.
  - **All other cases:** These are cases where the payment was made on time.
- When a customer applies for a loan, there are four possible outcomes:
  - **Approved:** The company has approved the loan application.
  - **Cancelled:** The customer cancelled the application during the approval process.
  - **Refused:** The company rejected the loan.
  - **Unused Offer:** The loan was approved but the customer did not use it.

Bank Loan : Dataset Information

- The main aim is to observe the dataset as a data analyst at a financial company specializing in lending loans. Use Exploratory Data Analysis (EDA) to identify patterns in the data to make informed loan approval decisions.
  - **previous_applications.csv :** It contains information about the previous loan applications. “NAME_CONTRACT_STATUS” is the important attribute for this dataset.
  - **application_data.csv :** It provides the details about the current loan applications. “TARGET” is the primary attribute which can be used for analysis.
  - **columns_description.csv :** It describes the columns present in the other two datasets, explaining what each column represents in detail.

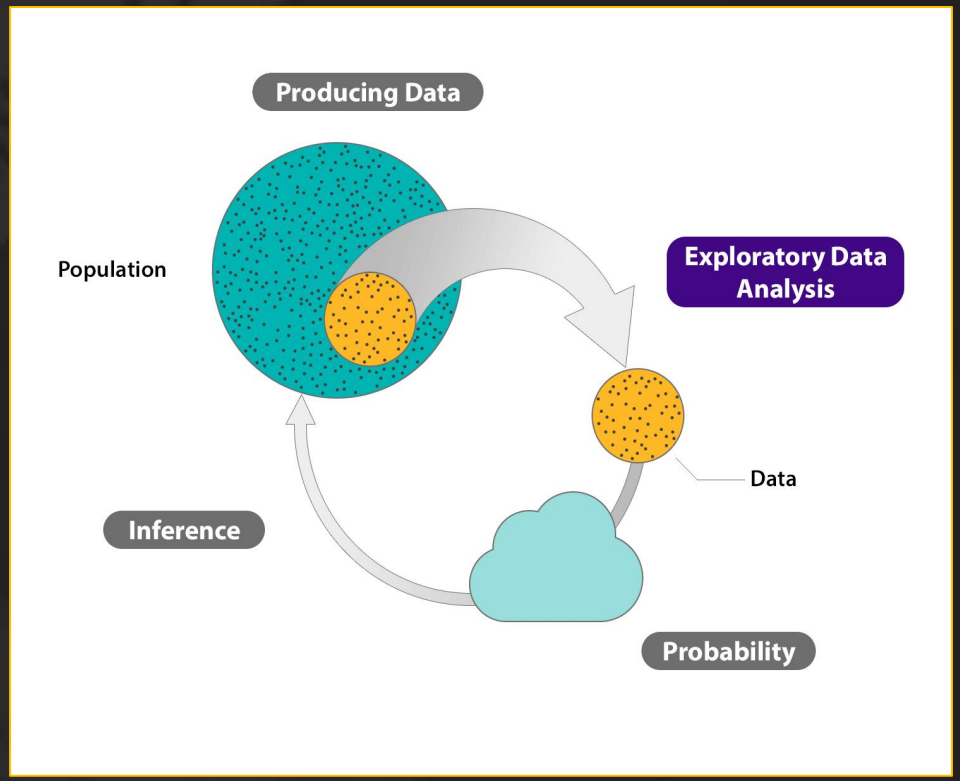
Project Goal / Objectives

- The main aim of this project is to identify patterns that indicate if a customer will have difficulty paying their installments.
- This information can be used to make decisions such as denying the loan, reducing the amount of loan, or lending at a higher interest rate to risky applicants.
- The company wants to understand the key factors behind loan default so it can make better decisions about loan approval.

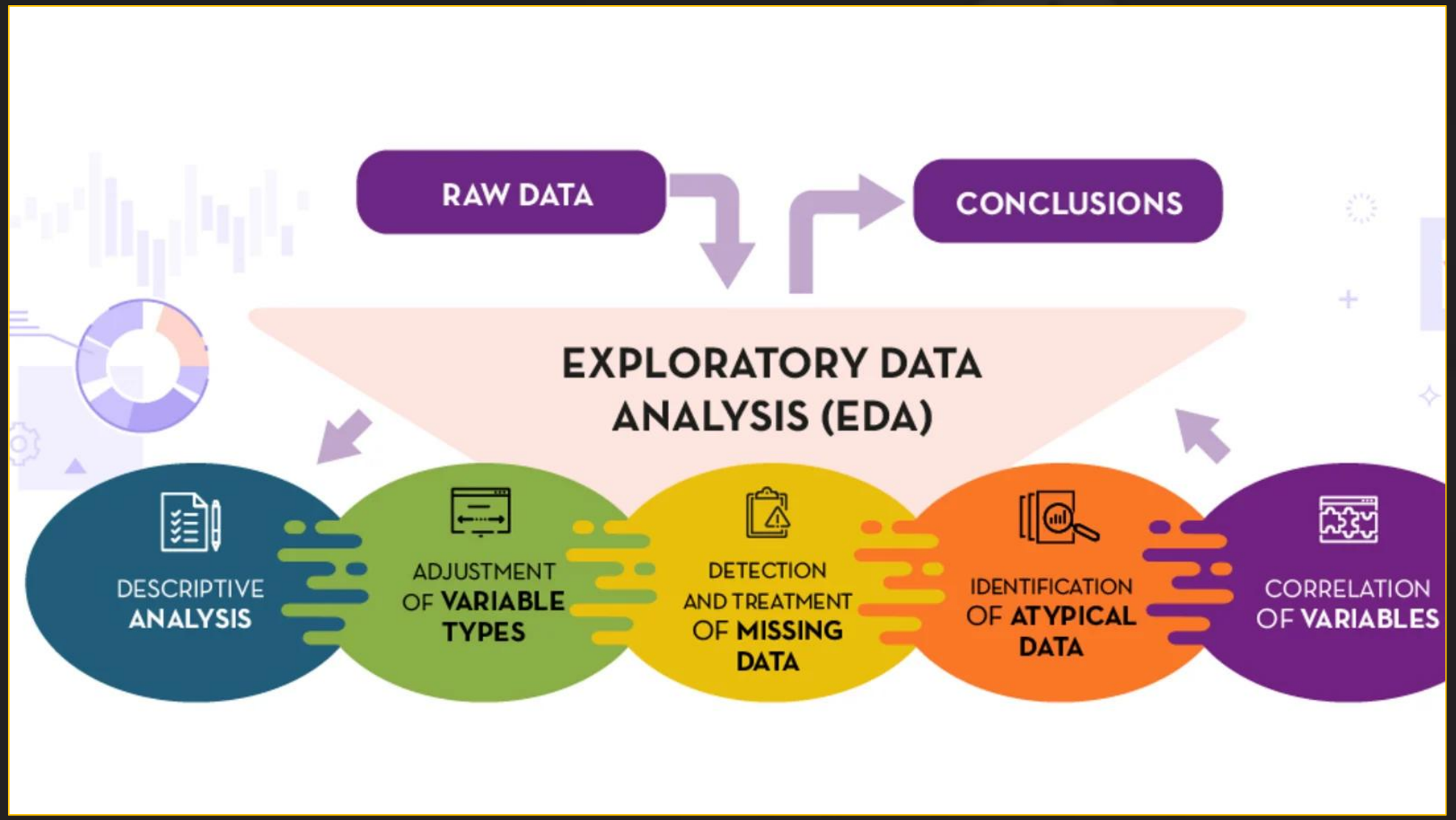
Pre-requisite : To research a bit about risk analytics in banking and financial services and understand the types of variables and their significance should be enough.

Exploratory Data Analysis (EDA)

- Exploratory Data Analysis (EDA) refers to the method of studying and exploring record sets to apprehend their predominant traits, discover patterns, locate outliers, and identify relationships between variables.
- It is normally carried out as a preliminary step before undertaking extra formal statistical analyses or modeling.
- It refers back to the method of analyzing and analyzing information units to uncover styles, pick out relationships, and gain insights.
- There are various sorts of EDA strategies that can be hired relying on the nature of the records and the desires of the evaluation.

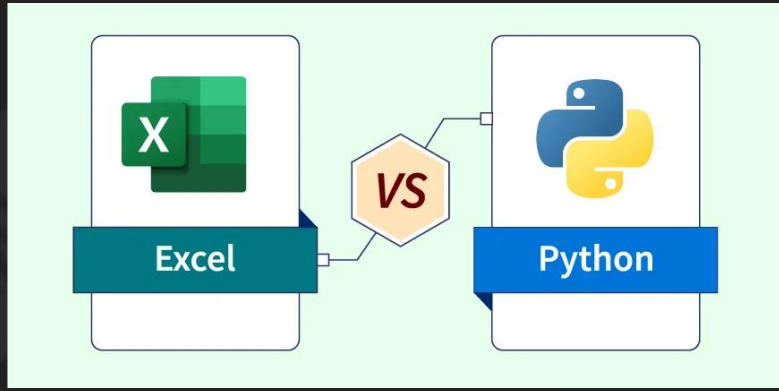


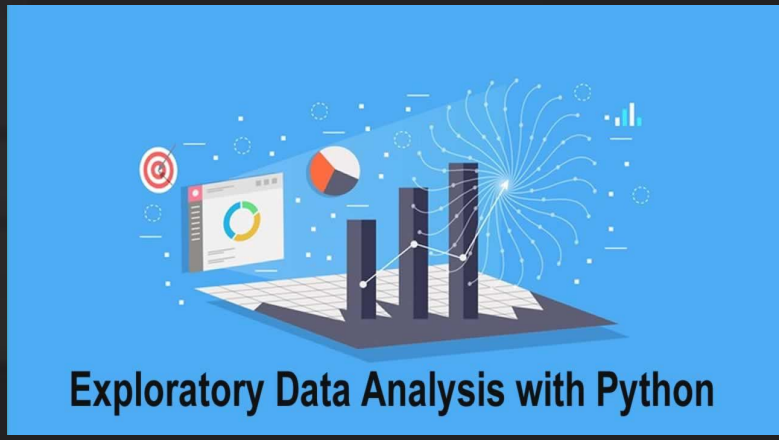
Exploratory Data Analysis (EDA) - Process



Project : Tech Stack Used

- **Why Python over Excel ?** As a data scientist, we often work with lots of different types of data from different sources and in large amounts. While Excel can manage data from multiple sources, Python has libraries that allows to easily access and process data from lots of other sources.
- **Python :** Python libraries and packages can level-up how to analyze, visualize, and understand complex data. It is easy to learn and simple to read, you can start mastering more complicated concepts quicker. To incorporate Machine Learning Predictions, it is much more straightforward with Python
- **Jupyter Notebook :** The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text.





Project : Tech Stack Used - Python Libraries

- **Pandas** is an open-source library in Python that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. Pandas is fast and it has high performance & productivity for users.
- **NumPy** is a general-purpose array-processing package. It provides a high-performance multidimensional array object and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It is open-source software.
- **Math** module provides various the value of various constants . Having such constants saves the time of writing the value of each constant every time we want to use it and that too with great precision. Constants provided by the math module are – Euler's Number, Pi, Tau, Infinity, Not a Number (NaN).
- **Matplotlib** is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.
- **Seaborn** is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It provides dataset-oriented APIs so that we can switch between different visual representations for the same variables for a better understanding of the dataset.

Project 6

Bank Loan - Case Study

Data Pre-Processing Steps

Importing Required Libraries

Importing the required libraries and project setup

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt, seaborn as sns
import math
```

In [2]:
```
#To ignore the warnings
import warnings
warnings.filterwarnings("ignore")
```

In [3]:
```
#To get all the data about rows and columns without limits
get_ipython().run_line_magic('matplotlib', 'inline')
pd.set_option('display.max_columns',None)
pd.set_option('display.max_rows',None)
```

Loading the Dataset

Loading the dataset

In [4]:
```
data = pd.read_csv("/Users/swat/Desktop/Qatar/Trainity-DA/Excel_Projects/application_data.csv")
data.head()
```

Out[4]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDI
0	100002	1	Cash loans	M	N	Y	0	202500.0	406597.
1	100003	0	Cash loans	F	N	N	0	270000.0	1293502.
2	100004	0	Revolving loans	M	Y	Y	0	67500.0	135000.
3	100006	0	Cash loans	F	N	Y	0	135000.0	312682.
4	100007	0	Cash loans	M	N	Y	0	121500.0	513000.

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 49999 entries, 0 to 49998
Columns: 122 entries, SK_ID_CURR to AMT_REQ_CREDIT_BUREAU_YEAR
dtypes: float64(64), int64(42), object(16)
memory usage: 46.5+ MB
```

Exploring the Data Types

Understanding the data

```
In [5]: print(data.shape)
(49999, 122)

In [6]: print(data.dtypes)
```

```
SK_ID_CURR                      int64
TARGET                          int64
NAME_CONTRACT_TYPE             object
CODE_GENDER                    object
FLAG_OWN_CAR                   object
FLAG_OWN_REALTY                object
CNT_CHILDREN                    int64
AMT_INCOME_TOTAL              float64
AMT_CREDIT                    float64
AMT_ANNUITY                   float64
AMT_GOODS_PRICE               float64
NAME_TYPE_SUITE                object
NAME_INCOME_TYPE               object
NAME_EDUCATION_TYPE            object
NAME_FAMILY_STATUS             object
NAME_HOUSING_TYPE              object
REGION_POPULATION_RELATIVE    float64
DAYS_BIRTH                      int64
DAYS_EMPLOYED                   int64
```

```
DAYS_REGISTRATION               int64
DAYS_ID_PUBLISH                 int64
OWN_CAR_AGE                   float64
FLAG_MOBIL                      int64
FLAG_EMP_PHONE                  int64
FLAG_WORK_PHONE                 int64
FLAG_CONT_MOBILE                int64
FLAG_PHONE                      int64
FLAG_EMAIL                      int64
OCCUPATION_TYPE                object
CNT_FAM_MEMBERS               float64
REGION_RATING_CLIENT            int64
REGION_RATING_CLIENT_W_CITY     int64
WEEKDAY_APPR_PROCESS_START     object
HOUR_APPR_PROCESS_START         int64
REG_REGION_NOT_LIVE_REGION      int64
REG_REGION_NOT_WORK_REGION      int64
LIVE_REGION_NOT_WORK_REGION     int64
REG_CITY_NOT_LIVE_CITY          int64
```

```
REG_CITY_NOT_WORK_CITY          int64
LIVE_CITY_NOT_WORK_CITY         int64
ORGANIZATION_TYPE              object
EXT_SOURCE_1                  float64
EXT_SOURCE_2                  float64
EXT_SOURCE_3                  float64
APARTMENTS_AVG                float64
BASEMENTAREA_AVG              float64
YEARS_BEGINEXPLUATATION_AVG   float64
YEARS_BUILD_AVG               float64
COMMONAREA_AVG                float64
ELEVATORS_AVG                 float64
ENTRANCES_AVG                 float64
FLOORSMAX_AVG                 float64
FLOORSMIN_AVG                 float64
LANDAREA_AVG                  float64
LIVINGAPARTMENTS_AVG          float64
LIVINGAREA_AVG                float64
NONLIVINGAPARTMENTS_AVG       float64
NONLIVINGAREA_AVG             float64
```

```
APARTMENTS_MODE               float64
BASEMENTAREA_MODE             float64
YEARS_BEGINEXPLUATATION_MODE  float64
YEARS_BUILD_MODE              float64
COMMONAREA_MODE               float64
ELEVATORS_MODE                float64
ENTRANCES_MODE                float64
FLOORSMAX_MODE                float64
FLOORSMIN_MODE                float64
LANDAREA_MODE                 float64
LIVINGAPARTMENTS_MODE         float64
LIVINGAREA_MODE               float64
NONLIVINGAPARTMENTS_MODE      float64
NONLIVINGAREA_MODE            float64
APARTMENTS_MEDI               float64
BASEMENTAREA_MEDI             float64
YEARS_BEGINEXPLUATATION_MEDI  float64
YEARS_BUILD_MEDI              float64
COMMONAREA_MEDI               float64
ELEVATORS_MEDI                float64
```

```
ELEVATORS_MEDI                float64
ENTRANCES_MEDI                float64
FLOORSMAX_MEDI                float64
FLOORSMIN_MEDI                float64
LANDAREA_MEDI                 float64
LIVINGAPARTMENTS_MEDI         float64
LIVINGAREA_MEDI               float64
NONLIVINGAPARTMENTS_MEDI      float64
NONLIVINGAREA_MEDI            float64
FONDKAPREMONT_MODE             object
HOUSETYPE_MODE                 object
TOTALAREA_MODE                float64
WALLSMATERIAL_MODE             object
EMERGENCYSTATE_MODE            object
OBS_30_CNT_SOCIAL_CIRCLE      float64
DEF_30_CNT_SOCIAL_CIRCLE      float64
OBS_60_CNT_SOCIAL_CIRCLE      float64
DEF_60_CNT_SOCIAL_CIRCLE      float64
DAYS_LAST_PHONE_CHANGE        float64
```

```
FLAG_DOCUMENT_2                 int64
FLAG_DOCUMENT_3                 int64
FLAG_DOCUMENT_4                 int64
FLAG_DOCUMENT_5                 int64
FLAG_DOCUMENT_6                 int64
FLAG_DOCUMENT_7                 int64
FLAG_DOCUMENT_8                 int64
FLAG_DOCUMENT_9                 int64
FLAG_DOCUMENT_10                int64
FLAG_DOCUMENT_11                int64
FLAG_DOCUMENT_12                int64
FLAG_DOCUMENT_13                int64
FLAG_DOCUMENT_14                int64
FLAG_DOCUMENT_15                int64
FLAG_DOCUMENT_16                int64
FLAG_DOCUMENT_17                int64
FLAG_DOCUMENT_18                int64
FLAG_DOCUMENT_19                int64
FLAG_DOCUMENT_20                int64
```

```
FLAG_DOCUMENT_21                int64
AMT_REQ_CREDIT_BUREAU_HOUR    float64
AMT_REQ_CREDIT_BUREAU_DAY     float64
AMT_REQ_CREDIT_BUREAU_WEEK    float64
AMT_REQ_CREDIT_BUREAU_MON     float64
AMT_REQ_CREDIT_BUREAU_QRT     float64
AMT_REQ_CREDIT_BUREAU_YEAR    float64
dtype: object
```

Statistical Analysis of Data - Sample Output

```
data.describe()
```

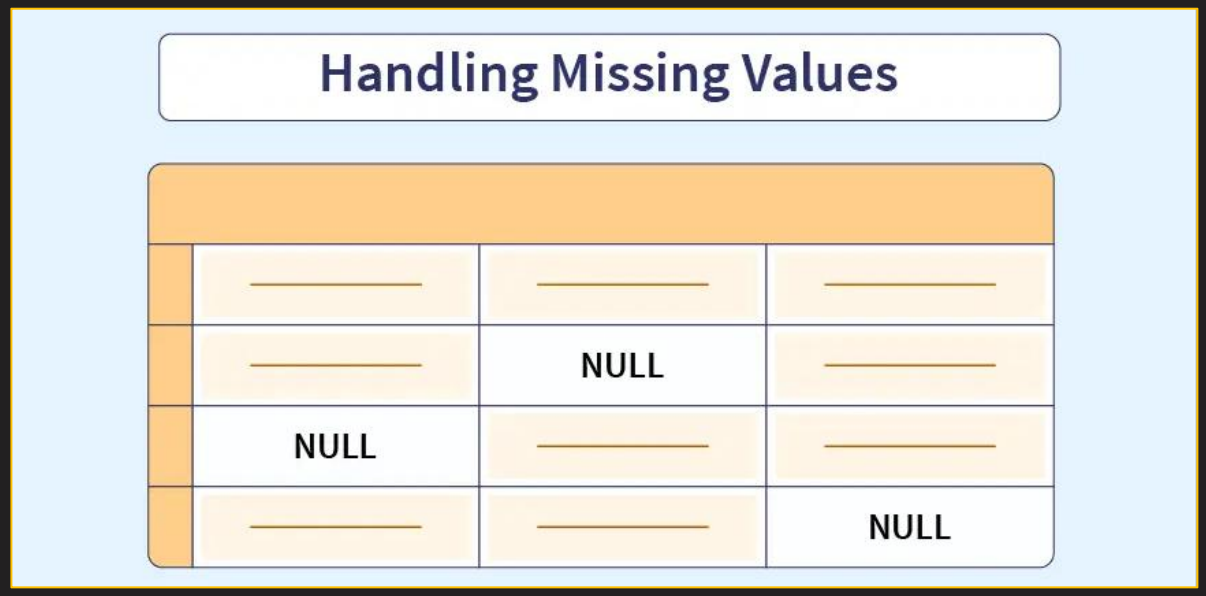
	SK_ID_CURR	TARGET	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_PRICE	REGION_POPULATION_RELATIVE
count	49999.000000	49999.000000	49999.000000	4.999900e+04	4.999900e+04	49998.000000	4.996100e+04	49999.000000
mean	129013.210584	0.080522	0.419848	1.707676e+05	5.997006e+05	27107.377355	5.390600e+05	0.020798
std	16690.512048	0.272102	0.724039	5.318191e+05	4.024154e+05	14562.944435	3.698533e+05	0.013761
min	100002.000000	0.000000	0.000000	2.565000e+04	4.500000e+04	2052.000000	4.500000e+04	0.000533
25%	114570.500000	0.000000	0.000000	1.125000e+05	2.700000e+05	16456.500000	2.385000e+05	0.010006
50%	129076.000000	0.000000	0.000000	1.458000e+05	5.147775e+05	24939.000000	4.500000e+05	0.018850
75%	143438.500000	0.000000	1.000000	2.025000e+05	8.086500e+05	34596.000000	6.795000e+05	0.028663
max	157875.000000	1.000000	11.000000	1.170000e+08	4.050000e+06	258025.500000	4.050000e+06	0.072508

DAYS_BIRTH	DAYS_EMPLOYED	DAYS_REGISTRATION	DAYS_ID_PUBLISH	OWN_CAR_AGE	FLAG_MOBIL	FLAG_EMP_PHONE	FLAG_WORK_PHONE
49999.000000	49999.000000	49999.000000	49999.000000	17049.000000	49999.000000	49999.000000	49999.000000
-16022.042081	63219.424488	-4977.282666	-2996.797176	12.025749	0.999980	0.821476	0.199264
4361.400270	140794.605668	3525.548305	1509.235410	11.887519	0.004472	0.382957	0.399451
-25184.000000	-17531.000000	-22392.000000	-6232.000000	0.000000	0.000000	0.000000	0.000000
-19644.000000	-2786.000000	-7463.500000	-4297.000000	5.000000	1.000000	1.000000	0.000000
-15731.000000	-1221.000000	-4490.000000	-3261.000000	9.000000	1.000000	1.000000	0.000000
-12378.500000	-292.000000	-1998.000000	-1722.000000	15.000000	1.000000	1.000000	0.000000
-7680.000000	365243.000000	0.000000	0.000000	65.000000	1.000000	1.000000	1.000000

Project 6

Bank Loan - Case Study

Task 1 : Handling Missing Data



Task 1 : Handling Missing Data

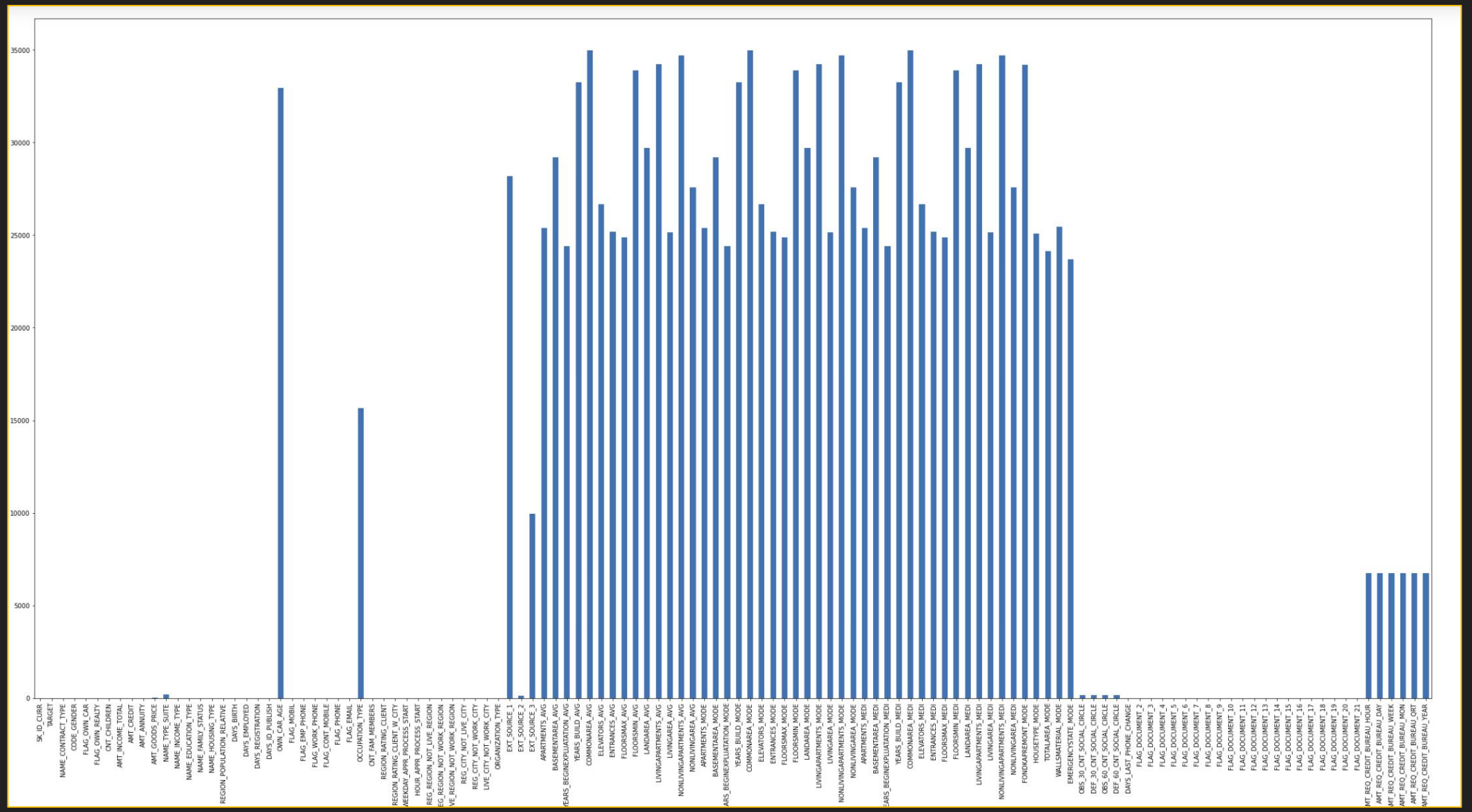
- **Objective :** To identify the missing data in the dataset and decide on the appropriate methods to deal with those using the Excel functions and features.
- **Approach :** Calculating the percentage of missing values for all the columns. Number of missing values for row-wise is also calculated.
  - Value >50% : Columns are removed as they don't have any impact for further analysis.
  - Value ~= 13% : Imputation Technique is chosen for filling the missed data. (4 Columns)
- **Data Visualisation :** Using bar-chart visualisation, the percentage of missing values for each column is plotted.

Task 1 : Handling Missing Data

```
missing_cols =
pd.DataFrame(data.isnull().mean().round(4)*100,
columns=['Percentage_Missing_Value']).sort_val
ues(by=['Percentage_Missing_Value'])
print(missing_cols)
```

	Percentage_Missing_Value
SK_ID_CURR	0.00
HOUR_APPR_PROCESS_START	0.00
REG_REGION_NOT_WORK_REGION	0.00
LIVE_REGION_NOT_WORK_REGION	0.00
REG_CITY_NOT_LIVE_CITY	0.00
REG_CITY_NOT_WORK_CITY	0.00
LIVE_CITY_NOT_WORK_CITY	0.00
ORGANIZATION_TYPE	0.00
FLAG_DOCUMENT_21	0.00
FLAG_DOCUMENT_20	0.00
FLAG_DOCUMENT_19	0.00
FLAG_DOCUMENT_18	0.00
FLAG_DOCUMENT_17	0.00
FLAG_DOCUMENT_16	0.00
WEEKDAY_APPR_PROCESS_START	0.00
FLAG_DOCUMENT_15	0.00
FLAG_DOCUMENT_13	0.00
YEARS_BUILD_MODE	66.48
YEARS_BUILD_AVG	66.48
YEARS_BUILD_MEDI	66.48
FLOORSMIN_AVG	67.79
FLOORSMIN_MODE	67.79
FLOORSMIN_MEDI	67.79
FONDKAPREMONT_MODE	68.38
LIVINGAPARTMENTS_AVG	68.45
LIVINGAPARTMENTS_MODE	68.45
LIVINGAPARTMENTS_MEDI	68.45
NONLIVINGAPARTMENTS_AVG	69.43
NONLIVINGAPARTMENTS_MEDI	69.43
NONLIVINGAPARTMENTS_MODE	69.43
COMMONAREA_MODE	69.92
COMMONAREA_AVG	69.92

Task 1 : Handling Missing Data - Visualisation



Task 1 : Handling Missing Data - Column Removal

Removing columns with >50% missing values

```
new_data = data.drop(data.columns[data.apply(lambda col : (col.isnull().sum()/len(data)*100) >50)], axis=1)
print(new_data.columns)
```

```
Index(['SK_ID_CURR', 'TARGET', 'NAME_CONTRACT_TYPE', 'CODE_GENDER',
       'FLAG_OWN_CAR', 'FLAG_OWN_REALTY', 'CNT_CHILDREN', 'AMT_INCOME_TOTAL',
       'AMT_CREDIT', 'AMT_ANNUITY', 'AMT_GOODS_PRICE', 'NAME_TYPE_SUITE',
       'NAME_INCOME_TYPE', 'NAME_EDUCATION_TYPE', 'NAME_FAMILY_STATUS',
       'NAME_HOUSING_TYPE', 'REGION_POPULATION_RELATIVE', 'DAYS_BIRTH',
       'DAYS_EMPLOYED', 'DAYS_REGISTRATION', 'DAYS_ID_PUBLISH', 'FLAG_MOBIL',
       'FLAG_EMP_PHONE', 'FLAG_WORK_PHONE', 'FLAG_CONT_MOBILE', 'FLAG_PHONE',
       'FLAG_EMAIL', 'OCCUPATION_TYPE', 'CNT_FAM_MEMBERS',
       'REGION_RATING_CLIENT', 'REGION_RATING_CLIENT_W_CITY',
       'WEEKDAY_APPR_PROCESS_START', 'HOUR_APPR_PROCESS_START',
       'REG_REGION_NOT_LIVE_REGION', 'REG_REGION_NOT_WORK_REGION',
       'LIVE_REGION_NOT_WORK_REGION', 'REG_CITY_NOT_LIVE_CITY',
       'REG_CITY_NOT_WORK_CITY', 'LIVE_CITY_NOT_WORK_CITY',
       'ORGANIZATION_TYPE', 'EXT_SOURCE_2', 'EXT_SOURCE_3',
       'YEARS_BEGINEXPLUATATION_AVG', 'FLOORSMAX_AVG',
       'YEARS_BEGINEXPLUATATION_MODE', 'FLOORSMAX_MODE',
       'YEARS_BEGINEXPLUATATION_MEDI', 'FLOORSMAX_MEDI', 'TOTALAREA_MODE',
       'EMERGENCYSTATE_MODE', 'OBS_30_CNT_SOCIAL_CIRCLE',
       'DEF_30_CNT_SOCIAL_CIRCLE', 'OBS_60_CNT_SOCIAL_CIRCLE',
       'DEF_60_CNT_SOCIAL_CIRCLE', 'DAYS_LAST_PHONE_CHANGE', 'FLAG_DOCUMENT_2',
       'FLAG_DOCUMENT_3', 'FLAG_DOCUMENT_4', 'FLAG_DOCUMENT_5',
       'FLAG_DOCUMENT_6', 'FLAG_DOCUMENT_7', 'FLAG_DOCUMENT_8',
       'FLAG_DOCUMENT_9', 'FLAG_DOCUMENT_10', 'FLAG_DOCUMENT_11',
       'FLAG_DOCUMENT_12', 'FLAG_DOCUMENT_13', 'FLAG_DOCUMENT_14',
       'FLAG_DOCUMENT_15', 'FLAG_DOCUMENT_16', 'FLAG_DOCUMENT_17',
       'FLAG_DOCUMENT_18', 'FLAG_DOCUMENT_19', 'FLAG_DOCUMENT_20',
       'FLAG_DOCUMENT_21', 'AMT_REQ_CREDIT_BUREAU_HOUR',
       'AMT_REQ_CREDIT_BUREAU_DAY', 'AMT_REQ_CREDIT_BUREAU_WEEK',
       'AMT_REQ_CREDIT_BUREAU_MON', 'AMT_REQ_CREDIT_BUREAU_QRT',
       'AMT_REQ_CREDIT_BUREAU_YEAR'],
```

```
new_data.shape
```

```
(49999, 81)
```

Task 1 : Handling Missing Data - Imputation

Imputation Technique for columns having ~13% missing values

```
Focus Columns would be the following :
1. AMT_REQ_CREDIT_BUREAU_HOUR                   13.47
2. AMT_REQ_CREDIT_BUREAU_MON                    13.47
3. AMT_REQ_CREDIT_BUREAU_WEEK                   13.47
4. AMT_REQ_CREDIT_BUREAU_DAY                    13.47
5. AMT_REQ_CREDIT_BUREAU_YEAR                   13.47
6. AMT_REQ_CREDIT_BUREAU_QRT                    13.47
```

```
focus_columns = ['AMT_REQ_CREDIT_BUREAU_MON','AMT_REQ_CREDIT_BUREAU_WEEK','AMT_REQ_CREDIT_BUREAU_DAY','AMT_REQ_CREDIT_B
new_data[focus_columns].dtypes
```

```
AMT_REQ_CREDIT_BUREAU_MON     float64
AMT_REQ_CREDIT_BUREAU_WEEK    float64
AMT_REQ_CREDIT_BUREAU_DAY     float64
AMT_REQ_CREDIT_BUREAU_HOUR    float64
AMT_REQ_CREDIT_BUREAU_QRT     float64
AMT_REQ_CREDIT_BUREAU_YEAR    float64
dtype: object
```

Task 1 : Handling Missing Data - Imputation

Imputation on AMT_REQ_CREDIT_BUREAU_MON ¶

```
new_data['AMT_REQ_CREDIT_BUREAU_MON'].value_counts()
```

```
0.0     36125
1.0      5370
2.0       897
3.0       341
4.0       180
5.0        97
7.0        53
6.0        49
8.0        37
9.0        32
11.0       24
10.0       19
12.0       11
16.0       10
13.0        6
14.0        4
15.0        4
17.0        3
19.0        2
24.0        1
Name: AMT_REQ_CREDIT_BUREAU_MON, dtype: int64
```

Number of Missing Data

```
new_data['AMT_REQ_CREDIT_BUREAU_MON'].isnull().sum()
```

```
6734
```

```
new_data['AMT_REQ_CREDIT_BUREAU_MON'].value_counts(normalize=True)*100
```

```
0.0     85.719714
1.0     10.740215
2.0      1.794036
3.0      0.682014
4.0      0.360007
5.0      0.194004
7.0      0.106002
6.0      0.098002
8.0      0.074001
9.0      0.064001
11.0     0.048001
10.0     0.038001
12.0     0.022000
16.0     0.020000
13.0     0.012000
14.0     0.008000
15.0     0.008000
17.0     0.006000
19.0     0.004000
24.0     0.002000
Name: AMT_REQ_CREDIT_BUREAU_MON, dtype: float64
```

For the column **'AMT_REQ_CREDIT_BUREAU_MON'** either we can exclude the missing values or impute the column with the value as zero (0) which is present in more than 85% of the rows.
Hence, we can replace the **missing values by zero**.

Task 1 : Handling Missing Data - Imputation

Imputation on AMT_REQ_CREDIT_BUREAU_WEEK

```
new_data['AMT_REQ_CREDIT_BUREAU_WEEK'].value_counts()
```

```
0.0    41951
1.0     1259
2.0       36
3.0       10
4.0        6
5.0        2
6.0        1
Name: AMT_REQ_CREDIT_BUREAU_WEEK, dtype: int64
```

```
new_data['AMT_REQ_CREDIT_BUREAU_WEEK'].isnull().sum()
```

```
6734
```

```
new_data['AMT_REQ_CREDIT_BUREAU_WEEK'].value_counts(normalize=True)*100
```

```
0.0    96.962903
1.0     2.909973
2.0     0.083208
3.0     0.023113
4.0     0.013868
5.0     0.004623
6.0     0.002311
Name: AMT_REQ_CREDIT_BUREAU_WEEK, dtype: float64
```

For the column **'AMT_REQ_CREDIT_BUREAU_WEEK"** either we can exclude the missing values or impute the column with the value as zero (0) which is present in more than 96% of the rows.
Hence, we can replace the **missing values by zero**.

Task 1 : Handling Missing Data - Imputation

Imputation on AMT_REQ_CREDIT_BUREAU_DAY

```
new_data['AMT_REQ_CREDIT_BUREAU_DAY'].value_counts()
```

```
0.0    42993
1.0      242
2.0       17
3.0        7
4.0        3
5.0        2
6.0        1
Name: AMT_REQ_CREDIT_BUREAU_DAY, dtype: int64
```

```
new_data['AMT_REQ_CREDIT_BUREAU_DAY'].isnull().sum()
```

```
6734
```

```
new_data['AMT_REQ_CREDIT_BUREAU_DAY'].value_counts(normalize=True)*100
```

```
0.0    99.371316
1.0     0.559344
2.0     0.039293
3.0     0.016179
4.0     0.006934
5.0     0.004623
6.0     0.002311
Name: AMT_REQ_CREDIT_BUREAU_DAY, dtype: float64
```

For the column **'AMT_REQ_CREDIT_BUREAU_DAY"** either we can exclude the missing values or impute the column with the value as zero (0) which is present in more than 99% of the rows.
Hence, we can replace the **missing values by zero**.

Task 1 : Handling Missing Data - Imputation

Imputation on AMT_REQ_CREDIT_BUREAU_HOUR

```
new_data['AMT_REQ_CREDIT_BUREAU_HOUR'].value_counts()
```

```
0.0    42970
1.0      285
2.0        8
3.0        2
Name: AMT_REQ_CREDIT_BUREAU_HOUR, dtype: int64
```

```
new_data['AMT_REQ_CREDIT_BUREAU_HOUR'].isnull().sum()
```

```
6734
```

```
new_data['AMT_REQ_CREDIT_BUREAU_HOUR'].value_counts(normalize=True)*100
```

```
0.0    99.318156
1.0     0.658731
2.0     0.018491
3.0     0.004623
Name: AMT_REQ_CREDIT_BUREAU_HOUR, dtype: float64
```

For the column **'AMT_REQ_CREDIT_BUREAU_HOUR"** either we can exclude the missing values or impute the column with the value as zero (0) which is present in more than 99% of the rows.
Hence, we can replace the **missing values by zero**.

Task 1 : Handling Missing Data - Imputation

Imputation on AMT_REQ_CREDIT_BUREAU_QRT

```
new_data['AMT_REQ_CREDIT_BUREAU_QRT'].value_counts()
```

```
0.0    35131
1.0     5452
2.0     2324
3.0      269
4.0       74
5.0        8
6.0        3
8.0        2
7.0        2
Name: AMT_REQ_CREDIT_BUREAU_QRT, dtype: int64
```

```
new_data['AMT_REQ_CREDIT_BUREAU_QRT'].isnull().sum()
```

```
6734
```

```
new_data['AMT_REQ_CREDIT_BUREAU_QRT'].value_counts(normalize=True)*100
```

```
0.0    81.199584
1.0    12.601410
2.0     5.371547
3.0     0.621750
4.0     0.171039
5.0     0.018491
6.0     0.006934
8.0     0.004623
7.0     0.004623
Name: AMT_REQ_CREDIT_BUREAU_QRT, dtype: float64
```

For the column **'AMT_REQ_CREDIT_BUREAU_QRT"** either we can exclude the missing values or impute the column with the value as zero (0) which is present in more than 80% of the rows.
Hence, we can replace the **missing values by zero**.

Task 1 : Handling Missing Data - Imputation

Imputation on AMT_REQ_CREDIT_BUREAU_YEAR

```
new_data['AMT_REQ_CREDIT_BUREAU_YEAR'].value_counts()
```

```
0.0     11754
1.0     10534
2.0      8059
3.0      5394
4.0      3374
5.0      1867
6.0      1114
7.0       617
8.0       354
9.0       173
11.0        8
10.0        4
12.0        4
13.0        2
16.0        2
15.0        2
25.0        1
23.0        1
14.0        1
Name: AMT_REQ_CREDIT_BUREAU_YEAR, dtype: int64
```

```
new_data['AMT_REQ_CREDIT_BUREAU_YEAR'].isnull().sum()
```

```
6734
```

```
new_data['AMT_REQ_CREDIT_BUREAU_YEAR'].value_counts(normalize=True)*100
```

```
0.0     27.167456
1.0     24.347625
2.0     18.627066
3.0     12.467352
4.0      7.798451
5.0      4.315266
6.0      2.574830
7.0      1.426095
8.0      0.818213
9.0      0.399861
11.0     0.018491
10.0     0.009245
12.0     0.009245
13.0     0.004623
16.0     0.004623
15.0     0.004623
25.0     0.002311
23.0     0.002311
14.0     0.002311
Name: AMT_REQ_CREDIT_BUREAU_YEAR, dtype: float64
```

For the column **'AMT_REQ_CREDIT_BUREAU_YEAR"** the proportion of values 0,1,2, and 3 are divided equally. Hence, we cannot impute the value based on this normalised percentage. For this column, it is better that, we can **exclude the rows with missing data.**

Task 1 : Handling Missing Data - Imputation (Cross-Checking)

```
#Checking for missing values
round(100.0 *
new_data.isnull().sum()/len(new_dat
a),2).sort_values()
```

SK_ID_CURR	0.00
AMT_REQ_CREDIT_BUREAU_YEAR	0.00
AMT_REQ_CREDIT_BUREAU_QRT	0.00
AMT_REQ_CREDIT_BUREAU_MON	0.00
AMT_REQ_CREDIT_BUREAU_WEEK	0.00
AMT_REQ_CREDIT_BUREAU_DAY	0.00
AMT_REQ_CREDIT_BUREAU_HOUR	0.00
DAYS_LAST_PHONE_CHANGE	0.00
ORGANIZATION_TYPE	0.00
REGION_RATING_CLIENT_W_CITY	0.00
REGION_RATING_CLIENT	0.00
CNT_FAM_MEMBERS	0.00
DAYS_ID_PUBLISH	0.00
AGE_GROUP	0.00
DAYS_EMPLOYED	0.00
DAYS_BIRTH	0.00
DAYS_REGISTRATION	0.00
NAME_FAMILY_STATUS	0.00
NAME_EDUCATION_TYPE	0.00
NAME_INCOME_TYPE	0.00
TARGET	0.00

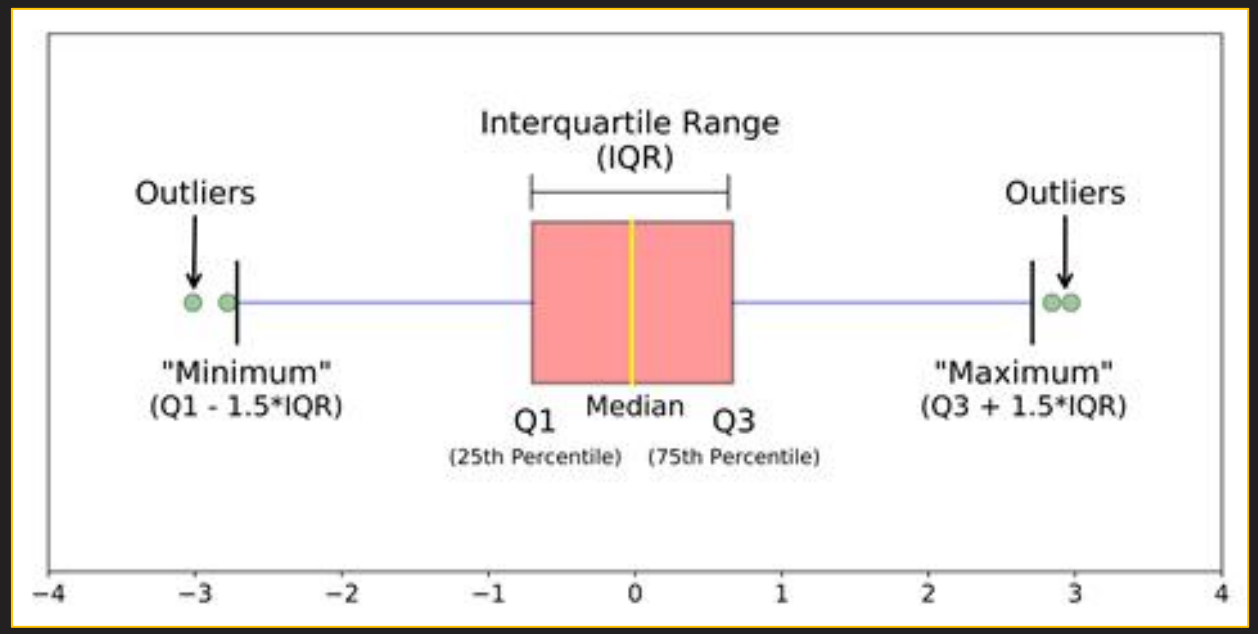
NAME_CONTRACT_TYPE	0.00
AMT_ANNUITY	0.00
AMT_CREDIT	0.00
CODE_GENDER	0.00
FLAG_OWN_CAR	0.00
NAME_HOUSING_TYPE	0.00
FLAG_OWN_REALTY	0.00
AMT_INCOME_TOTAL	0.00
CNT_CHILDREN	0.00
AMT_GOODS_PRICE	0.08
OBS_60_CNT_SOCIAL_CIRCLE	0.34
DEF_30_CNT_SOCIAL_CIRCLE	0.34
OBS_30_CNT_SOCIAL_CIRCLE	0.34
DEF_60_CNT_SOCIAL_CIRCLE	0.34
NAME_TYPE_SUITE	0.38
AMT_CATEGORY	0.57
OCCUPATION_TYPE	31.31
dtype: float64	

Missing Values for the columns in Cleaned Data (The above mentioned imputed attributes are now having the null values as zero)

Project 6

Bank Loan - Case Study

Task 2 : Handling Outliers



Task 2 : Handling Outliers

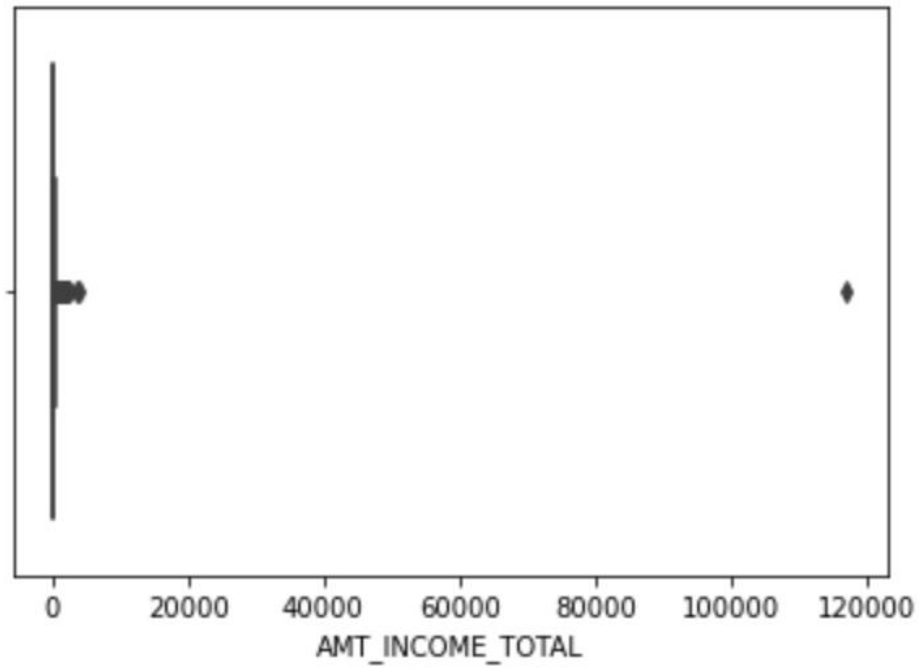
- **Objective :** To detect and identify outliers in the dataset , focusing on numerical variables.
- **Approach :** To identify potential outliers of numerical columns and analyse it. For the outlier analysis, we will focus on the below numerical columns of the dataset.
    - AMT_INCOME_TOTAL
    - AMT_CREDIT
    - AMT_ANNUITY
    - AMT_GOODS_PRICE
    - FLOORSMAX_AVG
- **Data Visualisation :** Using box-plot visualisation, the outliers are highlighted and the distribution of the outlier data is observed on the above columns.

Task 2 : Handling Outliers

Outlier Analysis for AMT_INCOME_TOTAL

```
sns.boxplot(new_data['AMT_INCOME_TOTAL']/1000.0)
```

```
<AxesSubplot:xlabel='AMT_INCOME_TOTAL'>
```




```
(new_data['AMT_INCOME_TOTAL']/1000.0).describe()
```

```
count     49999.000000
mean        170.767591
std         531.819095
min          25.650000
25%         112.500000
50%         145.800000
75%         202.500000
max      117000.000000
Name: AMT_INCOME_TOTAL, dtype: float64
```

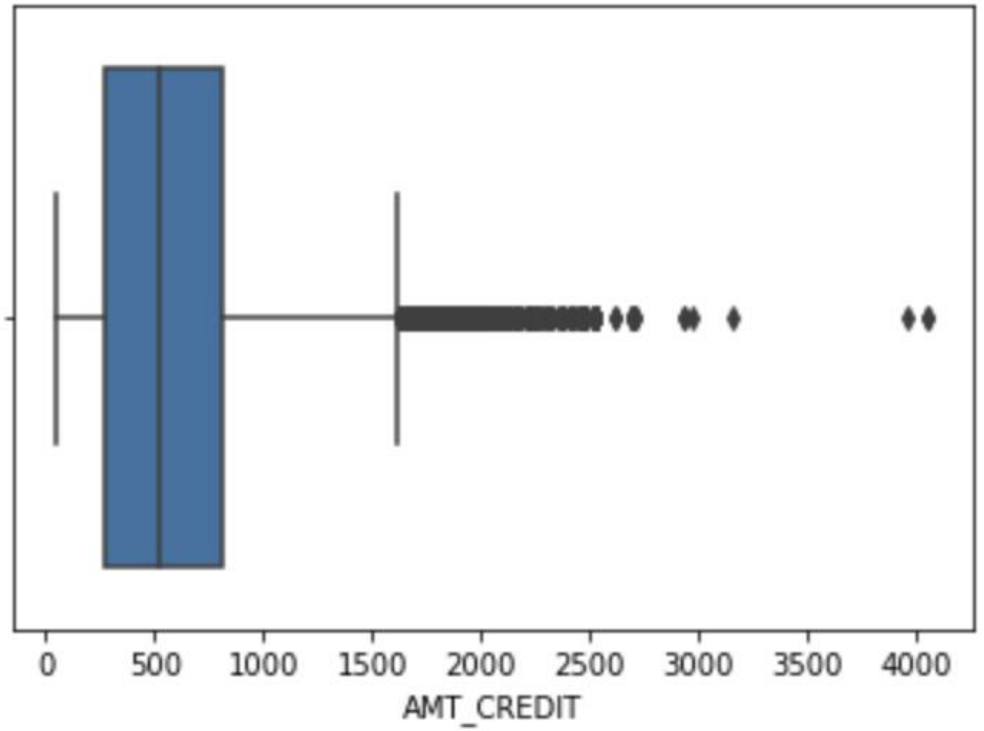
```
------ AMT_INCOME_TOTAL ------
IQR :  90.0
Upper Limit :  337.5
Lower Limit :  -22.5
Outlier in Percentage :  4.59
-----------------------------
```

Task 2 : Handling Outliers

Outlier Analysis for AMT_CREDIT

```
sns.boxplot(new_data['AMT_CREDIT']/1000.0)
```

```
<AxesSubplot:xlabel='AMT_CREDIT'>
```




```
(new_data['AMT_CREDIT']/1000.0).describe()
```

```
count    49999.000000
mean       599.700582
std        402.415434
min         45.000000
25%        270.000000
50%        514.777500
75%        808.650000
max       4050.000000
Name: AMT_CREDIT, dtype: float64
```

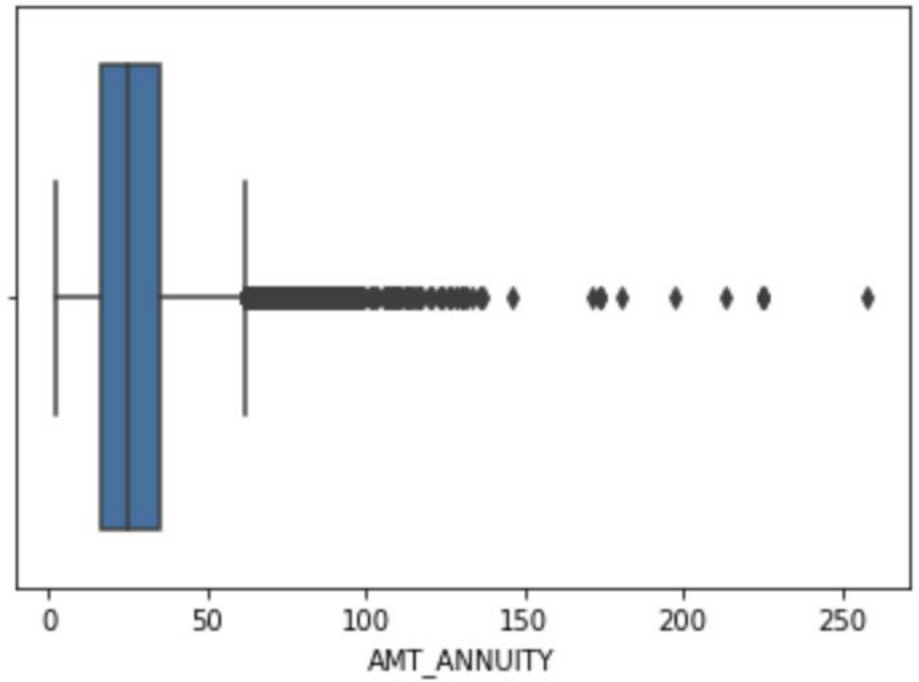
```
------ AMT_CREDIT ------
IQR :  538.65
Upper Limit :  1616.625
Lower Limit :  -537.9749999999999
Outlier in Percentage :  2.13
------------------------------
```

Task 2 : Handling Outliers

Outlier Analysis for AMT_ANNUITY ¶

```
sns.boxplot(new_data['AMT_ANNUITY']/1000.0)
```

```
<AxesSubplot:xlabel='AMT_ANNUITY'>
```




```
(new_data['AMT_ANNUITY']/1000.0).describe()
```

```
count    49998.000000
mean        27.107377
std         14.562944
min          2.052000
25%         16.456500
50%         24.939000
75%         34.596000
max        258.025500
Name: AMT_ANNUITY, dtype: float64
```

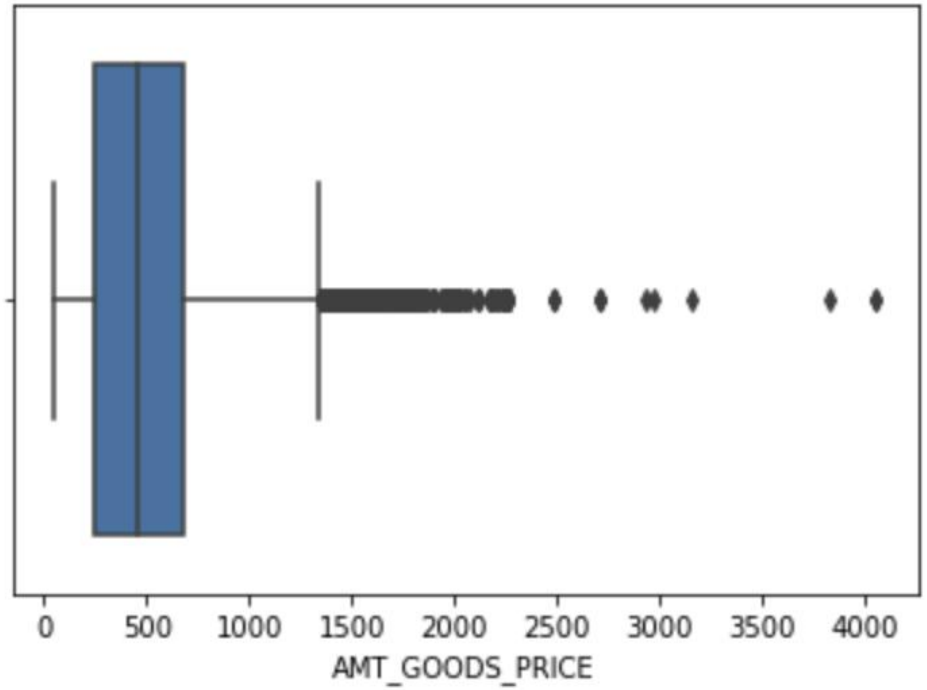
```
------ AMT_ANNUITY ------
IQR :  18.139499999999998
Upper Limit :  61.805249999999994
Lower Limit :  -10.752749999999999
Outlier in Percentage :  2.38
-------------------------------
```

Task 2 : Handling Outliers

Outlier Analysis for AMT_GOODS_PRICE

```
sns.boxplot(new_data['AMT_GOODS_PRICE']/1000.0)
```

```
<AxesSubplot:xlabel='AMT_GOODS_PRICE'>
```




```
(new_data['AMT_GOODS_PRICE']/1000.0).describe()
```

```
count    49961.000000
mean       539.060036
std        369.853253
min         45.000000
25%        238.500000
50%        450.000000
75%        679.500000
max       4050.000000
Name: AMT_GOODS_PRICE, dtype: float64
```

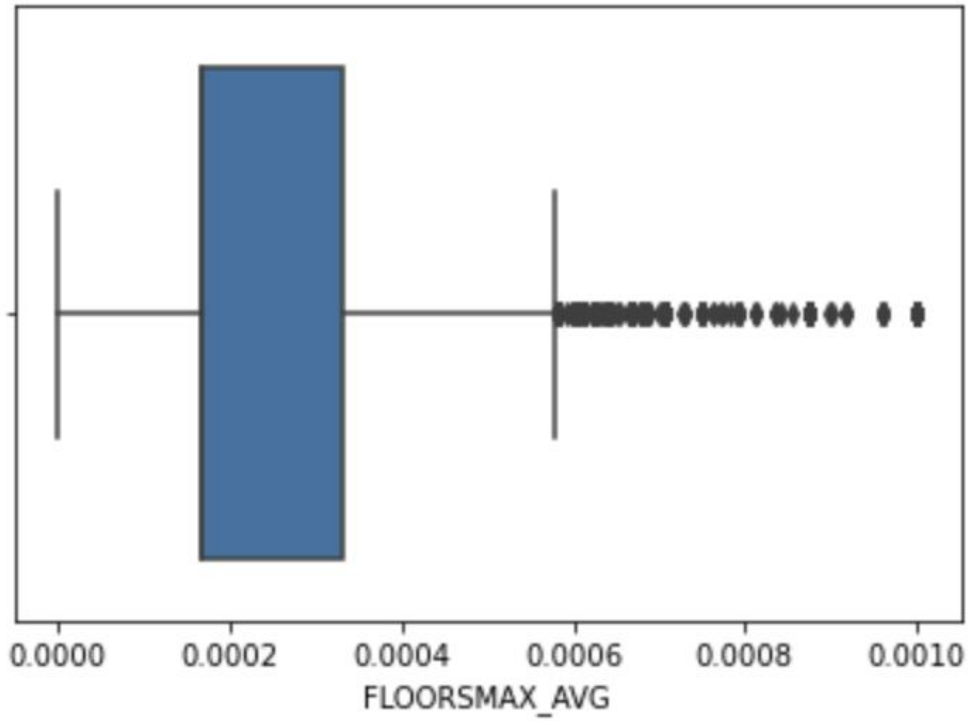
```
------ AMT_GOODS_PRICE ------
IQR :  441.0
Upper Limit :  1341.0
Lower Limit :  -423.0
Outlier in Percentage :  4.77
-----------------------------
```

Task 2 : Handling Outliers

Outlier Analysis for FLOORSMAX_AVG

```
sns.boxplot(new_data['FLOORSMAX_AVG']/1000.0)
```

```
<AxesSubplot:xlabel='FLOORSMAX_AVG'>
```




```
(new_data['FLOORSMAX_AVG']/1000.0).describe()
```

```
count    25124.000000
mean         0.000225
std          0.000145
min          0.000000
25%          0.000167
50%          0.000167
75%          0.000333
max          0.001000
Name: FLOORSMAX_AVG, dtype: float64
```

```
------ FLOORSMAX_AVG ------
IQR :  0.0001665999999999998
Upper Limit :  0.0005831999999999999
Lower Limit :  -8.319999999999996e-05
Outlier in Percentage :  1.72
---------------------------
```

Task 2 : Handling Outliers - Bins for Continuous Data

Bins for Continuous Variables

For bins, we will use the following columns :
* AGE_GROUP
* AMT_CATEGORY

```
# Fixing the column with negative data and deriving the number of years for DAYS_BIRTH
new_data['DAYS_BIRTH'] = data['DAYS_BIRTH'].div(365).round(2).abs()
```

```
new_data['AGE_GROUP'] = pd.cut(x=new_data.DAYS_BIRTH,
bins=[0,19,29,39,49,59,69,79,89],
labels=['10s','20s','30s','40s','50s','60s','70s','80s'])
```

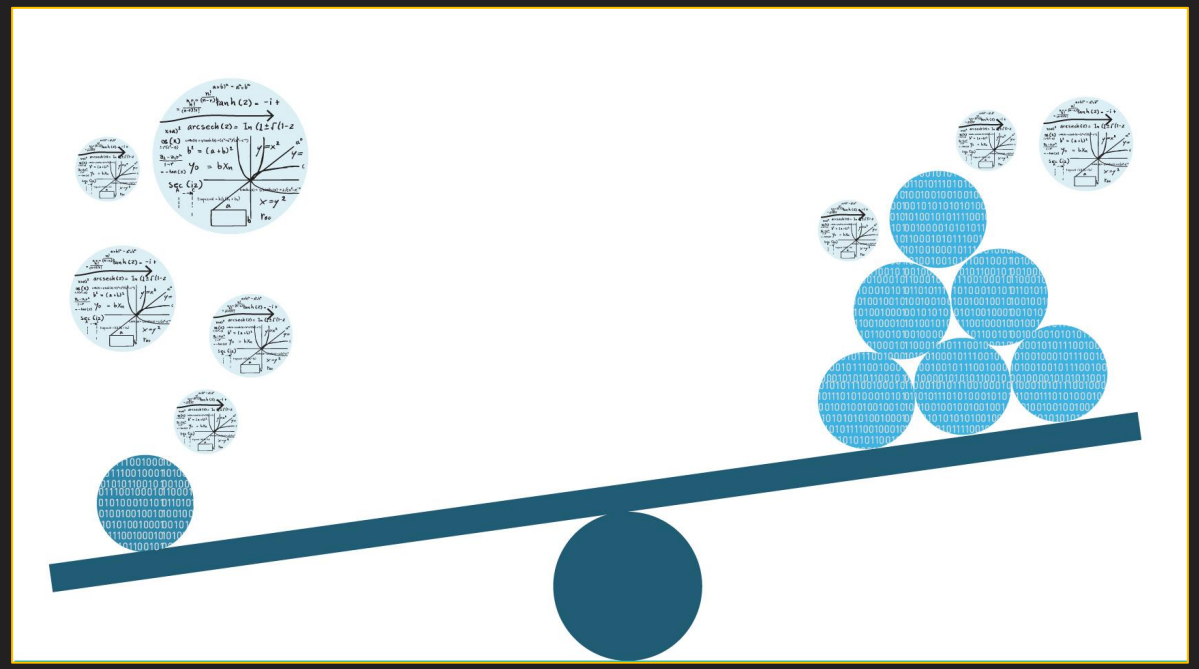
```
new_data['AMT_CATEGORY'] =
pd.cut(x=new_data.AMT_INCOME_TOTAL,
bins=[0,100000,200000,300000,400000,500000,600000],
labels=['Low','Average','Good','Best','High','Very High'])
```

AGE_GROUP	AMT_CATEGORY
20s	Good
40s	Good
50s	Low
50s	Average
50s	Average

Project 6

Bank Loan - Case Study

Task 3 : Analysing Data Imbalance



Task 3 : Analysing Data Imbalance

- **Objective :** For binary classification, data imbalance can affect the accuracy of the analysis. Hence, to understand the data distribution is crucial for building reliable models.
- **Approach :**
  - To identify if there is data imbalance in the dataset and analyse the ratio of imbalance.
  - Dividing the dataset into two different data segments one for Target 0 and Target 1 for further analysis
- **Data Visualisation :** Using box-plot visualisation, the data imbalance is identified and checked.

Task 3 : Analysing Data Imbalance

Removing unnecessary data for analysis

```
drop_columns = ['FLAG_MOBIL','FLAG_EMP_PHONE', 'FLAG_WORK_PHONE', 'FLAG_CONT_MOBILE', 'FLAG_PHONE','FLAG_EMAIL',
                'WEEKDAY_APPR_PROCESS_START', 'HOUR_APPR_PROCESS_START', 'EXT_SOURCE_2', 'EXT_SOURCE_3',
               'FLOORSMAX_AVG', 'FLOORSMAX_MODE', 'FLOORSMAX_MEDI','TOTALAREA_MODE', 'EMERGENCYSTATE_MODE',
               'REGION_POPULATION_RELATIVE', 'YEARS_BEGINEXPLUATATION_AVG', 'YEARS_BEGINEXPLUATATION_MODE',
                'YEARS_BEGINEXPLUATATION_MEDI', 'REG_REGION_NOT_LIVE_REGION', 'REG_REGION_NOT_WORK_REGION',
               'LIVE_REGION_NOT_WORK_REGION', 'REG_CITY_NOT_LIVE_CITY','REG_CITY_NOT_WORK_CITY','LIVE_CITY_NOT_WORK_CIT
               'FLAG_DOCUMENT_2','FLAG_DOCUMENT_3','FLAG_DOCUMENT_4','FLAG_DOCUMENT_5','FLAG_DOCUMENT_6',
               'FLAG_DOCUMENT_7','FLAG_DOCUMENT_8','FLAG_DOCUMENT_9','FLAG_DOCUMENT_10','FLAG_DOCUMENT_11',
               'FLAG_DOCUMENT_12','FLAG_DOCUMENT_13','FLAG_DOCUMENT_14','FLAG_DOCUMENT_15','FLAG_DOCUMENT_16',
               'FLAG_DOCUMENT_17','FLAG_DOCUMENT_18','FLAG_DOCUMENT_19','FLAG_DOCUMENT_20','FLAG_DOCUMENT_21']
```

```
new_cleaned_data = new_data.drop(columns=drop_columns, axis=1)
print(new_cleaned_data.shape)
```

```
(49999, 38)
```

Before proceeding with the analysis, the unnecessary columns of the dataset are removed.

Task 3 : Analysing Data Imbalance

```
#Checking for missing values
round(100.0 *
new_cleaned_data.isnull().sum()/len
(new_cleaned_data),2).sort_values()
```

SK_ID_CURR	0.00
AMT_REQ_CREDIT_BUREAU_YEAR	0.00
AMT_REQ_CREDIT_BUREAU_QRT	0.00
AMT_REQ_CREDIT_BUREAU_MON	0.00
AMT_REQ_CREDIT_BUREAU_WEEK	0.00
AMT_REQ_CREDIT_BUREAU_DAY	0.00
AMT_REQ_CREDIT_BUREAU_HOUR	0.00
DAYS_LAST_PHONE_CHANGE	0.00
ORGANIZATION_TYPE	0.00
REGION_RATING_CLIENT_W_CITY	0.00
REGION_RATING_CLIENT	0.00
CNT_FAM_MEMBERS	0.00
DAYS_ID_PUBLISH	0.00
AGE_GROUP	0.00
DAYS_EMPLOYED	0.00
DAYS_BIRTH	0.00
DAYS_REGISTRATION	0.00
NAME_FAMILY_STATUS	0.00
NAME_EDUCATION_TYPE	0.00
NAME_INCOME_TYPE	0.00
TARGET	0.00
NAME_CONTRACT_TYPE	0.00
AMT_ANNUITY	0.00
AMT_CREDIT	0.00
CODE_GENDER	0.00
FLAG_OWN_CAR	0.00
NAME_HOUSING_TYPE	0.00
FLAG_OWN_REALTY	0.00
AMT_INCOME_TOTAL	0.00
CNT_CHILDREN	0.00
AMT_GOODS_PRICE	0.08
OBS_60_CNT_SOCIAL_CIRCLE	0.34
DEF_30_CNT_SOCIAL_CIRCLE	0.34
OBS_30_CNT_SOCIAL_CIRCLE	0.34
DEF_60_CNT_SOCIAL_CIRCLE	0.34
NAME_TYPE_SUITE	0.38
AMT_CATEGORY	0.57
OCCUPATION_TYPE	31.31
dtype: float64	

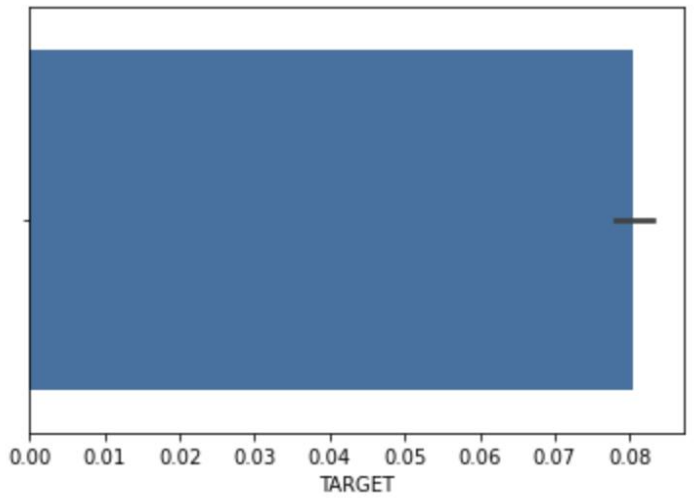
Missing Values for the columns in Cleaned Data

Task 3 : Analysing Data Imbalance

Checking Data Imbalance

```
sns.barplot(x = new_cleaned_data["TARGET"])
```

```
<AxesSubplot:xlabel='TARGET'>
```




```
#Finding % of people with outstanding dues and no outstanding dues
target_0_percentage = (round(len(new_cleaned_data.query('TARGET== 0'))/len(new_cleaned_data),4))*100
print("Target 0 Percentage : ",target_0_percentage," %")

target_1_percentage = (round(len(new_cleaned_data.query('TARGET== 1'))/len(new_cleaned_data),4))*100
print("Target 1 Percentage : ",target_1_percentage," %")
```

```
Target 0 Percentage :  91.95  %
Target 1 Percentage :  8.05  %
```

From the above analysis, we can observe that, the data distribution between **Target 0 (92%)** and **Target 1 (8%)** are **completely imbalanced.** Hence, we will be creating two different datasets for each target values for further analysis.

Task 3 : Analysing Data Imbalance - Dividing into Data Segments

```
#Creating dataset for people with no outstanding dues (target=0)
target_0_df = new_cleaned_data.query('TARGET==0')
print("Shape : ",target_0_df.shape )
target_0_df.head()
```

```
Shape :  (45973, 38)
```

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN
1	100003	0	Cash loans	F	N	N	0
2	100004	0	Revolving loans	M	Y	Y	0
3	100006	0	Cash loans	F	N	Y	0
4	100007	0	Cash loans	M	N	Y	0
5	100008	0	Cash loans	M	N	Y	0

Dataset for customers with no outstanding dues (Target = 0)

Shape : (45973,38)

```
#Creating dataset for people with some outstanding dues (target=1)
target_1_df = new_cleaned_data.query('TARGET==1')
print("Shape : ",target_1_df.shape )
target_1_df.head()
```

```
Shape :  (4026, 38)
```

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN
0	100002	1	Cash loans	M	N	Y	0
26	100031	1	Cash loans	F	N	Y	0
40	100047	1	Cash loans	M	N	Y	0
42	100049	1	Cash loans	F	N	N	0
81	100096	1	Cash loans	F	N	Y	0

Dataset for customers with some outstanding dues (Target = 1)

Shape : (4026,38)

Project 6

Bank Loan - Case Study

Task 4 : Univariate & Bivariate Analysis

Task 4 : Univariate, Segmented Univariate & Bivariate Analysis

- **Objective :** To gain insights into the driving factors of loan default which is important to conduct various analyses on consumer and loan attributes.
- **Approach :**
  - To perform univariate analysis to understand the distribution of individual variables and segmented univariate analysis to compare variable distributions for different scenarios
  - To perform bivariate analysis to explore relationships between variables and the target variable
- **Data Visualisation :** Using box-plot and bar-chart visualisations, the univariate, segmented univariate and bi-variate relationship between the attributes are observed.

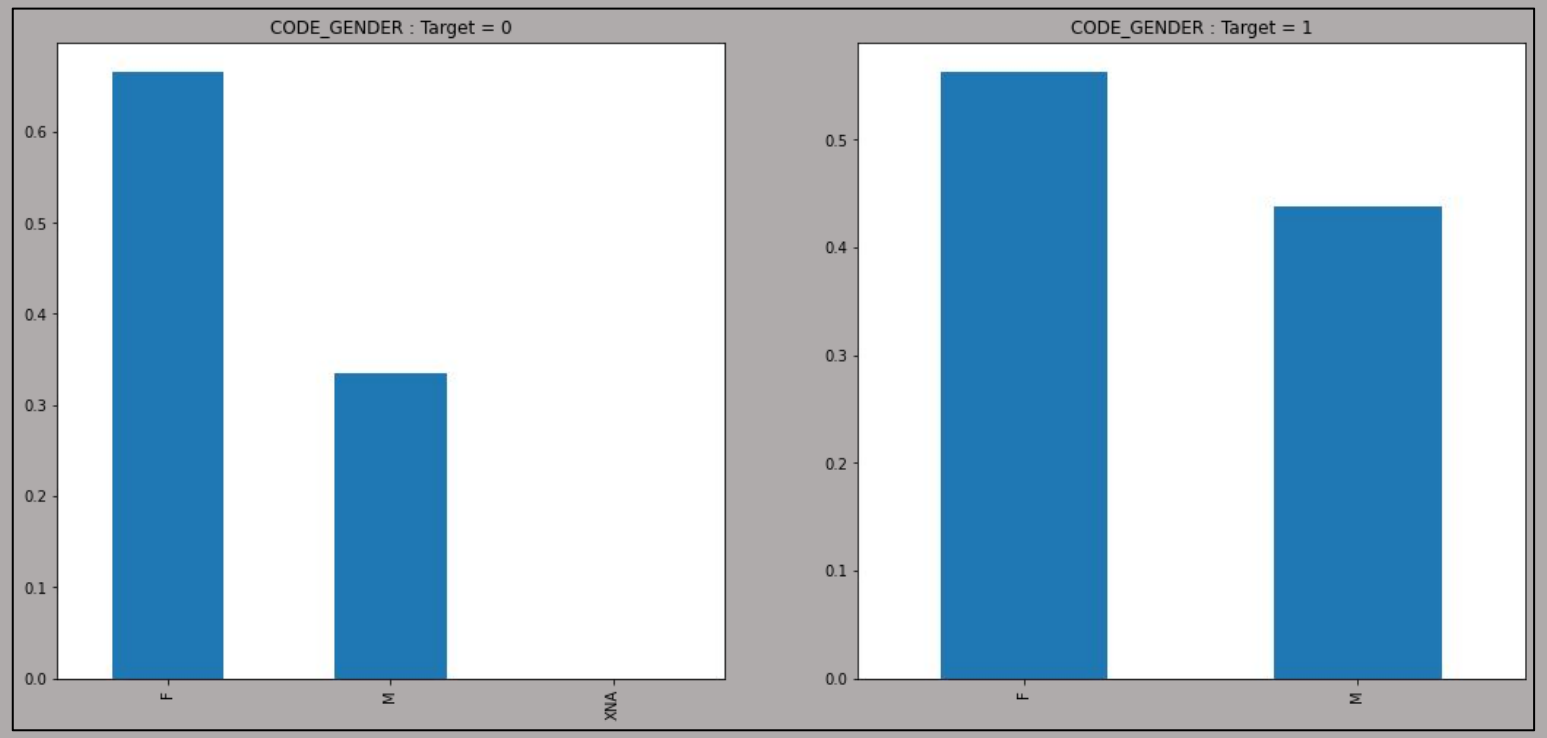
Task 4 : Univariate, Segmented Univariate, Bivariate Analysis

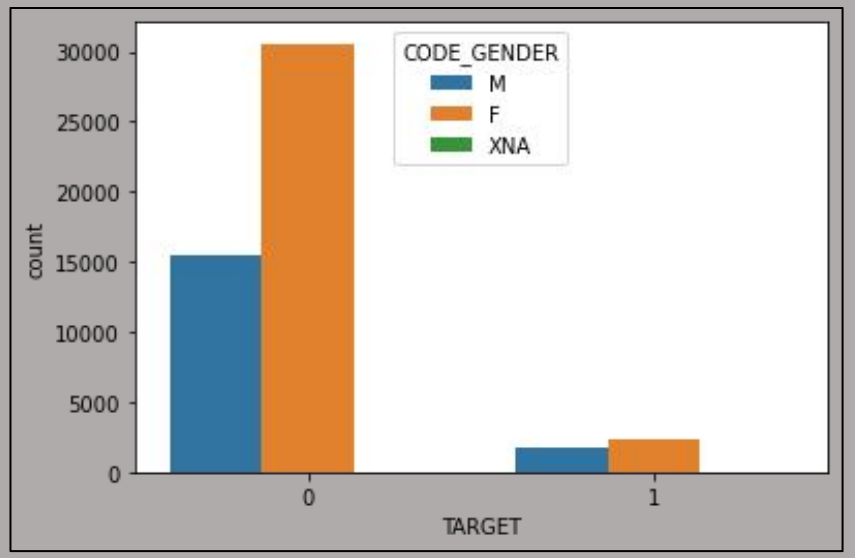
```
#list of all categorical columns
categorical_columns = ['NAME_CONTRACT_TYPE',
                       'FLAG_OWN_CAR',
                       'FLAG_OWN_REALTY',
                       'CODE_GENDER',
                       'NAME_EDUCATION_TYPE',
                       'AMT_CATEGORY',
                       'AGE_GROUP',
                       'NAME_FAMILY_STATUS',
                       'NAME_HOUSING_TYPE',
                       'NAME_TYPE_SUITE',
                       'NAME_INCOME_TYPE',
                       'OCCUPATION_TYPE',
                       'ORGANIZATION_TYPE',
                       'REGION_RATING_CLIENT_W_CITY',
                       'REGION_RATING_CLIENT',
                       'AMT_REQ_CREDIT_BUREAU_HOUR',
                       'DEF_60_CNT_SOCIAL_CIRCLE',
                       'AMT_REQ_CREDIT_BUREAU_WEEK',
                       'AMT_REQ_CREDIT_BUREAU_DAY',
                       'DEF_30_CNT_SOCIAL_CIRCLE',
                       'AMT_REQ_CREDIT_BUREAU_QRT',
                       'CNT_CHILDREN',
                       'CNT_FAM_MEMBERS',
                       'AMT_REQ_CREDIT_BUREAU_MON',
                       'AMT_REQ_CREDIT_BUREAU_YEAR',
                       'OBS_30_CNT_SOCIAL_CIRCLE',
                       'OBS_60_CNT_SOCIAL_CIRCLE',
                      ]
```

```
# list of all continuous numerical column
numerical_columns= ['AMT_GOODS_PRICE',
                    'DAYS_LAST_PHONE_CHANGE',
                    'DAYS_ID_PUBLISH',
                    'AMT_INCOME_TOTAL',
                    'DAYS_EMPLOYED',
                    'DAYS_REGISTRATION',
                    'DAYS_BIRTH',
                    'AMT_CREDIT',
                    'AMT_ANNUITY'
                   ]
```

Before we proceeded with the analysis, the attributes of the dataset are divided into **Categorical and Numerical Variables**

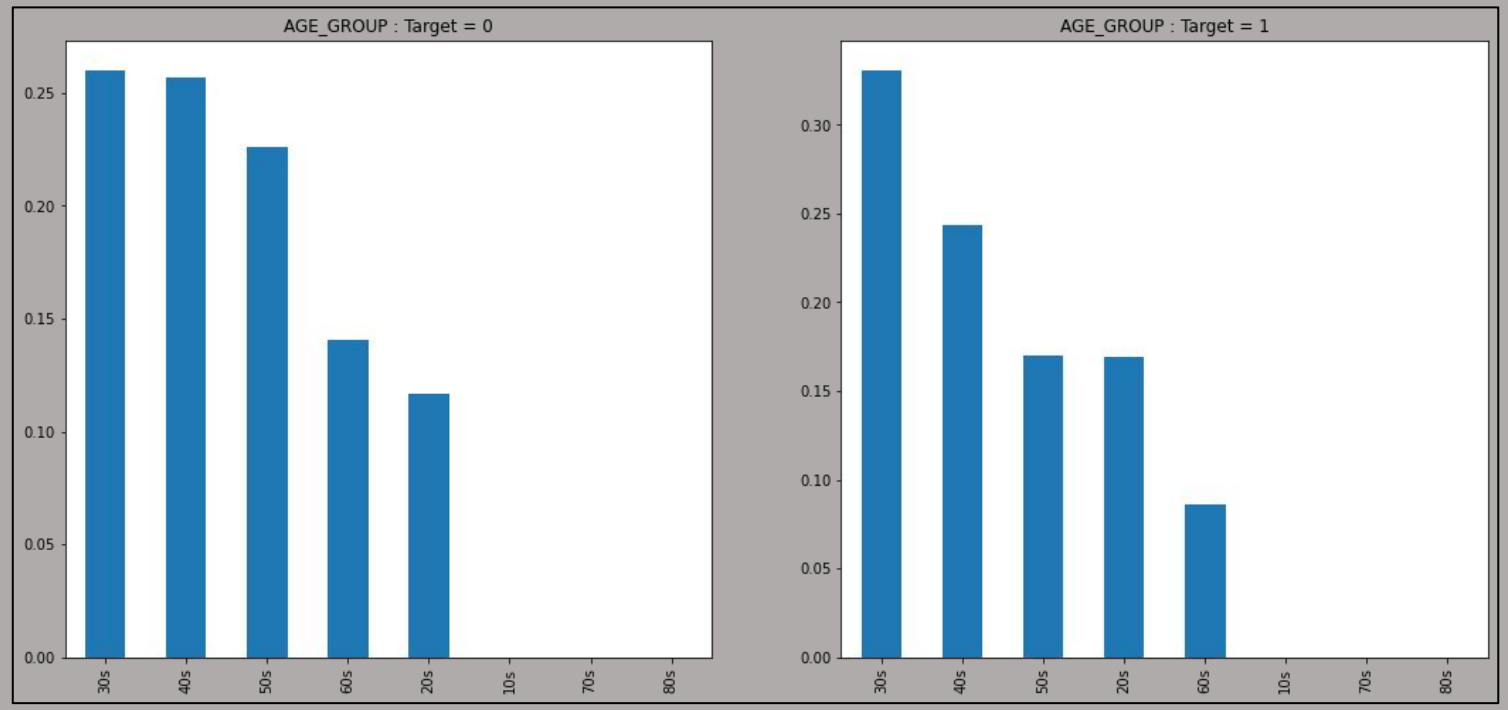
Task 4 : Univariate / Segmented Univariate Analysis of Data

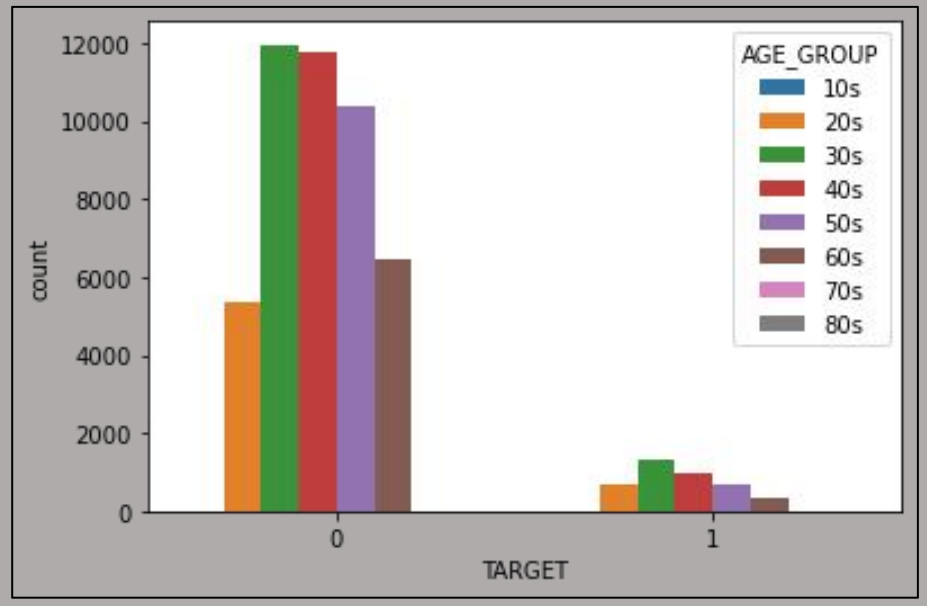




Inference : From the figures above, it is observed that, the proportion of female is more than the male customers.

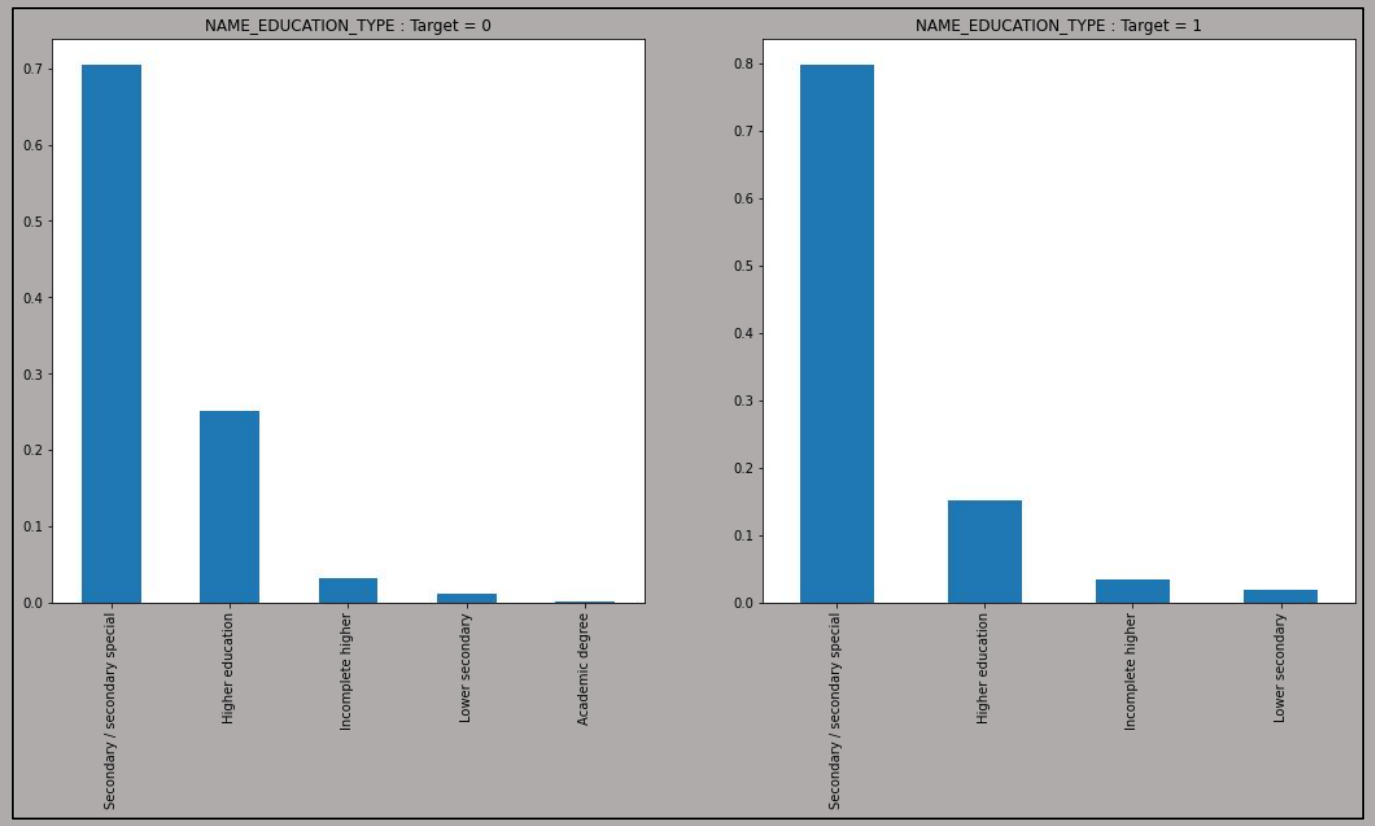
Task 4 : Univariate / Segmented Univariate Analysis of Data

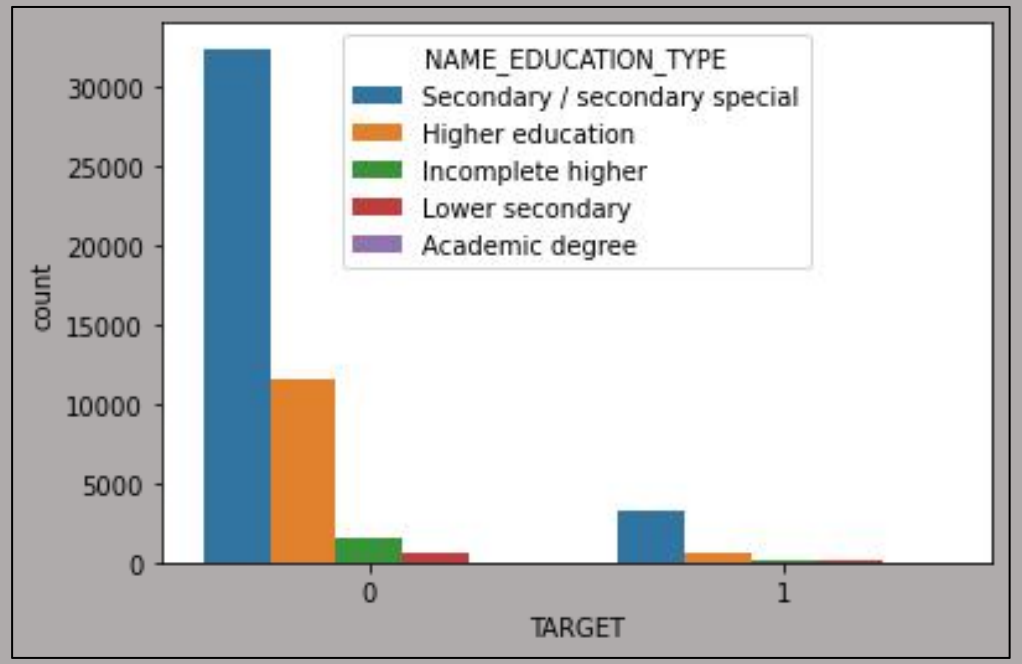




Inference : From the figures above, it is observed that, customers of age group 30's and 40's are almost equal in proportion for "no dues" category. The customers of age group 30's dominate the "with dues" category.

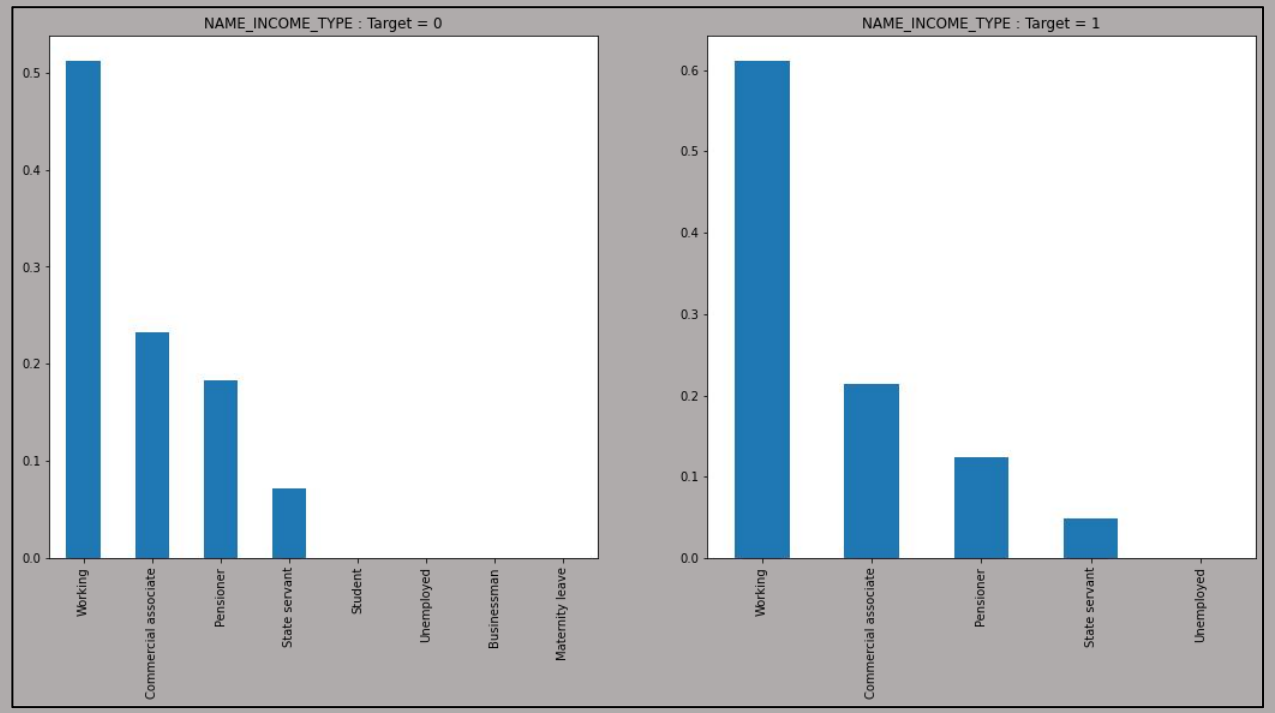
Task 4 : Univariate / Segmented Univariate Analysis of Data

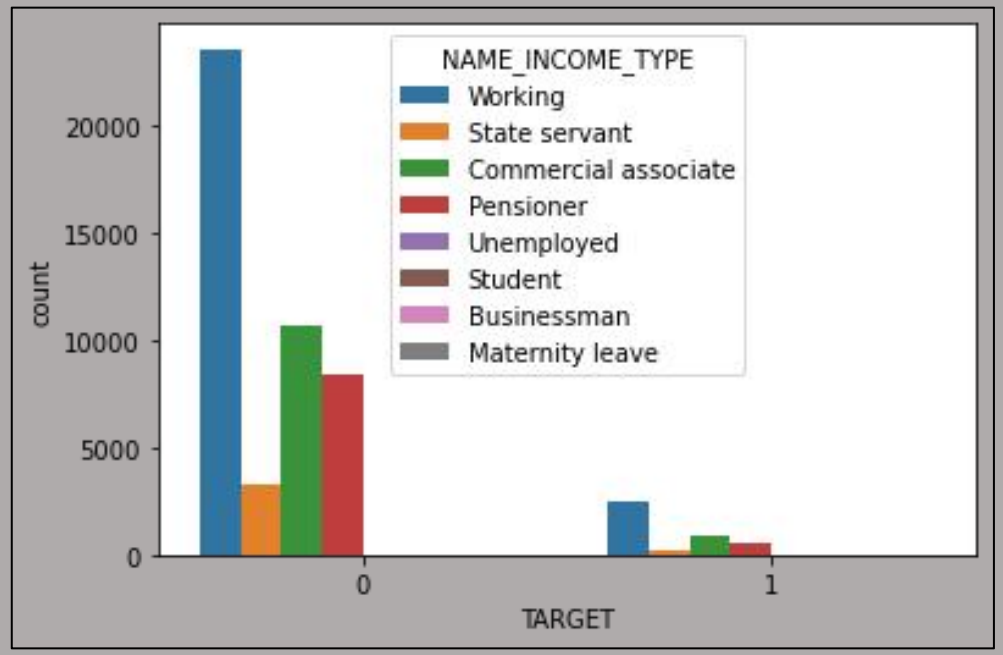




Inference : From the figures above, it is observed that, customers of education type Secondary / Secondary Special dominate on both the category of data.

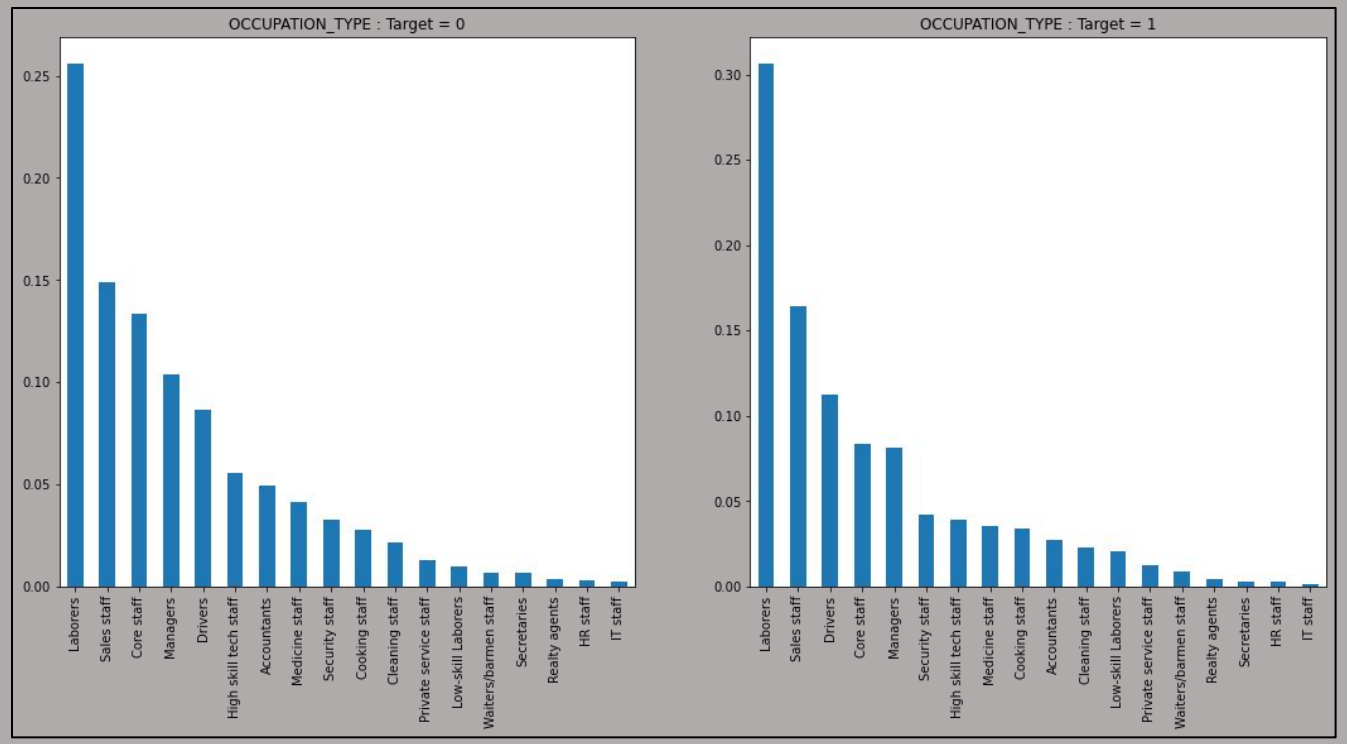
Task 4 : Univariate / Segmented Univariate Analysis of Data

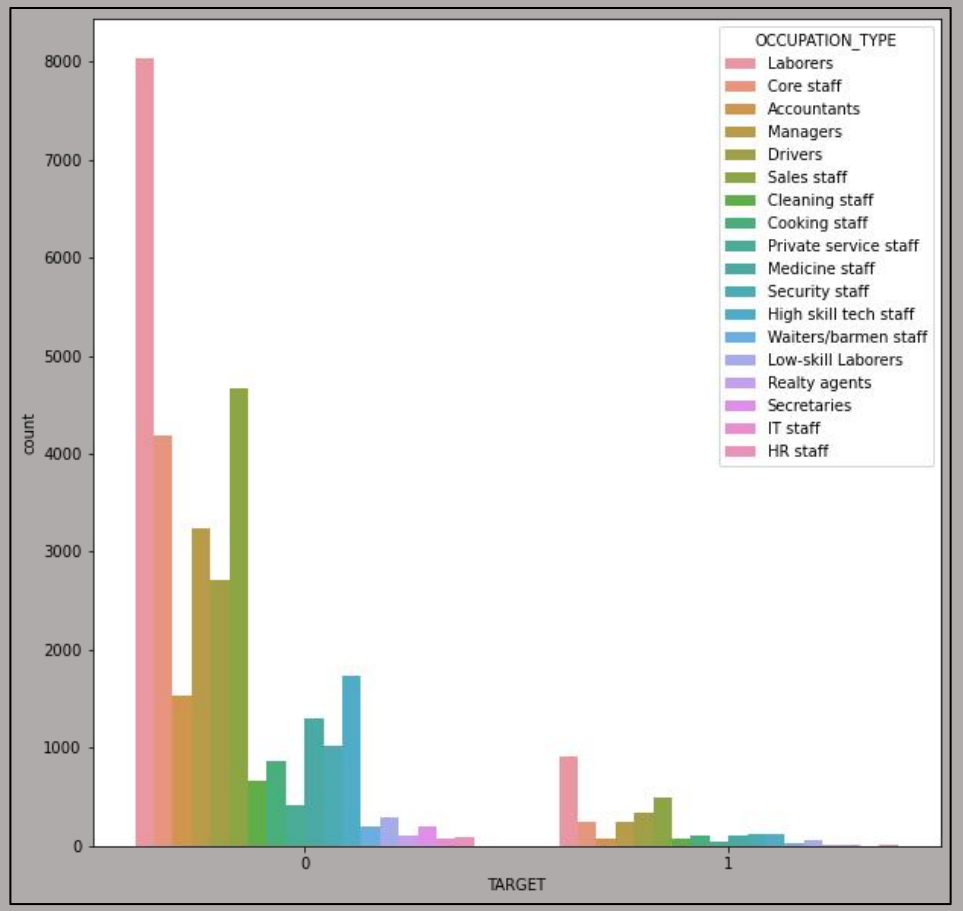




Inference : From the figures above, it is observed that, customers of working type have a lots of possibility for bank loans. The second most common is commercial agents. Comparitively, commercial agents have less number of “with dues” customers

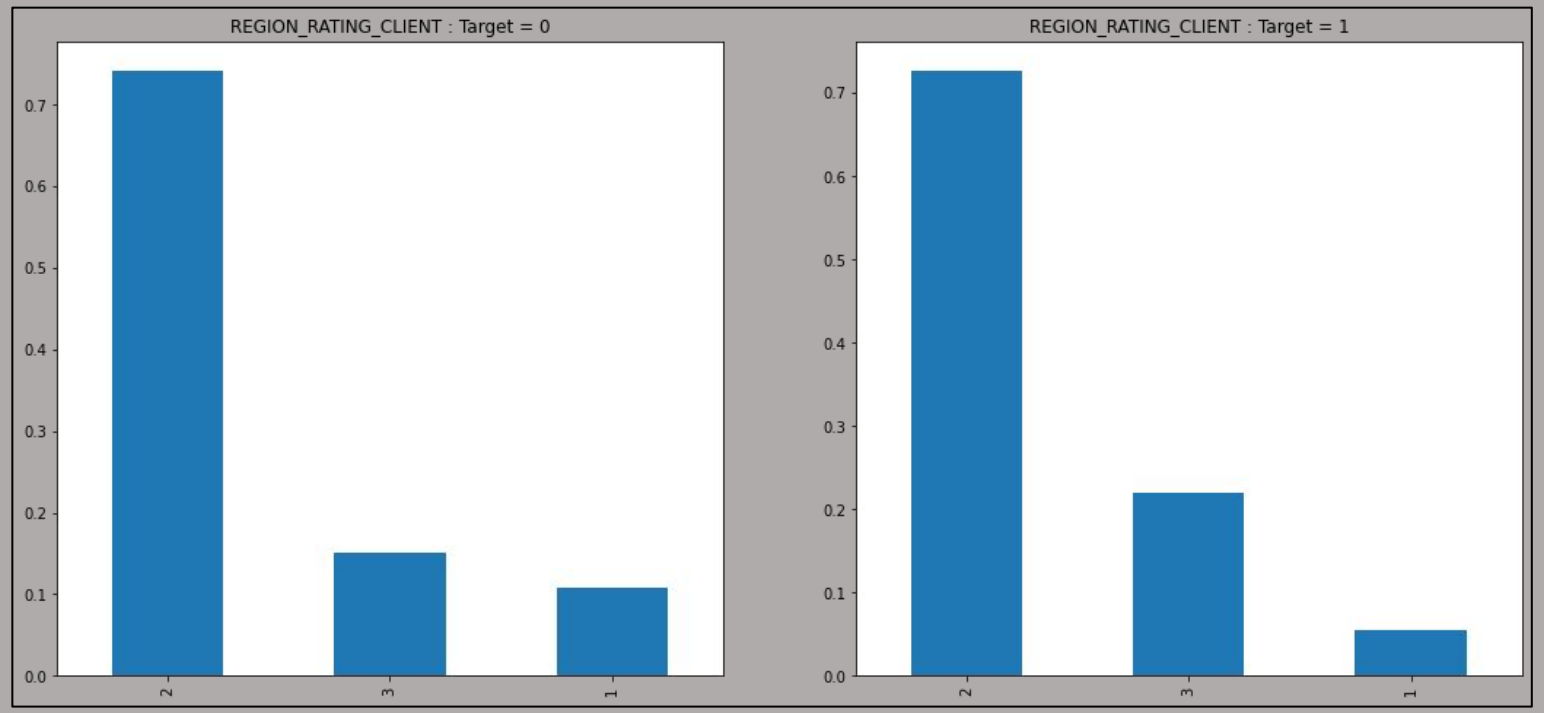
Task 4 : Univariate / Segmented Univariate Analysis of Data

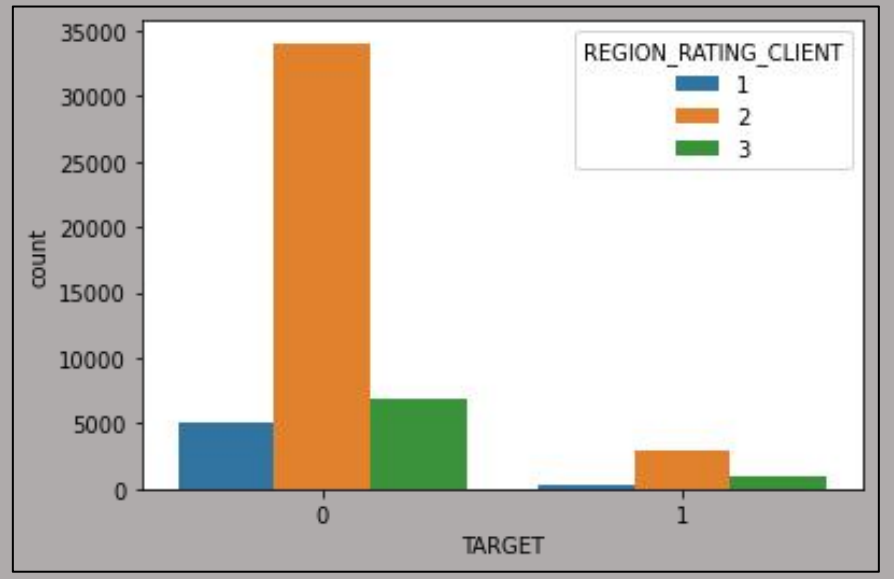




Inference : From the figures above, it is observed that, customers whose occupation is labourers are more prone for bank loans.

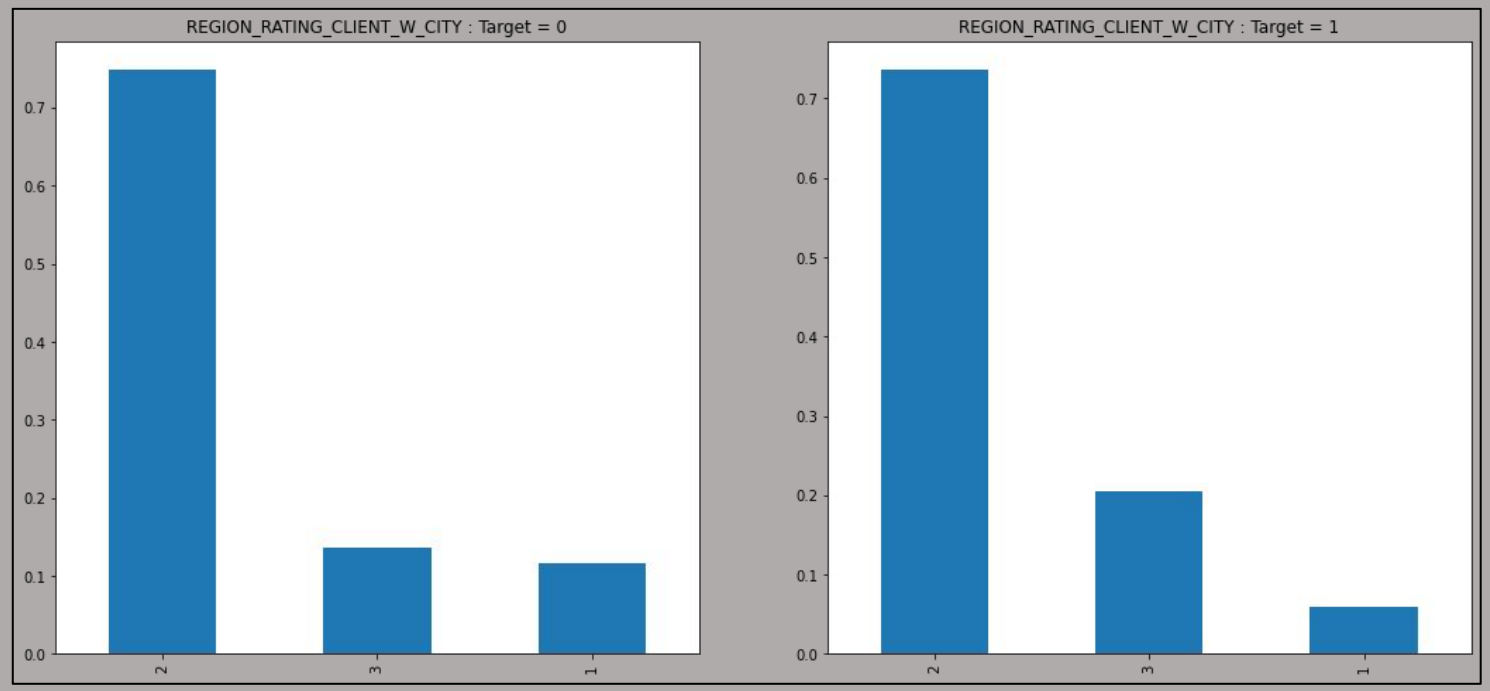
Task 4 : Univariate / Segmented Univariate Analysis of Data

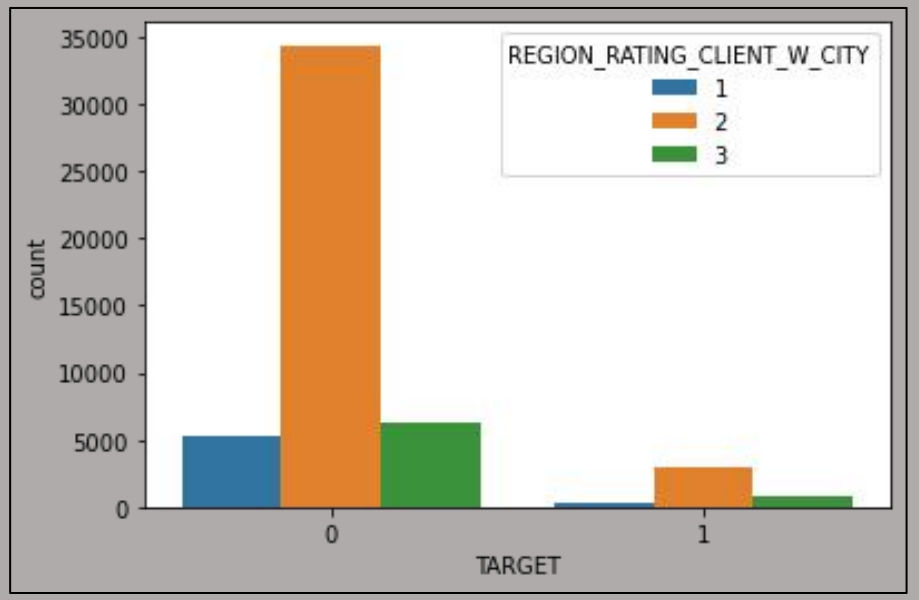




Inference : From the figures above, it is observed that, customers with region rating as 2 dominate on both the categories of data.

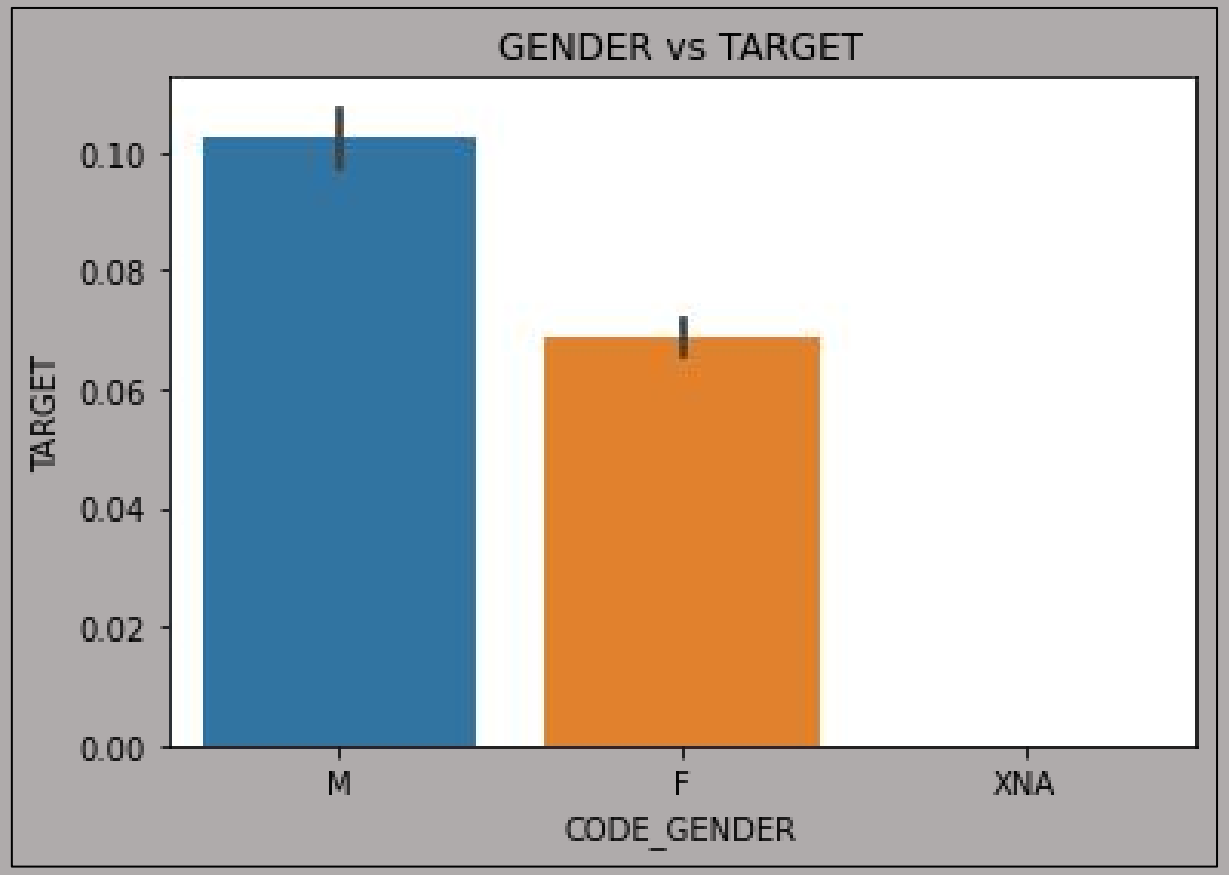
Task 4 : Univariate / Segmented Univariate Analysis of Data





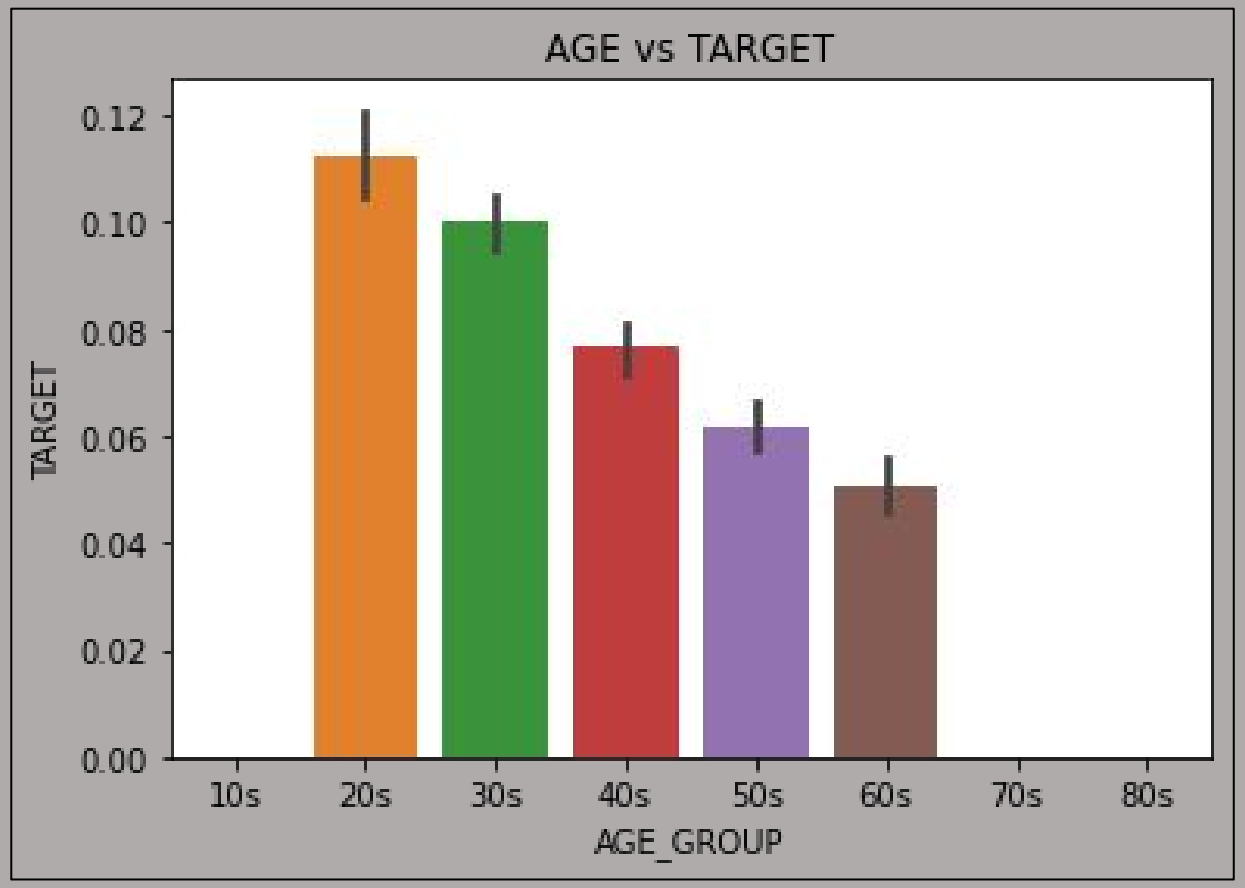
Inference : From the figures above, it is observed that, customers of with region rating as 2 dominate on both the categories of data.

Task 4 : Bi-Variate Analysis of Data



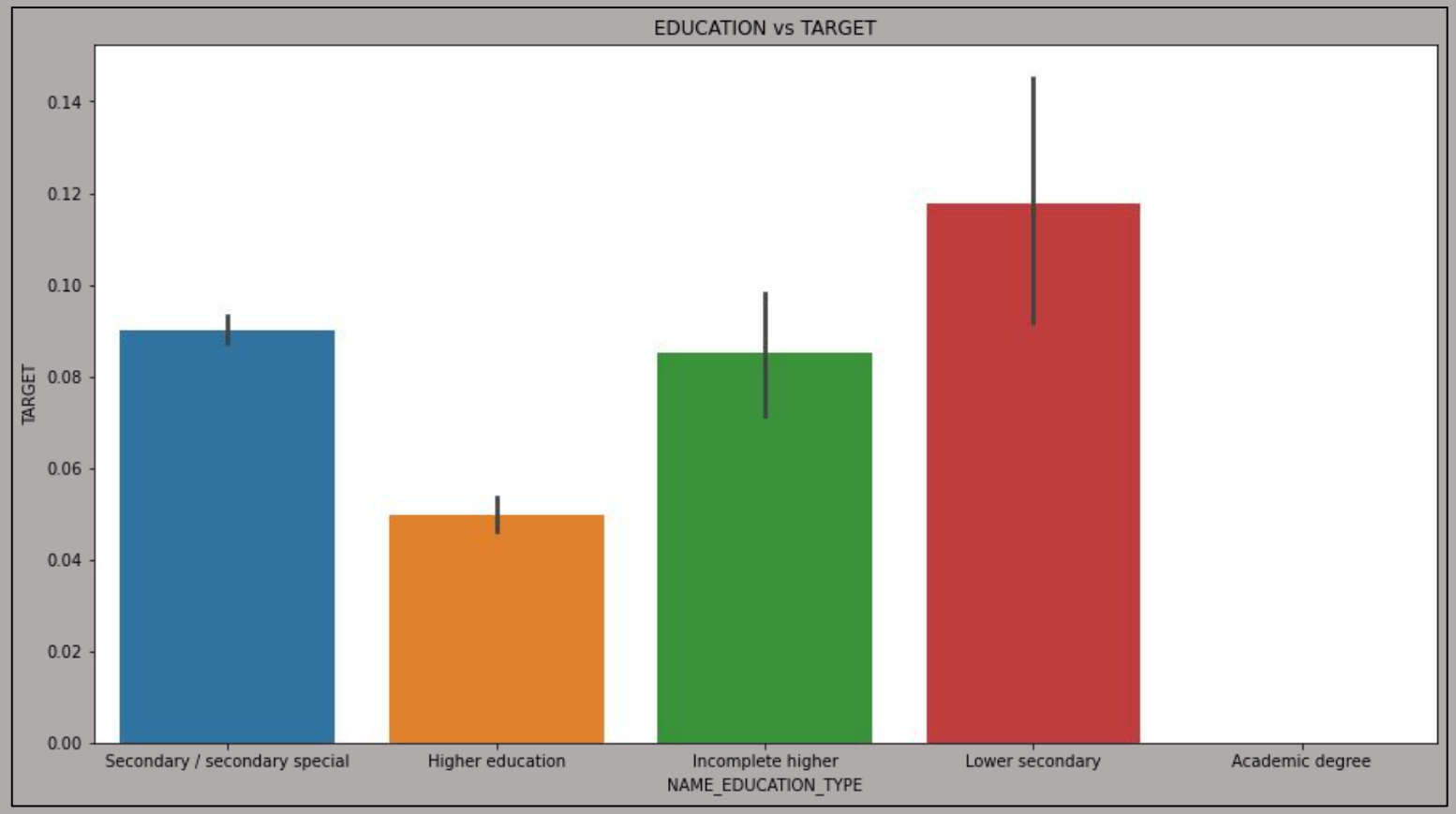
Inference : From the figures it is observed that, male customers have higher probability than the female customers for dues.

Task 4 : Bi-Variate Analysis of Data



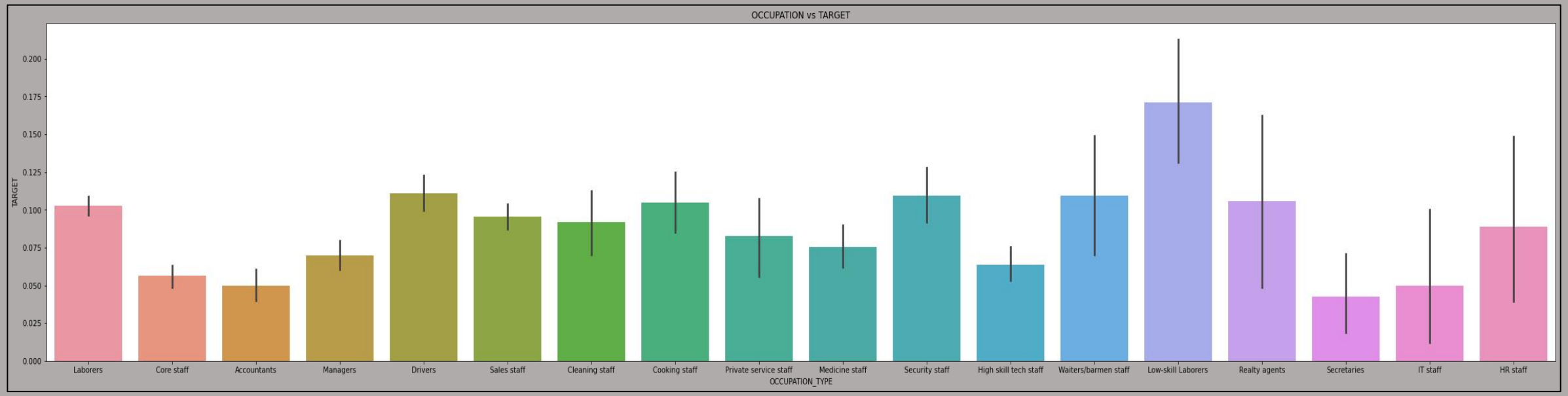
Inference : From the figures it is observed that, customers of age group 20's and 30's have higher probability for dues. Hence, they can be identified as "Risky" customers.

Task 4 : Bi-Variate Analysis of Data



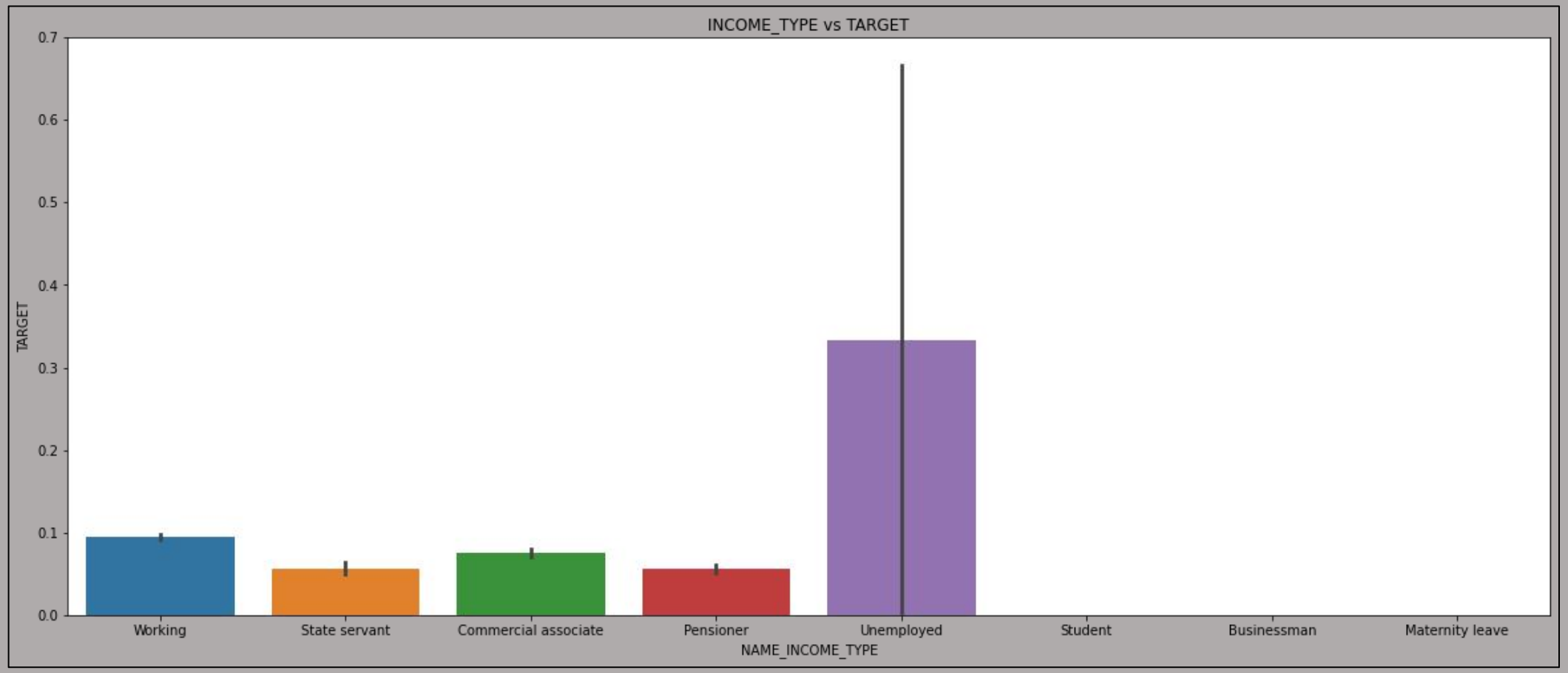
Inference : From the figures it is observed that, customers with lower secondary education have higher probability for dues.

Task 4 : Bi-Variate Analysis of Data



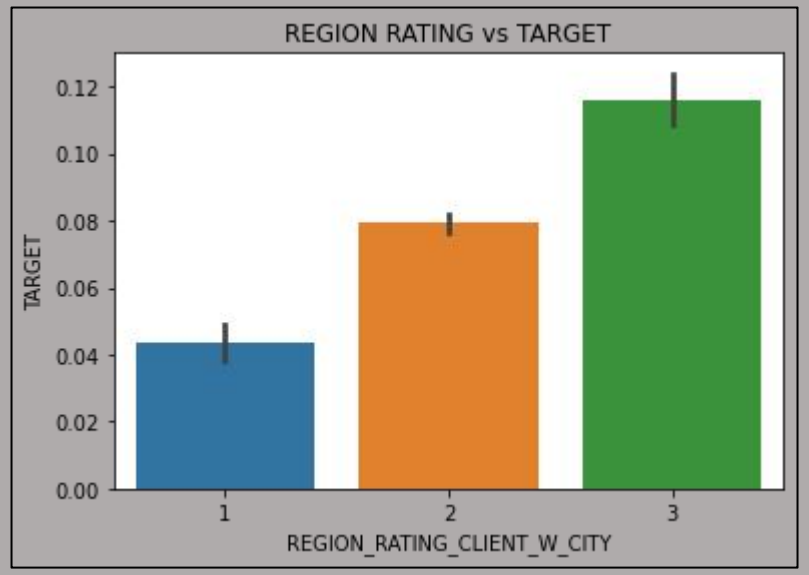
Inference : From the figures it is observed that, customers whose occupation are Low-Skill Labours have high risk for dues.

Task 4 : Bi-Variate Analysis of Data



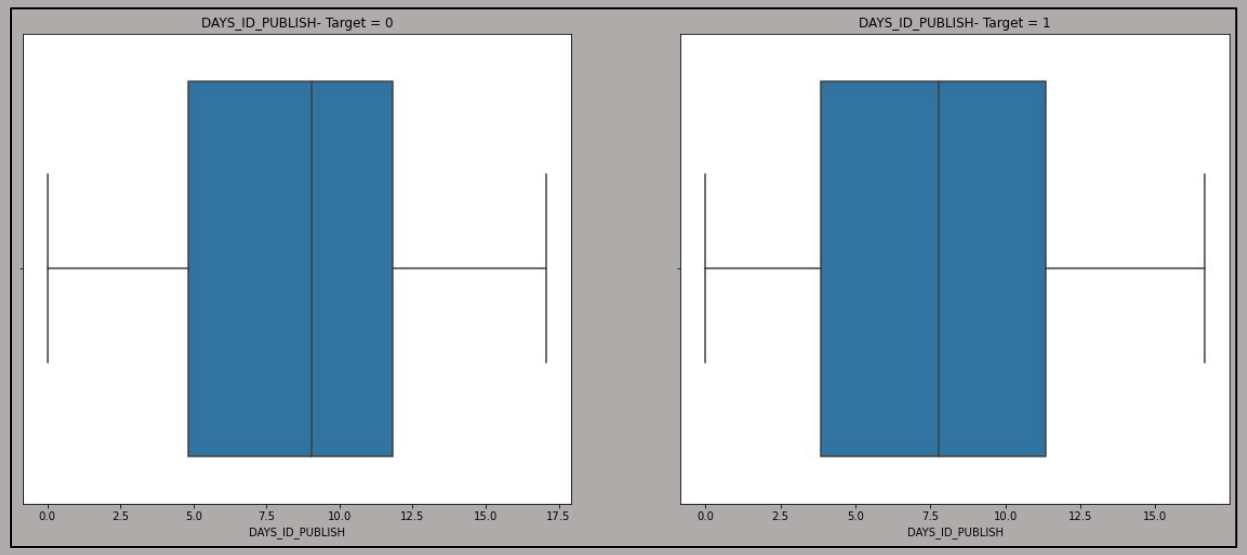
Inference : From the figures it is observed that, customers who are unemployed have high risk for due payments.

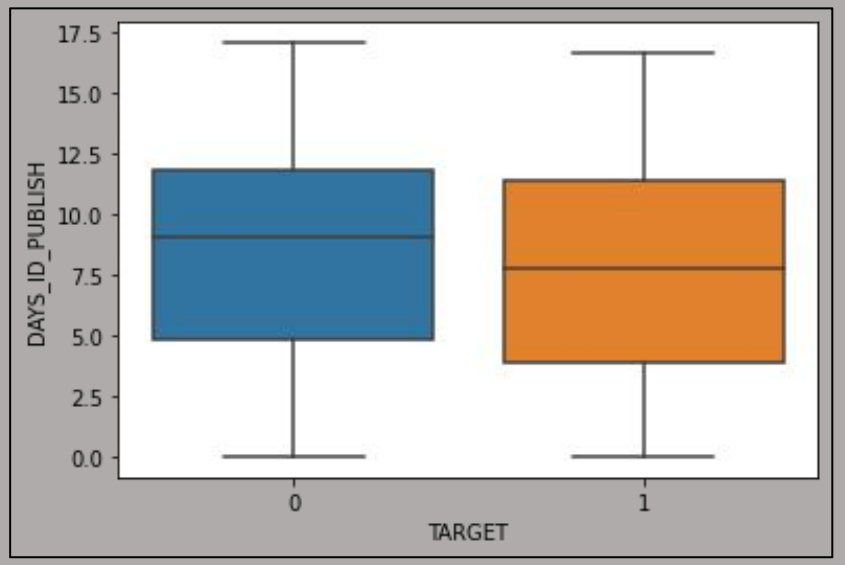
Task 4 : Bi-Variate Analysis of Data



Inference : From the figures it is observed that, customers with region rating as 3 have a higher risk for the due payments.

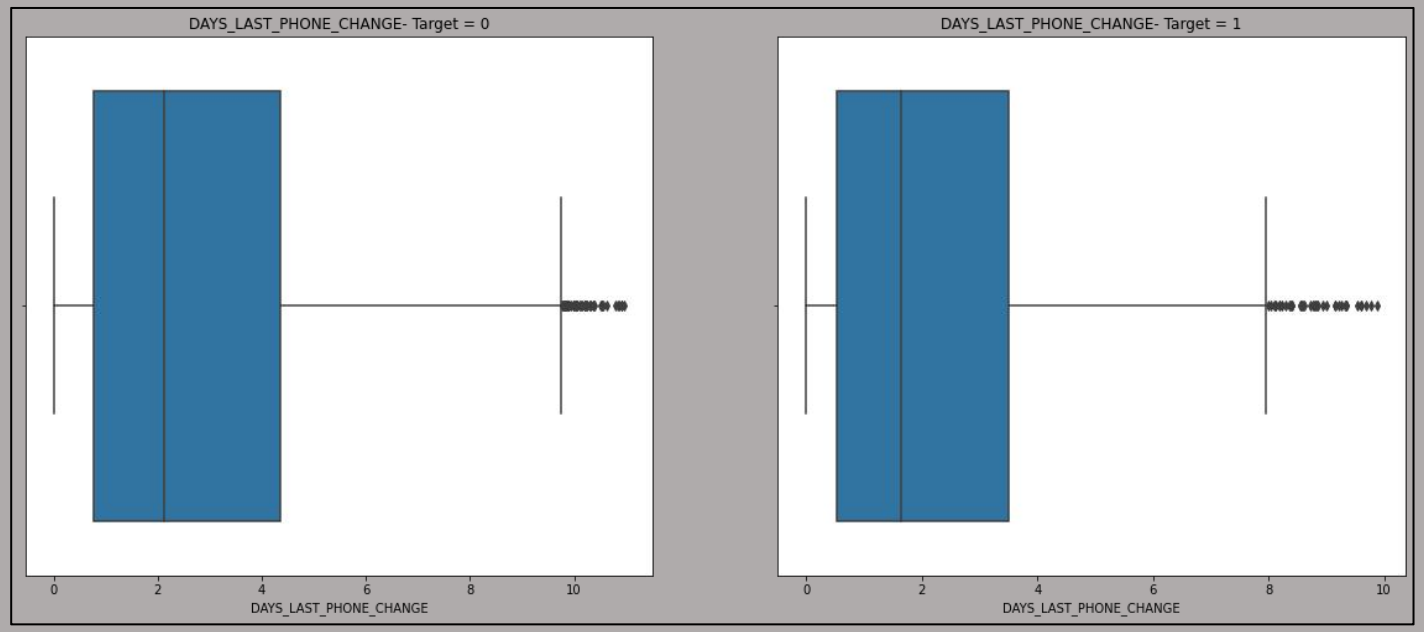
Task 4 : Univariate & Bi-Variate Analysis of Data (Numerical)

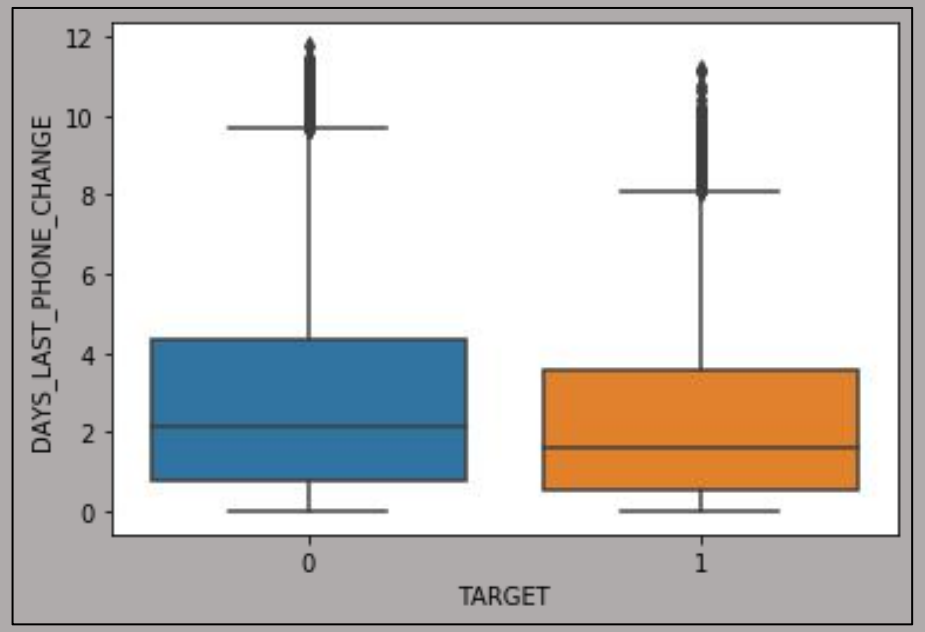




Inference : From the figures it is observed that, customers with no difficulty (Target 0) change their identity documents closer to submission.

Task 4 : Univariate & Bi-Variate Analysis of Data (Numerical)

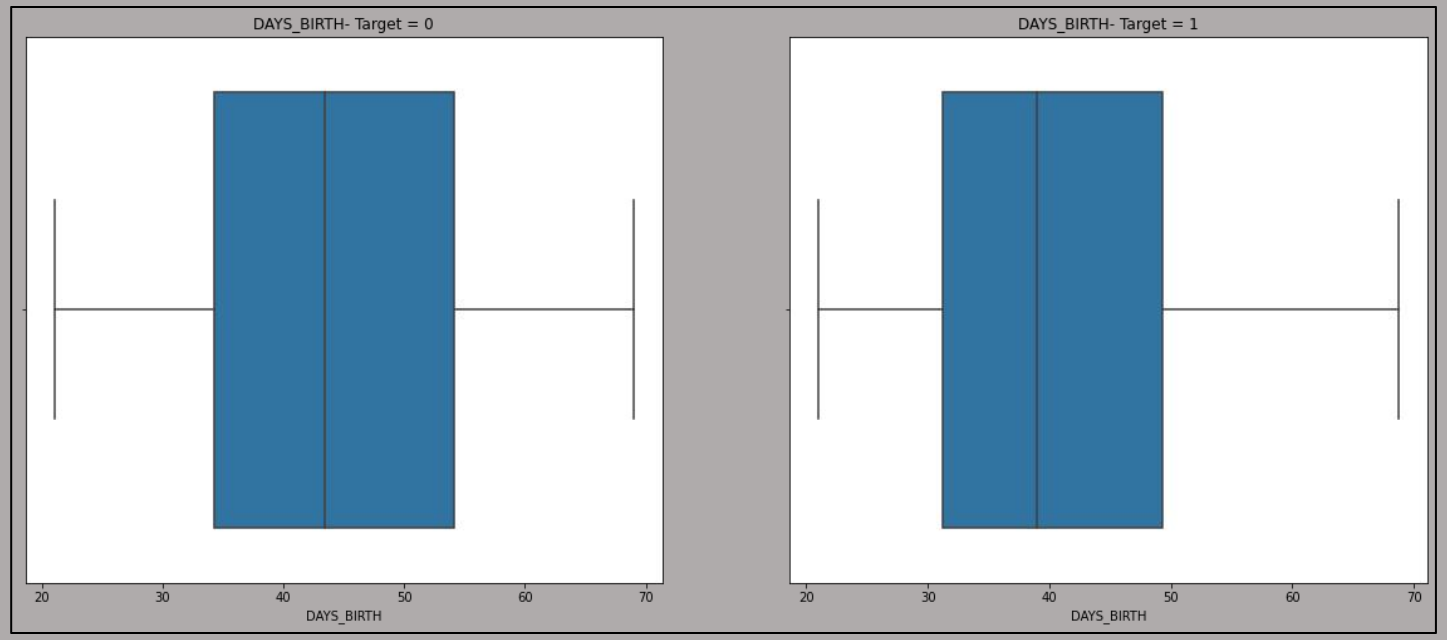


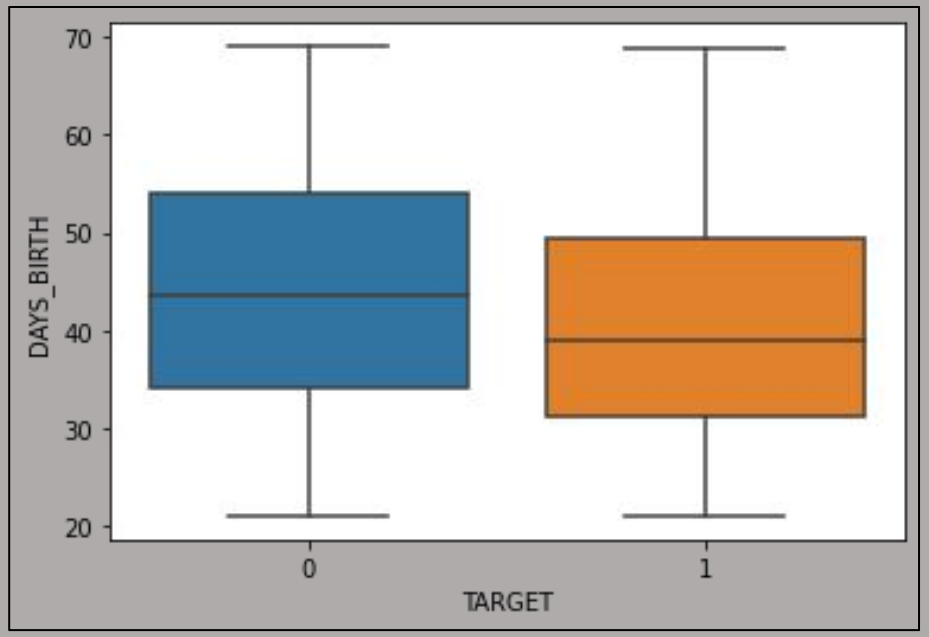


Inference : From the figures it is observed that, customers with no difficulty (Target 0) change their phone number closer to submission.

Also, the outlier data present for customers with dues (Target 1) is comparitively more.

Task 4 : Univariate & Bi-Variate Analysis of Data (Numerical)





Inference : From the figures it is observed that, customers with age group > 40 are non-risky. Customers between 30 to 45 age group are having both the possibilities of being risky and non-risky.

Project 6

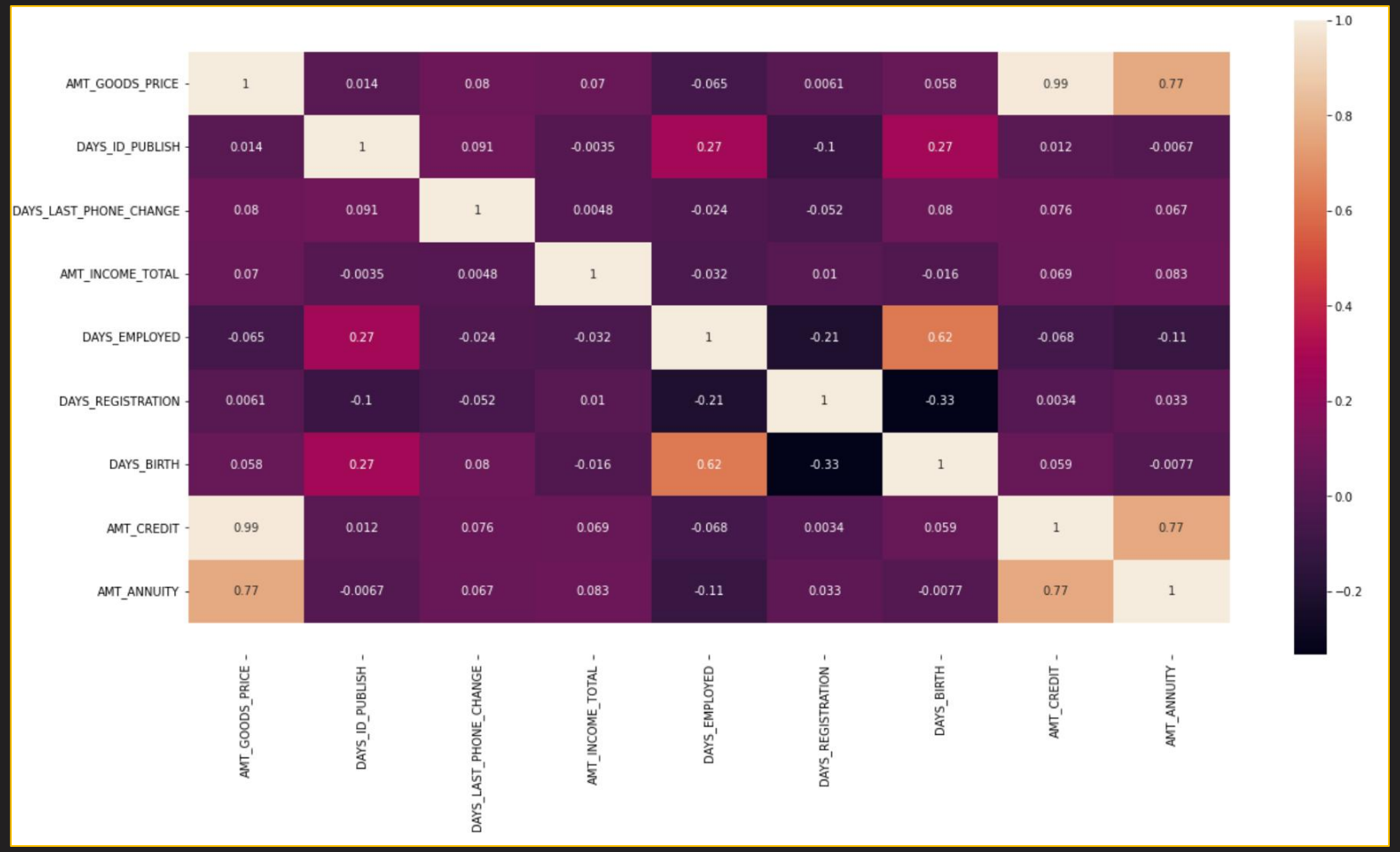
Bank Loan - Case Study

Task 5 : Correlational Analysis

Task 5 : Correlational Analysis

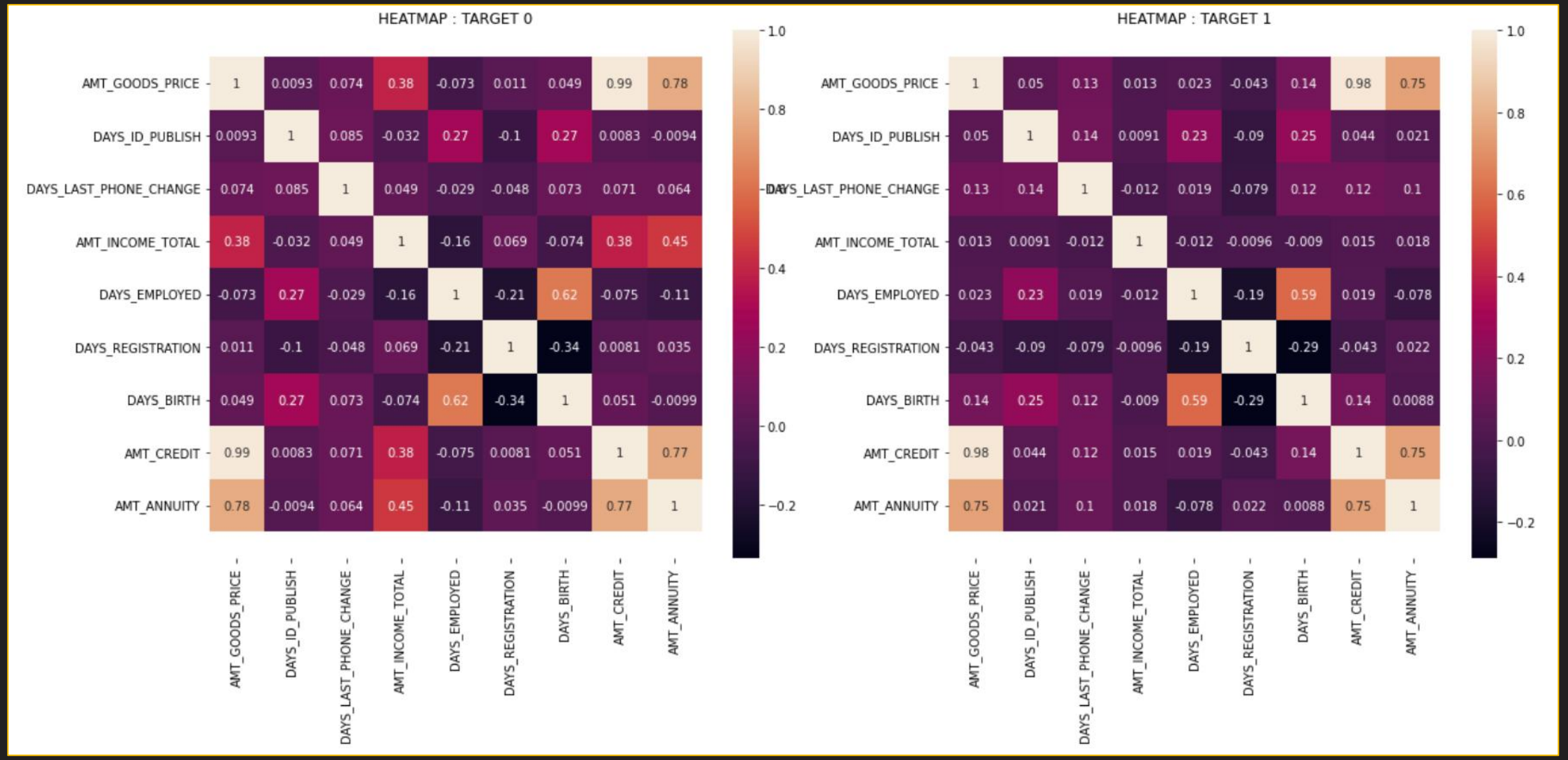
- **Objective :** To understand the correlation between variables and the target variable can provide insights into strong indicators of loan default.
- **Approach :** To segment data on different scenarios and identify the top correlated factors for each segment based on the dues.
    - » Correlation Calculation for Customers with No dues (Target = 0)
    - » Correlation Calculation for Customers with Some dues (Target = 1)
    - » Identifying the top 10 Correlated factors for both the groups
- **Data Visualisation :** Using heatmaps the correlations between variables within each segment are observed. The top 10 correlated factors are identified.

Task 5 : Correlation Analysis of Data - Entire Dataset



- **Heatmap Visualisation for Correlational Analysis for the entire dataset.**
- **Top Correlated Factors**
- **AMT_GOODS_PRICE vs AMT_ANNUITY**
- **AMT_ANNUITY vs AMT_CREDIT**
- **AMT_ANNUITY vs AMT_GOODS_PRICE**

Task 5 : Correlation Analysis of Data - Target 0 and Target 1 Segments



Task 5 : Correlation Analysis of Data - Top Correlated Factors

Fig.1

	VAR1	VAR2	Correlation	Correlation_abs
398	OBS_60_CNT_SOCIAL_CIRCLE	OBS_30_CNT_SOCIAL_CIRCLE	1.00	1.00
549	AMT_REQ_CREDIT_BUREAU_MON	AMT_REQ_CREDIT_BUREAU_WEEK	1.00	1.00
148	AMT_GOODS_PRICE	AMT_CREDIT	0.99	0.99
324	REGION_RATING_CLIENT_W_CITY	REGION_RATING_CLIENT	0.95	0.95
266	CNT_FAM_MEMBERS	CNT_CHILDREN	0.88	0.88
423	DEF_60_CNT_SOCIAL_CIRCLE	DEF_30_CNT_SOCIAL_CIRCLE	0.85	0.85
149	AMT_GOODS_PRICE	AMT_ANNUITY	0.78	0.78
124	AMT_ANNUITY	AMT_CREDIT	0.77	0.77
199	DAYS_EMPLOYED	DAYS_BIRTH	0.62	0.62
123	AMT_ANNUITY	AMT_INCOME_TOTAL	0.45	0.45

Fig.2

	VAR1	VAR2	Correlation	Correlation_abs
398	OBS_60_CNT_SOCIAL_CIRCLE	OBS_30_CNT_SOCIAL_CIRCLE	1.00	1.00
549	AMT_REQ_CREDIT_BUREAU_MON	AMT_REQ_CREDIT_BUREAU_WEEK	1.00	1.00
148	AMT_GOODS_PRICE	AMT_CREDIT	0.99	0.99
324	REGION_RATING_CLIENT_W_CITY	REGION_RATING_CLIENT	0.95	0.95
266	CNT_FAM_MEMBERS	CNT_CHILDREN	0.88	0.88
423	DEF_60_CNT_SOCIAL_CIRCLE	DEF_30_CNT_SOCIAL_CIRCLE	0.85	0.85
149	AMT_GOODS_PRICE	AMT_ANNUITY	0.78	0.78
124	AMT_ANNUITY	AMT_CREDIT	0.77	0.77
199	DAYS_EMPLOYED	DAYS_BIRTH	0.62	0.62
123	AMT_ANNUITY	AMT_INCOME_TOTAL	0.45	0.45

For further observation, using the correlation formula, the top 10 correlated factors are identified for both the segment of data. It is observed that mostly the factors of correlation are similar between both the data segments.

Fig.1 : Customers with no dues (Target = 0)

Fig.2 : Customers with some dues (Target = 1)

Project 6

Bank Loan - Case Study

EDA : Combined Dataset

- Univariate Analysis
  - Categorical
    - Contact_Type, Education Type, Gender, Age, Amount Category
  - Numerical
    - Goods price, Credit Amount, Amount Total, Amount Annuity, Days : Birth, Employed, Last Phone Change, Registration
- Bivariate Analysis
  - Categorical
    - Contract Type, Gender, Target, Age Group, Amount Category
  - Numerical
    - Region Rating vs Credit, Region Rating City vs Credit, Family Size vs Credit

Combined Analysis of Bank Loan Study - Data Preprocessing

Combined Analysis of Data

```
#Loading the previous_application data
prev_data = pd.read_csv("/Users/swat/Desktop/Qatar/Trainity-DA/Excel_Projects/previous_application.csv")
prev_data.head()
```

```
#Merging the datasets
final_data = pd.merge(left = new_cleaned_data, right = prev_data, left_on='SK_ID_CURR', right_on='SK_ID_CURR', how ='in
final_data.head()
```

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE_x	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CRE
0	100007	0	Cash loans	M	N	Y	0	121500.0	513
1	100009	0	Cash loans	F	Y	Y	1	171000.0	1560
2	100012	0	Revolving loans	M	N	Y	0	135000.0	405
3	100026	0	Cash loans	F	N	N	1	450000.0	497
4	100027	0	Cash loans	F	N	Y	0	83250.0	239

Loading the data, Merging both the datasets and dividing the entire data attributes into numerical and categorical.

Dividing the data into numerical and categorical columns for analysis

```
final_category = ['NAME_CONTRACT_TYPE_x','FLAG_OWN_CAR','FLAG_OWN_REALTY','CODE_GENDER','NAME_EDUCATION_TYPE',
                  'AMT_CATEGORY', 'AGE_GROUP']
final_numeric = ['AMT_GOODS_PRICE_x', 'DAYS_BIRTH','AMT_CREDIT_x','AMT_ANNUITY_x','DAYS_REGISTRATION',
                 'DAYS_LAST_PHONE_CHANGE','DAYS_ID_PUBLISH', 'AMT_INCOME_TOTAL', 'DAYS_EMPLOYED']
```

Combined Analysis of Bank Loan Study - Dataset Division for Status

Getting the different Status Values for Application and Creating respective datasets for analysis

```
final_data.NAME_CONTRACT_STATUS.unique()
```

```
array(['Approved', 'Refused', 'Canceled', 'Unused offer'], dtype=object)
```

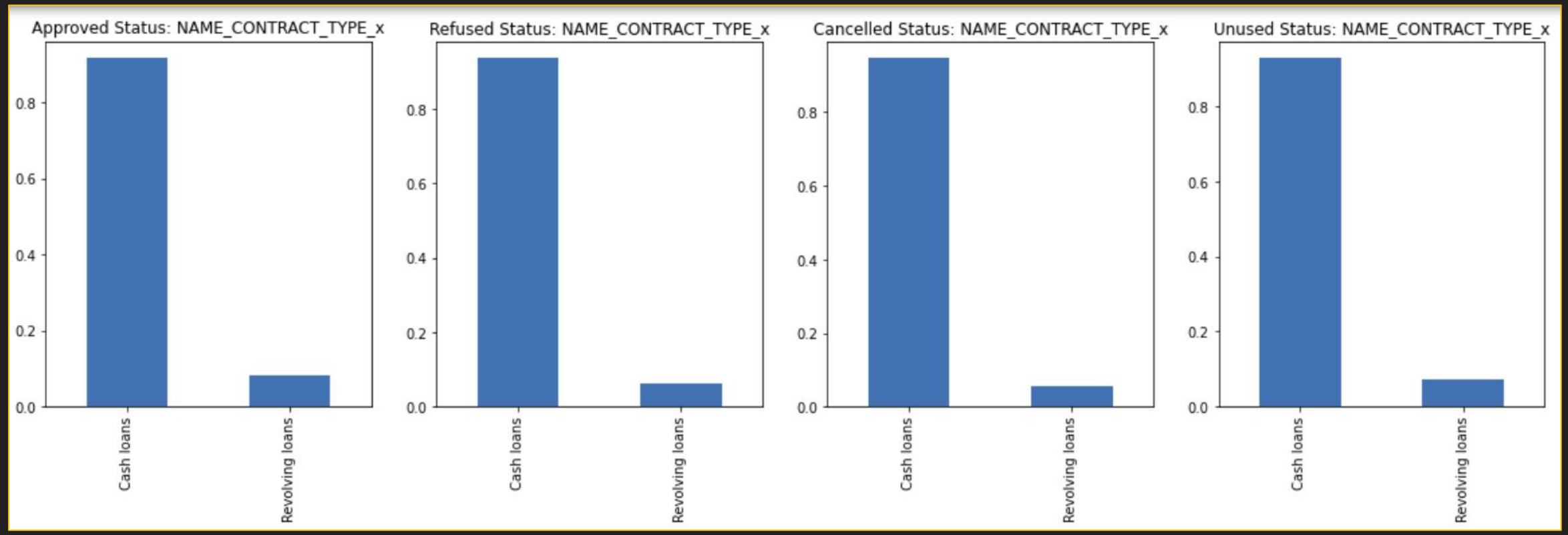
```
approved_data = final_data[final_data.NAME_CONTRACT_STATUS == 'Approved']
refused_data = final_data[final_data.NAME_CONTRACT_STATUS == 'Refused']
cancelled_data = final_data[final_data.NAME_CONTRACT_STATUS == 'Canceled']
unused_data = final_data[final_data.NAME_CONTRACT_STATUS == 'Unused offer']

print("--------LIST OF NUMBER OF DATA FOR EACH CATEGORY---------------")
print("Approved Data : ", approved_data.shape)
print("Refused Data : ", refused_data.shape)
print("Cancelled Data : ", cancelled_data.shape)
print("Unused Data : ", unused_data.shape)
print("----------------------------------------------------------------")
```

```
--------LIST OF NUMBER OF DATA FOR EACH CATEGORY---------------
Approved Data :  (4376, 75)
Refused Data :  (1218, 75)
Cancelled Data :  (1123, 75)
Unused Data :  (124, 75)
----------------------------------------------------------------
```

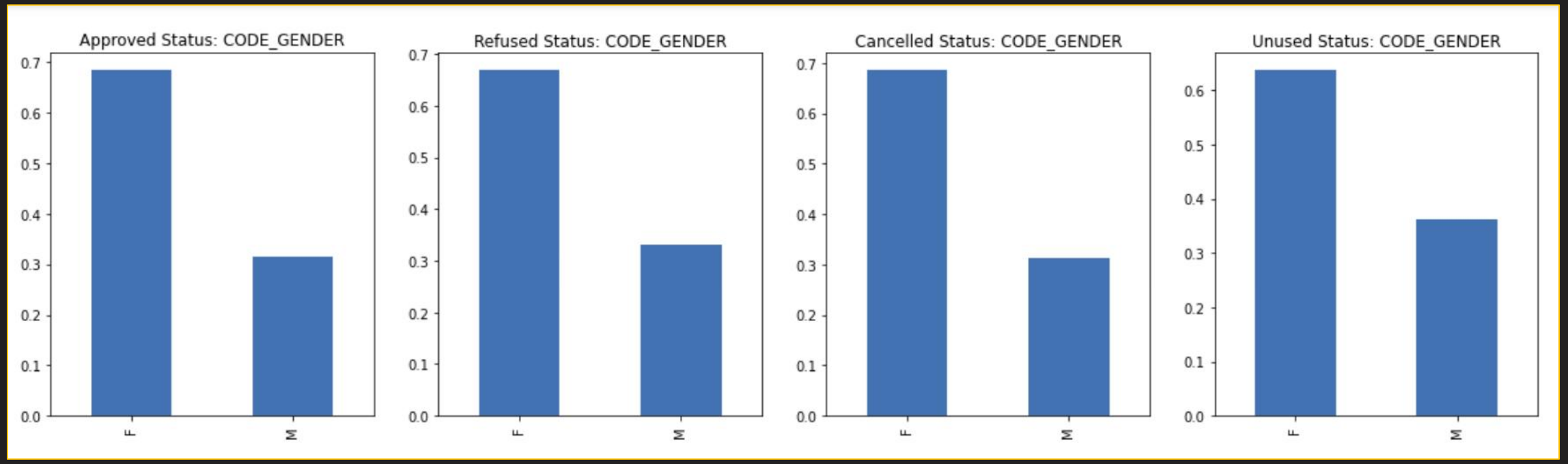
Four Datasets created based on the status as : Approved, Cancelled, Refused and Unused.

Combined Univariate Analysis for Categorical Columns



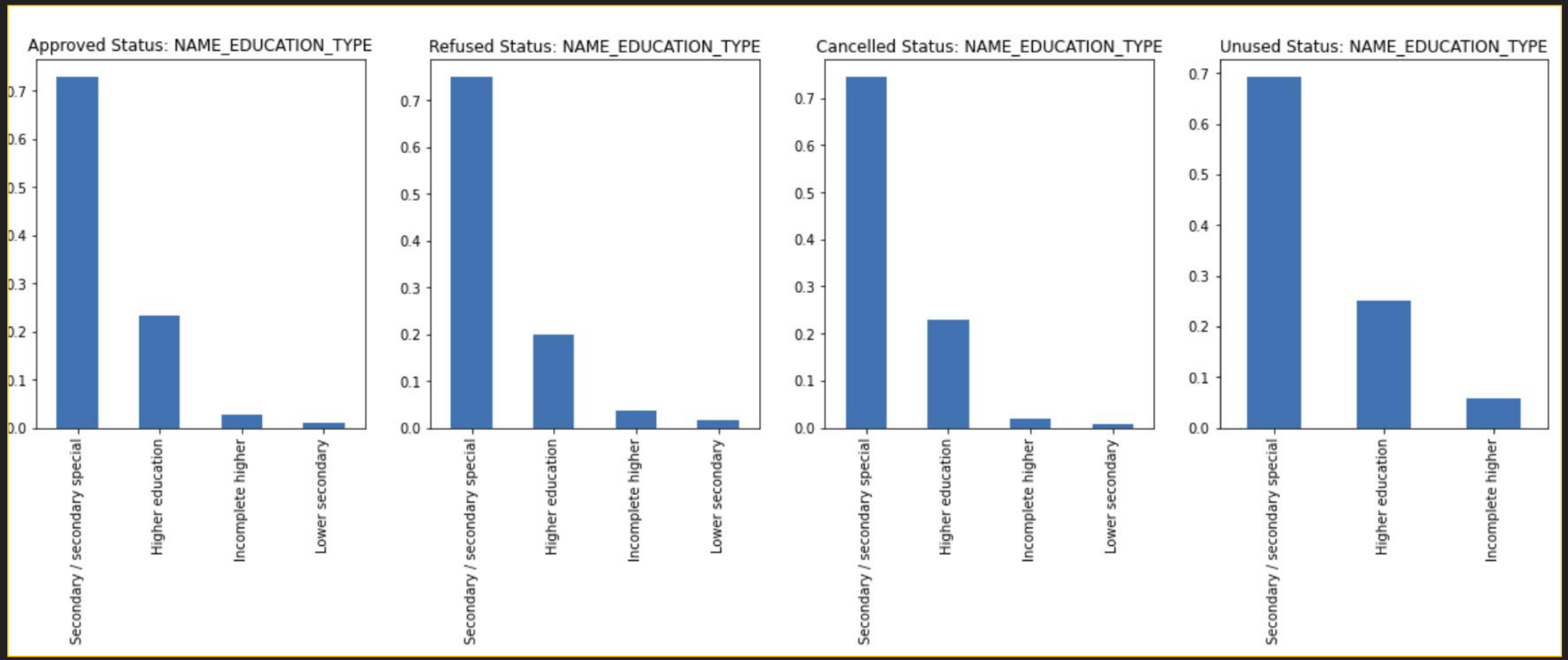
Inference : For all the loan status, it is observed that the Cash Loans are higher than the revolving loans.

Combined Univariate Analysis for Categorical Columns

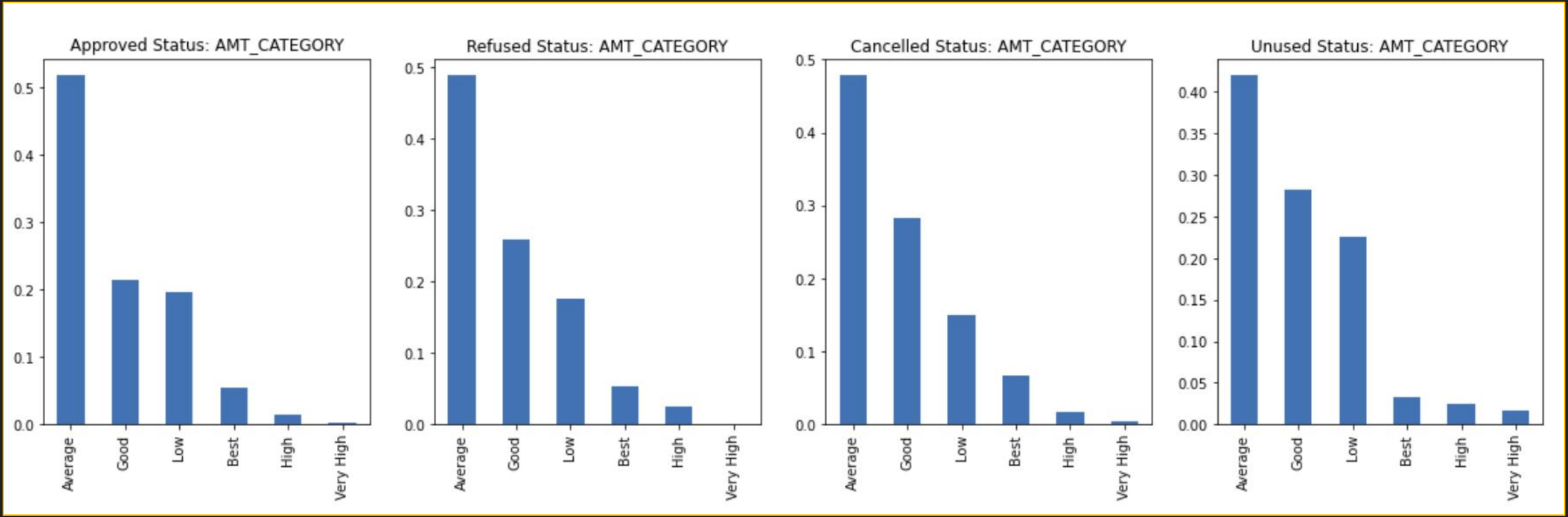


Inference : For all the loan status, it is observed that the female customers have a larger impact than the male customers

Combined Univariate Analysis for Categorical Columns

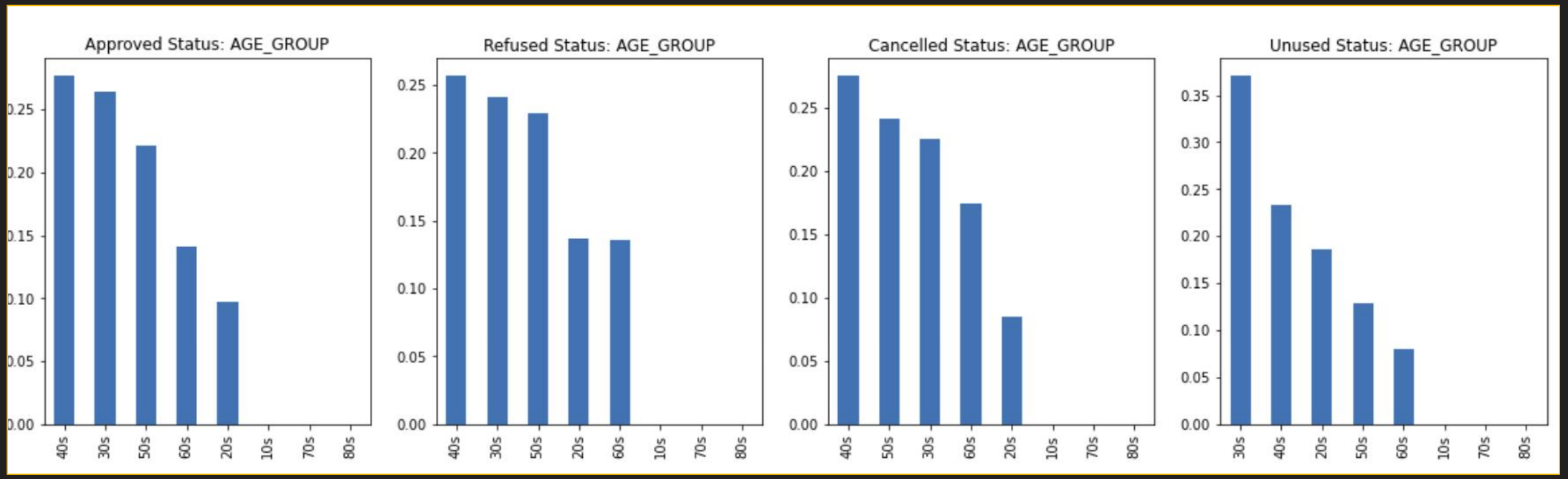


Combined Univariate Analysis for Categorical Columns



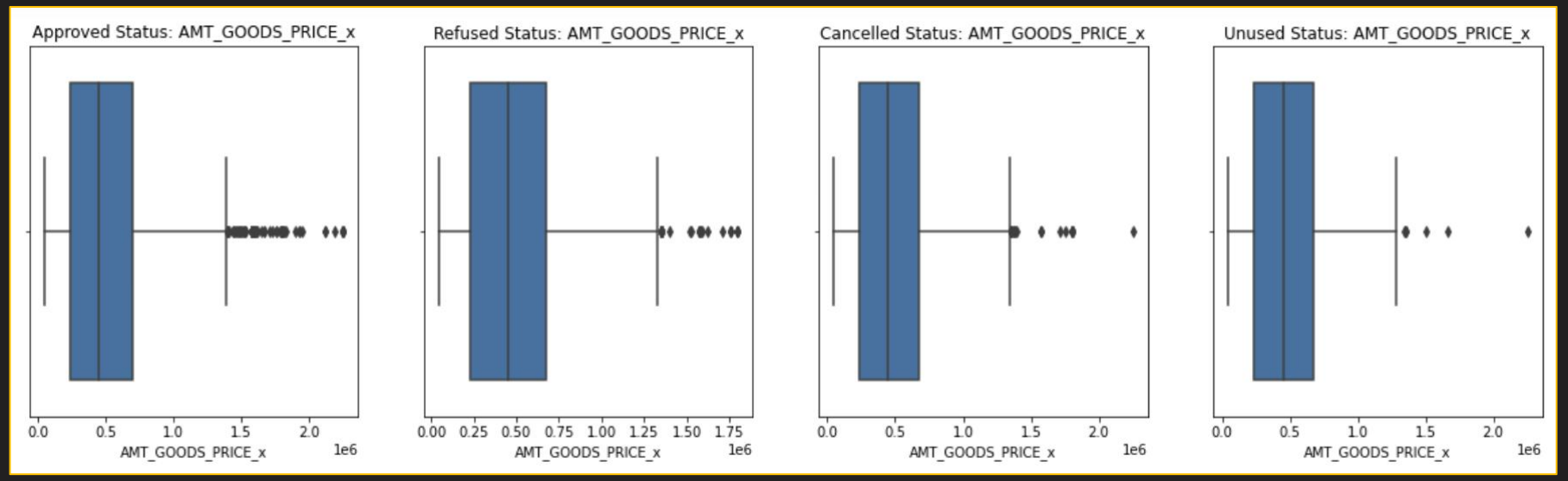
Inference : For all the loan status, it is observed that the average group customers dominate the other group based on the amount.

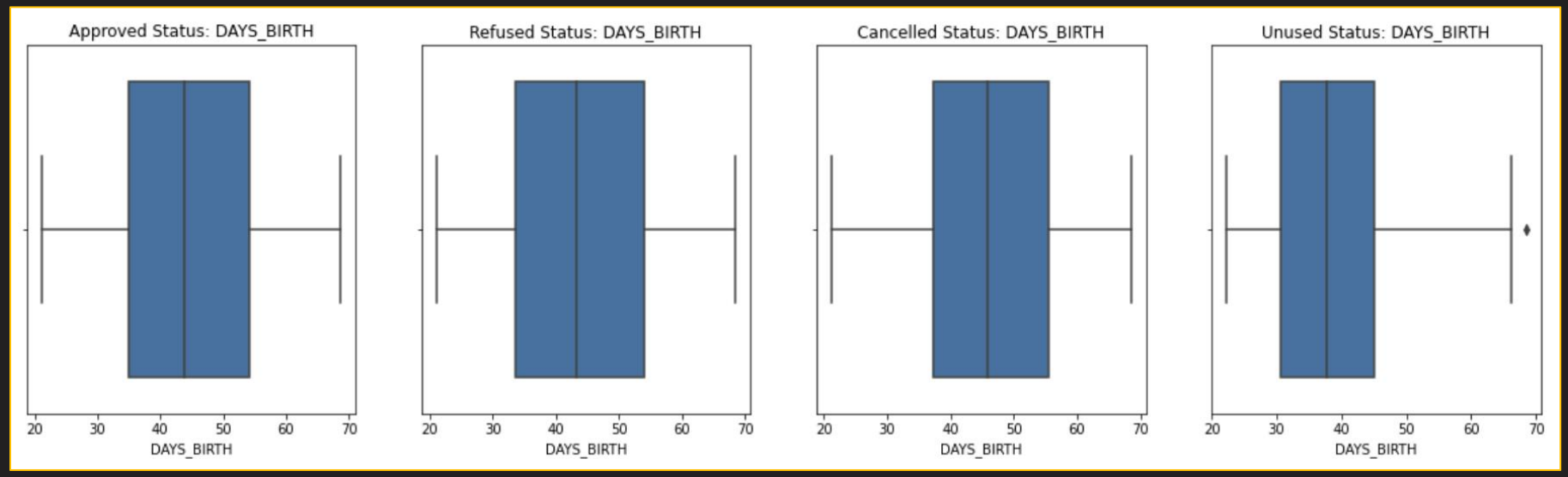
Combined Univariate Analysis for Categorical Columns



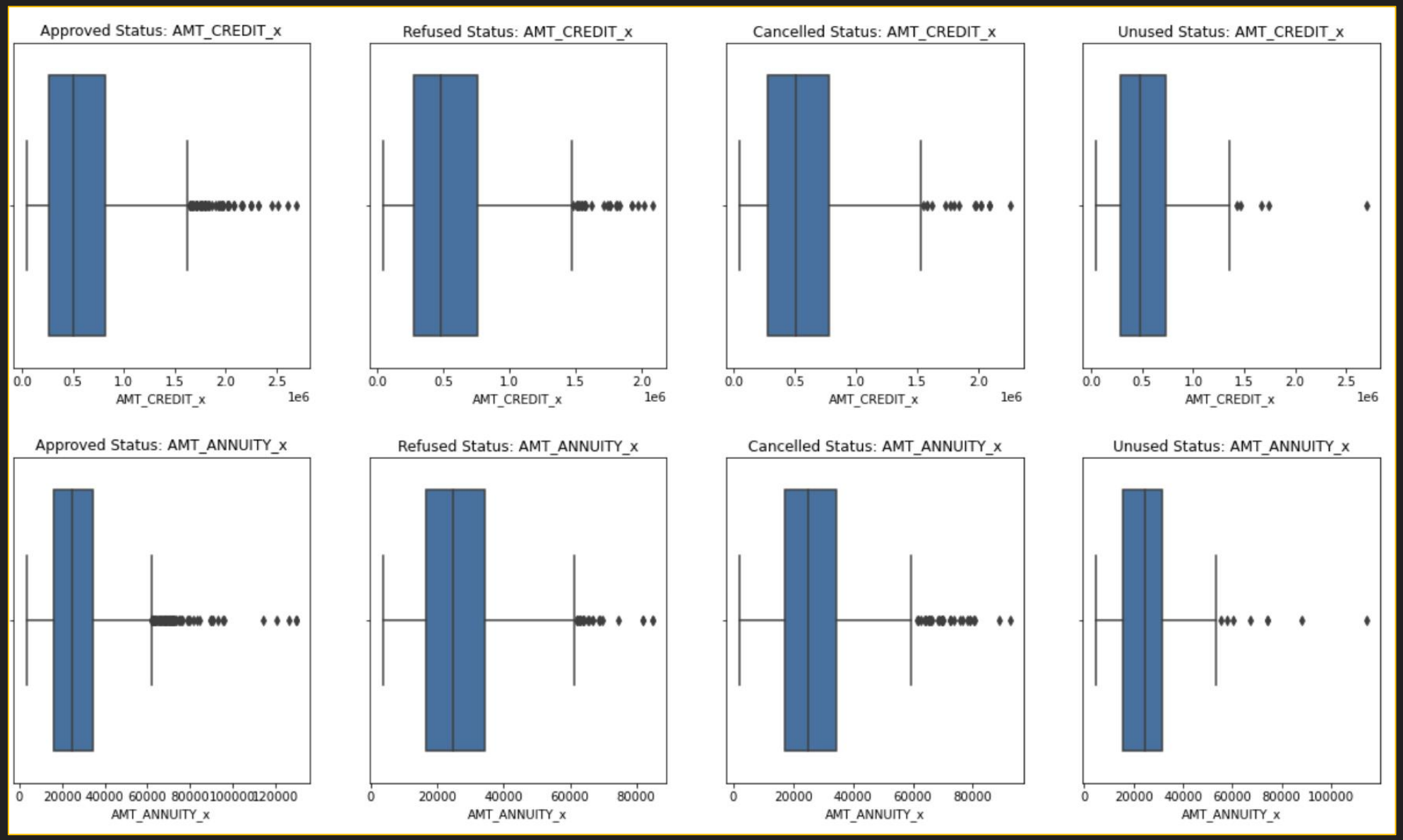
Inference : The 40's group customers mostly have the loan status as approved, refused and cancelled. The 50's group customers mostly come second highest in the cancelled loans and 30's group customers mostly dominate the unused loans category

Combined Univariate Analysis for Numerical Columns

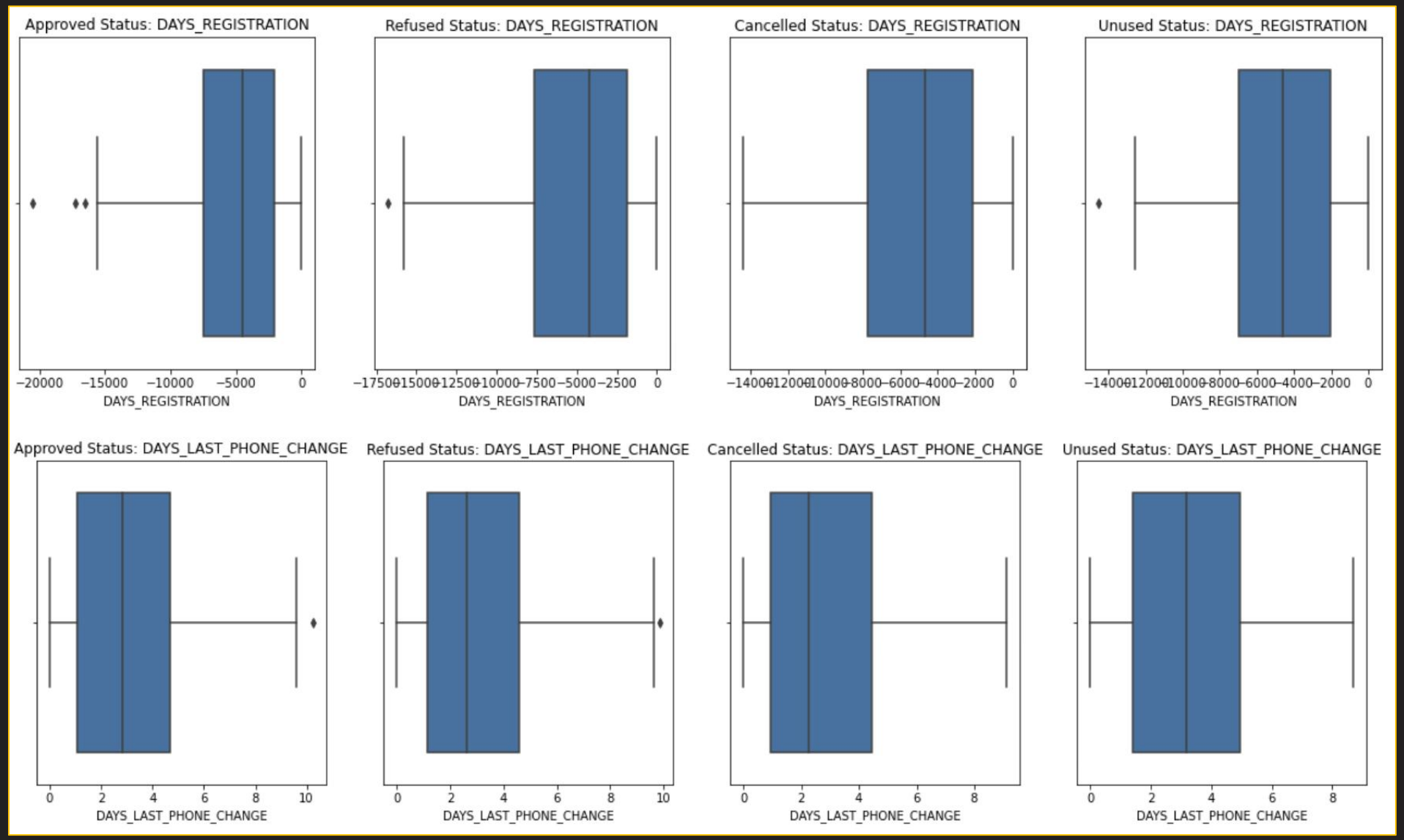




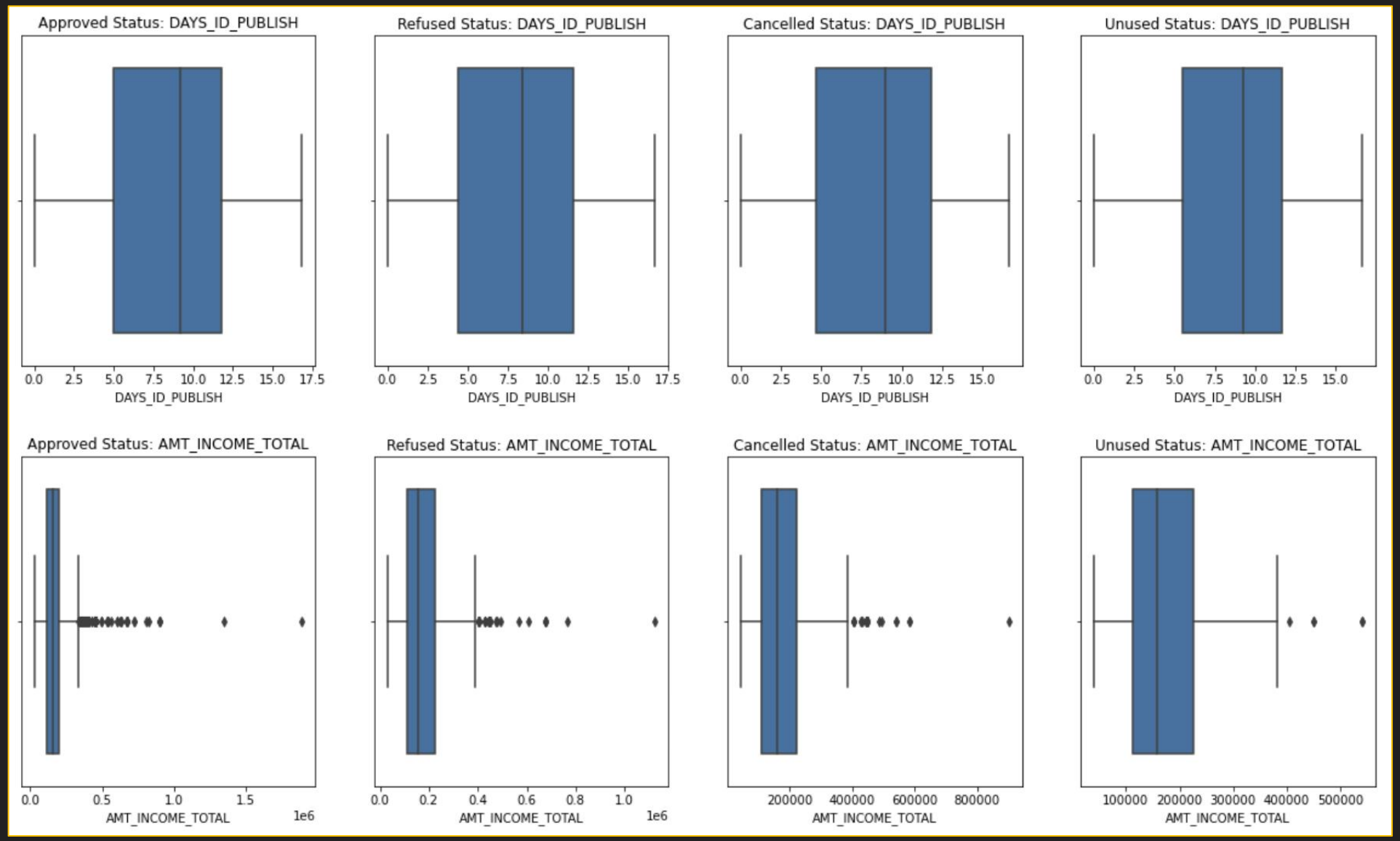
Combined Univariate Analysis for Numerical Columns



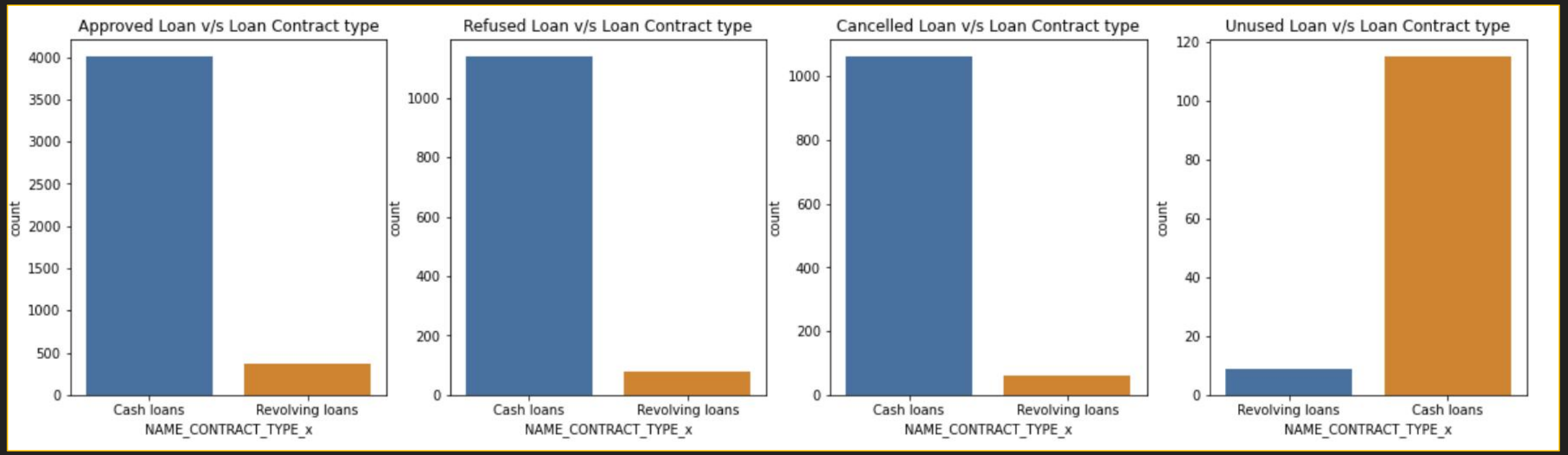
Combined Univariate Analysis for Categorical Columns



Combined Univariate Analysis for Categorical Columns

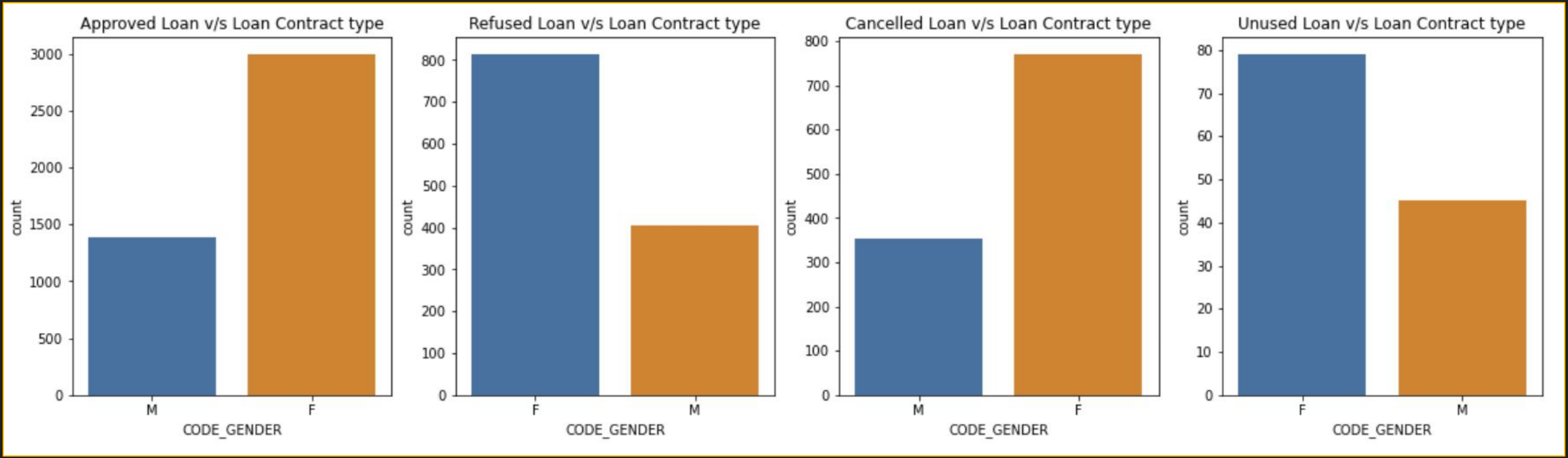


Combined Bi-Variate Analysis for Categorical Columns



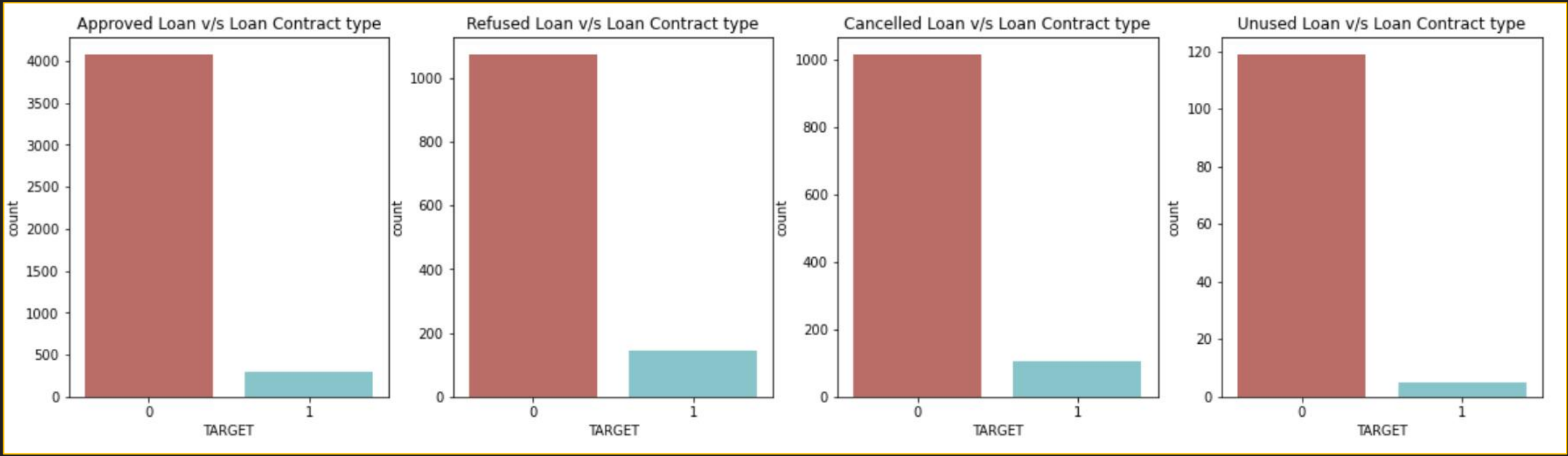
Inference : For all the loan status, it is observed that the Cash Loans are higher than the revolving loans.

Combined Bi-Variate Analysis for Categorical Columns



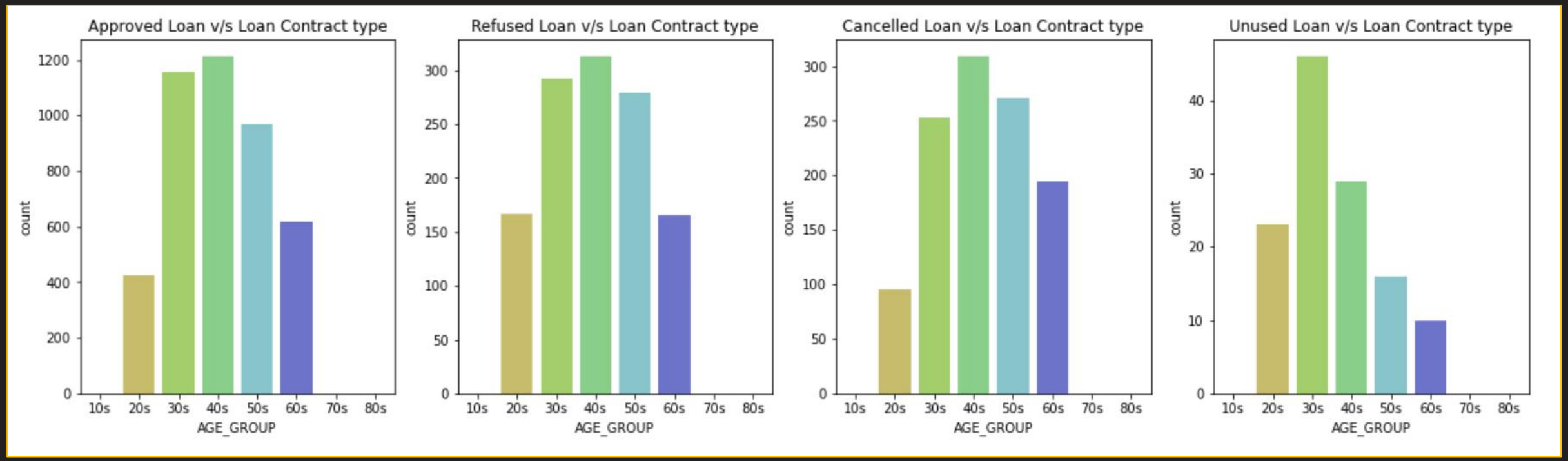
Inference : Female Customer loans is larger in all the loan category than the male customers.

Combined Bi-Variate Analysis for Categorical Columns



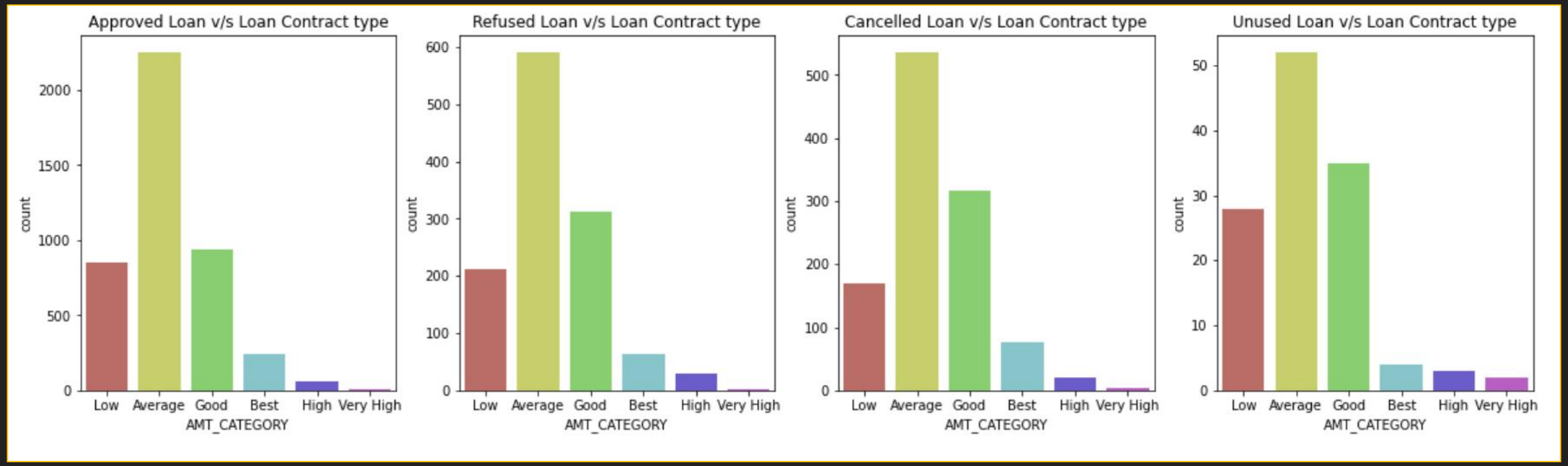
Inference : For all the loan status, it is observed that the customers with no dues are dominating the customers with some dues.

Combined Bi-Variate Analysis for Categorical Columns



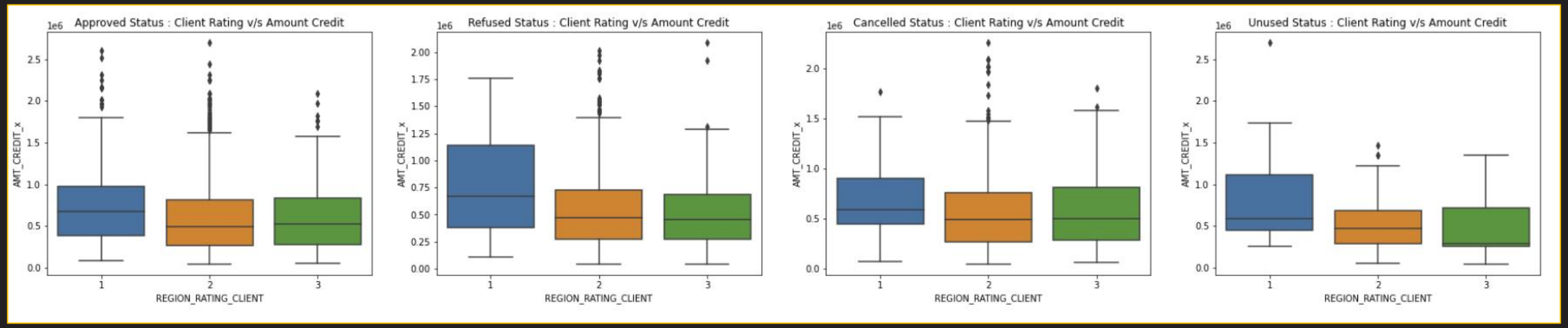
Inference : The 40's group customers mostly have the loan status as approved, refused and cancelled. The 50's group customers mostly come second highest in the cancelled loans and 30's group customers mostly dominate the unused loans category

Combined Bi-Variate Analysis for Categorical Columns



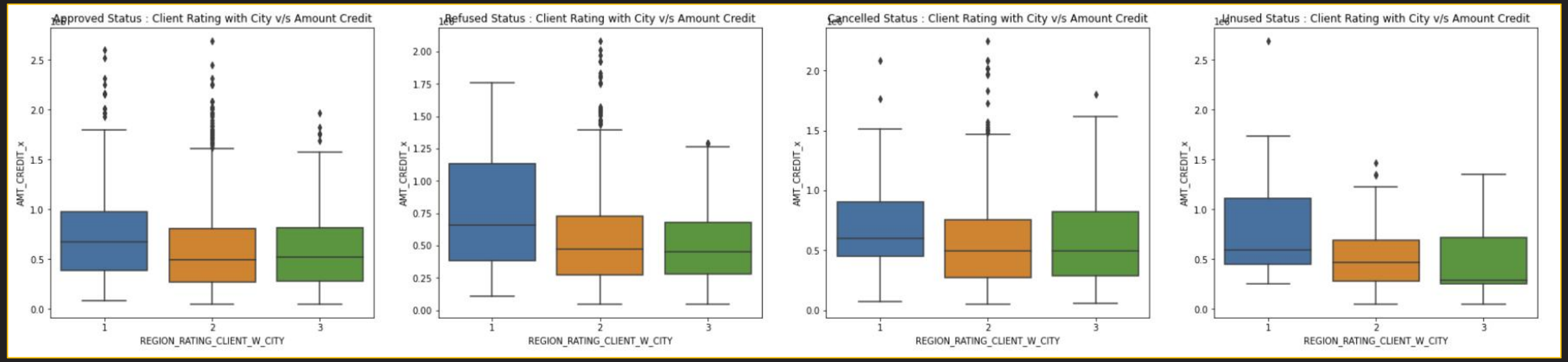
Inference : For all the loan status, it is observed that the average group customers dominate the other group based on the amount.

Combined Bi-Variate Analysis for Numerical Columns



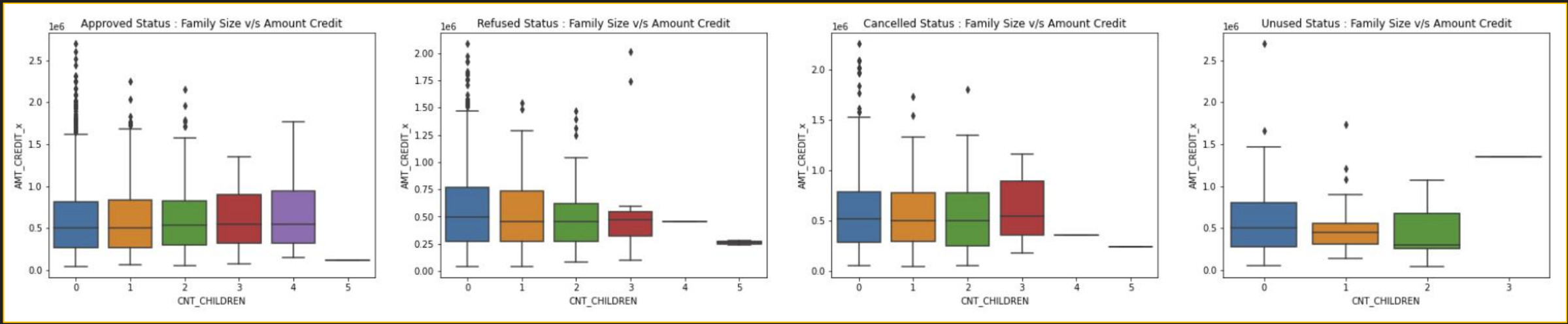
Inference : For all the loan status, client rating with 2 have the most outliers. For approved status, client rating with 1 and 3 have a larger outlier count than the other loan statuses.

Combined Bi-Variate Analysis for Numerical Columns



Inference : For all the loan status, client rating with 2 have the most outliers. For approved status, client rating with 1 and 3 have a larger outlier count than the other loan statuses.

Combined Bi-Variate Analysis for Numerical Columns



Inference : It is observed that, customers with no children have a greater outlier count with the loan status. For customers with 2 children, the outliers are more on approved and refused statuses. For customers with 1 children, there is a stabilised box plot on the loan status.

Summary of Analysis

- ✓ The dataset is observed completely before analysis for the pre-processing steps. The various attributes are learnt and identified.
- ✓ Missing Values and outliers are handled appropriately for the required attributes analysis.
- ✓ Through the univariate and bivariate analysis, different attribute relationships are identified which is used for analysing the risk associated with a customer.
- ✓ Correlation Analysis is made to identify the closely related attributes and the top 10 attributes are listed.
- ✓ Combined Dataset EDA is performed and the relationship are identified between the attributes for more detailed conclusions.
- ✓ Based on these evaluations, a list of Categorical and Numerical Attributes are identified which can be used for prediction of risk customers.

Project 6

Bank Loan - Case Study