TRAINITY DATA ANALYTICS PROJECT – 6

BANK LOAN
CASE STUDY

BY
ASHWIN K

CONTENTS

PROJECT DESCRIPTION

- THIS PROJECT AIMS TO LEVERAGE EXPLORATORY DATA ANALYSIS (EDA) TECHNIQUES TO ANALYZE PATTERNS IN DATA AND MINIMIZE THE RISK OF LOSING MONEY WHILE LENDING TO CUSTOMERS IN THE CONSUMER FINANCE INDUSTRY.

- THE FOCUS IS ON URBAN CUSTOMERS WHO OFTEN FACE DIFFICULTIES IN OBTAINING LOANS DUE TO INSUFFICIENT OR NON-EXISTENT CREDIT HISTORIES.

- BY CONDUCTING AN IN-DEPTH ANALYSIS OF THE AVAILABLE DATA, WE AIM TO IDENTIFY PATTERNS AND INDICATORS THAT CAN HELP PREDICT THE REPAYMENT CAPABILITY OF LOAN APPLICANTS.

- THIS ANALYSIS WILL ENABLE THE COMPANY TO MAKE INFORMED DECISIONS AND AVOID REJECTING ELIGIBLE APPLICANTS.

APPROACH

1. DATA COLLECTION AND UNDERSTANDING
2. DATA PREPARATION
3. EXPLORATORY DATA ANALYSIS (EDA)
4. FEATURE ENGINEERING
5. CORRELATION ANALYSIS
6. RISK ANALYTICS
7. REPORTING AND RECOMMENDATIONS

TECH-STACK USED

JUPYTER NOTEBOOK

PYTHON

MICROSOFT EXCEL

INSIGHTS

CREDIT HISTORY LIMITATIONS:

- URBAN CUSTOMERS WITH INSUFFICIENT OR NON-EXISTENT CREDIT HISTORIES POSE CHALLENGES FOR LOAN PROVIDERS.

KEY VARIABLES:

- VARIABLES SUCH AS LOAN AMOUNT, INCOME, CREDIT SCORE, AND EMPLOYMENT HISTORY ARE CRUCIAL FOR RISK ASSESSMENT.

DEMOGRAPHIC FACTORS:

- DEMOGRAPHIC FACTORS LIKE AGE, LOCATION, AND OCCUPATION MAY IMPACT LOAN REPAYMENT CAPABILITY.

RISK FACTORS IDENTIFICATION:

- THROUGH EXPLORATORY ANALYSIS, POTENTIAL RISK FACTORS ASSOCIATED WITH LOAN DEFAULTS CAN BE IDENTIFIED.

PROBLEM STATEMENT

WHEN THE COMPANY RECEIVES A LOAN APPLICATION, THE COMPANY HAS TO DECIDE FOR LOAN APPROVAL BASED ON THE APPLICANT'S PROFILE. TWO TYPES OF RISKS ARE ASSOCIATED WITH THE BANK'S DECISION:

IF THE APPLICANT IS LIKELY TO REPAY THE LOAN, THEN NOT APPROVING THE LOAN RESULTS IN A LOSS OF BUSINESS TO THE COMPANY.

IF THE APPLICANT IS NOT LIKELY TO REPAY THE LOAN, I.E. HE/SHE IS LIKELY TO DEFAULT, THEN APPROVING THE LOAN MAY LEAD TO A FINANCIAL LOSS FOR THE COMPANY.

SOLUTIONS

IMPORTING DATA SETS

In a Jupyter Notebook, the provided code snippets are employed to import data from two distinct files, specifically "previous_application.csv" and "application_data.csv".

```
In [3]: previous_application=pd.read_csv('E:/VCET/Trainity Intern/Project 6 Bank Loan Case Study/previous_application.csv')
        application_data=pd.read_csv('E:/VCET/Trainity Intern/Project 6 Bank Loan Case Study/application_data.csv')
```

DATA INSIGHTS

```
In [4]: previous_application.shape
Out[4]: (1670214, 37)

In [5]: application_data.shape
Out[5]: (307511, 122)
```

The previous_application data set consists of 1670214 rows and 37 columns.

The application_data data set consists of 307511 rows and 122 columns.

```
In [6]: previous_application.info(verbose=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1670214 entries, 0 to 1670213
Data columns (total 37 columns):
 #   Column                       Non-Null Count    Dtype
---  ------                       --------------    -----
 0   SK_ID_PREV                   1670214 non-null  int64
 1   SK_ID_CURR                   1670214 non-null  int64
 2   NAME_CONTRACT_TYPE           1670214 non-null  object
 3   AMT_ANNUITY                  1297979 non-null  float64
 4   AMT_APPLICATION              1670214 non-null  float64
 5   AMT_CREDIT                   1670213 non-null  float64
 6   AMT_DOWN_PAYMENT             774370 non-null   float64
 7   AMT_GOODS_PRICE              1284699 non-null  float64
 8   WEEKDAY_APPR_PROCESS_START   1670214 non-null  object
 9   HOUR_APPR_PROCESS_START      1670214 non-null  int64
 10  FLAG_LAST_APPL_PER_CONTRACT  1670214 non-null  object
 11  NFLAG_LAST_APPL_IN_DAY       1670214 non-null  int64
 12  RATE_DOWN_PAYMENT            774370 non-null   float64
 13  RATE_INTEREST_PRIMARY        5951 non-null     float64
 14  RATE_INTEREST_PRIVILEGED     5951 non-null     float64
 15  NAME_CASH_LOAN_PURPOSE       1670214 non-null  object
 16  NAME_CONTRACT_STATUS         1670214 non-null  object
 17  DAYS_DECISION                1670214 non-null  int64
 18  NAME_PAYMENT_TYPE            1670214 non-null  object
```

```
In [7]: application_data.info(verbose=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Data columns (total 122 columns):
 #    Column                       Dtype
---   ------                       -----
 0    SK_ID_CURR                   int64
 1    TARGET                       int64
 2    NAME_CONTRACT_TYPE           object
 3    CODE_GENDER                  object
 4    FLAG_OWN_CAR                 object
 5    FLAG_OWN_REALTY              object
 6    CNT_CHILDREN                 int64
 7    AMT_INCOME_TOTAL             float64
 8    AMT_CREDIT                   float64
 9    AMT_ANNUITY                  float64
 10   AMT_GOODS_PRICE              float64
 11   NAME_TYPE_SUITE              object
 12   NAME_INCOME_TYPE             object
```

The information regarding the column name, number of non-null values, count, and data types of the previous_application and application_data files is presented.

```
In [9]: application_data.describe()
```

Out[9]:

	SK_ID_CURR	TARGET	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_PRICE	REGION_POPULATION_RELATIV
count	307511.000000	307511.000000	307511.000000	3.075110e+05	3.075110e+05	307499.000000	3.072330e+05	307511.0000(
mean	278180.518577	0.080729	0.417052	1.687979e+05	5.990260e+05	27108.573909	5.383962e+05	0.0208(
std	102790.175348	0.272419	0.722121	2.371231e+05	4.024908e+05	14493.737315	3.694465e+05	0.0138:
min	100002.000000	0.000000	0.000000	2.565000e+04	4.500000e+04	1615.500000	4.050000e+04	0.0002!
25%	189145.500000	0.000000	0.000000	1.125000e+05	2.700000e+05	16524.000000	2.385000e+05	0.0100(
50%	278202.000000	0.000000	0.000000	1.471500e+05	5.135310e+05	24903.000000	4.500000e+05	0.0188!
75%	367142.500000	0.000000	1.000000	2.025000e+05	8.086500e+05	34596.000000	6.795000e+05	0.0286(
max	456255.000000	1.000000	19.000000	1.170000e+08	4.050000e+06	258025.500000	4.050000e+06	0.0725(

```
In [8]: previous_application.describe()
```

Out[8]:

	SK_ID_PREV	SK_ID_CURR	AMT_ANNUITY	AMT_APPLICATION	AMT_CREDIT	AMT_DOWN_PAYMENT	AMT_GOODS_PRICE	HOUR_APPR_PROCESS_STA
count	1.670214e+06	1.670214e+06	1.297979e+06	1.670214e+06	1.670213e+06	7.743700e+05	1.284699e+06	1.670214e+
mean	1.923089e+06	2.783572e+05	1.595512e+04	1.752339e+05	1.961140e+05	6.697402e+03	2.278473e+05	1.248418e+
std	5.325980e+05	1.028148e+05	1.478214e+04	2.927798e+05	3.185746e+05	2.092150e+04	3.153966e+05	3.334028e+
min	1.000001e+06	1.000010e+05	0.000000e+00	0.000000e+00	0.000000e+00	-9.000000e-01	0.000000e+00	0.000000e+
25%	1.461857e+06	1.893290e+05	6.321780e+03	1.872000e+04	2.416050e+04	0.000000e+00	5.084100e+04	1.000000e+
50%	1.923110e+06	2.787145e+05	1.125000e+04	7.104600e+04	8.054100e+04	1.638000e+03	1.123200e+05	1.200000e+
75%	2.384280e+06	3.675140e+05	2.065842e+04	1.803600e+05	2.164185e+05	7.740000e+03	2.340000e+05	1.500000e+
max	2.845382e+06	4.562550e+05	4.180581e+05	6.905160e+06	6.905160e+06	3.060045e+06	6.905160e+06	2.300000e+

The descriptive statistics including count, mean, standard deviation, minimum value, 25th percentile, 50th percentile, 75th percentile, and maximum value for both previous_application and application_data files are computed using the describe function.

PERCENTAGE OF MISSING VALUES

```
In [12]: round(previous_application.isnull().sum()/previous_application.shape[0]*100,2).sort_values(ascending=False)
```

```
Out[12]: RATE_INTEREST_PRIVILEGED       99.64
         RATE_INTEREST_PRIMARY          99.64
         RATE_DOWN_PAYMENT              53.64
         AMT_DOWN_PAYMENT               53.64
         NAME_TYPE_SUITE                49.12
         NFLAG_INSURED_ON_APPROVAL      40.30
         DAYS_FIRST_DRAWING             40.30
         DAYS_FIRST_DUE                 40.30
         DAYS_LAST_DUE_1ST_VERSION      40.30
         DAYS_LAST_DUE                  40.30
         DAYS_TERMINATION               40.30
         AMT_GOODS_PRICE                23.08
         AMT_ANNUITY                    22.29
         CNT_PAYMENT                    22.29
         PRODUCT_COMBINATION             0.02
         CHANNEL_TYPE                    0.00
         NAME_PRODUCT_TYPE               0.00
         NAME_YIELD_GROUP                0.00
         SELLERPLACE_AREA                0.00
```

```
In [13]: round(application_data.isnull().sum()/application_data.shape[0]*100,2).sort_values(ascending=False)
```

```
Out[13]: COMMONAREA_MEDI                69.87
         COMMONAREA_AVG                 69.87
         COMMONAREA_MODE                69.87
         NONLIVINGAPARTMENTS_MODE       69.43
         NONLIVINGAPARTMENTS_AVG        69.43
         NONLIVINGAPARTMENTS_MEDI       69.43
         FONDKAPREMONT_MODE             68.39
         LIVINGAPARTMENTS_MODE          68.35
         LIVINGAPARTMENTS_AVG           68.35
         LIVINGAPARTMENTS_MEDI          68.35
         FLOORSMIN_AVG                  67.85
         FLOORSMIN_MODE                 67.85
         FLOORSMIN_MEDI                 67.85
         YEARS_BUILD_MEDI               66.50
         YEARS_BUILD_MODE               66.50
         YEARS_BUILD_AVG                66.50
         OWN_CAR_AGE                    65.99
         LANDAREA_MEDI                  59.38
         LANDAREA_MODE                  59.38
```

The calculation of the percentage of missing values in each column of the application_data and previous_application data files has been performed.

DROPPING OF COLUMNS

```
In [14]: application_data_up=application_data.loc[: ,application_data.isnull().mean()<=0.45]
```

Dropping columns of application_data whose missing value percentage is grater than or equal to 45%.

```
In [28]: not_required=['FLAG_DOCUMENT_21','FLAG_DOCUMENT_20','FLAG_DOCUMENT_19','FLAG_DOCUMENT_18','FLAG_DOCUMENT_17','FLAG_DOCUMENT_16',
                       'FLAG_DOCUMENT_15','FLAG_DOCUMENT_14','FLAG_DOCUMENT_13','FLAG_DOCUMENT_12','FLAG_DOCUMENT_11',
                       'FLAG_DOCUMENT_10','FLAG_DOCUMENT_9','FLAG_DOCUMENT_8','FLAG_DOCUMENT_7','FLAG_DOCUMENT_6','FLAG_DOCUMENT_5',
                       'FLAG_DOCUMENT_4','FLAG_DOCUMENT_3','FLAG_DOCUMENT_2','OBS_30_CNT_SOCIAL_CIRCLE','DEF_30_CNT_SOCIAL_CIRCLE',
                       'OBS_60_CNT_SOCIAL_CIRCLE','DEF_60_CNT_SOCIAL_CIRCLE','AMT_REQ_CREDIT_BUREAU_YEAR','AMT_REQ_CREDIT_BUREAU_MON',
                       'AMT_REQ_CREDIT_BUREAU_WEEK','AMT_REQ_CREDIT_BUREAU_DAY','AMT_REQ_CREDIT_BUREAU_HOUR','AMT_REQ_CREDIT_BUREAU_QRT']

In [29]: application_data_up.drop(labels=not_required,axis=1,inplace=True)

In [30]: application_data_up.shape

Out[30]: (307511, 43)
```

Dropping columns of application_data which are not required for analysis.

```
In [138]: req_columns=['SK_ID_CURR','AMT_APPLICATION','NAME_CASH_LOAN_PURPOSE','NAME_CONTRACT_STATUS','DAYS_DECISION','NAME_PAYMENT_TYPE',
                       'CODE_REJECT_REASON','NAME_CLIENT_TYPE','NAME_GOODS_CATEGORY','NAME_PORTFOLIO','NAME_PRODUCT_TYPE','CHANNEL_TYPE',
                       'NAME_YIELD_GROUP']
          previous_application=previous_application.loc[:,req_columns]

In [139]: previous_application.shape

Out[139]: (1670214, 13)
```

Keeping only the required columns and deleting the other columns of previous_applicaation

IMPUTING DATA

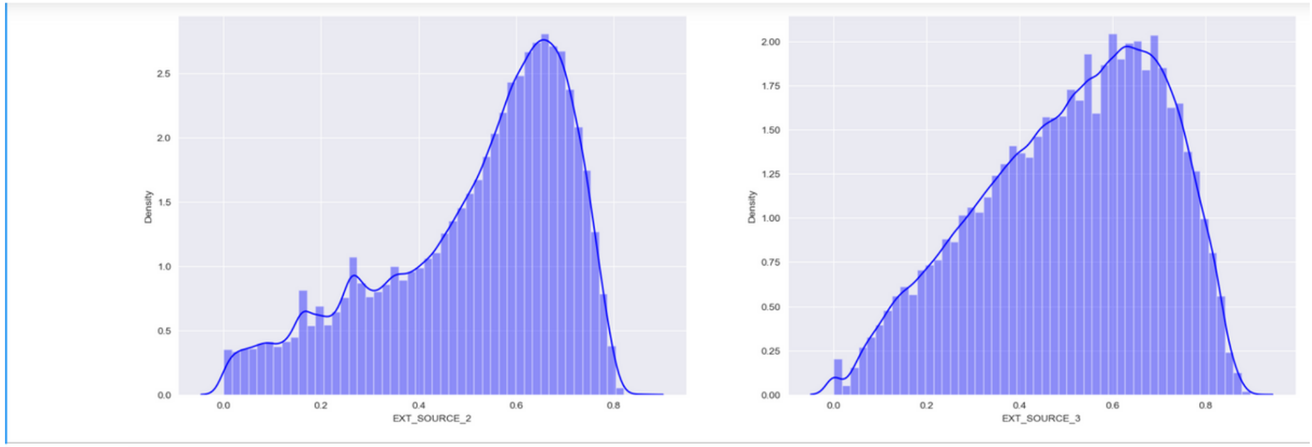
```
In [23]: application_data_up["OCCUPATION_TYPE"].isnull().sum()
Out[23]: 96391

In [24]: application_data_up["OCCUPATION_TYPE"].replace(np.NaN, "unknown", inplace=True)
```

To address the 96391 null values in the OCCUPATION_TYPE column, the plan is to substitute them with the term "unknown."

Correlation between the loan amount and the price of goods for which loan was given


```
In [36]: # Imputing the above mentioned logic

application_data_up['AMT_GOODS_PRICE']=np.where(application_data_up['AMT_GOODS_PRICE'].isnull()==True,
                                        application_data_up['AMT_CREDIT'],application_data_up['AMT_GOODS_PRICE'])
```

The scatter plot reveals a positive correlation between the AMT_GOODS_PRICE and AMT_CREDIT variables. Consequently, the decision is made to fill the null values in the AMT_GOODS_PRICE column with the corresponding values from the AMT_CREDIT column.




```
In [34]: #REPLACING THE MISSING VALUES WITH ITS CORRESPONDING MEDIAN
         for column in ['EXT_SOURCE_2','EXT_SOURCE_3']:
             application_data_up[column].fillna(application_data_up[column].median(), inplace=True)
```

Both EXT_SOURCE_2 and EXT_SOURCE_3 exhibit a right-skewed distribution in their respective graphs, the decision is made to substitute their values with the median.

```
In [41]: application_data_up['NAME_TYPE_SUITE'].isnull().sum()

Out[41]: 1292

In [42]: application_data_up['NAME_TYPE_SUITE'].mode()

Out[42]: 0    Unaccompanied
         Name: NAME_TYPE_SUITE, dtype: object

In [43]: #Replacing missing values with MODE

         application_data_up['NAME_TYPE_SUITE'].fillna(application_data_up['NAME_TYPE_SUITE'].mode()[0],inplace=True)
```

Since the number of null values int NAME_TYPE_SUITE column is very less we are imputing it with it's mode.

In [45]:
```
#REMAINING COLUMNS WITH NEGLIGIBLE NULL VALUES (LESS THAN 1%)

null_col=['CNT_FAM_MEMBERS','AMT_ANNUITY', 'DAYS_LAST_PHONE_CHANGE']
for column in null_col:
    application_data_up[column].fillna(application_data_up[column].median(), inplace=True)
```

Other columns that has negligible null values are replaced with it's corresponding median.

In [147]:
```
#CHANGING XNA TO UNKNOWN

XNA_col=['NAME_PAYMENT_TYPE','NAME_CLIENT_TYPE','NAME_PORTFOLIO']

for i in XNA_col:
    previous_application[i]=previous_application[i].str.replace('XMA','unknown')
```

In the previous_application dataset, the columns that contain null values are substituted with the term "UNKNOWN".

CHANGING DATA TYPES

```
In [48]: dayandcount=['CNT_FAM_MEMBERS','DAYS_REGISTRATION','DAYS_LAST_PHONE_CHANGE']
         application_data_up.loc[:, dayandcount]=application_data_up.loc[:,dayandcount].apply(lambda x: x.astype('int64',errors='ignore'))
```

The data type of the CNT_FAM_MEMBERS, DAYS_REGISTRATION, and DATA_LAST_PHONE_CHANGE columns is converted to integer.

```
In [50]: #LISTING OBJECT TYPE COLUMNS AND CONFIRMING THE VALUES TO BE IN STRING TYPE

         obj_col=list(application_data_up.select_dtypes(include='object').columns)

         application_data_up.loc[:,obj_col]=application_data_up.loc[:,obj_col].apply(lambda x: x.astype('str'))
```

After listing the columns of object type, it is verified that the values within those columns are predominantly in string data type. In the case where any value is not in string data type, it is subsequently converted to a string.

```
In [116]: #Creating two datasets for target=1 and target=0 (1=bad,0=good)

target1=application_data_up[application_data_up['TARGET']==1]
target0=application_data_up[application_data_up['TARGET']==0]

print(target1.shape,target0.shape,application_data_up.shape)
```

```
(24825, 48) (282686, 48) (307511, 48)
```

```
In [117]: print("The dataset with Target value 1 has :" + "{:.2%}".format(target1.shape[0]/application_data_up.shape[0])+ "data.")
print("The dataset with Target value 0 has :" + "{:.2%}".format(target0.shape[0]/application_data_up.shape[0])+ "data.")
```

```
The dataset with Target value 1 has :8.07%data.
The dataset with Target value 0 has :91.93%data.
```

The given dataset is split into two parts based on the target values, namely Target value 1 and Target value 2.Percentage of people who paid their loan are: 91.93 %
Percentage of people who did not paid their loan are: 8.07 %.

```
imb_ratio = round(len(target0_df)/len(target1_df),2)

print('Imbalance Ratio:', imb_ratio)
```

```
Imbalance Ratio: 11.39
```

The imbalance ratio is 11.39

NEGATIVE TO POSITIVE CONVRSION

```
In [67]: application_data_up.DAYS_BIRTH.unique()
Out[67]: array([ -9461, -16765, -19046, ...,  -7951,  -7857, -25061], dtype=int64)

In [68]: application_data_up.DAYS_EMPLOYED.unique()
Out[68]: array([  -637,  -1188,   -225, ..., -12971, -11084,  -8694], dtype=int64)

In [69]: #CHECKING 'DAYS_REGISTRATION' COLUMN

         application_data_up.DAYS_REGISTRATION.unique()
Out[69]: array([ -3648,  -1186,  -4260, ..., -16396, -14558, -14798], dtype=int64)

In [70]: #CHECKING 'DAYS_ID_PUBLISH' COLUMN

         application_data_up.DAYS_ID_PUBLISH.unique()
Out[70]: array([-2120,  -291, -2531, ..., -6194, -5854, -6211], dtype=int64)

In [71]: #CHECKING 'DAYS_LAST_PHONE_CHANGE' COLUMN

         application_data_up.DAYS_LAST_PHONE_CHANGE.unique()
Out[71]: array([-1134,  -828,  -815, ..., -3988, -3899, -3538], dtype=int64)
```

To transform the negative values in the application_data file into positive values, the abs function is utilized.

```
In [73]: #changing values to positive integer

         num_days=['DAYS_BIRTH','DAYS_EMPLOYED','DAYS_REGISTRATION','DAYS_ID_PUBLISH','DAYS_LAST_PHONE_CHANGE']

         for i in num_days:
             application_data_up[i]=abs(application_data_up[i])
```

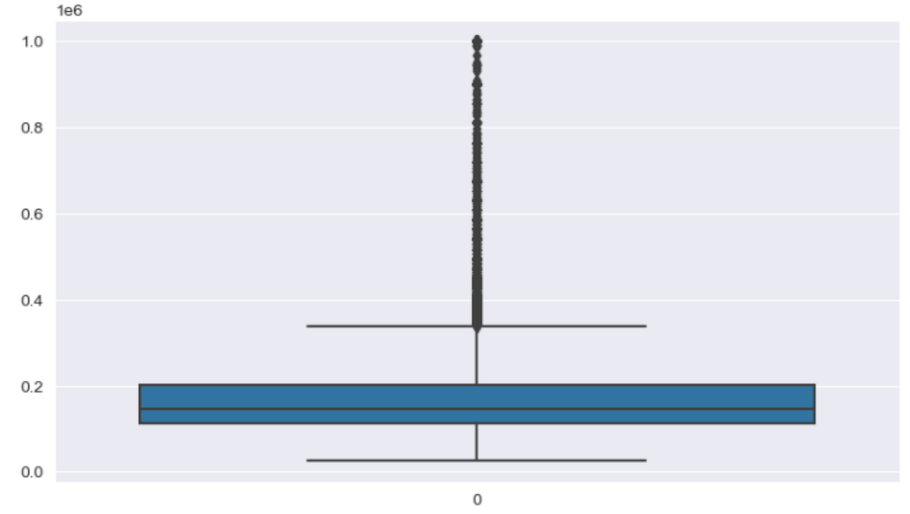
```
In [142]: previous_application.DAYS_DECISION.unique()
Out[142]: array([  -73,  -164,  -301, ..., -1967, -2389,    -1], dtype=int64)

In [143]: #changing values to positive integer

          previous_application['DAYS_DECISION']=abs(previous_application['DAYS_DECISION'])
```

The negative values in the DAYS_DECISION column are converted into positive values using the abs function.

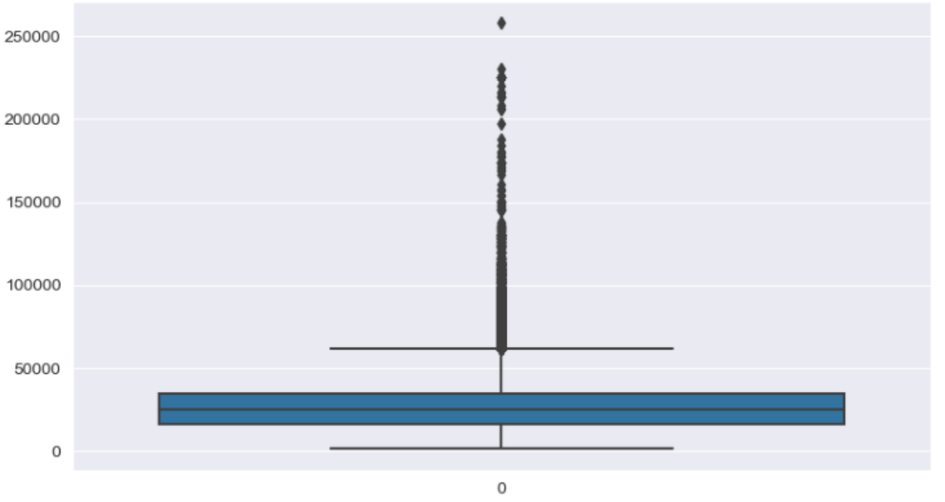
OUTLIERS

An outlier refers to a data point that significantly deviates from the expected pattern or distribution of a dataset. It can be either an unusually high or low value relative to the other data points. Outliers can greatly influence the decision-making process of an analyst by distorting statistical measures such as mean and standard deviation, impacting the accuracy of predictions and models. It is need to identify and handle outliers appropriately to ensure their impact is minimized and the insights drawn from the data are reliable and robust.

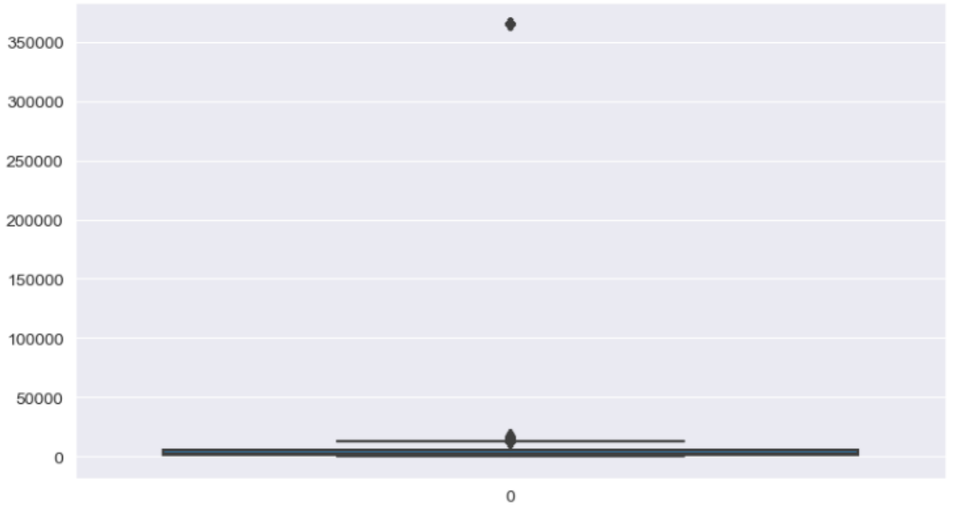
AMT_INCOME_TOTAL - BOX PLOT



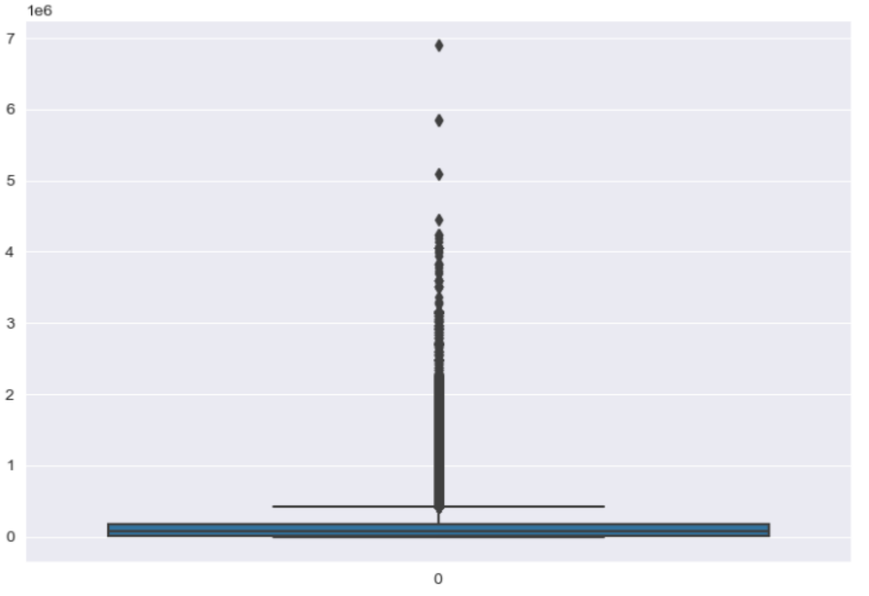
AMT_ANNUITY - BOX PLOT



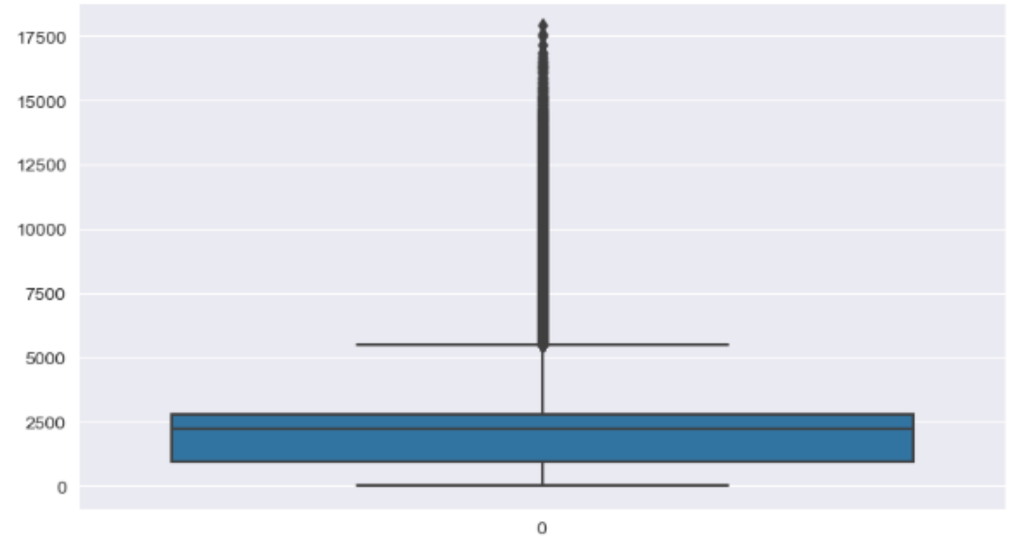
DAYS_EMPLOYED - BOX PLOT



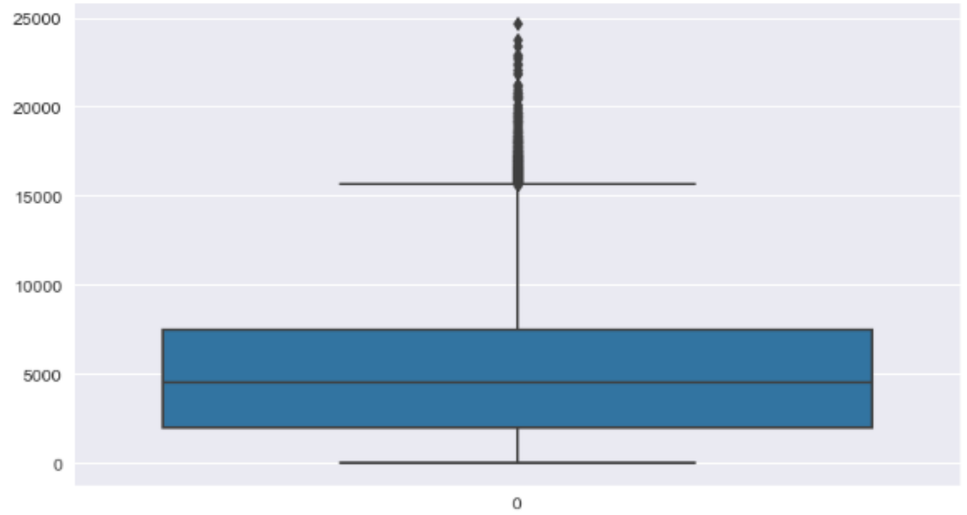
AMT_APPLICATION - BOXPLOT



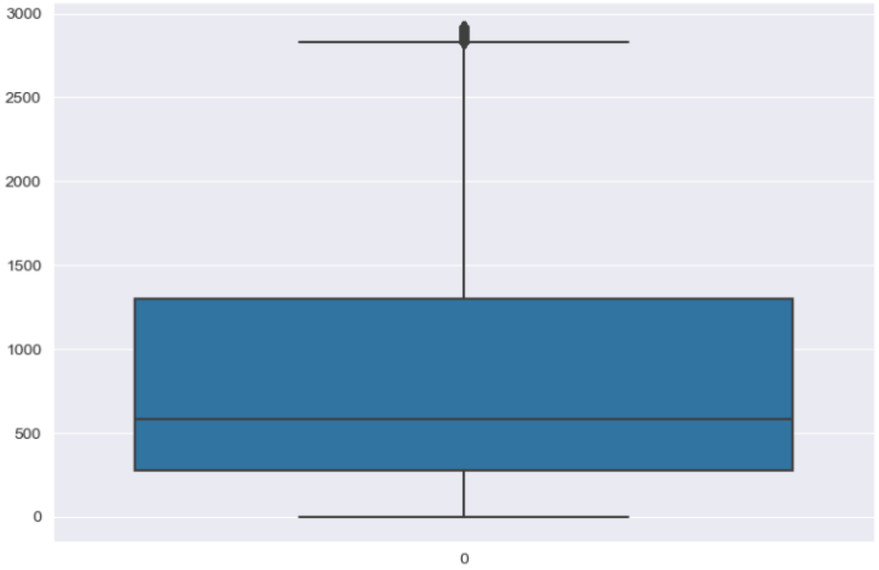
DAYS_EMPLOYED - BOX PLOT



DAYS_REGISTRATION - BOX PLOT



DAYS_DECISION - BOX PLOT



- In the AMT_INCOME_TOTAL column, there exists a single data point with a high value, indicating an outlier. Removing this outlier would have a significant impact on the box plot, causing notable changes in its distribution and potentially affecting subsequent analyses.
- For the variables AMT_ANNUITY, DAYS_EMPLOYED, and DAYS_REGISTRATION, the first quartile appears relatively smaller compared to the third quartile. This indicates that the data distribution is skewed towards the lower values of the first quartile.

BIN CREATION AND DERIVED METRICS

In [55]:
```
#Therefore, we eliminate the sub-category with the overall category

application_data_up.ORGANIZATION_TYPE = application_data_up.ORGANIZATION_TYPE.apply(lambda x: 'Industry' if 'Industry' in x else
application_data_up.ORGANIZATION_TYPE = application_data_up.ORGANIZATION_TYPE.apply(lambda x: 'Trade' if 'Trade' in x else x)
application_data_up.ORGANIZATION_TYPE = application_data_up.ORGANIZATION_TYPE.apply(lambda x: 'Transport' if 'Transport' in x els
application_data_up.ORGANIZATION_TYPE = application_data_up.ORGANIZATION_TYPE.apply(lambda x: 'Business' if 'Business' in x else
```

For the variables AMT_ANNUITY, DAYS_EMPLOYED, and DAYS_REGISTRATION, the first quartile appears relatively smaller compared to the third quartile. This indicates that the data distribution is skewed towards the lower values of the first quartile.

In [78]:
```
application_data_up['AMT_CREDIT_slab']=pd.qcut(application_data_up['AMT_CREDIT'], q=[0,0.2,0.5,0.75,0.95,1],
                                               labels=['VeryLow','Low','Medium','High','VeryHigh'])
```

In [76]:
```
application_data_up['INCOME_SLAB']=pd.qcut(application_data_up['AMT_INCOME_TOTAL'], q=[0,0.2,0.5,0.75,0.95,1],
                                           labels=['VeryLow','Low','Medium','High','VeryHigh'])
```

The AMT_INCOME_TOTAL and AMT_CREDIT columns are segmented into bins, which are then labeled as INCOME_SLAB and AMT_CREDIT_SLAB, respectively.

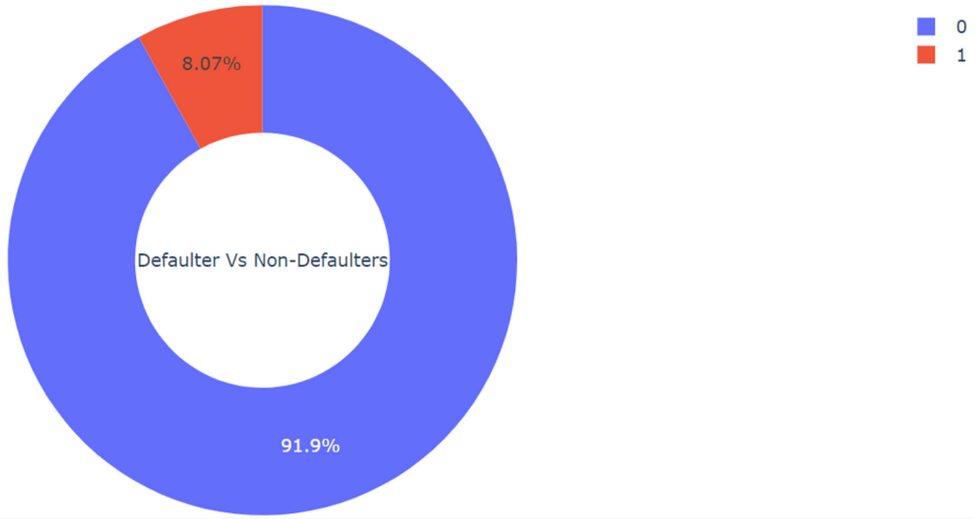
In [82]:
```
#CREATING 10 BINS

application_data_up['AGE_BINS']=pd.cut(application_data_up['AGE'],bins=np.arange(20,71,5))
```

The AGE column is divided into ten bins, referred to as AGE_BINS. The distribution of values within these bins is then presented using the value_counts function.

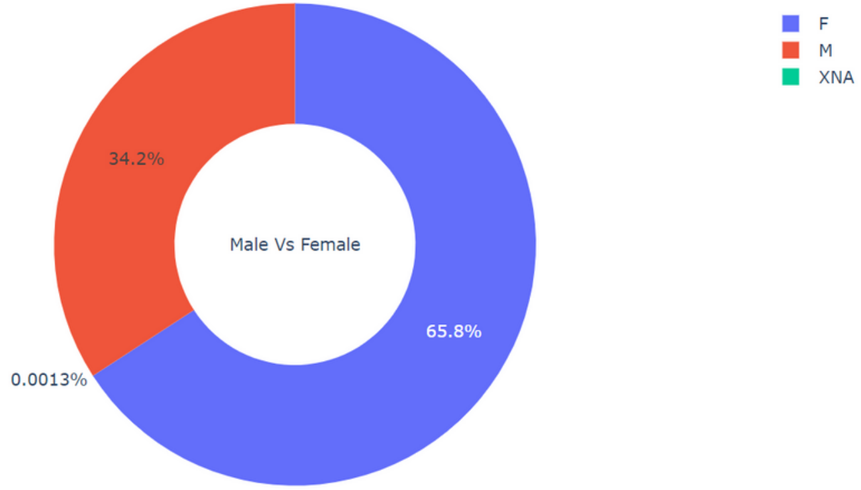
In [108]:
```
#CREATING ADDITIONAL COLUMNS FOR ANALYSIS
#CREATING A COLUMN WITH VALUES CREDIT TO INCOME RATIO - Derived Metrics

application_data_up['CREDIT_RATIO']=(application_data_up.AMT_CREDIT // application_data_up.AMT_INCOME_TOTAL).astype('int64')
```

To enhance analysis, a derived column called CREDIT_RATIO is generated by dividing the AMT_CREDIT by the AMT_INCOME_TOTAL. This new column provides a ratio that offers valuable insights into the relationship between credit and income.

DATA IMBALANCE



There is a significant data imbalance, where 91.9% of the data corresponds to Defaulter records, while only 8.07% of the data pertains to Non-Defaulters.



There is a significant data imbalance, where 65.8% of the data corresponds to Defaulter records, while only 34.2% of the data pertains to Non-Defaulters.

In [116]:
```
#Creating two datasets for target=1 and target=0 (1=bad,0=good)

target1=application_data_up[application_data_up['TARGET']==1]
target0=application_data_up[application_data_up['TARGET']==0]

print(target1.shape,target0.shape,application_data_up.shape)
```

```
(24825, 48) (282686, 48) (307511, 48)
```

In [117]:
```
print("The dataset with Target value 1 has :" + "{:.2%}".format(target1.shape[0]/application_data_up.shape[0])+ "data.")
print("The dataset with Target value 0 has :" + "{:.2%}".format(target0.shape[0]/application_data_up.shape[0])+ "data.")
```

```
The dataset with Target value 1 has :8.07%data.
The dataset with Target value 0 has :91.93%data.
```

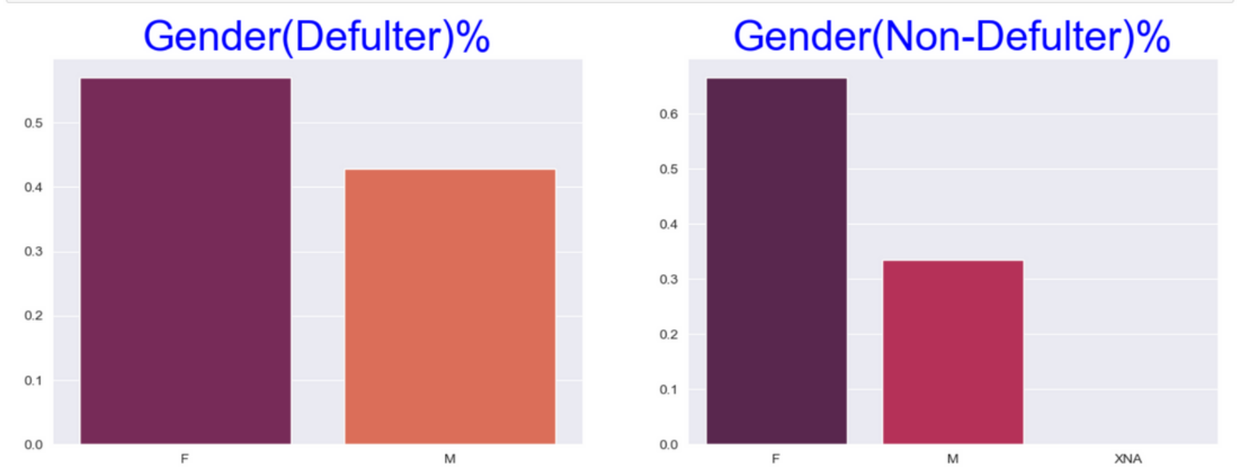
The dataset is split into two subsets, categorized by the target value of 0 and 1. The proportion of individuals who successfully paid their loan is 91.93%, while the proportion of those who did not fulfill their loan obligation is 8.07%.

```
imb_ratio = round(len(target0_df)/len(target1_df),2)

print('Imbalance Ratio:', imb_ratio)
```

```
Imbalance Ratio: 11.39
```

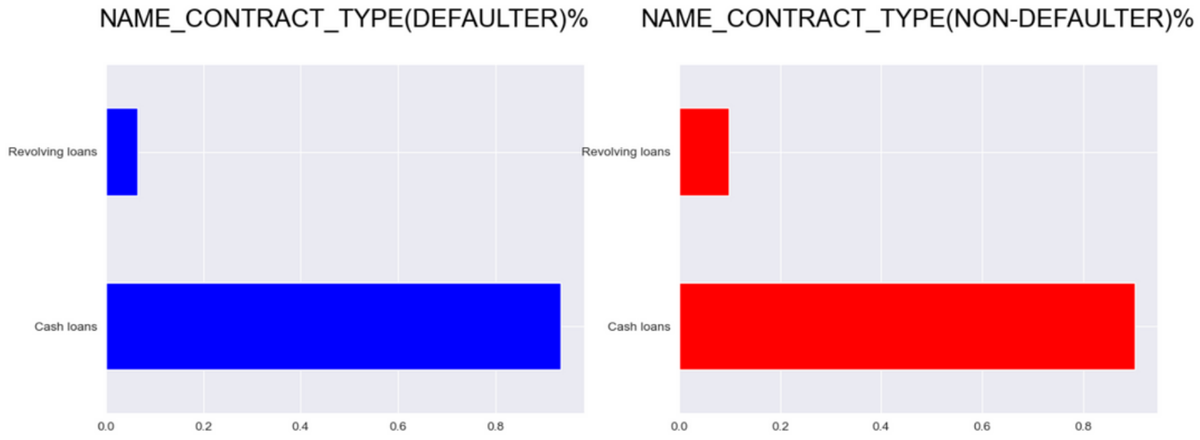
The imbalance ratio is 11.39

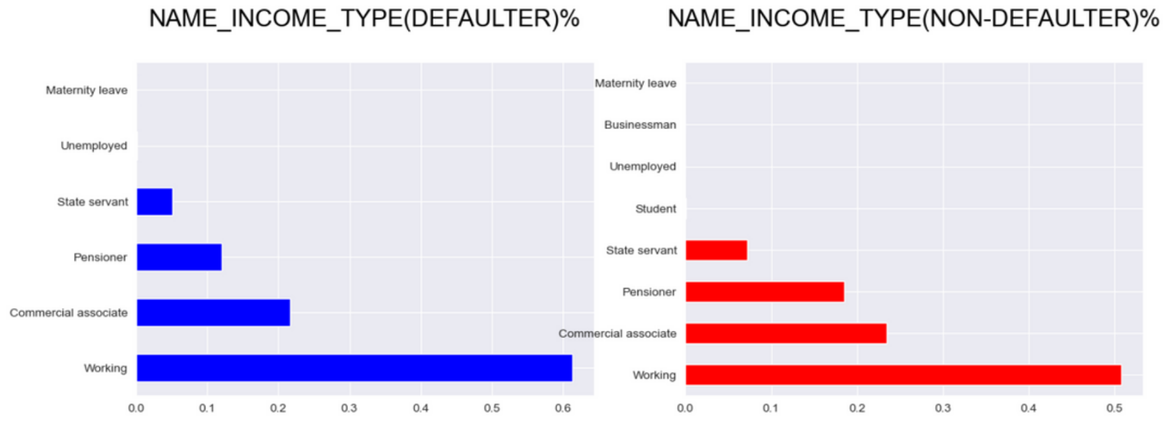
UNIVARIATE ANALYSIS



Men are more inclined to default on loans compared to women, as their default rate has risen by approximately 10%, whereas women have exhibited a corresponding decrease of 10% in their default rate.

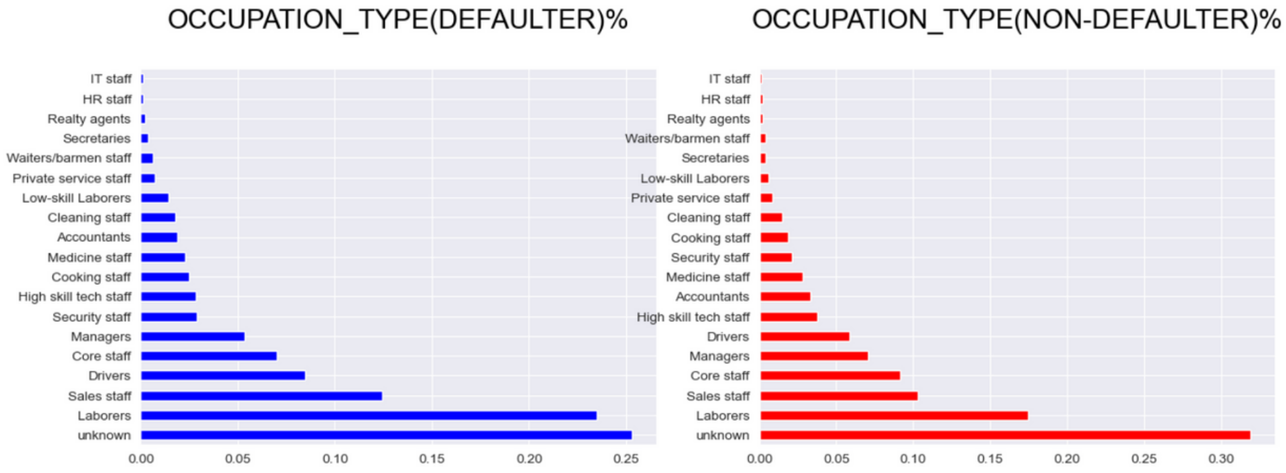
Based on the observed graphs, there appears to be a lack of significant distinction between the two, suggesting that the type of loan alone may not provide sufficient information to accurately predict whether a person will default or not.

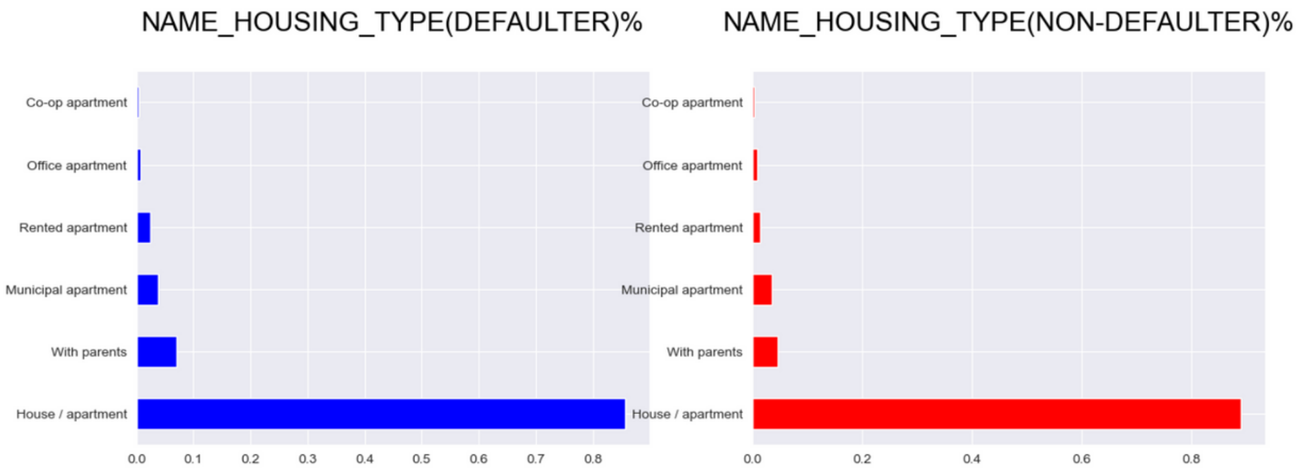




The analysis of the two graphs reveals a notable trend where individuals belonging to the "working" income type are found in higher proportions within both the defaulter and non-defaulter groups.

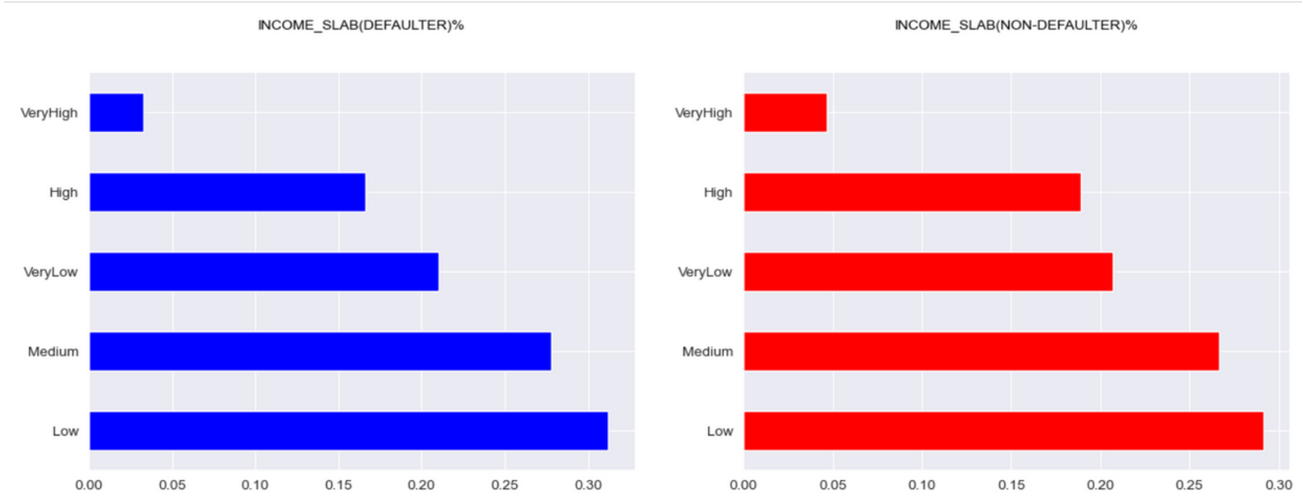
The graph indicates that individuals in occupations such as labours, sales staff, drivers, cleaning staff, and low-skill labours are more prone to default on loan payments. On the other hand, it is recommended to focus on targeting managers, core staff, and high-skill tech staff as they exhibit greater reliability as clients in this particular scenario.

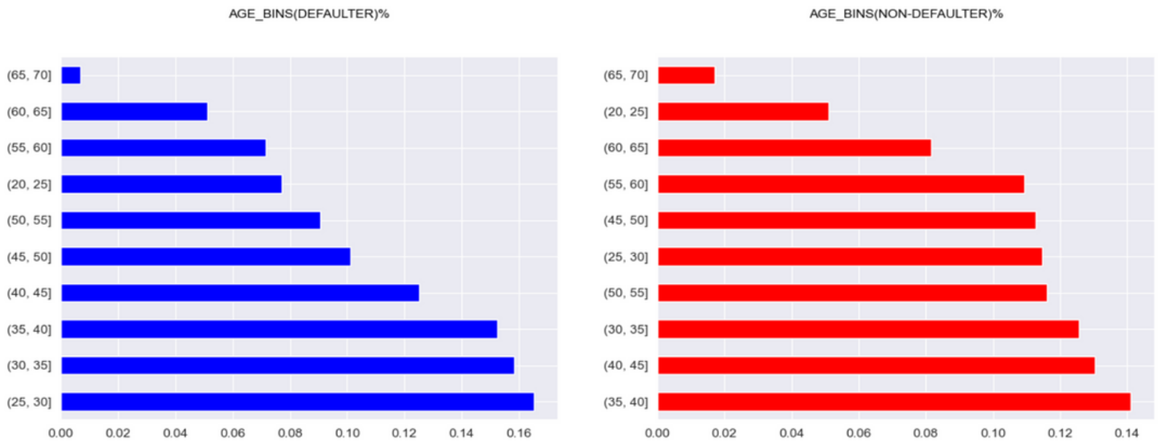




The "With Parents" category shows only a marginal change in percentage, indicating that clients who live with their parents may not be as financially established and could potentially face difficulties in loan repayment. On the other hand, individuals who own a house or apartment demonstrate a tendency to apply for loans more frequently.

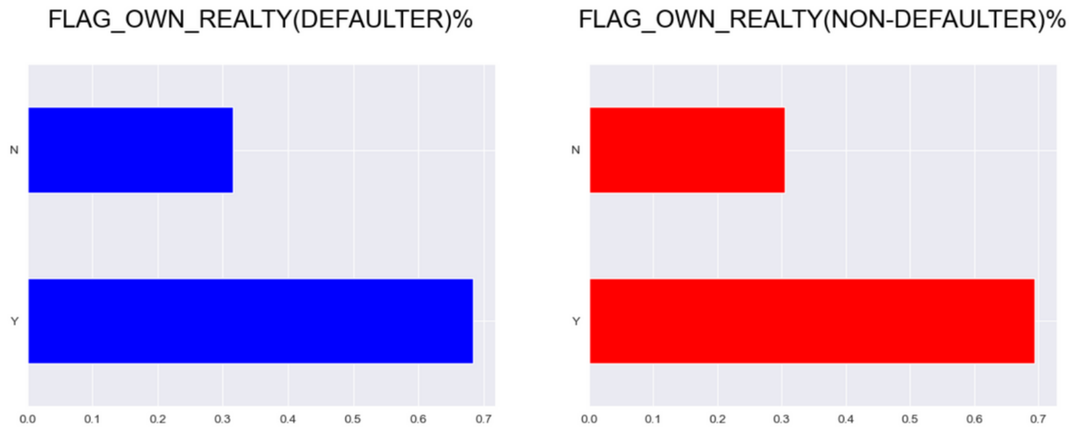
The data suggests that individuals who take out higher loan amounts are less likely to default. This can be attributed to the fact that clients who are wealthy or financially stable are more inclined to borrow larger sums. Conversely, when the loan amount is lower, it becomes apparent that individuals face challenges in repaying it.

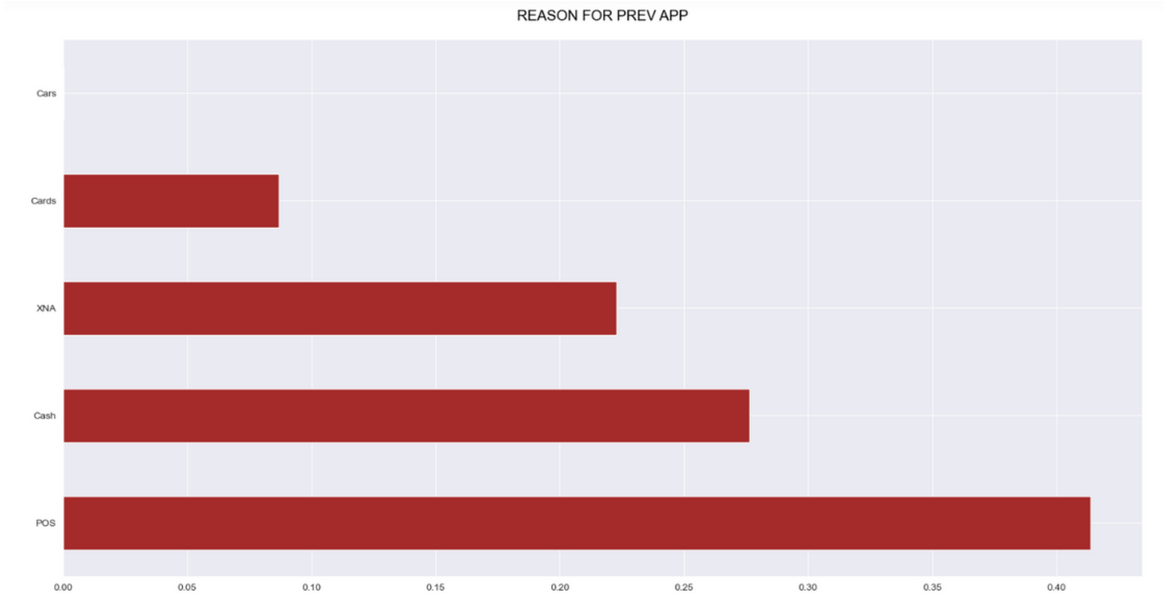




Based on the graph, it can be concluded that individuals within the age range of 25 to 40 are more prone to loan default. Conversely, those above 45 years of age are less likely to default. Additionally, there is a trend indicating that as the age group increases, the likelihood of loan default decreases.

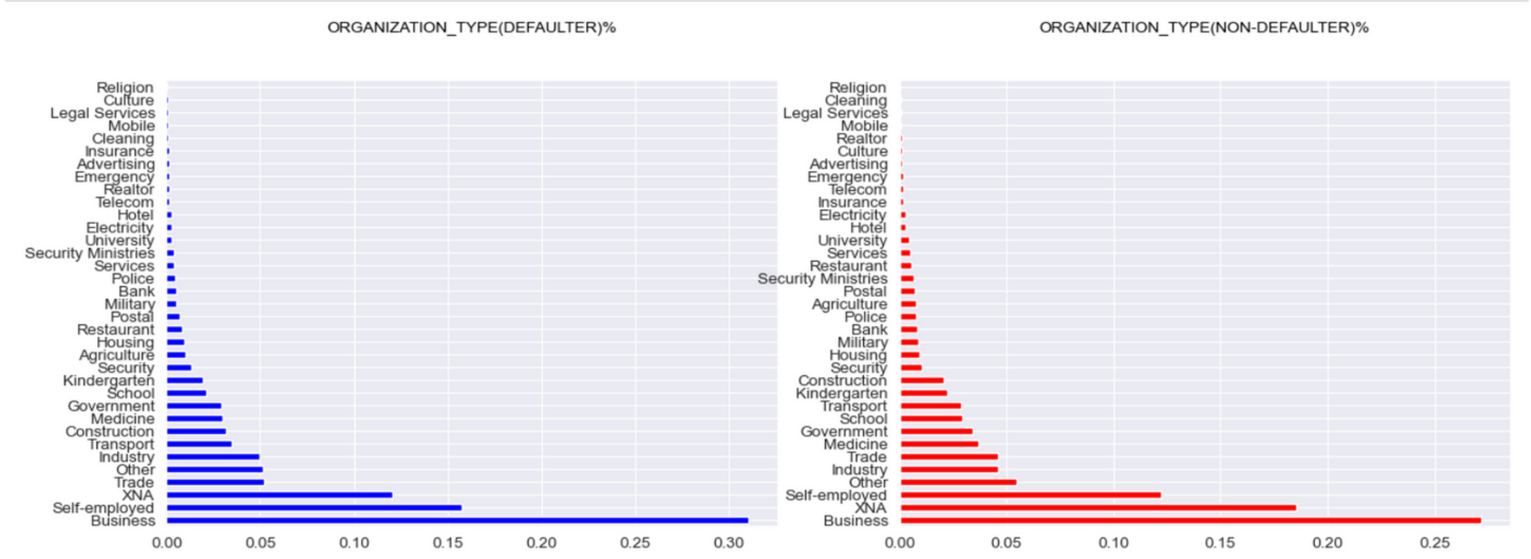
The graph indicates that the presence of real estate ownership does not strongly correlate with the likelihood of loan default. However, there is a subtle observation suggesting that individuals without a house may have a slightly increased chance of defaulting.





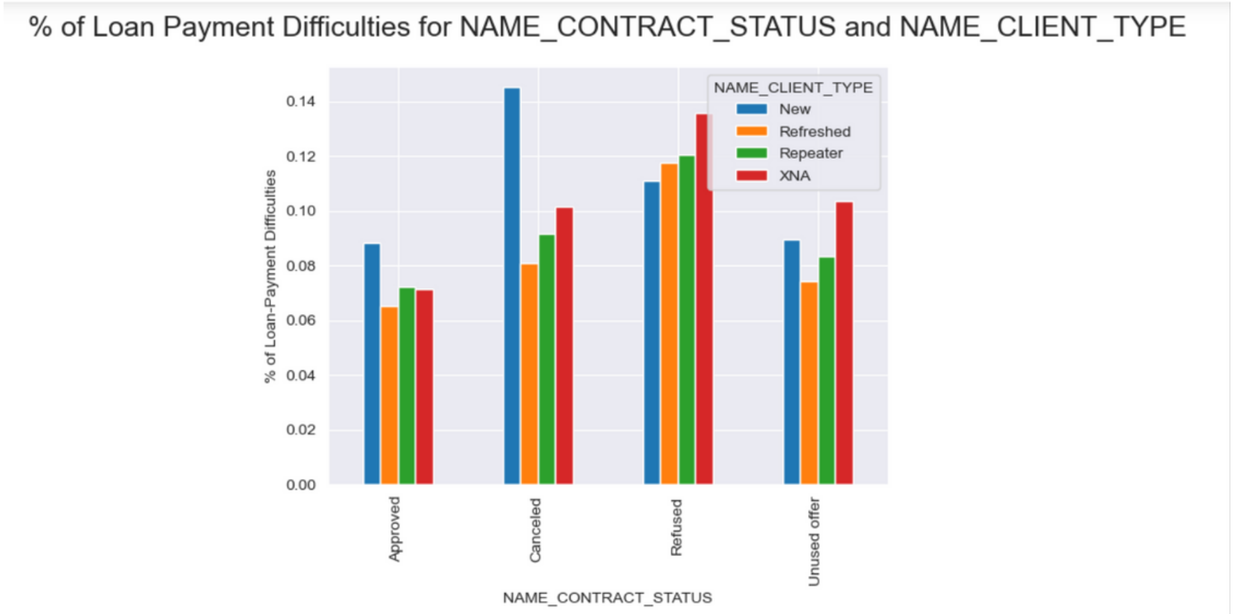
Based on the analysis of the graph, it can be deduced that individuals who are self-employed or engaged in trades, construction, security, transport, or business-related occupations are more prone to encountering challenges in loan repayment.

Based on the analysis of the graph, it can be deduced that individuals who are self-employed or engaged in trades, construction, security, transport, or business-related occupations are more prone to encountering challenges in loan repayment.



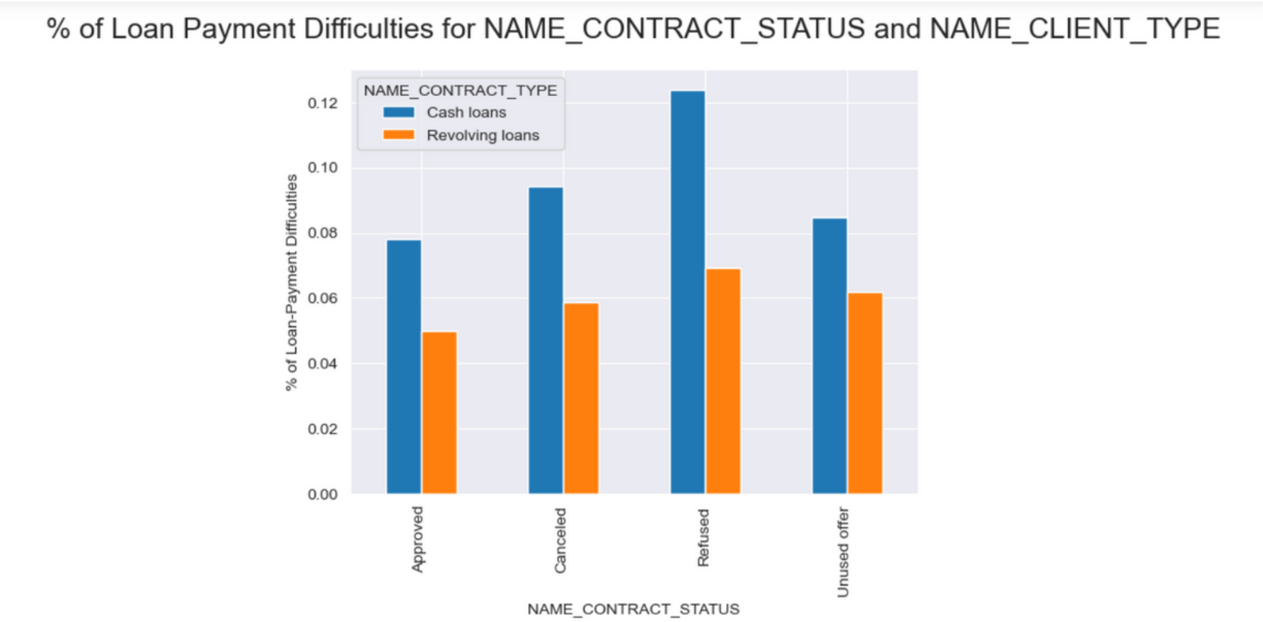
BIVARIATE ANALYSIS

NAME_CONTRACT_STATUS	Approved	Canceled	Refused	Unused offer
NAME_CLIENT_TYPE				
New	0.088216	0.145205	0.110940	0.089448
Refreshed	0.065158	0.081098	0.117412	0.074324
Repeater	0.072144	0.091767	0.120596	0.083338
XNA	0.071264	0.101377	0.135714	0.103448

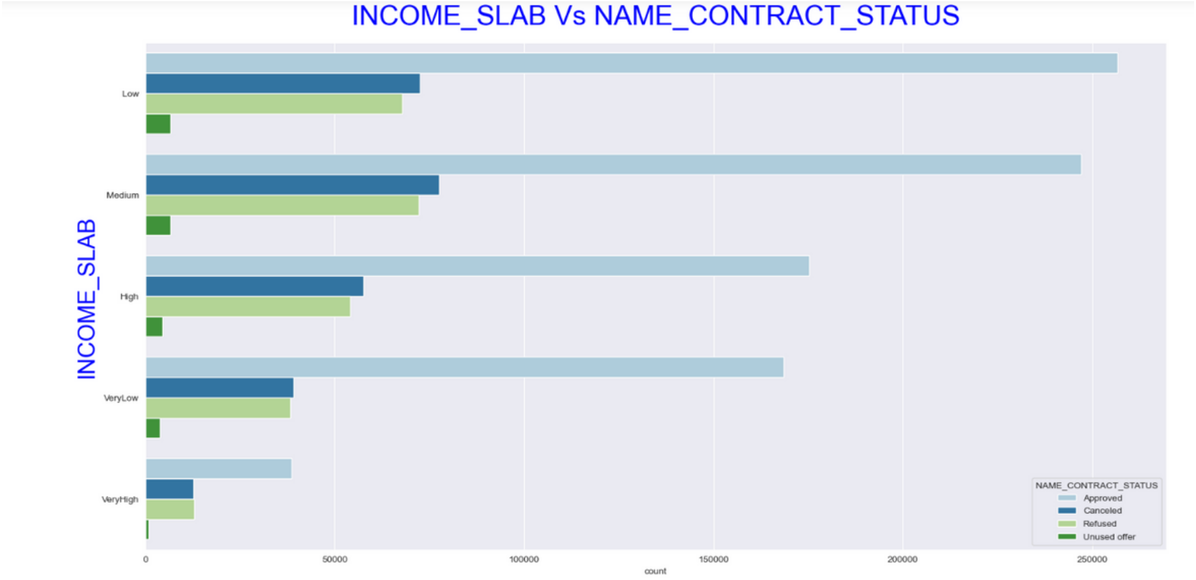


Based on the data and graph analysis, it can be deduced that new clients have a higher likelihood of loan cancellation. Additionally, new clients are more prone to having their loan amounts refused. Conversely, repeat clients exhibit a greater likelihood of loan refusal.

NAME_CONTRACT_STATUS	Approved	Canceled	Refused	Unused offer
NAME_CONTRACT_TYPE				
Cash loans	0.078105	0.094178	0.123735	0.084637
Revolving loans	0.049836	0.058751	0.069429	0.061972

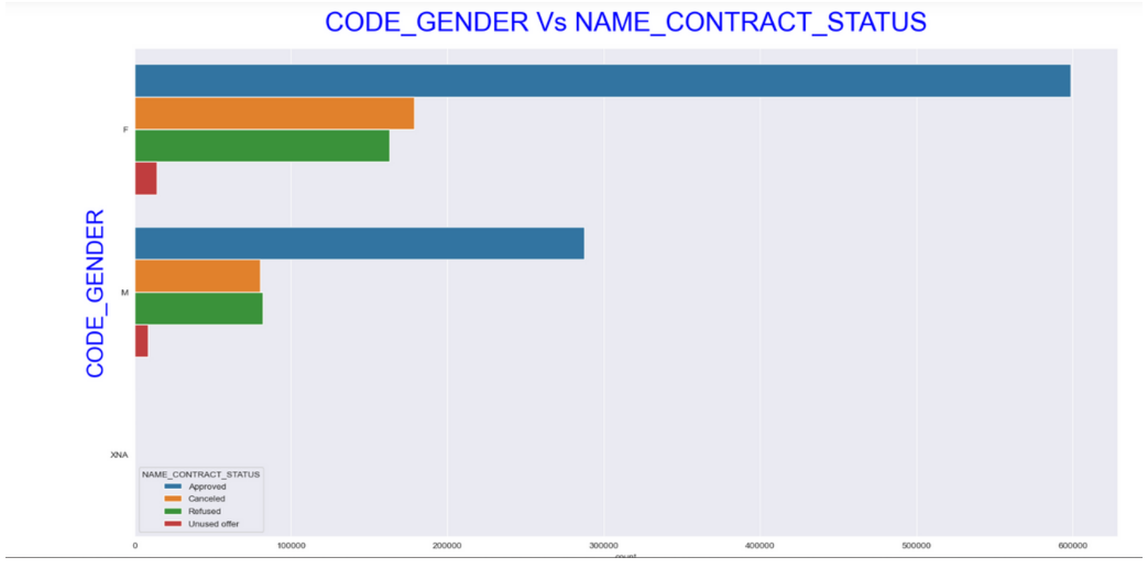


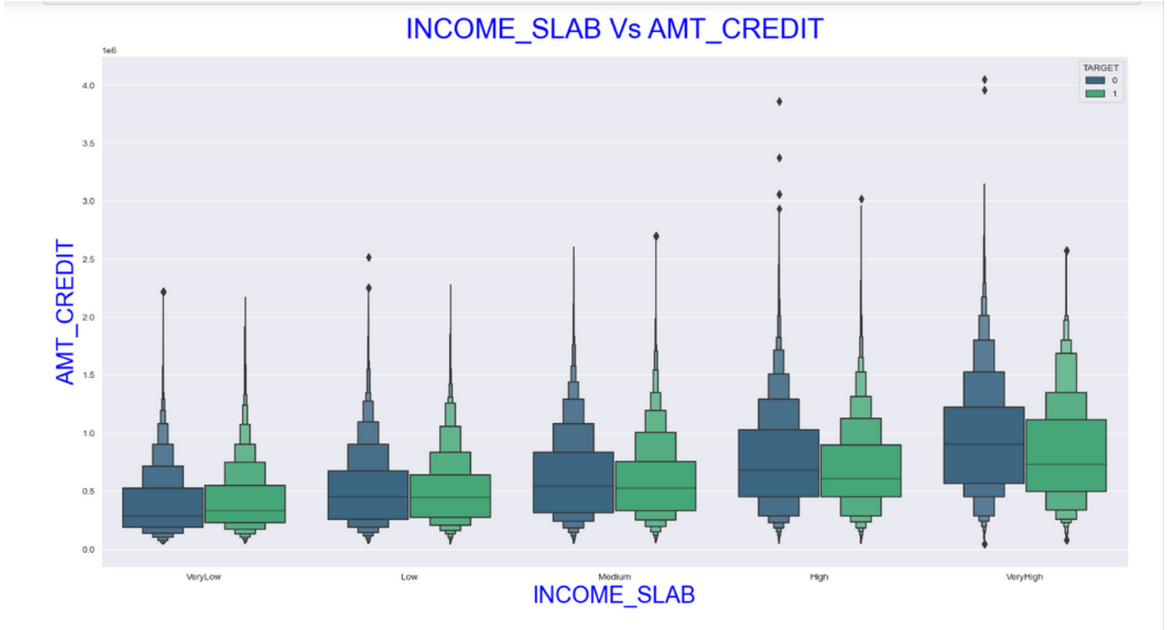
The graph indicates that the presence of real estate ownership does not strongly correlate with the likelihood of loan default. However, there is a subtle observation suggesting that individuals without a house may have a slightly increased chance of defaulting.



Based on the graph, it is apparent that individuals with extremely low or high income levels demonstrate a slightly higher likelihood of loan amount refusal, albeit with a minimal difference.

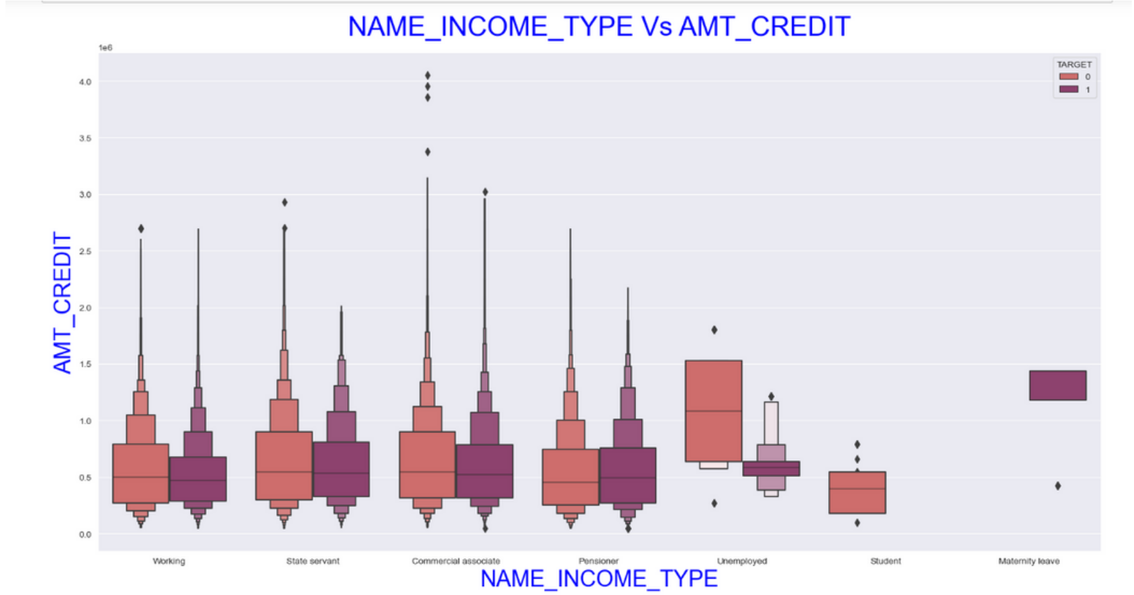
The data and graph indicate that new clients have a higher success rate in getting their loans approved. Conversely, repeater clients face a higher frequency of loan cancellations or refusals compared to new clients.



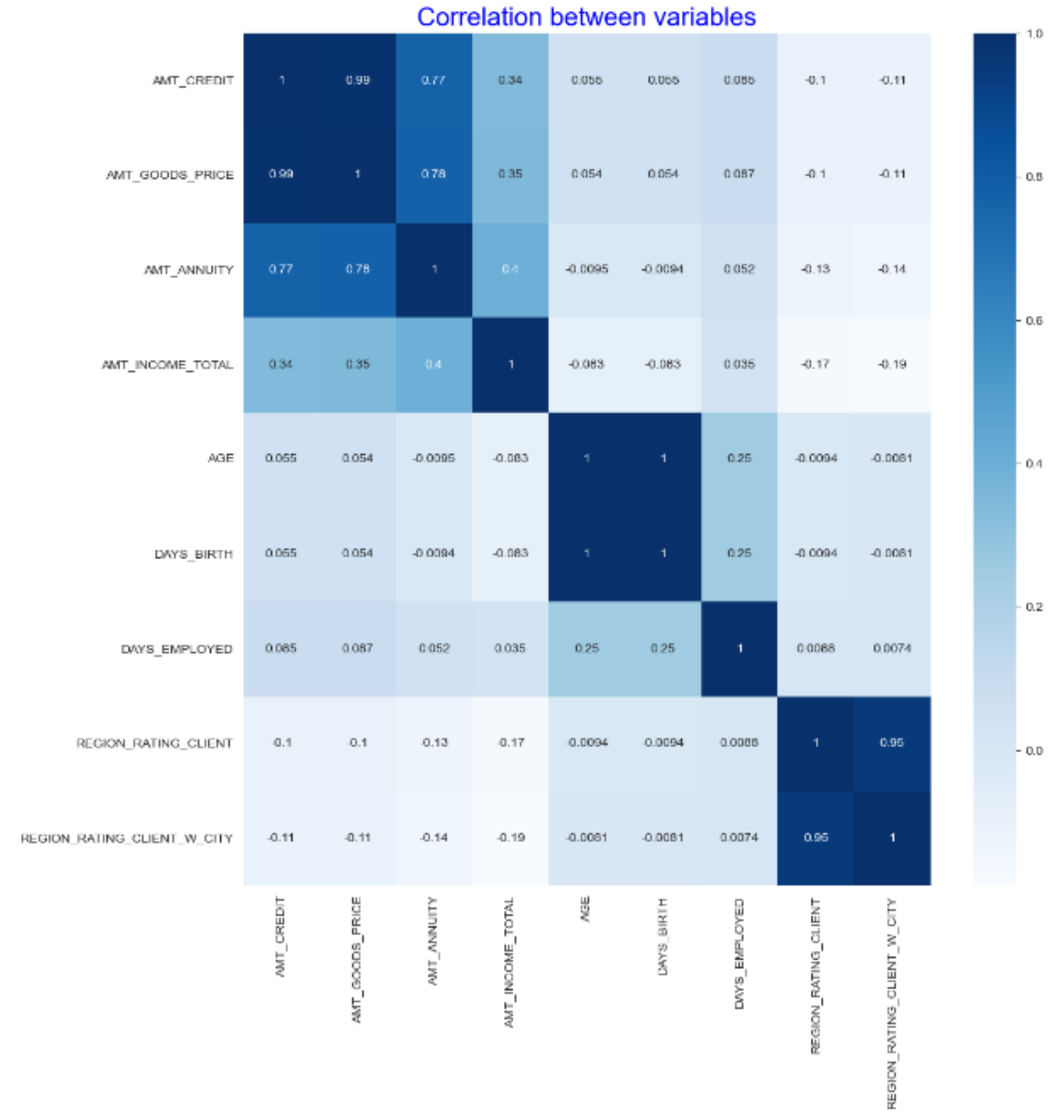


We can see that as the income slab increases, the credit amount of the loan also increases. We can also see that the people in the low and very low income slabs are likely to default more.

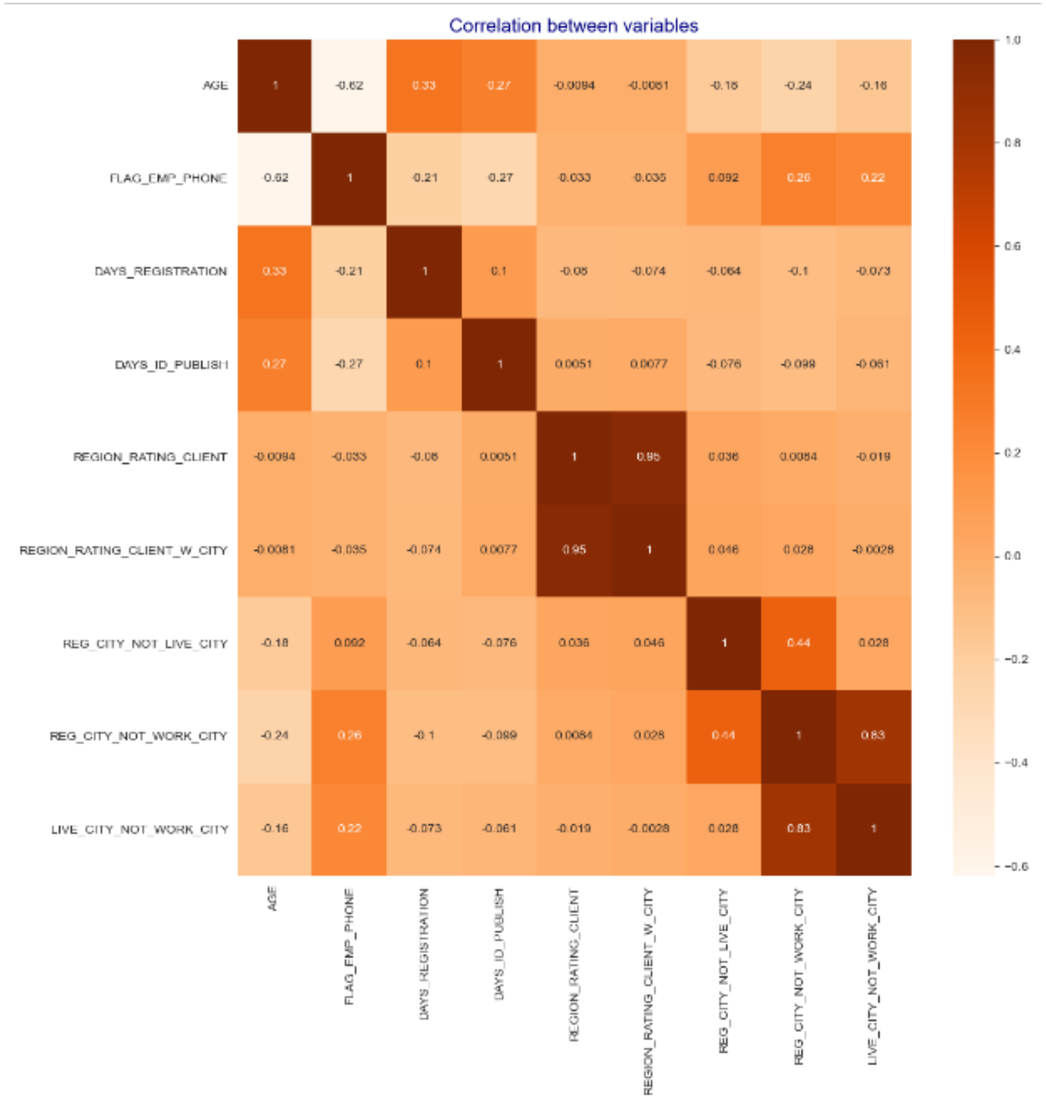
Based on the graph, it can be observed that individuals involved in business and those who are unemployed tend to request higher loan amounts. Interestingly, both groups exhibit better loan repayment behavior.



CORRELATION



From the observed data, a notable correlation can be observed between the price of goods and the loan amount, suggesting that the disbursed loan amount is generally equal to or slightly higher than the cost of the desired item. Furthermore, a significant association is apparent between the annuity amount and the loan amount, as well as the price of the goods. Moreover, there exists a negative correlation between the client's region and their income. This implies that individuals hailing from regions with higher ratings are more likely to have lower incomes.



A significant inverse relationship can be observed between an employee's phone number and their age. Conversely, a positive correlation exists between the number of days preceding a client's registration change and their age. These findings indicate that older individuals are less inclined to modify their registration details before applying for a loan. Moreover, clients who omit their phone numbers are also less prone to providing incorrect permanent and work addresses.

TOP 10 CORRELATED VARIABLES

Out[47]:

	Var1	Var2	Correlation
649	OBS_60_CNT_SOCIAL_CIRCLE	OBS_30_CNT_SOCIAL_CIRCLE	1.0000
184	AMT_GOODS_PRICE	AMT_CREDIT	0.9900
680	DEF_60_CNT_SOCIAL_CIRCLE	DEF_30_CNT_SOCIAL_CIRCLE	0.8600
464	LIVE_REGION_NOT_WORK_REGION	REG_REGION_NOT_WORK_REGION	0.8600
557	LIVE_CITY_NOT_WORK_CITY	REG_CITY_NOT_WORK_CITY	0.8300
185	AMT_GOODS_PRICE	AMT_ANNUITY	0.7800
154	AMT_ANNUITY	AMT_CREDIT	0.7700
278	DAYS_EMPLOYED	DAYS_BIRTH	0.6300
433	REG_REGION_NOT_WORK_REGION	REG_REGION_NOT_LIVE_REGION	0.4500
526	REG_CITY_NOT_WORK_CITY	REG_CITY_NOT_LIVE_CITY	0.4400

Target 0 data frame

Target 1 data frame

Out[48]:

	Var1	Var2	Correlation
649	OBS_60_CNT_SOCIAL_CIRCLE	OBS_30_CNT_SOCIAL_CIRCLE	1.0000
184	AMT_GOODS_PRICE	AMT_CREDIT	0.9800
680	DEF_60_CNT_SOCIAL_CIRCLE	DEF_30_CNT_SOCIAL_CIRCLE	0.8700
464	LIVE_REGION_NOT_WORK_REGION	REG_REGION_NOT_WORK_REGION	0.8500
557	LIVE_CITY_NOT_WORK_CITY	REG_CITY_NOT_WORK_CITY	0.7800
185	AMT_GOODS_PRICE	AMT_ANNUITY	0.7500
154	AMT_ANNUITY	AMT_CREDIT	0.7500
278	DAYS_EMPLOYED	DAYS_BIRTH	0.5800
433	REG_REGION_NOT_WORK_REGION	REG_REGION_NOT_LIVE_REGION	0.5000
526	REG_CITY_NOT_WORK_CITY	REG_CITY_NOT_LIVE_CITY	0.4700

CONCLUSION

- After conducting a thorough analysis and cleaning of both the application dataset and the previous_application dataset, several significant observations and insights have emerged. These findings provide valuable guidance for banks when evaluating loan applications and assessing the risk of default.

- Firstly, it is evident that granting loans to students, pensioners, and individuals with higher education degrees carries a lower risk of default. Therefore, banks can confidently extend loans to these groups.

- On the other hand, certain occupations such as laborers, sales staff, drivers, cleaning staff, and low-skilled workers exhibit a higher likelihood of defaulting on loan payments. To mitigate risk, the focus should be on targeting clients in more stable positions, such as managers, core staff, and high-skilled technical staff.

- Age also plays a crucial role in loan repayment behavior. Individuals between the ages of 20 and 30 have a higher propensity to default, while those above the age of 45 display a significantly lower default rate.

- Furthermore, the analysis revealed a strong indication that people belonging to low and very low income brackets are more prone to defaulting. This information underscores the importance of considering income levels when evaluating loan applications.

- Geographic location also influences loan repayment patterns. Individuals residing in less populated areas, such as villages or small towns, encounter greater difficulties in repaying their loan amounts. This factor should be taken into account during the assessment process.

- Lastly, clients who are more likely to default on loans tend to make last-minute changes to their registration details just prior to applying. Monitoring and analyzing such registration changes can provide valuable insights into identifying higher-risk applicants.

- By incorporating these parameters into the customer evaluation process, banks can effectively reduce the occurrence of default payments and make more informed decisions regarding loan approvals.

JUPYTER NOTEBOOK LINK

https://drive.google.com/file/d/1wP1QliJbfZ7r2Cl1C-pfbWCSDxUrebtT/view?usp=drive_link

THANK YOU