CREDIT EDA ANALYSIS

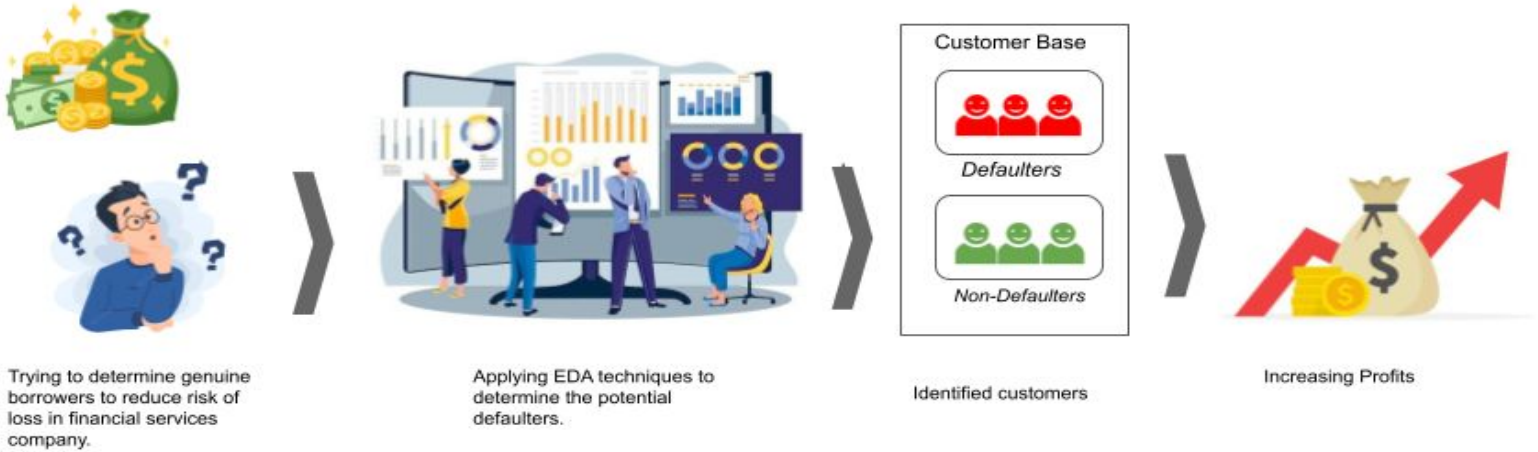
How Company Can Analyze Risk Associated With Loan Applications.

Submitted by:
SHYAM MOHAN AZAD

CONTENT

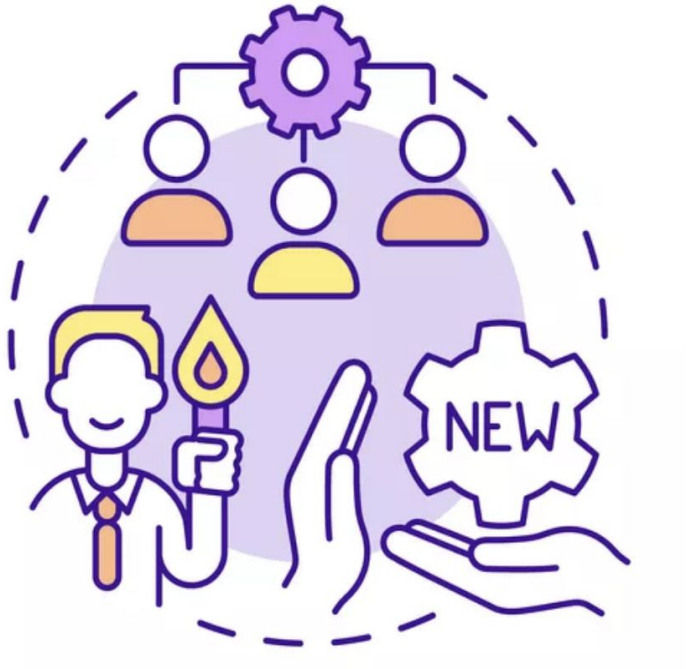
INTRODUCTION

The aim of this case study is to apply EDA techniques to a real business problem of companies in banking and financial services and understand the risk analytics in the said sector.This case study also aims to understand how data can be used to minimize the risk of losing money while lending to customers.



Business Problem Statement

- Businesses providing loans and financial services often fail to know which customers are defaulters and which are not.
- Sometime, an ample amount of data about customer's credit is not available with the company or with government agencies such as CIBIL.
- This information gap is used by defaulters ,to take loan from companies and creat loss for them.

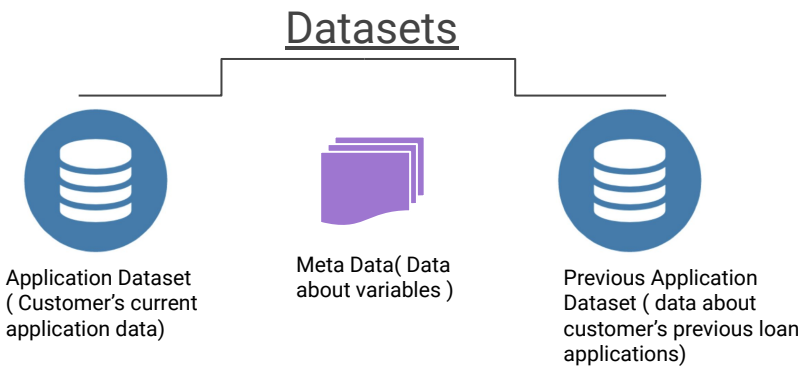


Insufficient Data leads to faulty decisions

Objective

- Apply EDA techniques to the available data with us and extract insights from it about user properties that can lead us to identify potential defaulters.
- To ensure that applications of customers who are capable to repay loans are not rejected.
- The customers who are likely to default, should be identified at early stages and company should be advised to take appropriate measures while lending loan to such customers.

Resources



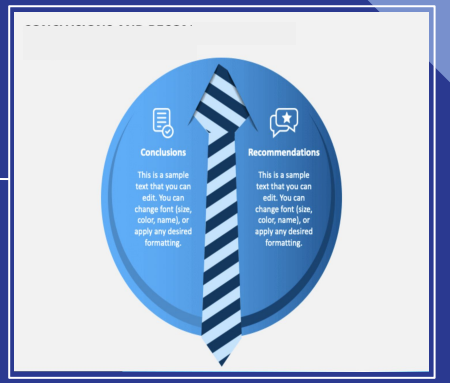
- OS - windows , linux, mac (capable to handle the load ,recommended latest model)
- Jupyter Notebook Environment./alternatives like Google Colab etc
- Python 3

The data source is one of the most important part of any analytical process to start, as without proper data no analytics can be performed. It may include anything but not limited to text, numbers, images, videos etc.

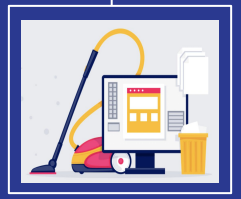
Workflow Diagram

Data Understanding and Exploration

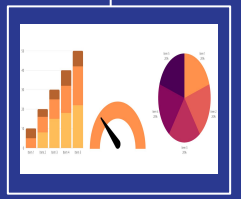
Variable Analysis and correlations



Data Collection



Data Cleaning and Manipulation



Analysis Visualization

Presentation of Conclusion and Recommendation

DataSets

Application Data

- This dataset consists of current application filled for loan to the company.
- The shape of this dataset is `307511, 122` that is 307511 Rows and 122 columns.

Prev. Application Data

- This dataset consists of previous applications filled for loan to the company.
- The shape of this dataset is `1670214, 37` that is 1670214 Rows and 37

Application Data Variables

```
['SK_ID_CURR',
'TARGET',
'NAME_CONTRACT_TYPE',
'CODE_GENDER',
       'FLAG_OWN_CAR',
'FLAG_OWN_REALTY',
'CNT_CHILDREN',
'AMT_INCOME_TOTAL',
       'AMT_CREDIT',
'AMT_ANNUITY',
       ...

'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'],
      dtype='object',
length=122)
```

```
[4] # HAving a check on data and it's shape.

    app.head(10)
```

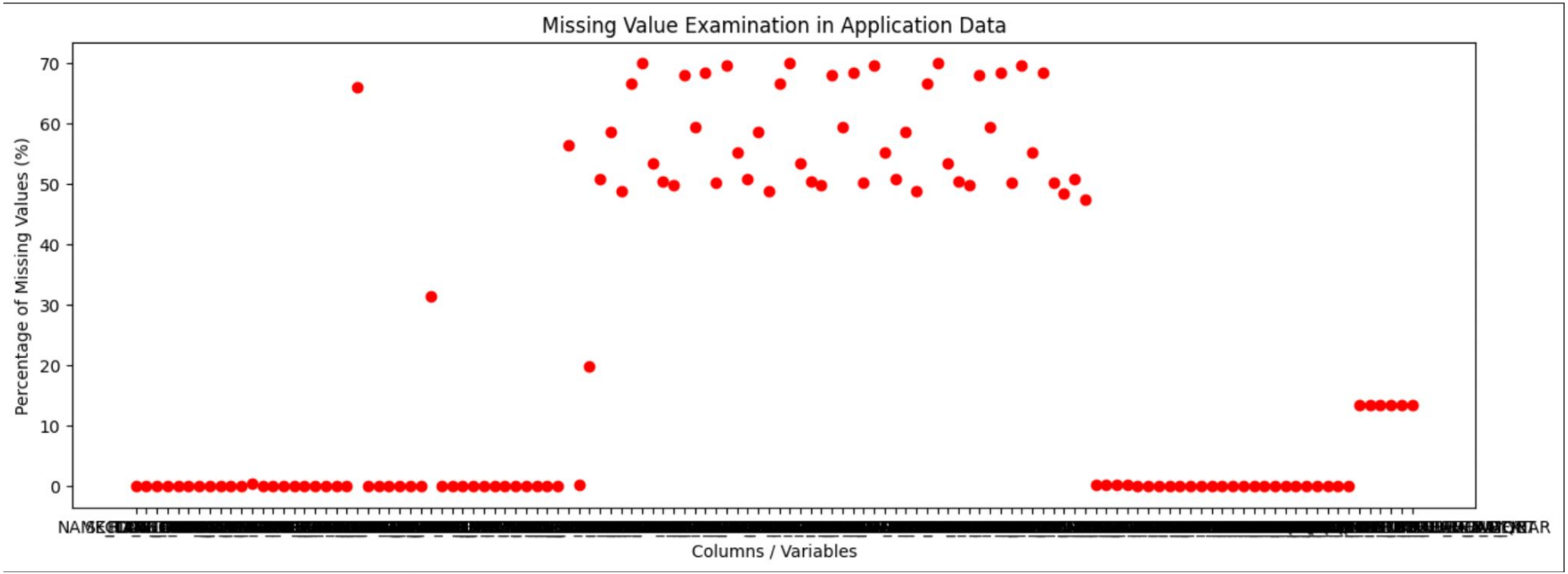
	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	...	FLAG_I
0	100002	1	Cash loans	M	N	Y	0	202500.0	406597.5	24700.5	...	
1	100003	0	Cash loans	F	N	N	0	270000.0	1293502.5	35698.5	...	
2	100004	0	Revolving loans	M	Y	Y	0	67500.0	135000.0	6750.0	...	
3	100006	0	Cash loans	F	N	Y	0	135000.0	312682.5	29686.5	...	
4	100007	0	Cash loans	M	N	Y	0	121500.0	513000.0	21865.5	...	
5	100008	0	Cash loans	M	N	Y	0	99000.0	490495.5	27517.5	...	
6	100009	0	Cash loans	F	Y	Y	1	171000.0	1560726.0	41301.0	...	
7	100010	0	Cash loans	M	Y	Y	0	360000.0	1530000.0	42075.0	...	
8	100011	0	Cash loans	F	N	Y	0	112500.0	1019610.0	33826.5	...	
9	100012	0	Revolving loans	M	N	Y	0	135000.0	405000.0	20250.0	...	

10 rows × 122 columns

```
# checking shape of app.
app.shape
```

```
(307511, 122)
```

Missing Value Examination



A lot of columns around 67 columns having NaNs , among which 65 columns having missing values greater than 10 %.

Target Variable Distribution

Inference:

A data Imbalance can be easily seen in Target Variable.

A majority of applicants are Non-Defaulters.

Out of 307511 , 24825 are Defaulters

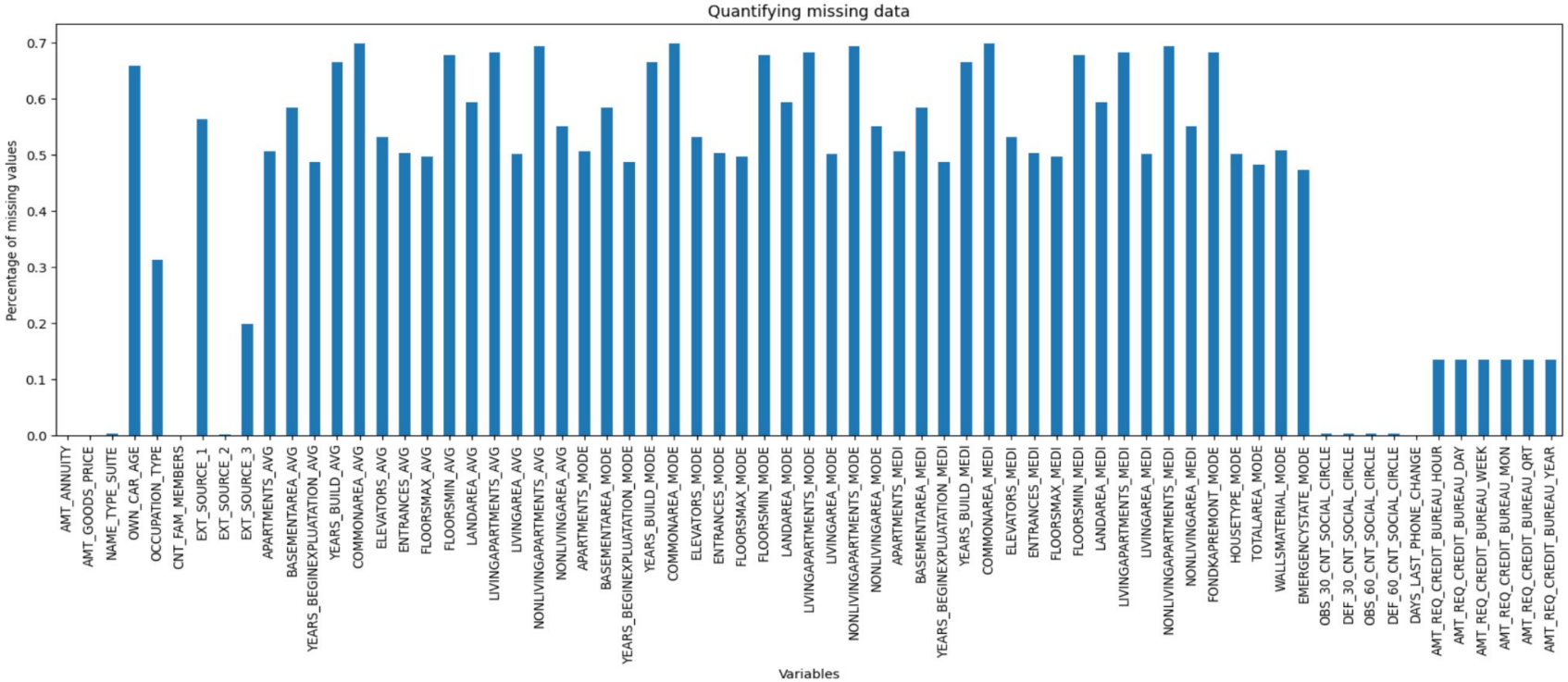
```
target=new_app.TARGET.value_counts()
res=target[0]/target[1]
print('Non-defaulter to defaulter Ratio',res)

#Checking for ratio of Defaulters and Non-Defaulters.
```

```
Non-defaulter to defaulter Ratio 11.387150050352467
```

Number of Non_Defaulter to Defaulter Ratio is 11.38

This implies that 1 in every 11 applicant is having difficulty in payment.

Data Cleaning



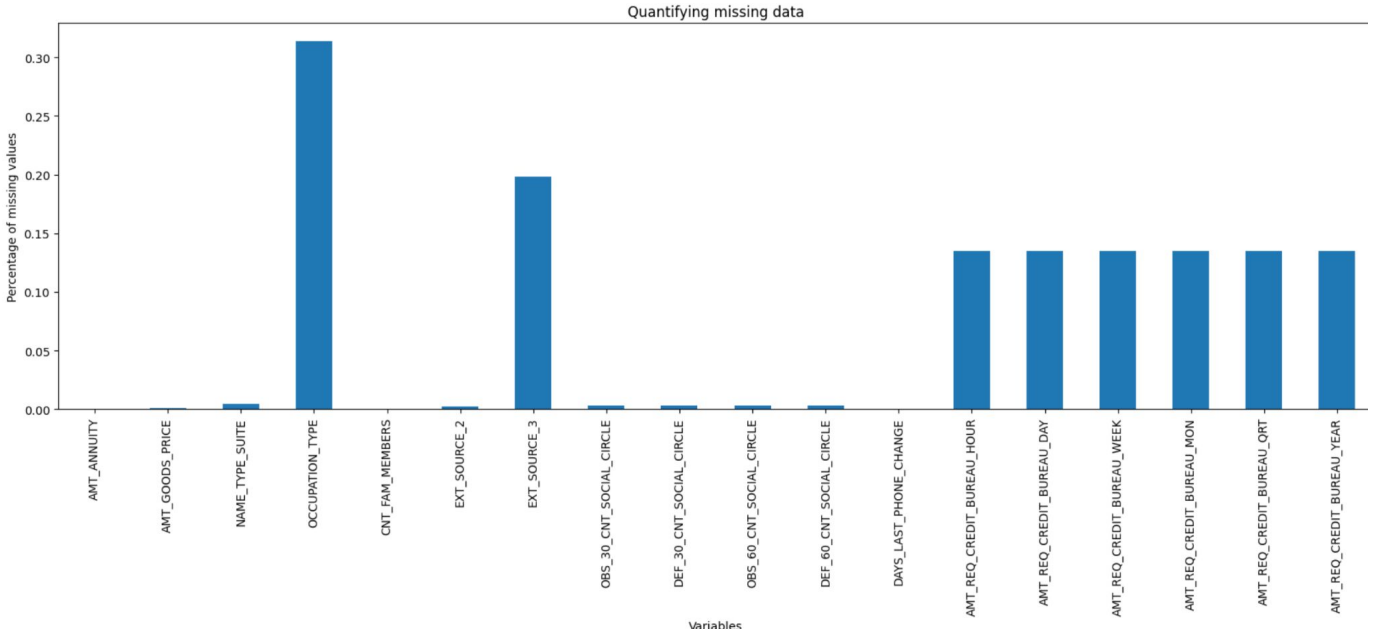
Displaying only Those Variables , which having Missing Values >0. Arround 67 entries.

Missing Values Treatment

. We observed that Many variables having missing values greater than our threshold of 40 %.

. We dropped those variables, as upon observation, it is noticed that all variables are related to applicants residence property.

.After dropping columns having missing values greater than 40% we are left with some more missing values.



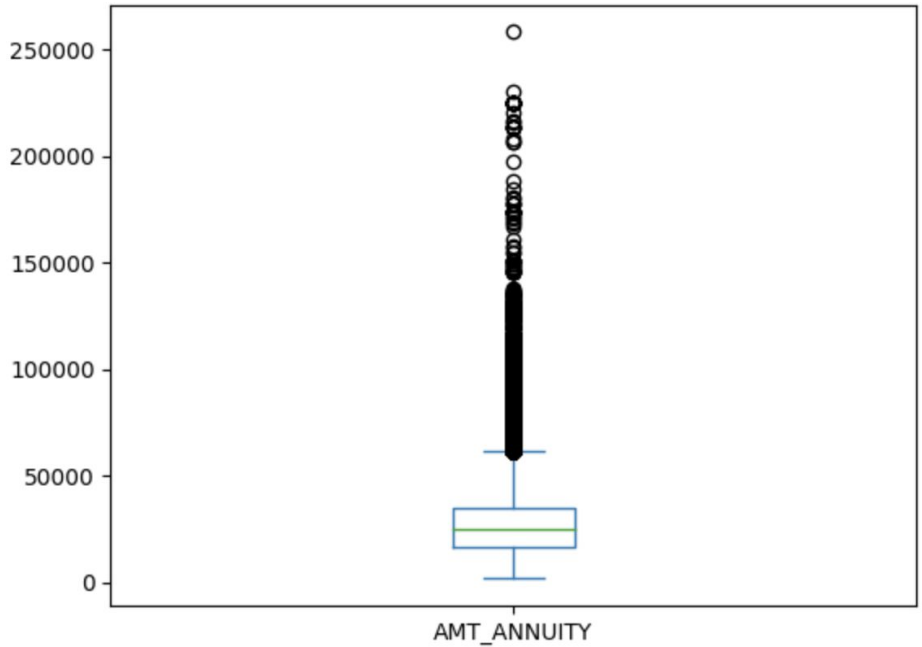
Missing Values Treatment

From here we will check each column one by one:
'AMT_ANNUITY','AMT_GOODS_PRICE','NAME_TYPE_SUITE','DAY_LAST_PHONE_CHANGE','EXT_SOURCE_2','OBS_30_CNT_SOCIAL_CIRCLE','DEF_30_CNT_SOCIAL_CIRCLE','OBS_60_CNT_SOCIAL_CIRCLE','DEF_60_CNT_SOCIAL_CIRCLE' as the percentage of missing values is much less . and also treat replace them if needed.

Inference:

AS it can be seen above that the column has outliers. though they seem to be outliers but actually they are not, because the distance between 99 percentile and outlier point is not significant.

They may be data points of some rich customers which is approximately above 49000. So taking mean() and imputing here may not work out here. therefore we go with the median() of AMT_ANNUITY i am leaving them for now and impute later.

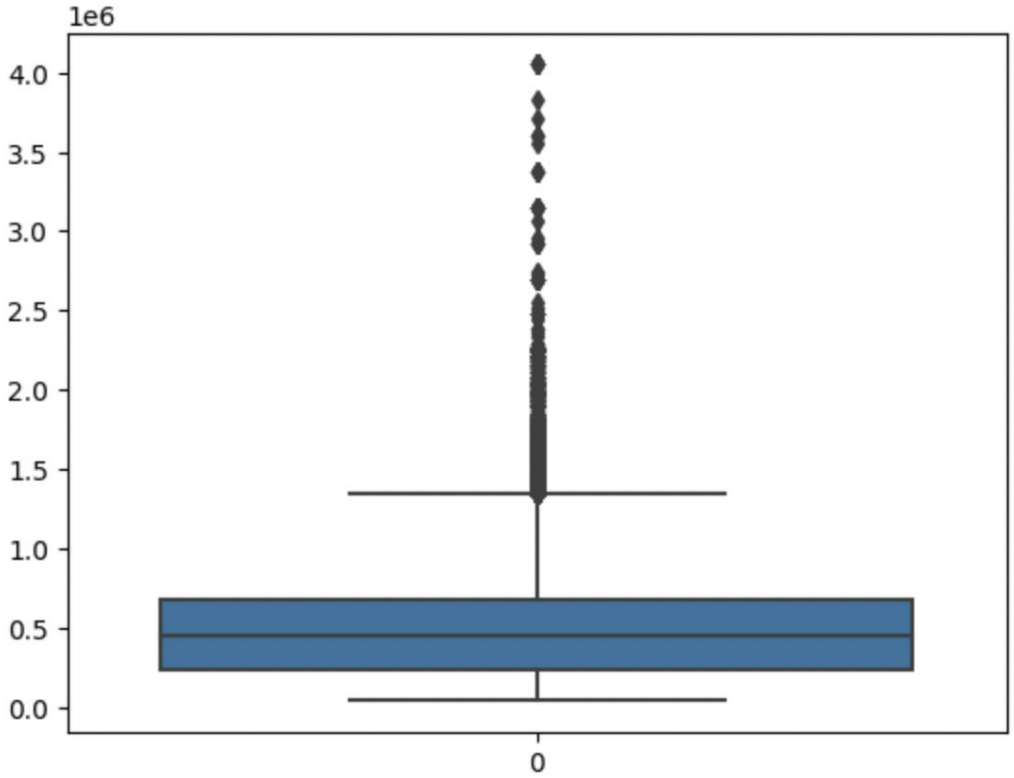


Checking for AMT_GOODS_PRICE

Inference:

A very similar type of boxplot as it was for "AMT_ANNUITY'.

These values also will be imputed by median() if needed.



As it can be seen above that AMT_GOODS_PRICE == 'NaN' for 'NAME_CONTRACT_TYPE'=="Revolving Loans'

Revolving_loans: Revolving credit allows you to borrow money up to a set credit limit, repay it and borrow again as needed. By contrast, installment credit lets you borrow one lump sum, which you pay back in scheduled payments until the loan is paid in full. Ex. Credit cards, home equity, etc.

This implies that in revolving loans no new product is sold/bought. There is a credit line established between financial institution and client, in which client take loan for some time, repay it and again he is eligible to borrow.

Therefore we can easily put AMT_GOODS_PRICE = 0 for here.

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
0	Revolving loans	F	N	Y	2	45000.0	135000.0	6750.0	NaN	
0	Revolving loans	F	N	N	0	157500.0	450000.0	22500.0	NaN	
0	Revolving loans	F	N	N	0	67500.0	202500.0	10125.0	NaN	
0	Revolving loans	F	N	N	1	121500.0	180000.0	9000.0	NaN	
0	Revolving loans	M	N	Y	0	180000.0	450000.0	22500.0	NaN	
...	...	...	...	...	...	...	...	...	...	
0	Revolving loans	F	N	Y	2	67500.0	202500.0	10125.0	NaN	
0	Revolving loans	M	N	Y	1	112500.0	270000.0	13500.0	NaN	
0	Revolving loans	F	N	Y	0	126000.0	270000.0	13500.0	NaN	
0	Revolving loans	M	N	Y	0	135000.0	270000.0	13500.0	NaN	
0	Revolving loans	M	N	Y	2	67500.0	202500.0	10125.0	NaN	

Checking for NAME_TYPE_SUITE

Inference:

NAME_TYPE_SUITE having 1292 missing values .which is very much less compared to the dataset.

Since it is a categorical variable therefore we cannot use mean() or median() here, instead we will use mode() here.

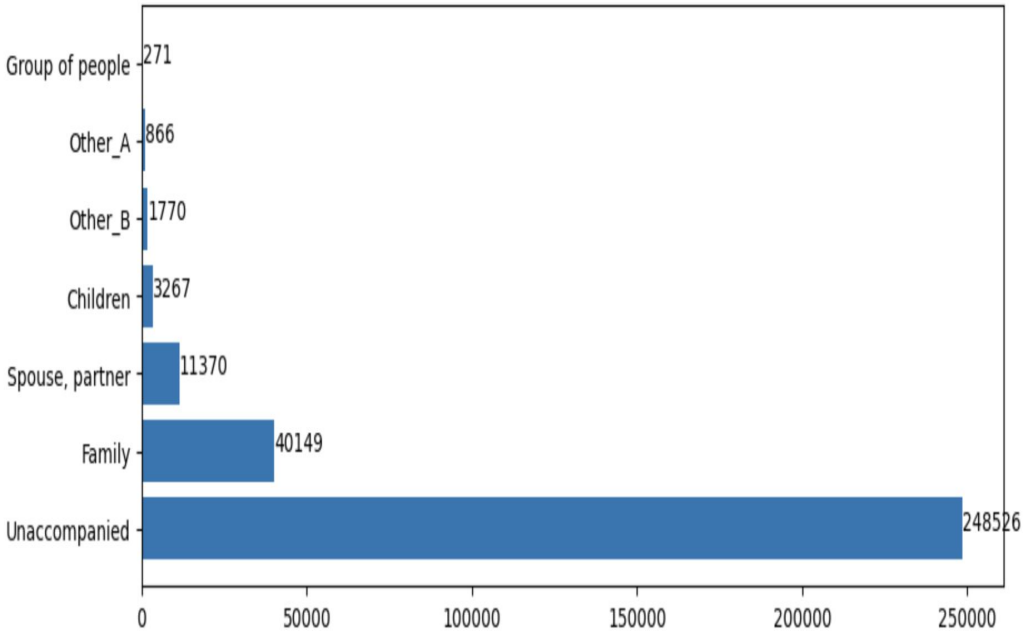
To impute values we can use:

- app['NAME_TYPE_SUITE'].fillna(value=(app['NAME_TYPE_SUITE'].mode()[0]), inplace=True)

*** we used index[0] because ,otherwise it will not write value.

- app['NAME_TYPE_SUITE'].value_counts()

Majority of applicants are Unaccompanied here. 248526



Checking for OCCUPATION_TYPE

Inference:

As a person's income is directly related it's occupation type, so i am looking for values in "NAME_INCOME_TYPE" where 'OCCUPATION_TYPE' == NaN

```
new_app[new_app.OCCUPATION_TYPE.isnull()]['NAME_INCOME_TYPE'].value_counts()
```

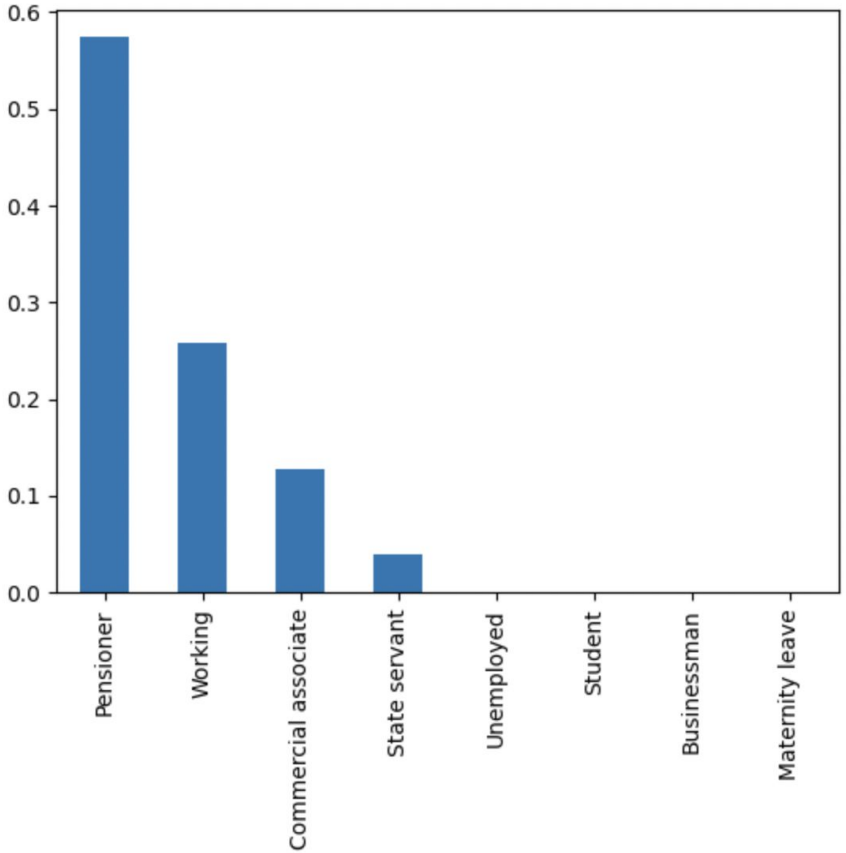
```
Pensioner               55357
Working                 24920
Commercial associate    12297
State servant            3787
Unemployed                 22
Student                     5
Businessman                 2
Maternity leave             1
Name: NAME_INCOME_TYPE, dtype: int64
```

AS it can be understood from above values that , only 22 applicants are actually unemployed, but null is 96391. where highest share is of pensioners.

```
new_app[new_app['NAME_INCOME_TYPE']=='Pensioner']['OCCUPATION_TYPE'].value_counts()
```

```
Cleaning staff    2
Laborers          1
Medicine staff    1
Sales staff       1
Name: OCCUPATION_TYPE, dtype: int64
```

But a very Few have provided information, this implies that this column is not mandatory.



Checking for 'EXT_SOURCE_2' and 'EXT_SOURCE_3'

Inference:

As per analysis , it can be easily seen above that , there is no significant difference between statistical Metrics of EXT_SOURCE_2 and EXT_SOURCE_3 , So there is no harm if we consider them replicates of each other.

Since these are Normalized scores from various external sources as described in meta_data , there may be high chances that those external sources may have utilised the same parameters to obtain these scores.

So we will keep only EXT_SOURCE_2 and discard EXT_SOURCE_3 as it have a high amount of missing values.

we will apply:

- new_app.drop(columns=['EXT_SOURCE_3'], inplace = True)

```
# For EXT_SOURCE_2

new_app['EXT_SOURCE_2'].info()
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 307511 entries, 0 to 307510
Series name: EXT_SOURCE_2
Non-Null Count   Dtype
--------------   -----
306851 non-null  float64
dtypes: float64(1)
memory usage: 2.3 MB
```

```
[50] new_app['EXT_SOURCE_2'].isnull().sum()
```

```
660
```

```
[51] new_app['EXT_SOURCE_2'].describe()
```

```
count    3.068510e+05
mean     5.143927e-01
std      1.910602e-01
min      8.173617e-08
25%      3.924574e-01
50%      5.659614e-01
75%      6.636171e-01
max      8.549997e-01
Name: EXT_SOURCE_2, dtype: float64
```

```
[52] np.round(new_app['EXT_SOURCE_2'].describe(),4)
```

```
count    306851.0000
mean          0.5144
std           0.1911
min           0.0000
25%           0.3925
50%           0.5660
75%           0.6636
max           0.8550
Name: EXT_SOURCE_2, dtype: float64
```

```
# For EXT_SOURCE_3

new_app['EXT_SOURCE_3'].info()
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 307511 entries, 0 to 307510
Series name: EXT_SOURCE_3
Non-Null Count   Dtype
--------------   -----
246546 non-null  float64
dtypes: float64(1)
memory usage: 2.3 MB
```

```
[54] new_app['EXT_SOURCE_3'].isnull().sum()
```

```
60965
```

```
[55] np.round(new_app['EXT_SOURCE_3'].describe(),4)
```

```
count    246546.0000
mean          0.5109
std           0.1948
min           0.0005
25%           0.3706
50%           0.5353
75%           0.6691
max           0.8960
Name: EXT_SOURCE_3, dtype: float64
```

Checking for 'AMT_REQ_CREDIT_BUREAU_HOUR' to "AMT_REQ_CREDIT_BUREAU_YEAR"

- AMT_REQ_CREDIT_BUREAU_HOUR Number of enquiries to Credit Bureau about the client one hour before application
- AMT_REQ_CREDIT_BUREAU_DAY Number of enquiries to Credit Bureau about the client one day before application (excluding one hour before application)
- AMT_REQ_CREDIT_BUREAU_WEEK Number of enquiries to Credit Bureau about the client one week before application (excluding one day before application)
- AMT_REQ_CREDIT_BUREAU_MON Number of enquiries to Credit Bureau about the client one month before application (excluding one week before application)
- AMT_REQ_CREDIT_BUREAU_QRT Number of enquiries to Credit Bureau about the client 3 month before application (excluding one month before application)
- AMT_REQ_CREDIT_BUREAU_YEAR Number of enquiries to Credit Bureau about the client one day year (excluding last 3 months before application)

Definition: A credit inquiry is a request for credit report information from a credit bureau. Credit inquiries are typically made by financial institutions to help them determine whether to approve you for credit.

```
[65] for i in new_app.columns:

       if i.startswith("AMT_REQ"):

         print(new_app[i].mean())
```

```
0.006402448193930645
0.007000210532647598 5
0.0343619356973142
0.26739526000781977
0.26547414959848414
1.899974435321363
```

```
for i in new_app.columns:

  if i.startswith("AMT_REQ"):

    print(new_app[i].mode())
```

```
0    0.0
Name: AMT_REQ_CREDIT_BUREAU_HOUR, dtype: float64
0    0.0
Name: AMT_REQ_CREDIT_BUREAU_DAY, dtype: float64
0    0.0
Name: AMT_REQ_CREDIT_BUREAU_WEEK, dtype: float64
0    0.0
Name: AMT_REQ_CREDIT_BUREAU_MON, dtype: float64
0    0.0
Name: AMT_REQ_CREDIT_BUREAU_QRT, dtype: float64
0    0.0
Name: AMT_REQ_CREDIT_BUREAU_YEAR, dtype: float64
```

```
for i in new_app.columns:
  if i.startswith('AMT_REQ_CREDIT_BUREAU'):
    print(new_app[i].describe())
    print("_____")
```

```
count    265992.000000
mean          0.006402
std           0.083849
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
max           4.000000
Name: AMT_REQ_CREDIT_BUREAU_HOUR, dtype: float64
_____
count    265992.000000
mean          0.007000
std           0.110757
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
max           9.000000
Name: AMT_REQ_CREDIT_BUREAU_DAY, dtype: float64
_____
count    265992.000000
mean          0.034362
std           0.204685
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
max           8.000000
Name: AMT_REQ_CREDIT_BUREAU_WEEK, dtype: float64
_____
count    265992.000000
mean          0.267395
std           0.916002
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
max          27.000000
Name: AMT_REQ_CREDIT_BUREAU_MON, dtype: float64
_____
count    265992.000000
mean          0.265474
std           0.794056
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
max         261.000000
Name: AMT_REQ_CREDIT_BUREAU_QRT, dtype: float64
_____
count    265992.000000
mean          1.899974
std           1.869295
min           0.000000
25%           0.000000
50%           1.000000
75%           3.000000
max          25.000000
Name: AMT_REQ_CREDIT_BUREAU_YEAR, dtype: float64
```

SO by definition it is clearly understood that 'AMT_REQ_CREDIT_BUREAU_['HOUR','DAY','WEEK','MONTH','YEAR'] are the inquiries made by this finance institution about the credit score of the applicant to analyse the risk associated with lending money to him/her.

NaN here may signify that information was not available or the applicant hasn't availed any loan yet.

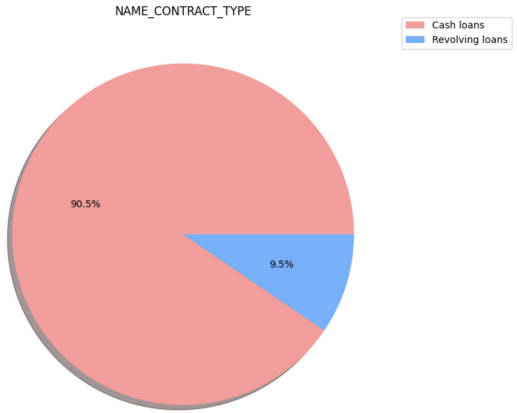
These NaNs can be replaced with mode() values (i.e 0) of respective columns not mean() values because means of these columns are approximately 0 but not 0. it may cause wrong analysis.

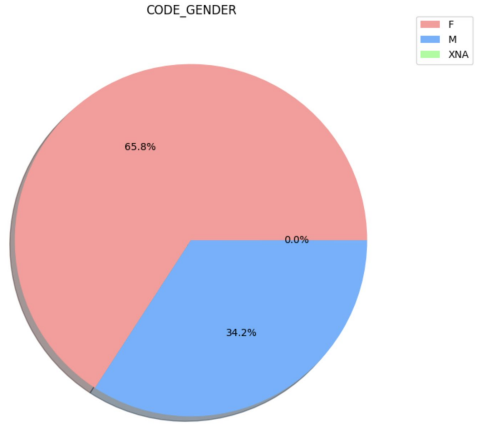
Univariate Analysis [Nominal-Categorical Variables]

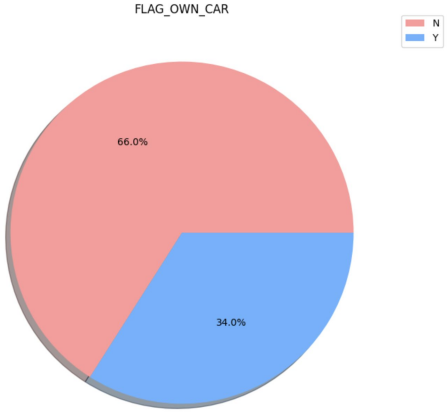
Variables:

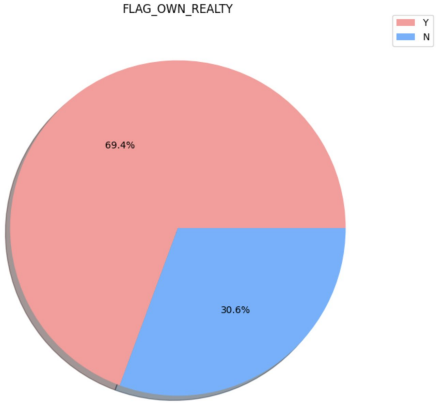
['NAME_CONTRACT_TYPE',
 'CODE_GENDER',
 'FLAG_OWN_CAR',
 'FLAG_OWN_REALTY',
 'NAME_TYPE_SUITE',
 'NAME_INCOME_TYPE',
 'NAME_EDUCATION_TYPE',
 'NAME_FAMILY_STATUS',
 'NAME_HOUSING_TYPE',
 'OCCUPATION_TYPE',
 'WEEKDAY_APPR_PROCESS_START',
 'ORGANIZATION_TYPE']

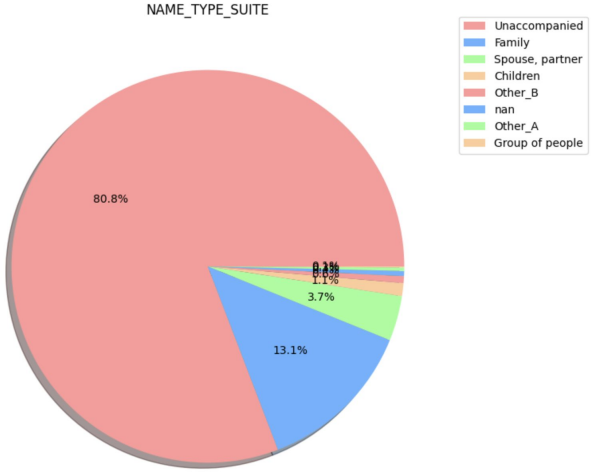
All these variables are showing their Unique values and their percentage of distribution.

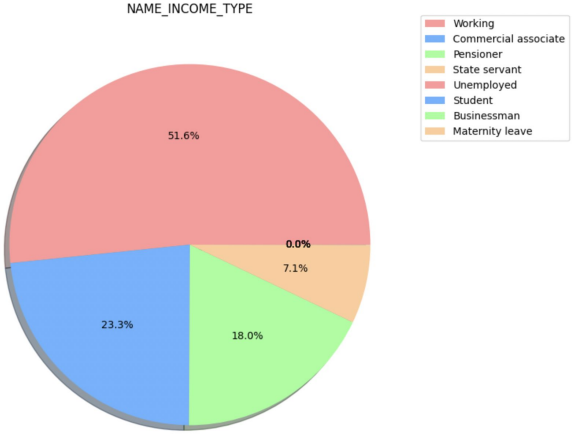


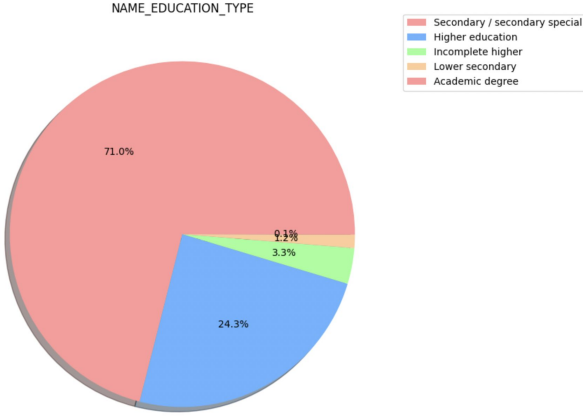


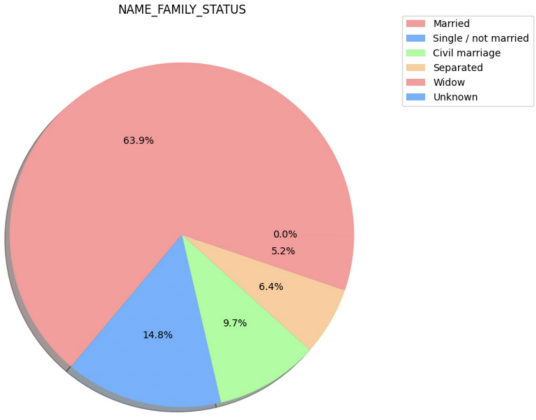


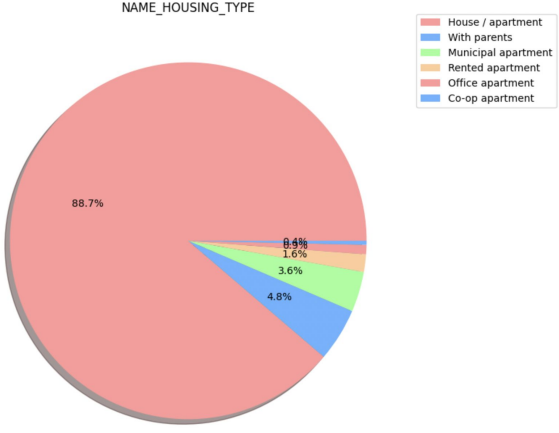


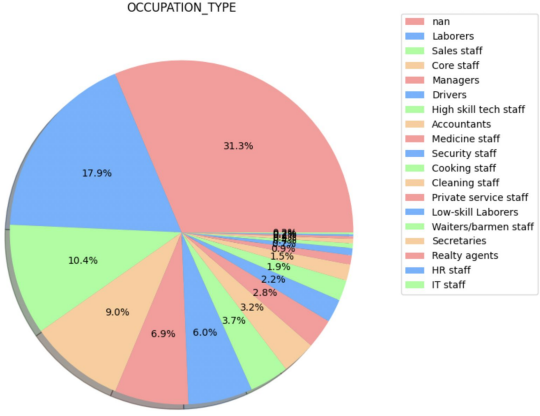

NAME_TYPE_SUITE
Unaccompanied
Family
Spouse, partner
Children
Other_B
nan
Other_A
Group of people
80.8%
13.1%
3.7%


NAME_INCOME_TYPE
Working
Commercial associate
Pensioner
State servant
Unemployed
Student
Businessman
Maternity leave
51.6%
23.3%
18.0%
7.1%
0.0%


NAME_EDUCATION_TYPE
Secondary / secondary special
Higher education
Incomplete higher
Lower secondary
Academic degree
71.0%
24.3%
3.3%

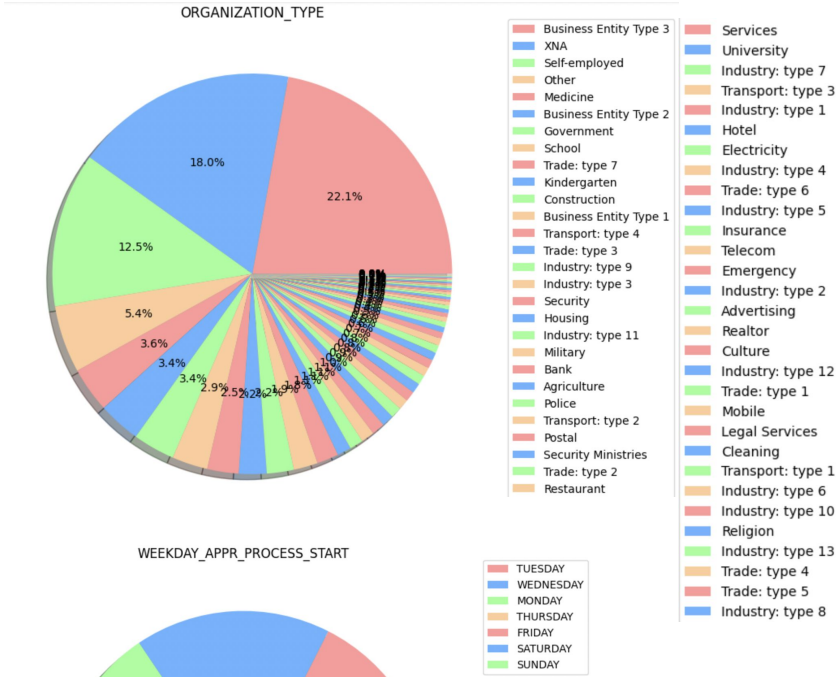

NAME_FAMILY_STATUS
Married
Single / not married
Civil marriage
Separated
Widow
Unknown
63.9%
14.8%
9.7%
6.4%
5.2%
0.0%


NAME_HOUSING_TYPE
House / apartment
With parents
Municipal apartment
Rented apartment
Office apartment
Co-op apartment
88.7%
4.8%
3.6%
1.6%


OCCUPATION_TYPE
nan
Laborers
Sales staff
Core staff
Managers
Drivers
High skill tech staff
Accountants
Medicine staff
Security staff
Cooking staff
Cleaning staff
Private service staff
Low-skill Laborers
Waiters/barmen staff
Secretaries
Realty agents
HR staff
IT staff
31.3%
17.9%
10.4%
9.0%
6.9%
6.0%
3.7%
3.2%
2.8%
2.2%
1.9%
1.5%

Inference from above charts:

1. Cash loans holds greater share(>90%) than Revolving Loans(< 10%)
2. Females have shown more interest towards taking loans which is >65% of total share , whereas men are at only 34.2%.
3. Approximately 66 % of applicants do not own cars.
4. Around 69 % of applicants do own a Property.
5. Approx 81 % of applicants were not accompanied by anyone at the time of application.
6. Half of the share (around 51.6 %) is of working class, 23% for Commercial Associates and 18% for pensioners.
7. Around 71% of applicants holds Secondary Education.
8. Approx 64% of Applicants are Married
9. 88 % of applicant live in their own property/House
10. 31 % of applicants haven't mentioned their OCCUpation type
11. No significant difference found in Application_process_start in any weekday



Handling Numerical Variables

Now we will perform Variable Selection.

As we our target is to find it the Cause of an applicant being defaulter, Therefor BY virtue of my subject matter Knowledge and belief that, some of the columns will not contribute in our Analysis.

These many columns we have found that will not contribute in our analysis, as All these columns are Flags .

Along with that we also renamed
'FLAG_OWN_CAR' to 'OWN_CAR' and
'FLAG_OWN_REALTY' to 'OWN_REALTY'
because we are to delete all the columns start with 'FLAG', so to save those from being deleted.

No use Columns :
['FLAG_MOBIL' , 'FLAG_EMP_PHONE' ,
'FLAG_WORK_PHONE', 'FLAG_CONT_MOBILE',
'FLAG_PHONE', 'FLAG_EMAIL' ,
'REGION_RATING_CLIENT' ,
'REGION_RATING_CLIENT_W_CITY'
,'CNT_FAM_MEMBERS'
,'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']

Checking for CNT_CHILDREN

Inference :

42 Rows have children greater than 5, which may be one of the cause of being defaulter due to inability of bearing Family Expenses.

```
# Checking for CNT_CHILDREN

new_app.CNT_CHILDREN.value_counts()
```

```
0     215371
1      61119
2      26749
3       3717
4        429
5         84
6         21
7          7
14         3
8          2
9          2
12         2
10         2
19         2
11         1
Name: CNT_CHILDREN, dtype: int64
```

```
[85] new_app['CNT_CHILDREN'].describe()
```

```
count    307511.000000
mean          0.417052
std           0.722121
min           0.000000
25%           0.000000
50%           0.000000
75%           1.000000
max          19.000000
Name: CNT_CHILDREN, dtype: float64
```

Standardization Process

Some of the columns such as :
DAYS_EMPLOYED, DAYS_BIRTH, DAYS_ID_PUBLISH, DAYS_REGISTRATION, are in days and some of them are represented negative. Converted to absolute values

```
# Changing DAYS_BIRTH to AGE_IN_YEARS and Removing negative sign.
new_app.DAYS_BIRTH=new_app.DAYS_BIRTH.apply(lambda x: abs(x//365.25))
new_app.rename({'DAYS_BIRTH':'AGE_IN_YEARS'}, axis=1, inplace=True)
new_app['AGE_IN_YEARS']
```

```
0         26.0
1         46.0
2         53.0
3         53.0
4         55.0
          ...
307506    26.0
307507    57.0
307508    41.0
307509    33.0
307510    47.0
Name: AGE_IN_YEARS, Length: 307511, dtype: float64
```

```
[83] new_app.DAYS_EMPLOYED=new_app.DAYS_EMPLOYED.apply(lambda x:abs(x//365.25))
     new_app.rename({'DAYS_EMPLOYED':'YEARS_OF_EMPLOYMENT'},axis=1,inplace=True)

     new_app.DAYS_ID_PUBLISH=new_app.DAYS_ID_PUBLISH.apply(lambda x:abs(x//365.25))
     new_app.rename({'DAYS_ID_PUBLISH':'AGE_OF_ID_IN_YEARS'},axis=1,inplace=True)

     new_app.DAYS_REGISTRATION=new_app.DAYS_REGISTRATION.apply(lambda x:abs(x//365.25))
     new_app.rename({'DAYS_REGISTRATION':'YEARS_OF_REGISTRATION'},axis=1,inplace=True)

     print(new_app.YEARS_OF_EMPLOYMENT,new_app.AGE_OF_ID_IN_YEARS,new_app.YEARS_OF_REGISTRATION)
```

```
0           2.0
1           4.0
2           1.0
3           9.0
4           9.0
           ...
307506      1.0
307507    999.0
307508     22.0
307509     14.0
307510      4.0
Name: YEARS_OF_EMPLOYMENT, Length: 307511, dtype: float64 0          6.0
1          1.0
2          7.0
3          7.0
4         10.0
          ...
307506     6.0
307507    12.0
307508    15.0
307509     3.0
307510     2.0
Name: AGE_OF_ID_IN_YEARS, Length: 307511, dtype: float64 0         10.0
1          4.0
2         12.0
3         27.0
4         12.0
          ...
307506    24.0
307507    13.0
307508    19.0
307509     8.0
307510    15.0
Name: YEARS_OF_REGISTRATION, Length: 307511, dtype: float64
```

Creating Bins for Age Group

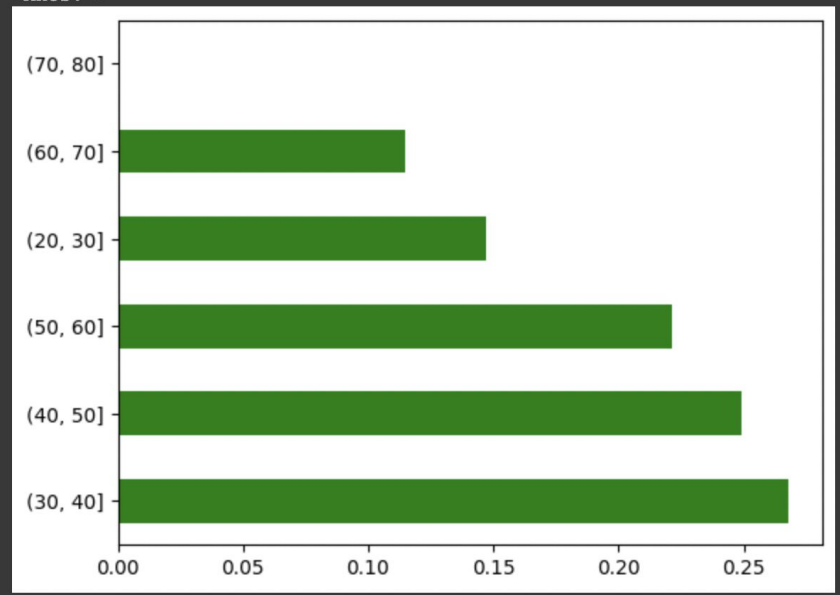
Inference :

As per the above Graph it can be concluded that people of AGE_GROUP 30-40 followed by AGE_GROUP 40-50, are holding greater share of loan applicants.

This may be Due to the fact that people of this age group are more attracted towards goods and services that are beyond necessary for survival or for traditional display of status

```
# Creating BINS for Age group

new_app['AGE_GROUP']= pd.cut(new_app.AGE_IN_YEARS,bins=[20,30,40,50,60,70,80])

new_app['AGE_GROUP'].value_counts(normalize=True).plot.barh(color='green')
```

```
<Axes: >
```



Creating Bins for Income Group

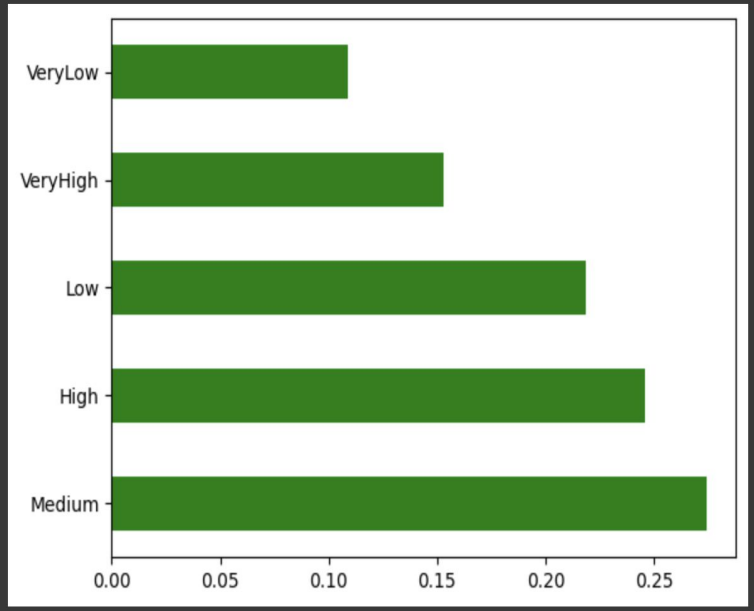
Inference :

As per the above Graph it can be concluded that people of Medium INCOME_GROUP 30%ile-60%ile are holding greater share of loan applicants.

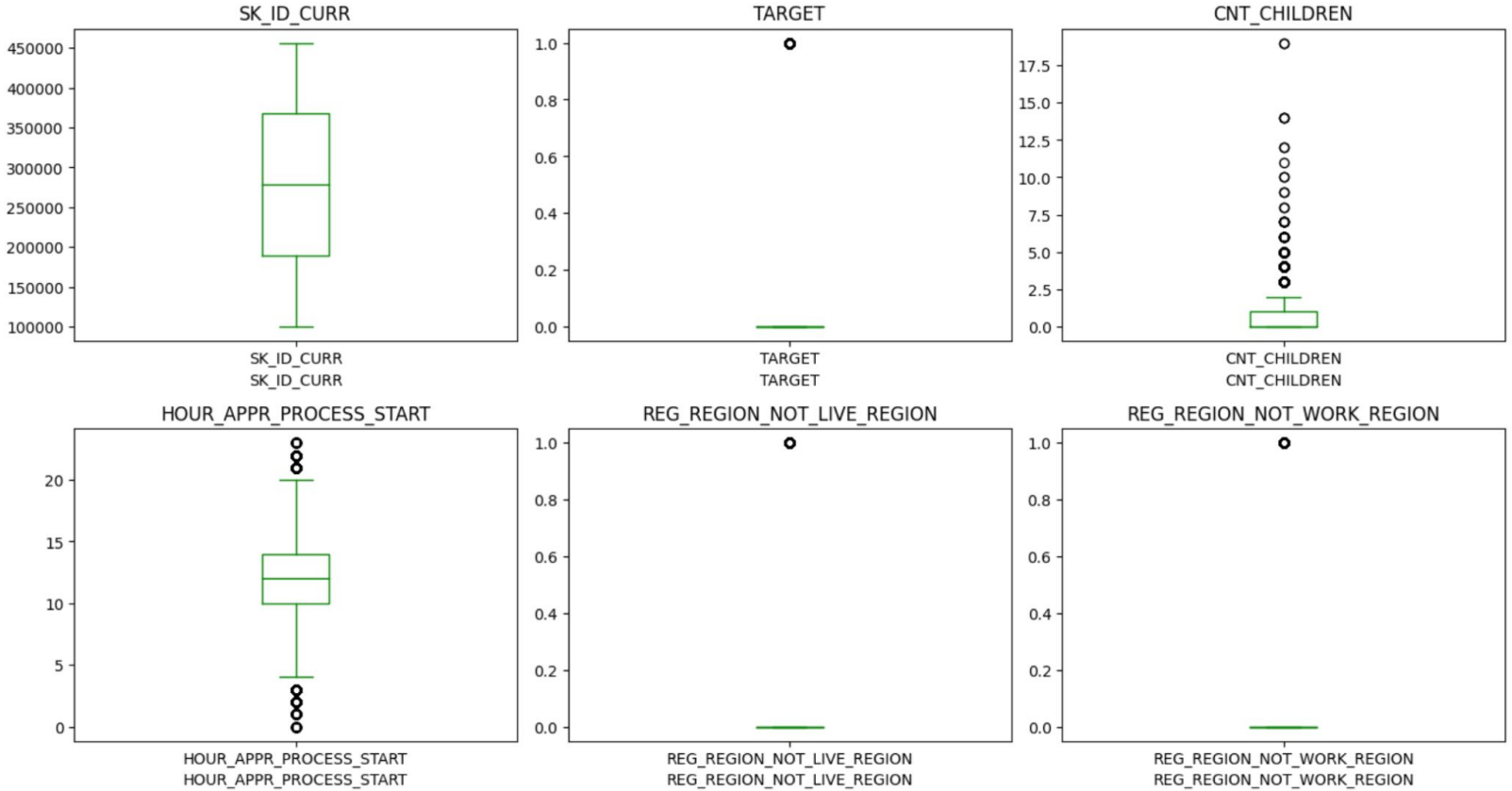
This may be Due to the fact that people of this INCOME_GROUP more attractive towards asset building(like cars , jewellery etc) or Luxury.

we have also seen earlier that , Majority of applicants do not own car, and are females.

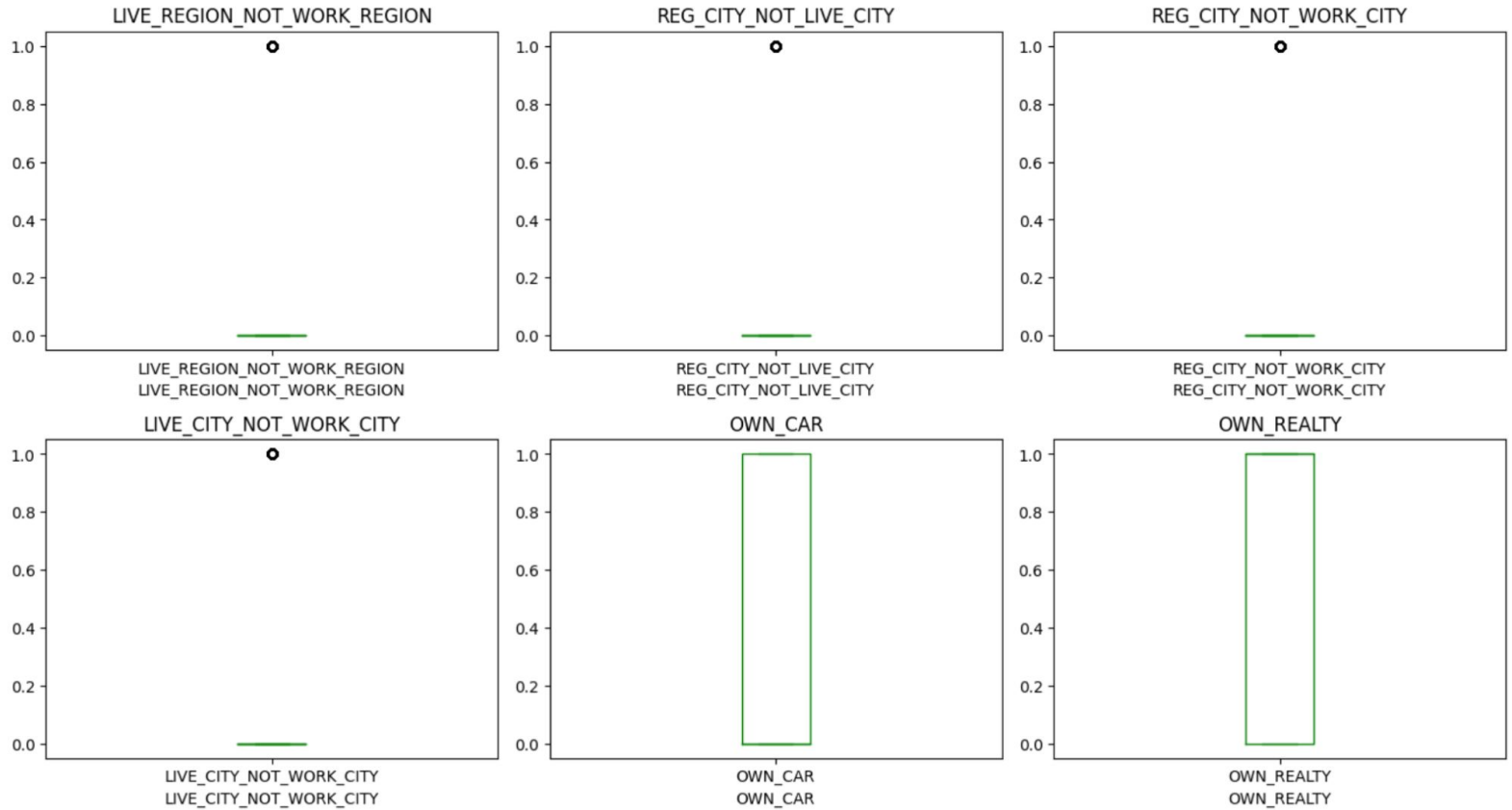
```
# Creating Bins for "INCOME_GROUP"

new_app['INCOME_GROUP']= pd.qcut(new_app['AMT_INCOME_TOTAL'], q=[0,0.1,0.3,0.6,0.8,1],
                                 labels=['VeryLow','Low','Medium','High','VeryHigh'])

new_app['INCOME_GROUP'].value_counts(normalize=True).plot.barh(color='green')
plt.show()
```



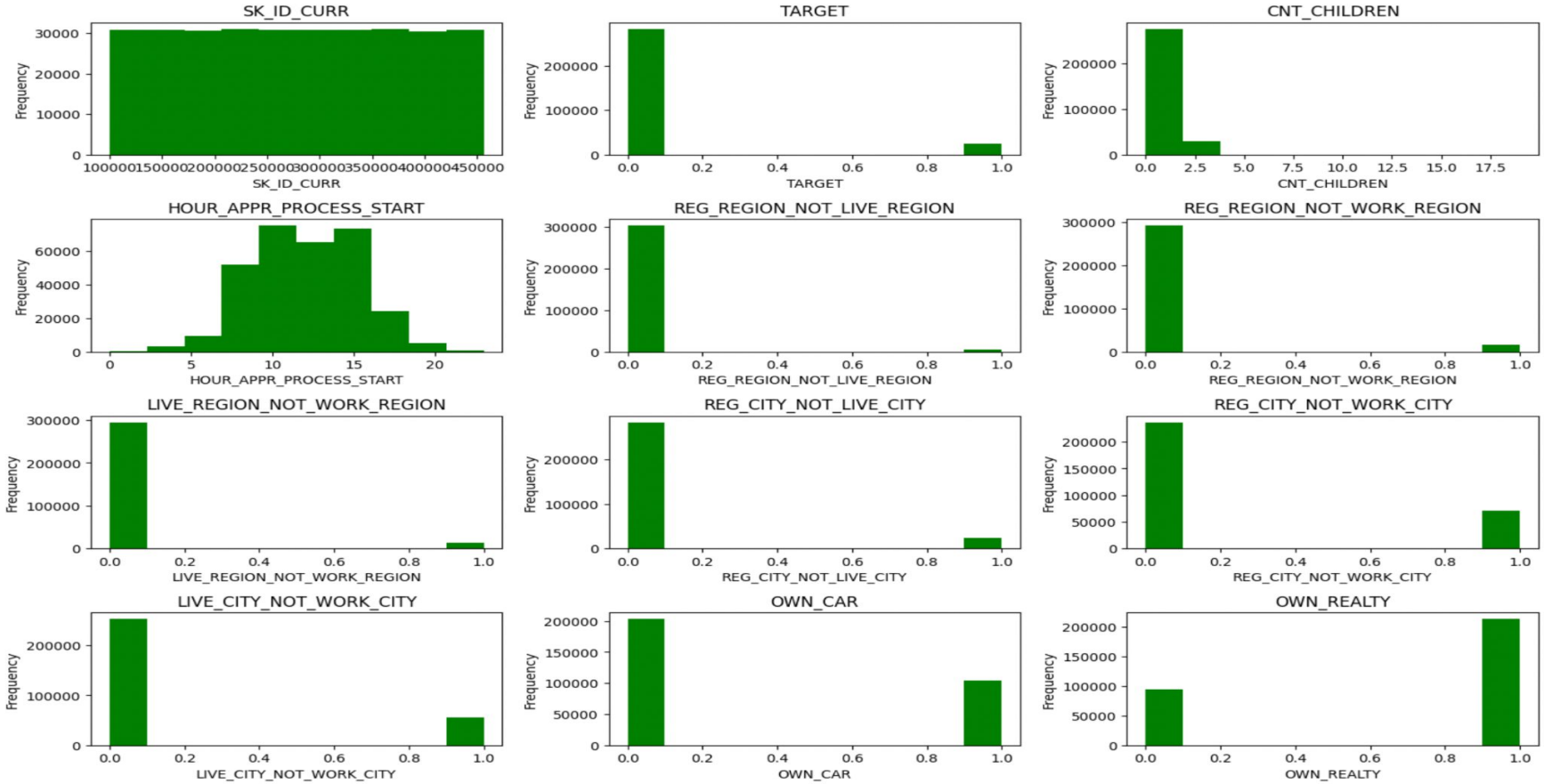
Creating Boxplot for Numerical Data types



Creating Boxplot for Numerical Data types



Creating Histplot for Numerical Data types

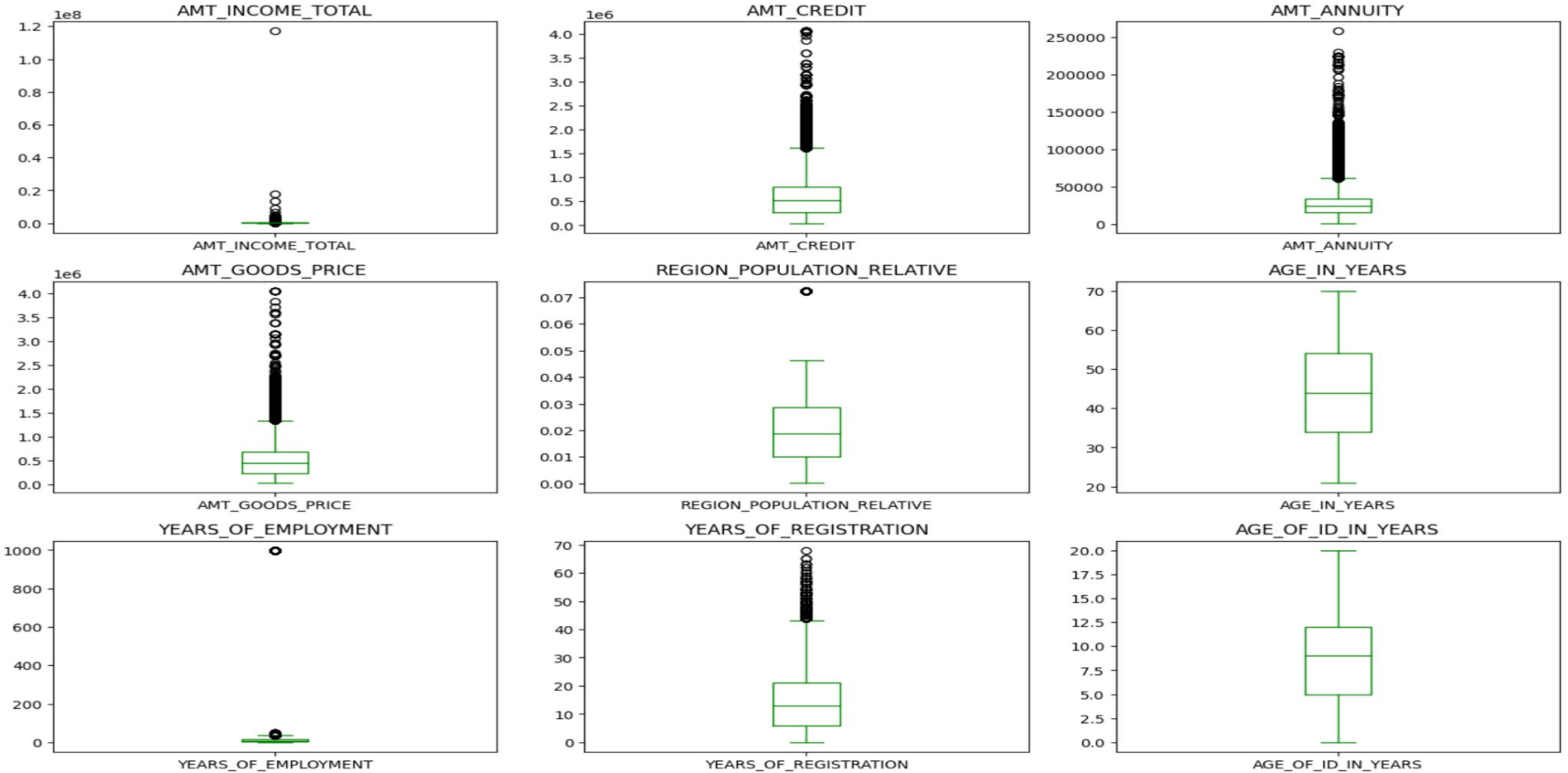


Creating Histplot for Numerical Data types

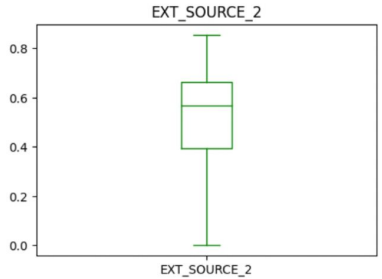
Inference :

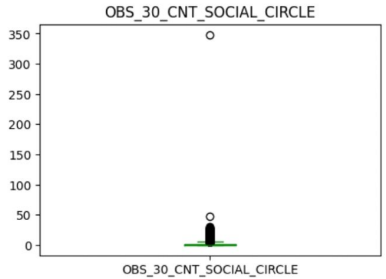
As per above analysis we can say that Majority of plots here are representing Flags(True/False)
ex. TARGET , REG_REGION_NOT_LIVE_REGION , REG_REGION_NOT_WORK_REGION , LIVE_REGION NOT_WORK_REGION , REG_CITY_NOT_LIVE_CITY , REG_CITY_NOT_WORK_CITY , LIVE_CITY_NOT_WORK_CITY , OWN CAR , OWN REALTY ,
we have kept them as int datatype for the sake of calculation.

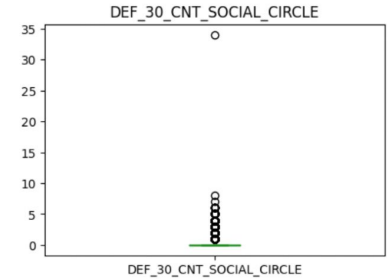
Creating Boxplot for Numerical (Float) Data types

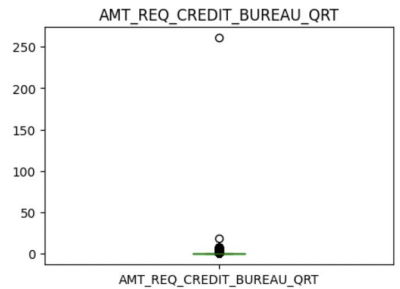


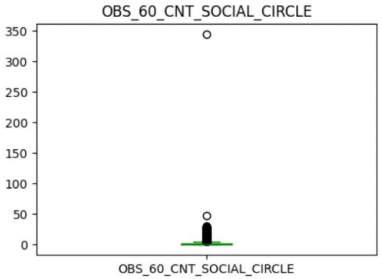
Creating Boxplot for Numerical (Float) Data types

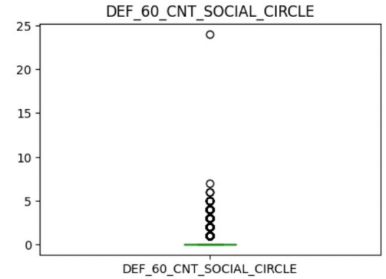


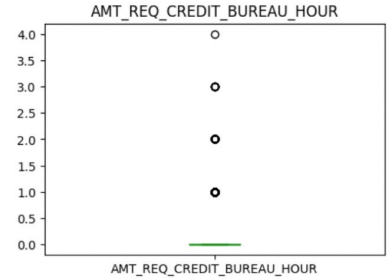


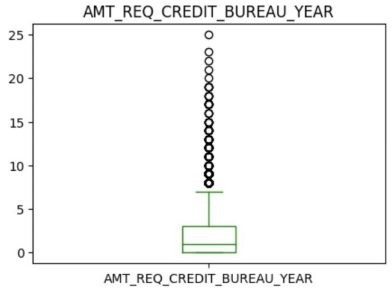


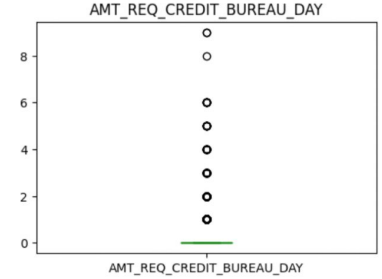


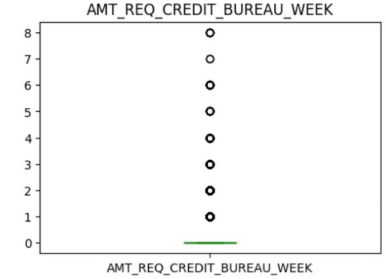


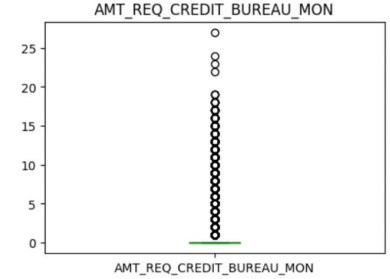












Outlier Analysis and Treatment

We observed that many variables have outliers, need to be treated.

As it has been discussed earlier as well some of these columns looks like they have outliers , but do they really be considered as outliers.

- for ex. ['AMT_CREDIT','AMT_ANNUITY','AMT_GOODS_PRICE','Year_OF_REGISTRATION'] and all columns related to CREDIT_BUREAU Ratings

We've used zscore to determine outliers.

We have taken zscore limit as 3 because , as it can be seen in the boxplot the datapoints seems as an outlier but very close to eachother,Though they have gap in between them but was not as significant as it can be considered as outlier. So we are considering 99.7% data ppoints, and will be substituted using Median values.

- Why Median not mean , because many of them are Flag columns, and Distribution of almost all columns is skwed, and in such case median is less sensitive to outliers as compared to mean.
- But leaving them as it is for now.

```
# Analysis using zscore

Out_col=['AMT_INCOME_TOTAL','AMT_CREDIT','AMT_ANNUITY','AMT_GOODS_PRICE',
         'REGION_POPULATION_RELATIVE','YEARS_OF_EMPLOYMENT','YEARS_OF_REGISTRATION',
         'OBS_30_CNT_SOCIAL_CIRCLE','DEF_30_CNT_SOCIAL_CIRCLE','OBS_60_CNT_SOCIAL_CIRCLE',
         'DEF_60_CNT_SOCIAL_CIRCLE','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']

for item in Out_col:
    print( item, ":", end= " ")
    print(new_app.iloc[np.where(np.abs(stat.zscore(new_app[item])>3))].shape)
```

```
AMT_INCOME_TOTAL : (454, 44)
AMT_CREDIT : (3255, 44)
AMT_ANNUITY : (0, 44)
AMT_GOODS_PRICE : (0, 44)
REGION_POPULATION_RELATIVE : (8412, 44)
YEARS_OF_EMPLOYMENT : (0, 44)
YEARS_OF_REGISTRATION : (645, 44)
OBS_30_CNT_SOCIAL_CIRCLE : (0, 44)
DEF_30_CNT_SOCIAL_CIRCLE : (0, 44)
OBS_60_CNT_SOCIAL_CIRCLE : (0, 44)
DEF_60_CNT_SOCIAL_CIRCLE : (0, 44)
AMT_REQ_CREDIT_BUREAU_HOUR : (0, 44)
AMT_REQ_CREDIT_BUREAU_DAY : (0, 44)
AMT_REQ_CREDIT_BUREAU_WEEK : (0, 44)
AMT_REQ_CREDIT_BUREAU_MON : (0, 44)
AMT_REQ_CREDIT_BUREAU_QRT : (0, 44)
AMT_REQ_CREDIT_BUREAU_YEAR : (0, 44)
```

Segmentation

We have divided TARGET variable in two Sub Dataframes

‘no_dif’ - applicants with no difficulty

‘dif’ - applicant with difficulty.

Further we divided application dataset into three sub dataframes

‘new_app_num’ - Numeric variables

‘new_app_obj’ - Object Type Variables

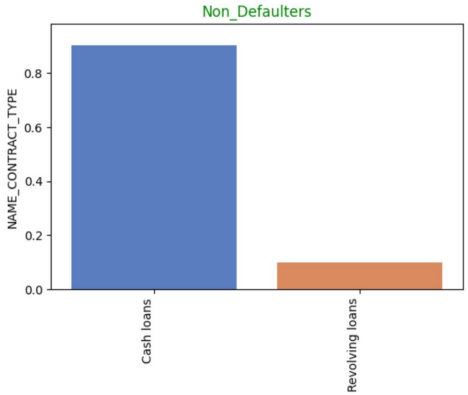
‘new_app_cat’ - Categorical variable

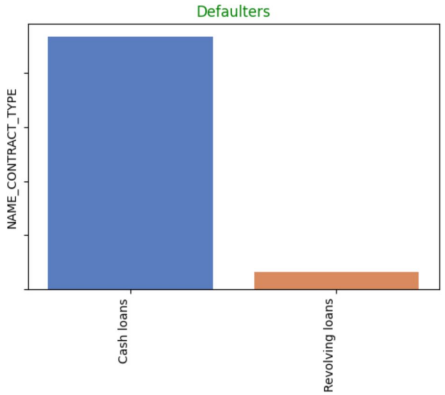
```
Numeric Variables:  ['CNT CHILDREN', 'AMT INCOME TOTAL',
'AMT CREDIT', 'AMT ANNUITY', 'AMT GOODS PRICE',
'REGION POPULATION RELATIVE', 'AGE IN YEARS',
'YEARS OF EMPLOYMENT', 'YEARS OF REGISTRATION',
'AGE OF ID IN YEARS', 'HOUR APPR PROCESS START',
'EXT SOURCE 2', 'OBS 30 CNT SOCIAL CIRCLE',
'DEF 30 CNT SOCIAL CIRCLE', 'OBS_60_CNT_SOCIAL_CIRCLE',
'DEF 60 CNT SOCIAL CIRCLE',
'AMT REQ CREDIT BUREAU 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']
```

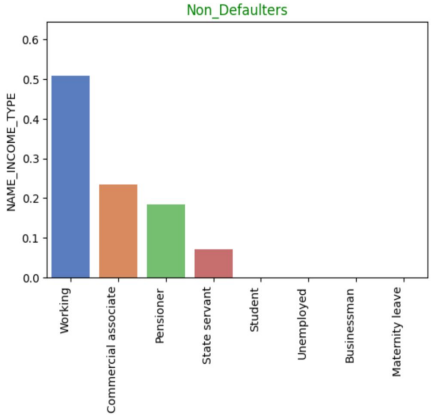
```
Object type Variables:  ['NAME CONTRACT TYPE',
'NAME TYPE SUITE', 'NAME INCOME TYPE',
'NAME EDUCATION TYPE', 'NAME FAMILY STATUS',
'NAME HOUSING TYPE', 'OCCUPATION TYPE',
'WEEKDAY_APPR_PROCESS_START', 'ORGANIZATION_TYPE']
```

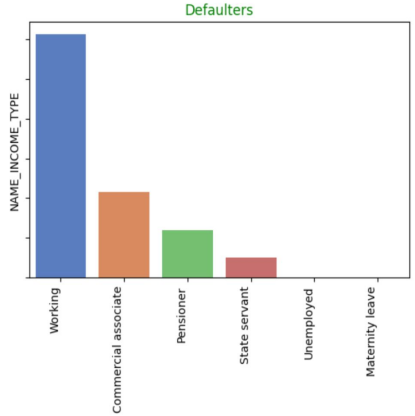
```
categorical Variables:  ['REG_REGION_NOT_LIVE_REGION',
'REG REGION NOT WORK REGION',
'LIVE REGION NOT WORK REGION', 'REG CITY NOT LIVE CITY',
'REG CITY NOT WORK CITY', 'LIVE CITY NOT_WORK_CITY',
'OWN_CAR', 'OWN_REALTY', 'CODE_GENDER']
```

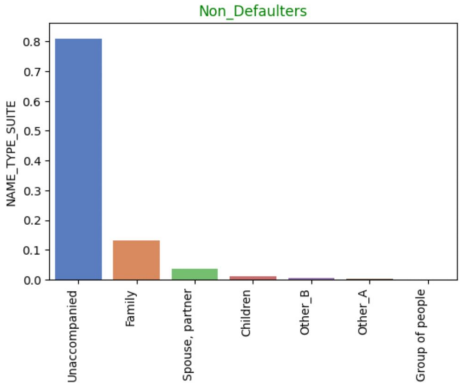
Analysing target Variable with Object Type Data

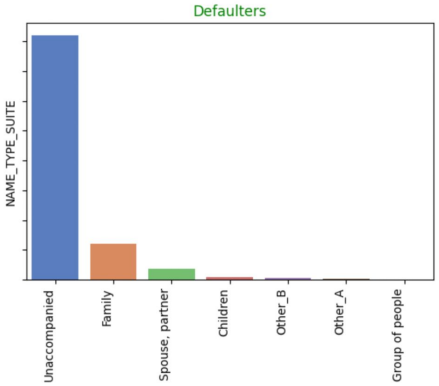


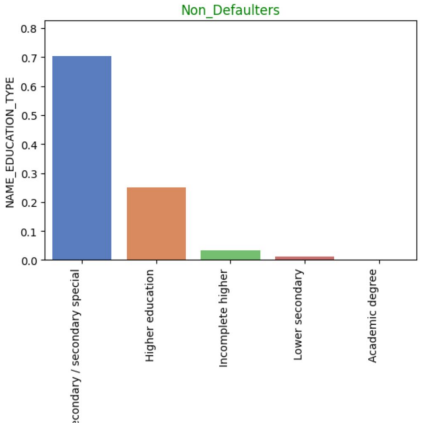


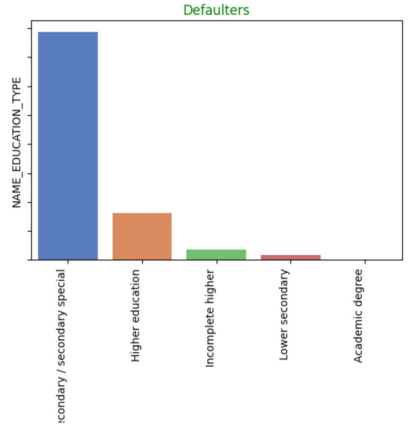




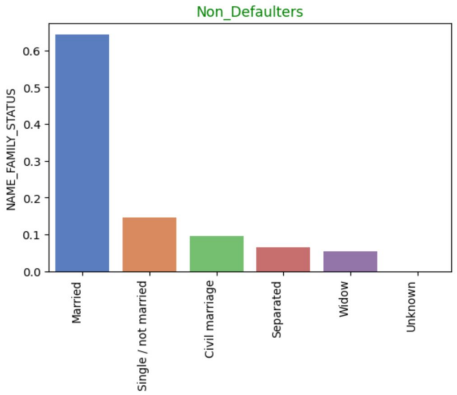


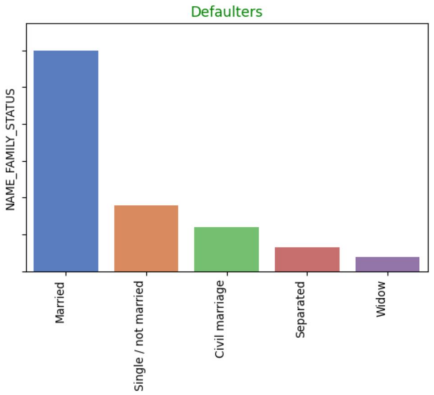


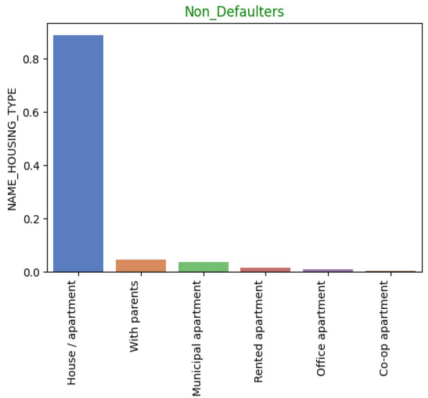


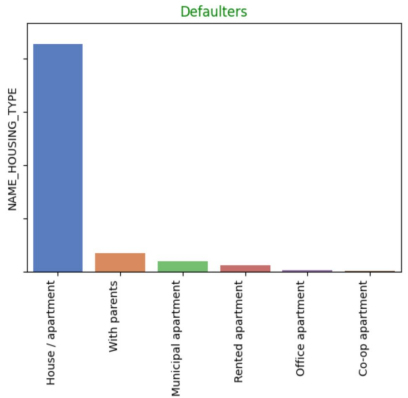


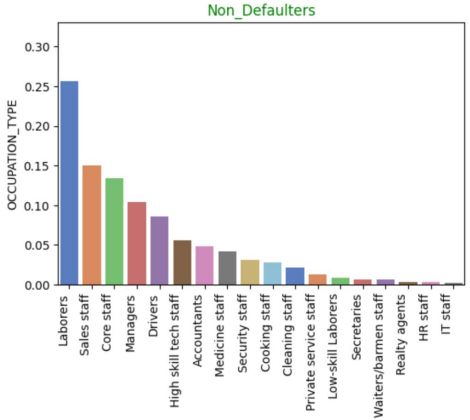
Analysing target Variable with Object Type Data

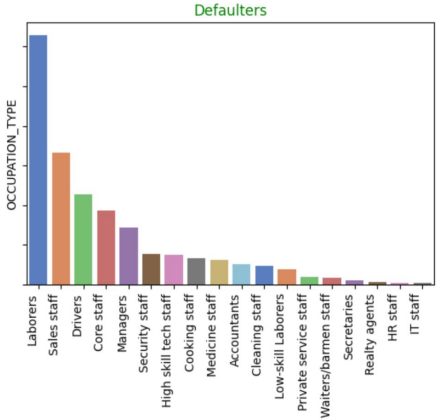


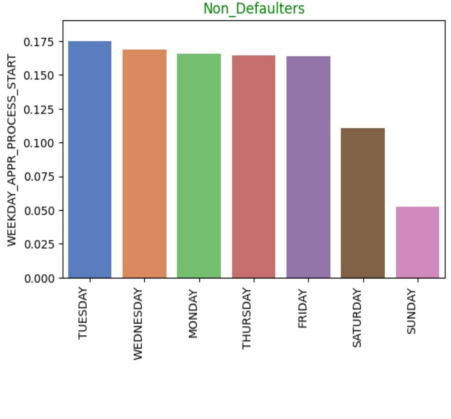


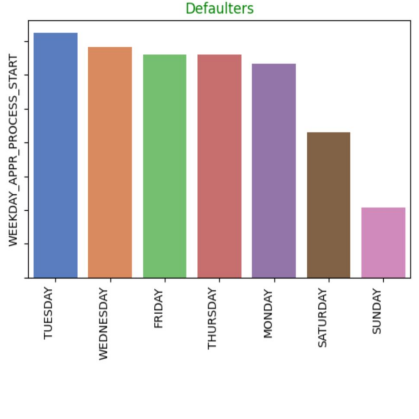




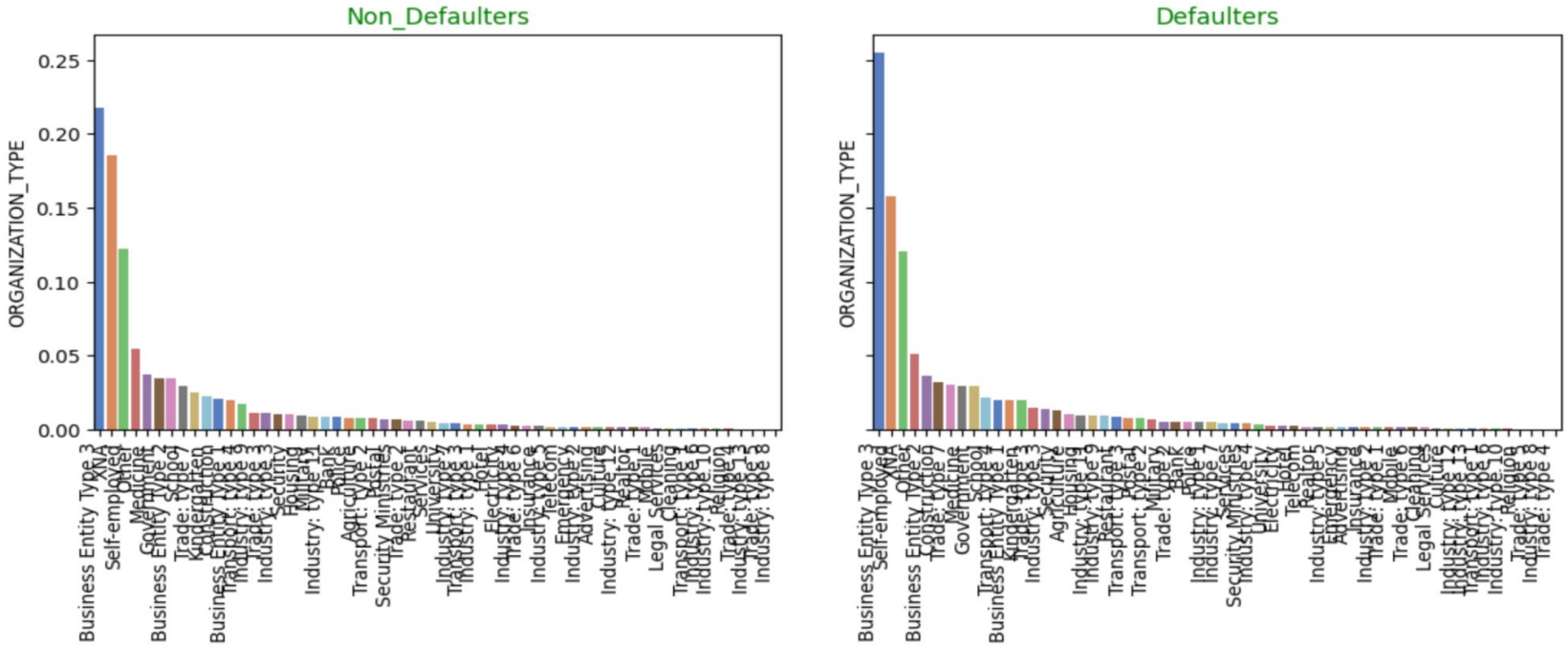








Analysing target Variable with Object Type Data



Analysing target Variable with Object Type Data

Inference :

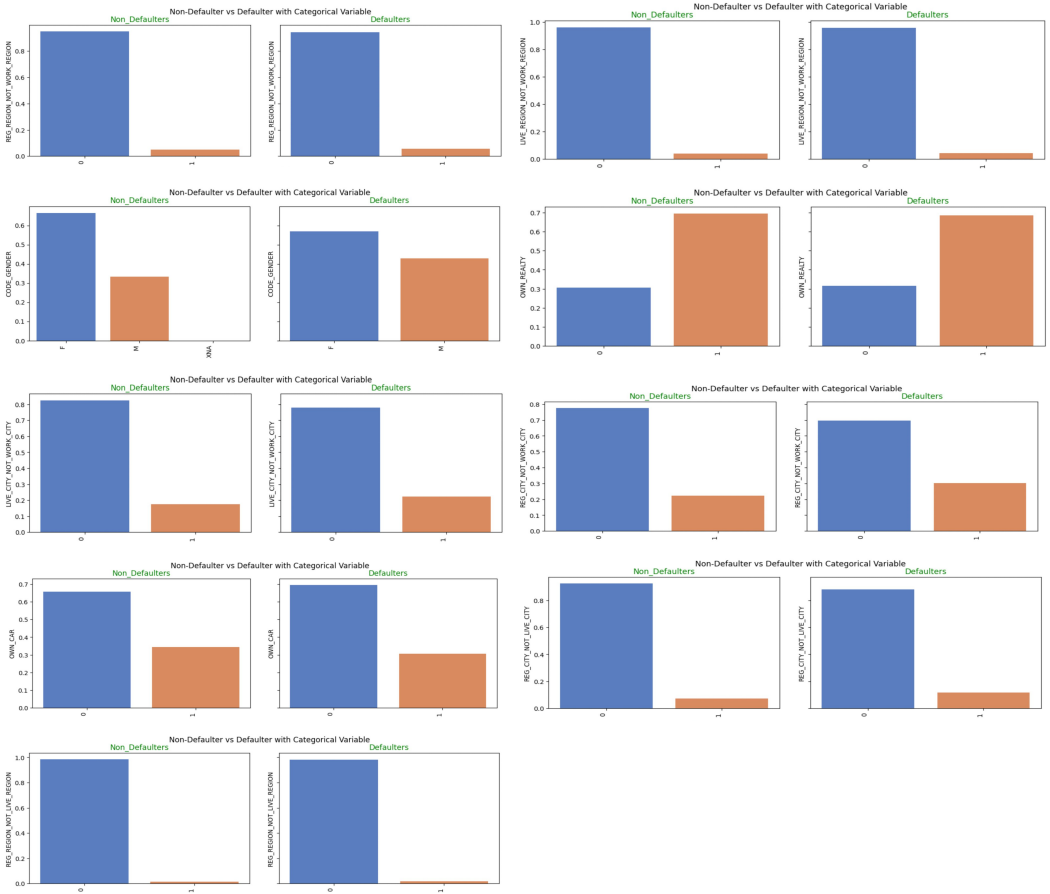
1. CAsh Loans a very large share For Non_defaulters (approx 85 %) and for Defaulters (approx 95%) in column NAME_CONTRACT_TYPE.
2. Approx 80% of Non_Defaulters and above 80% of Defaulters have applied for loan unaccompanied as per column NAME_TYPE_SUIT.
3. As per column NAME_INCOME_TYPE 50% of Non_Defaulters and 60% of Defaulters are working applicants.
4. Approx 70% of Non_Defaulters and 79% of Defaulters are Secondary Educated as per column NAME_EDUCATION_TYPE. Here difference between Positive class and Negative class is not significant to conclude anything.
5. Approximately 60% Defaulter Applicants are Married in column NAME_FAMILY_STATUS.
6. Housing Type data shows approx 85% of Defaulters own a Residential Property. So owning property can not guaranty that applicant will not have any difficulty in repayment of loan.
7. Approximately 60% of Defaulters having OCCUPATION_TYPE as ['Laborers, Sales staff,Drivers,Core staff]
8. Week days doesn't show any relevance with applicant being a defaulter.
9. From ORGAIZATION_TYPE chart is observed that above 25% of Defaulters are BUSINESS_ENTITY_TYPE_3 , approx 16% are XNA and around 12% are SELF_EMPLOYED.They constitute around 53% of Defaulters. Whereas in Non_Defaulters they constitute 52% , So from here also we cannot conclude anything as it is a chance of almost 50:50 .

Analysing target Variable with Categorical Type Data

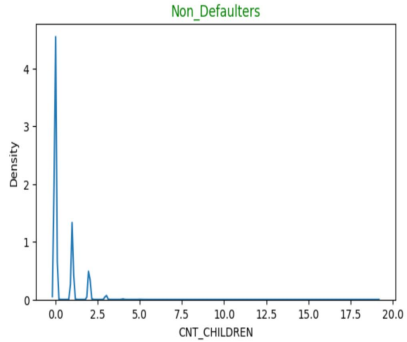
Inference :

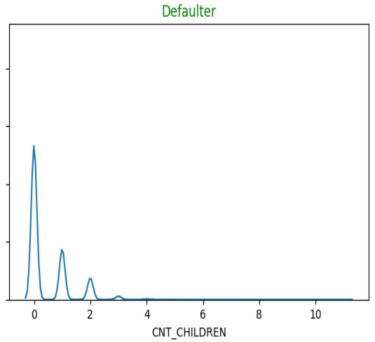
No conclusion can be made based upon above analysis except that Females cover larger share in Both Non_Defaulter and Defaulter Class. ie. above 65% in Non_Defaulter and approx 58% in Defaulters.

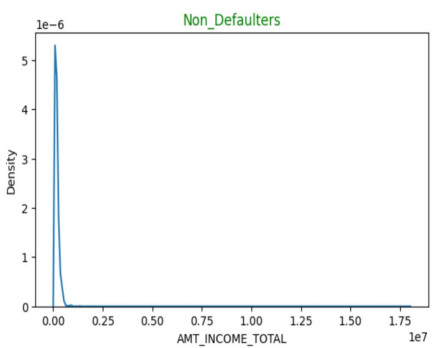
Similarly Reality Owners cover Approximately equal share in both Non_Defaulter and Defaulter Class.

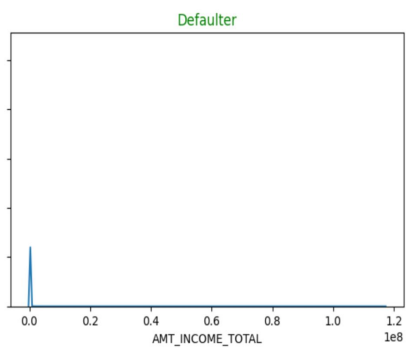


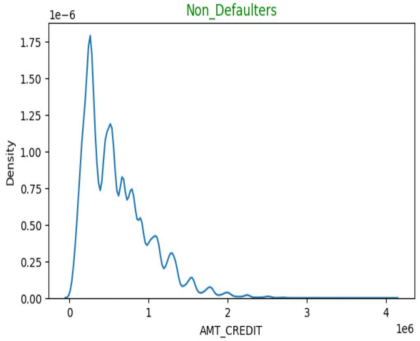
Analysing target Variable with Numerical Type Data

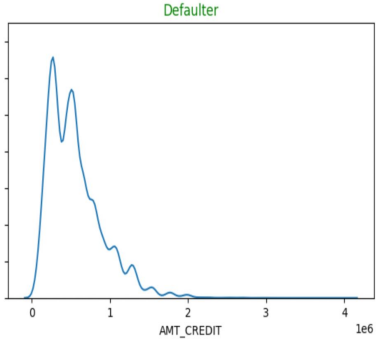


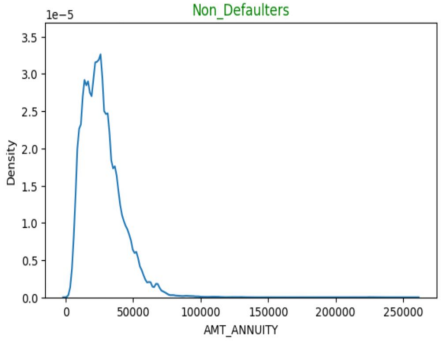


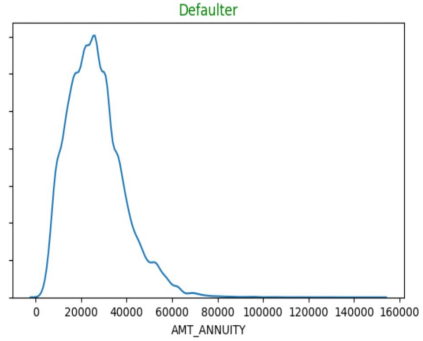




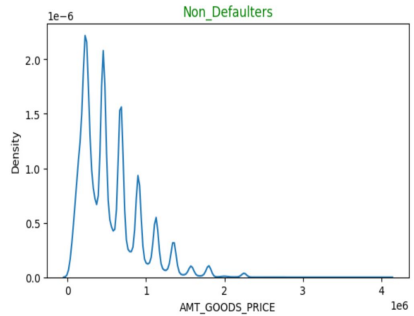


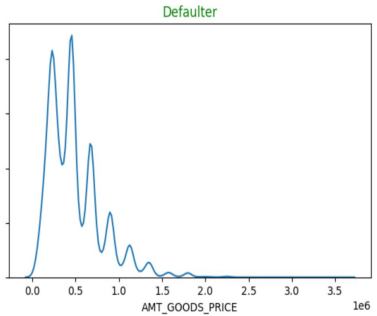


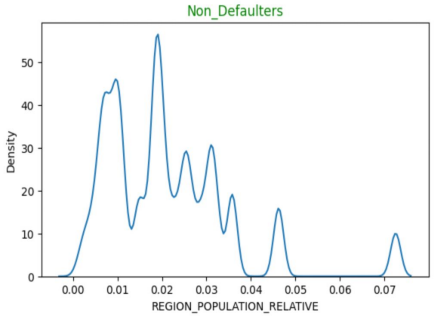


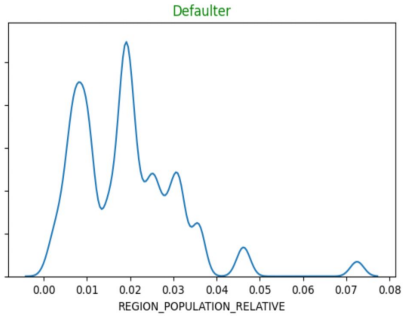


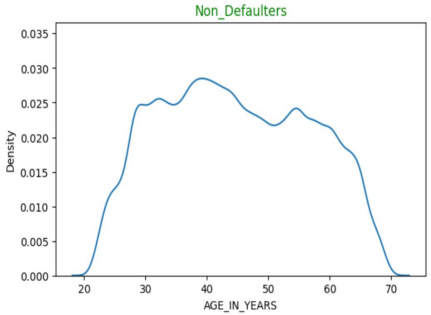
Analysing target Variable with Numerical Type Data

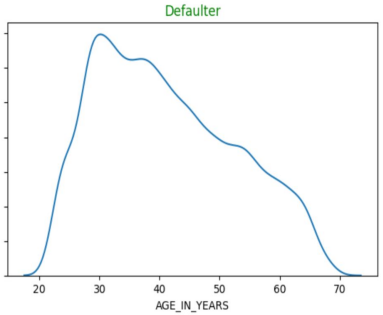


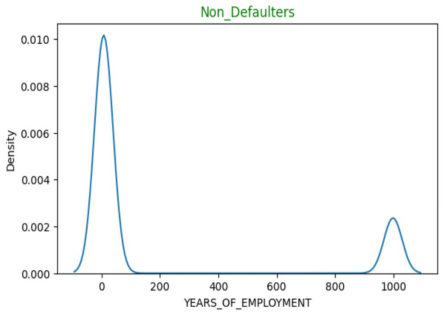


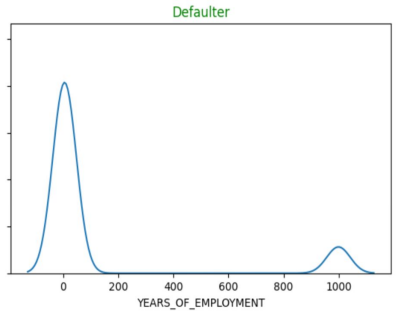




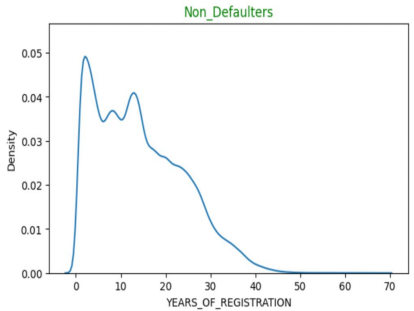


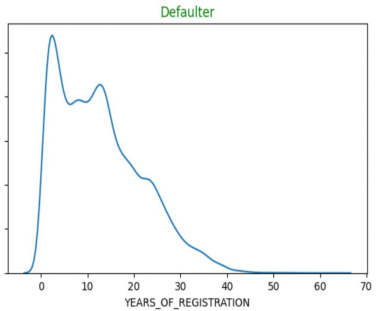


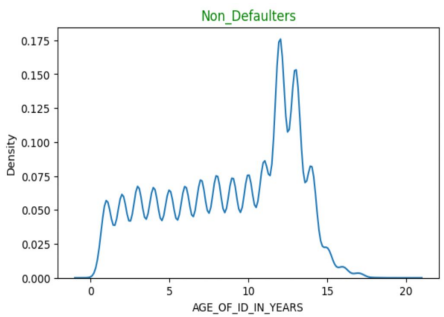


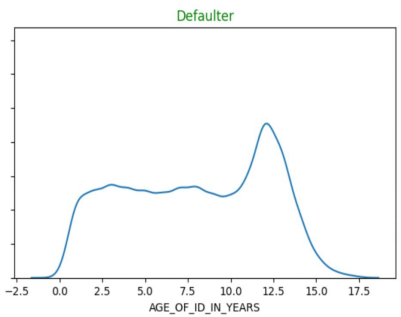


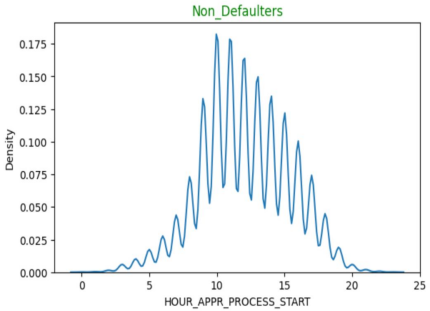
Analysing target Variable with Numerical Type Data

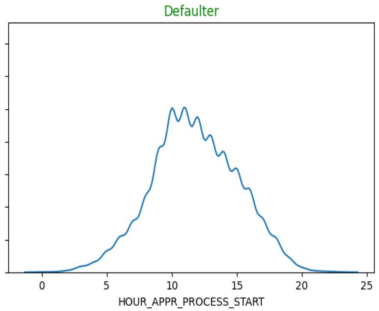


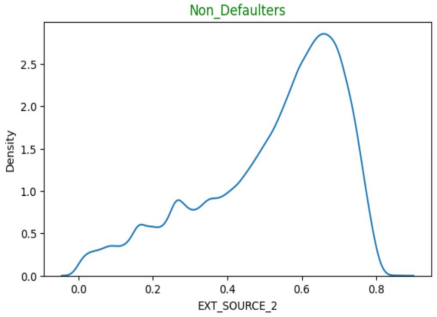


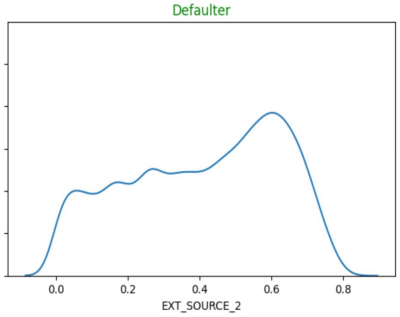




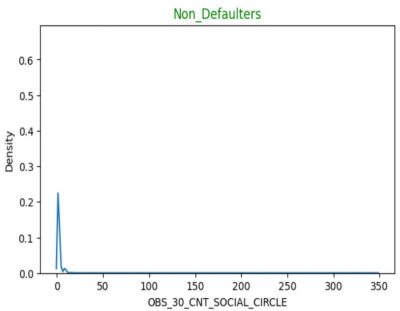


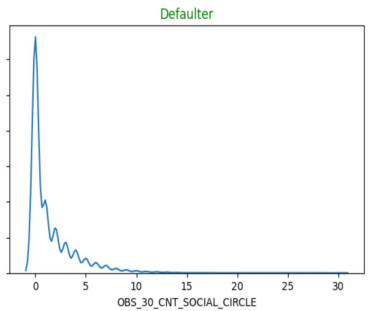


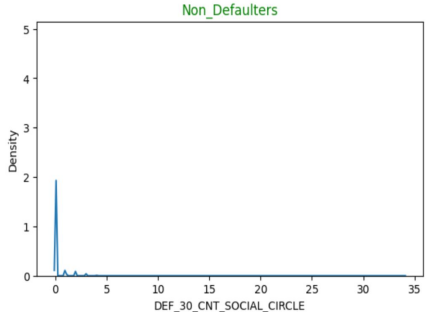


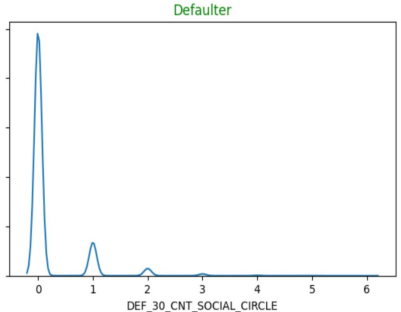


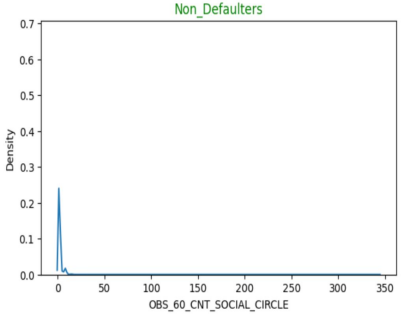
Analysing target Variable with Numerical Type Data

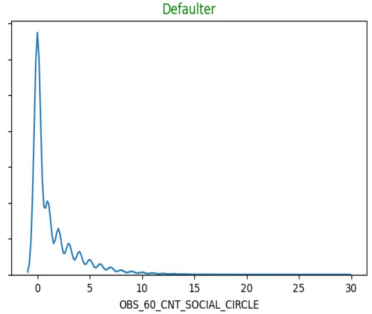


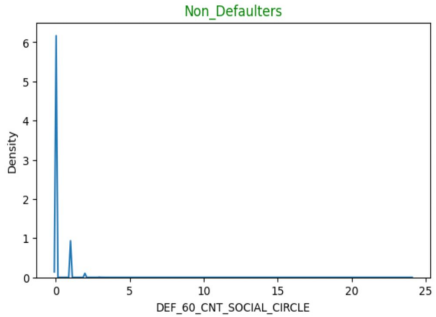


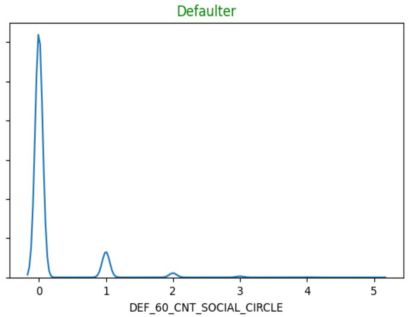




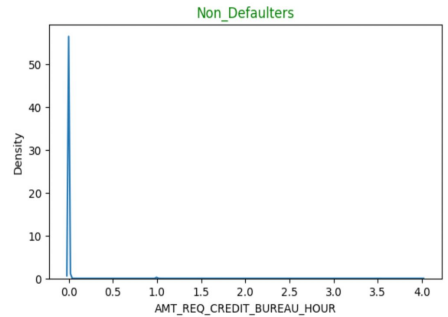


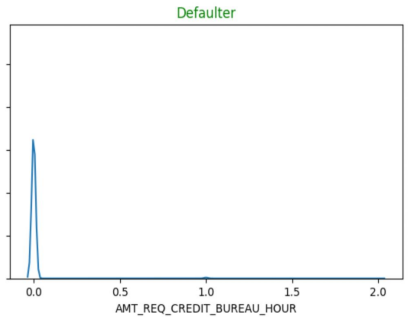


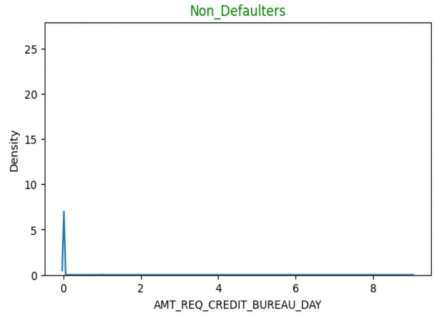


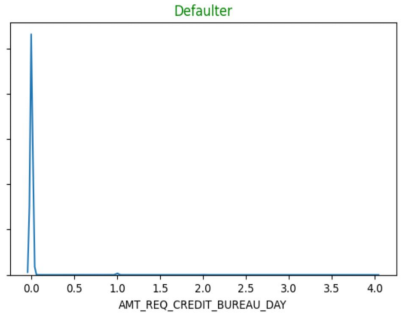


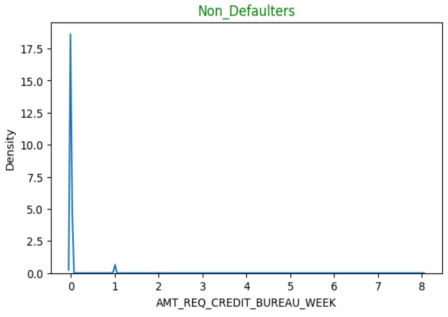
Analysing target Variable with Numerical Type Data

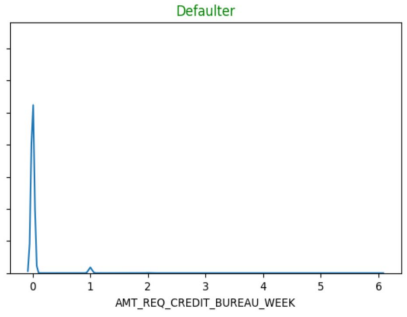


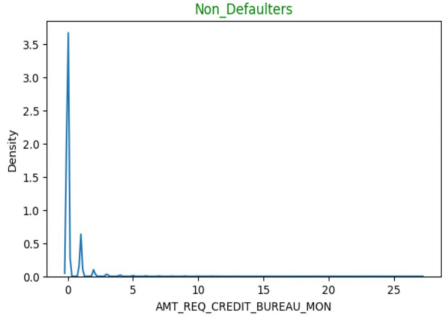


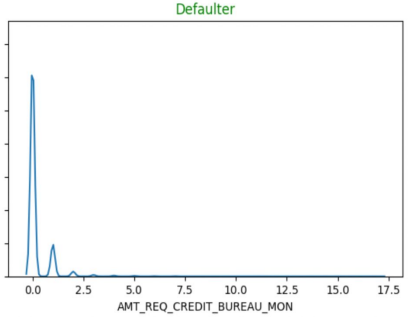




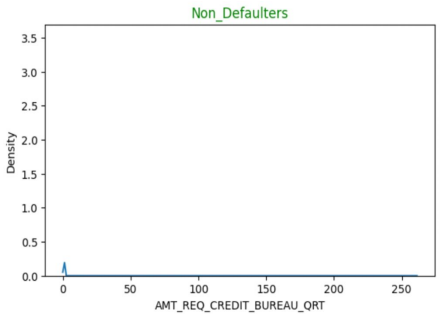


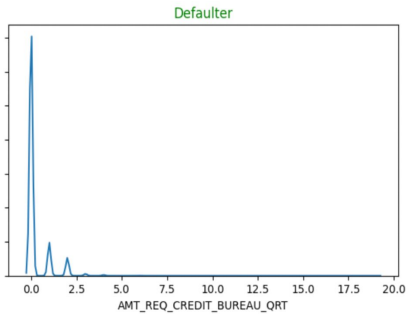


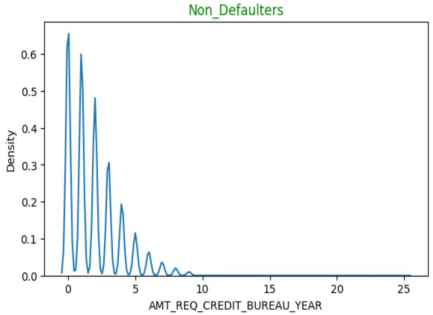


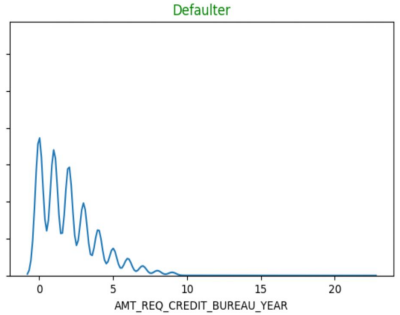


Analysing target Variable with Numerical Type Data





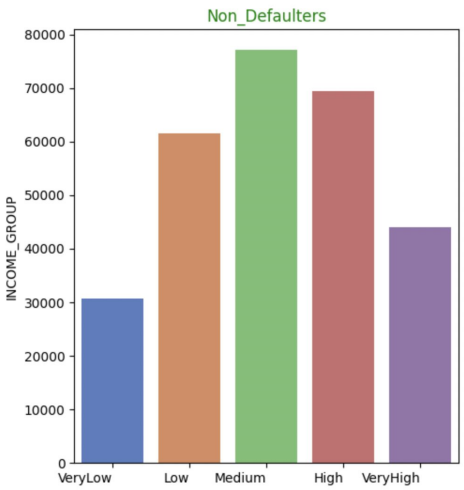


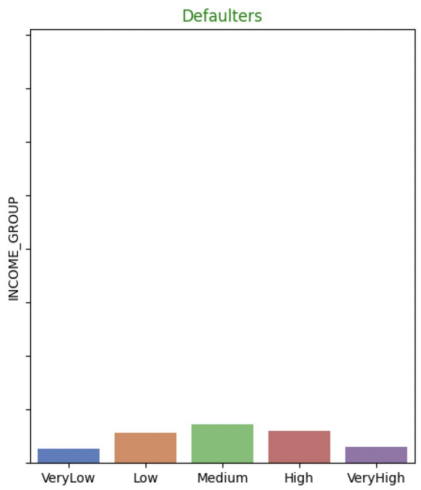


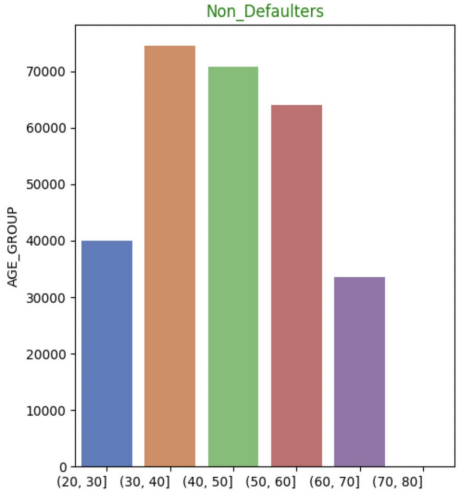
Inference :

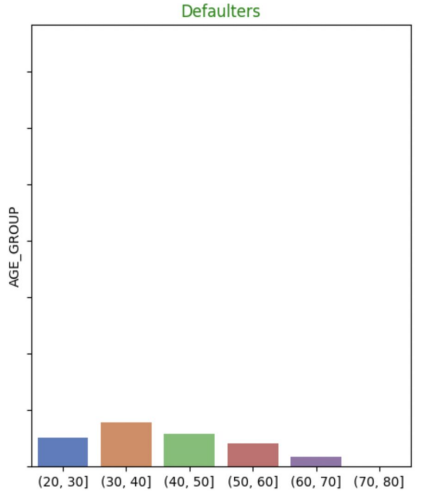
1. CNT_CHILDREN seems , it has no correlation of a person being defaulter. as higher share is of people having 0 children/No children.
2. AMT_CREDIT is lower for Defaulter , which shows lesser amount was lended to defaulters.
3. AMT_ANNUITY is higher for Defaulters, which may be a sign that means , higher Annuity/Installment amount lead person to be a defaulter.
4. AMT_GOODS_PRICE looks similar in distribution, so we cannot conclude anything from it.
5. Nothing can be inferred from REGION POPULATION RELATIVE as well.
6. AGE_IN_YEARS shows that maximum population of defaulter applicant lie in age bracket of 25-40 years.
7. Though there is not much difference in year of employment of Non_defaulters and defaulters, but as we can see Defaulters are more recenetly employed.(Years of employed has incorrect data also)
8. The Credit score obtained from EXT_SOURCE_2 is significantly lower for Defaulters in comparison to Non_Defaulters
9. OBS_30_CNT_SOCIAL_CIRCLE clearly shows that client's social surroundings observable 30 DPD is higher for Defaulters.
10. OBS_60_CNT_SOCIAL_CIRCLE clearly shows that client's social surroundings observable 60 DPD is higher for Defaulters.
11. similarly for DEF_30_CNT_SOCIAL_CIRCLE,DEF_60_CNT_SOCIAL_CIRCLE defaulted on 30 DPD and 60DPD are higher for Defaulters.
12. Credit Bureau ratings are also lesser for defaulters, this implies they are parallely looking for loan from other companies.

Checking with Binned Data of Age_group and income_group





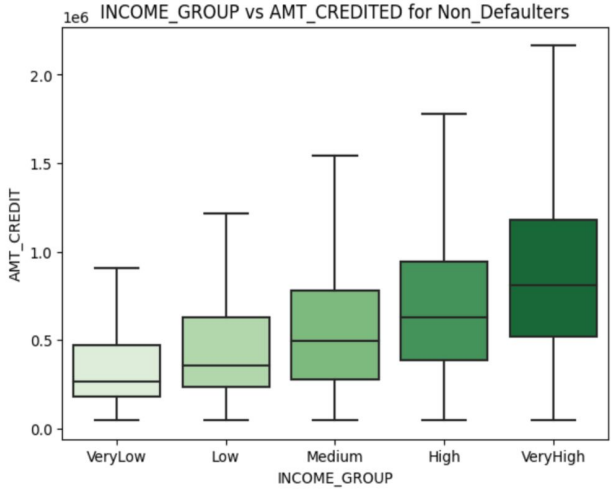


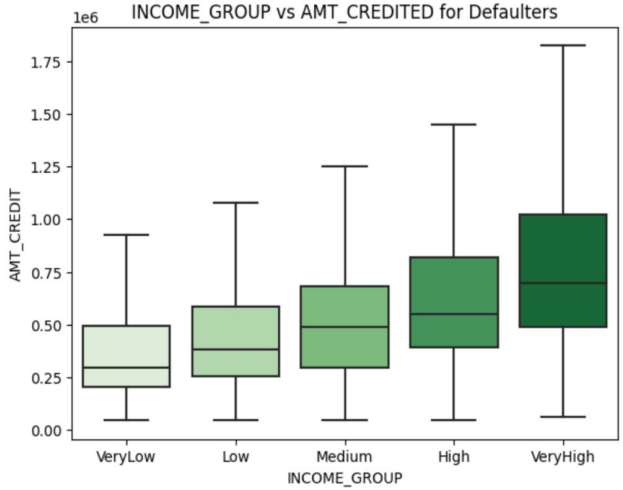


Inference :

1. Here also we can see Age group 30-40 years is higher in defaulting payments.
2. A Large share of Defaulters is hold by Low-High Income Grade People.

Checking with INCOME_GROUP , AMT_CREDIT vs Target

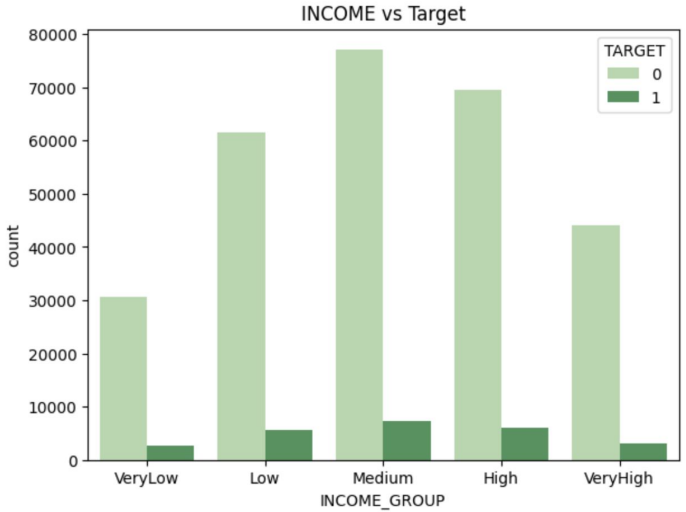


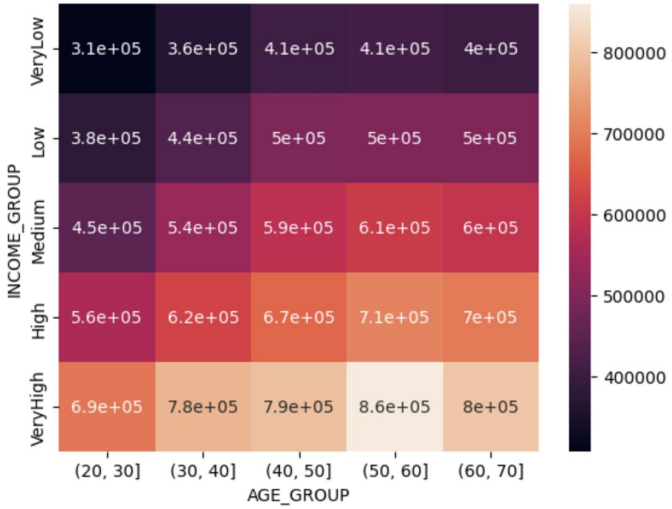


Inference :

we can see a large amount is credited to very_high Income Group in both graphs, let's check it's impact on Target variable.

very_high income group with Target variable





Inference :

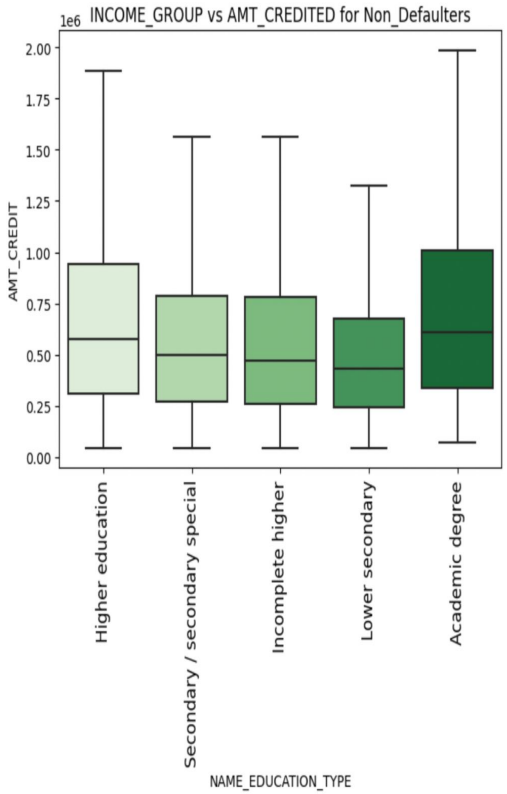
As it is observed from above two graphs that highest number of loans were given to Middle Income Group but the defaulters are larger in Very_high Income Group, and as a very high amount was lended to this group, so Defaulting by this group will lead to higher losses.

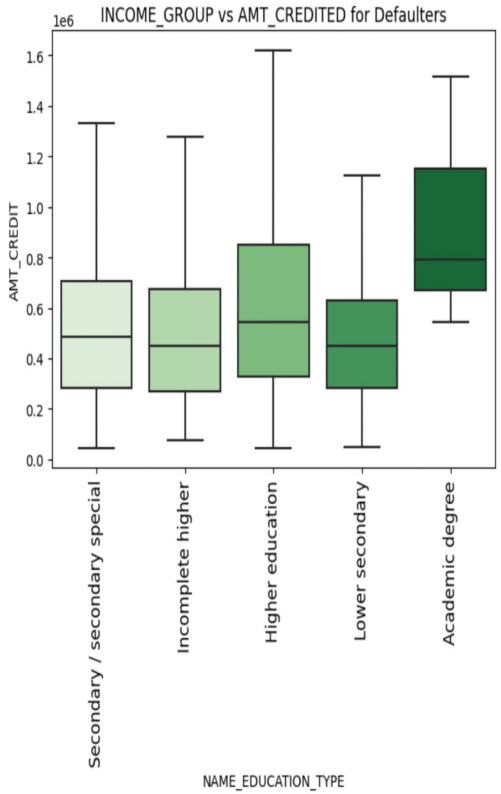
Checking with EDUCATION_TYPE , AMT_CREDIT vs Target

Inference :

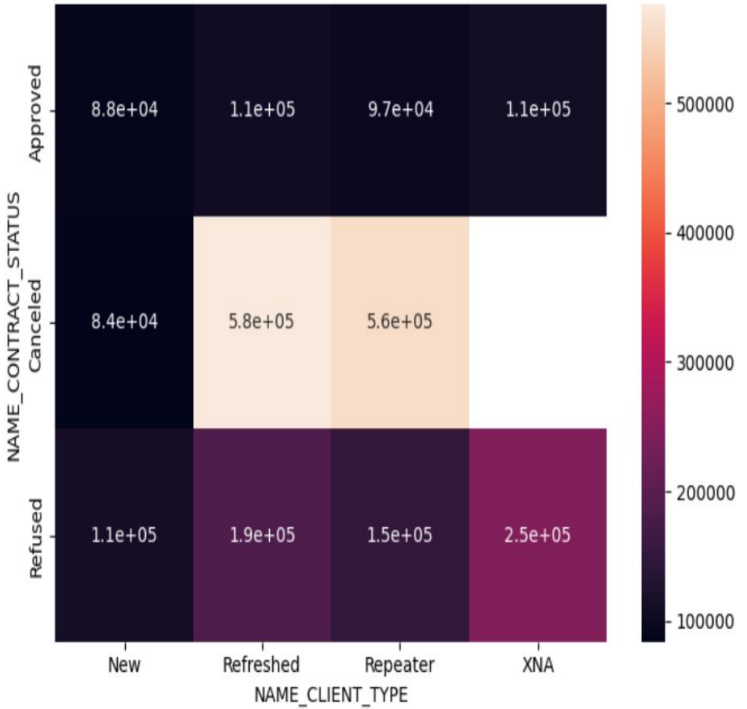
Though Highest defaulters are applicants holding ACADEMIC DEGREE but since their share in dataset is negligible in comparison to others, therefore their Impact is considered negligible.

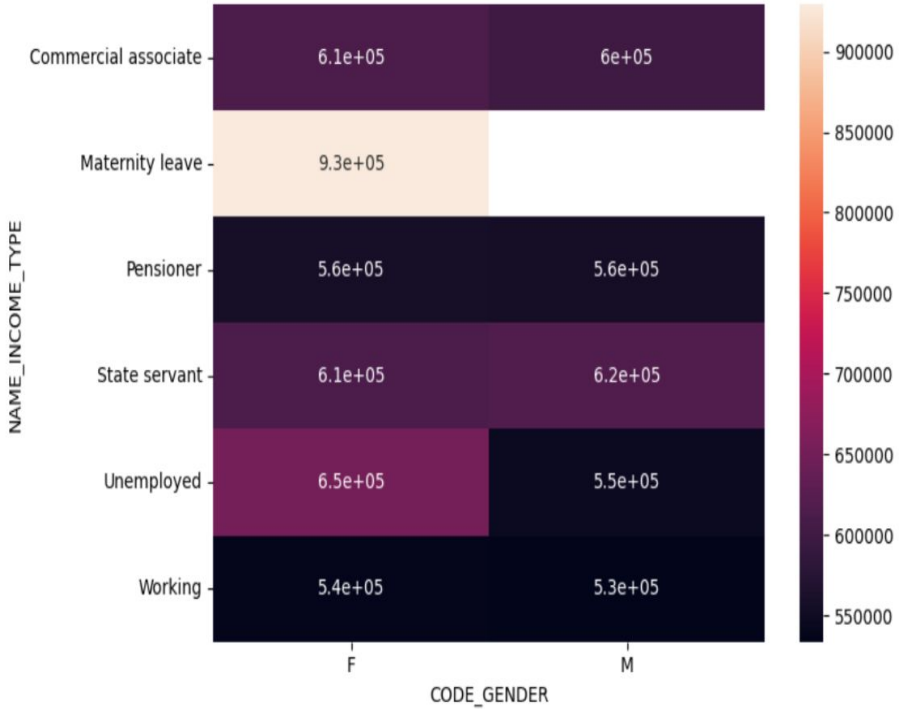
```
ADEMIC DEGREE 95
HIGHER EDUCATION 47514
INCOMPLETE HIGHER 6553
LOWER SECONDARY 1584
SECONADARY / Secondary Special 125333
```



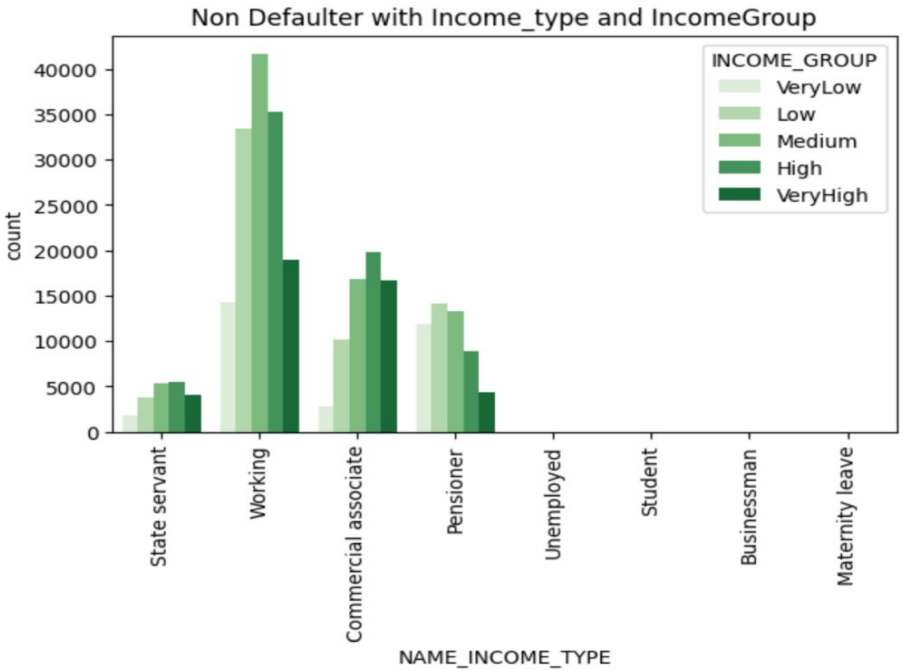


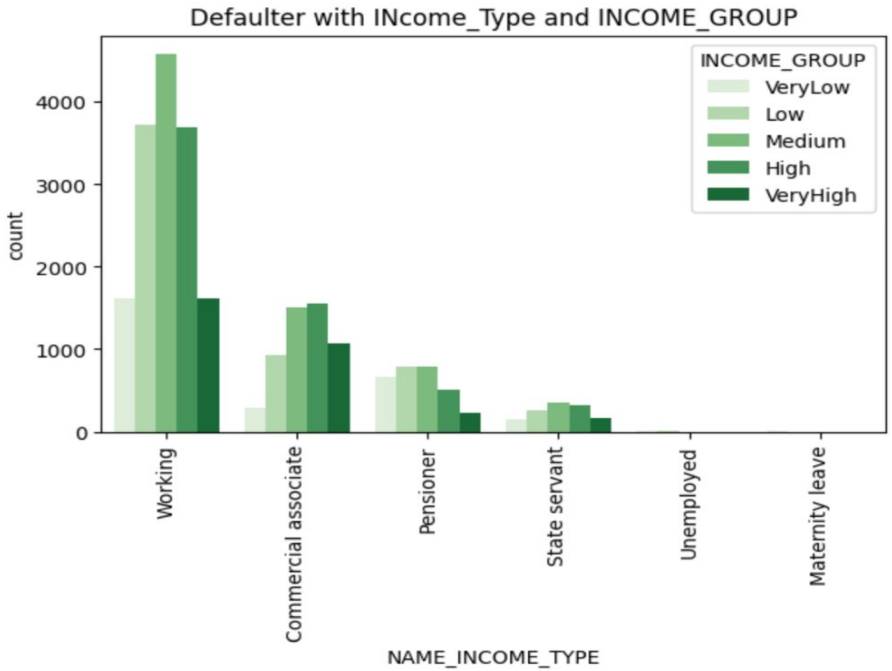
Relation between AMT_CREDIT ,GENDER_CODE, and INCOME_TYPE for Non_Defaulters and Defaulters





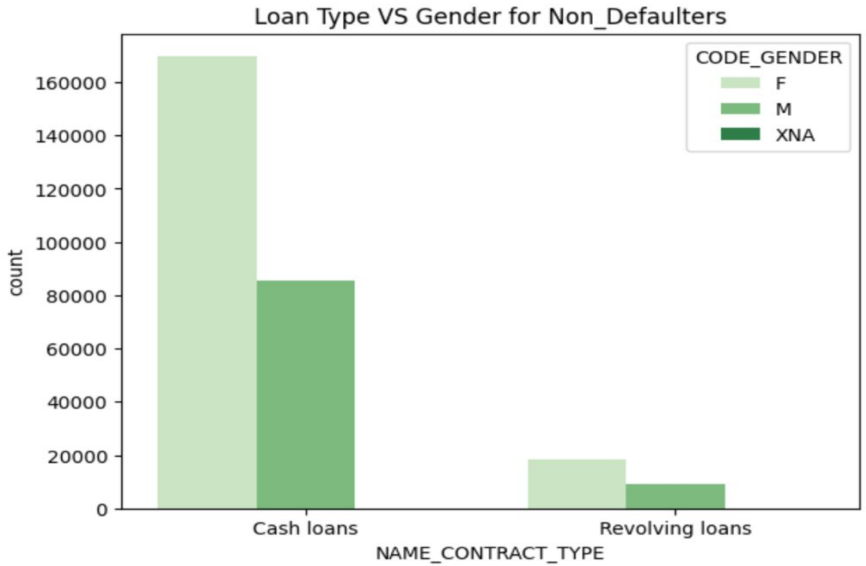
IncomeType with Income Group as Non Defaulter and Defaulter

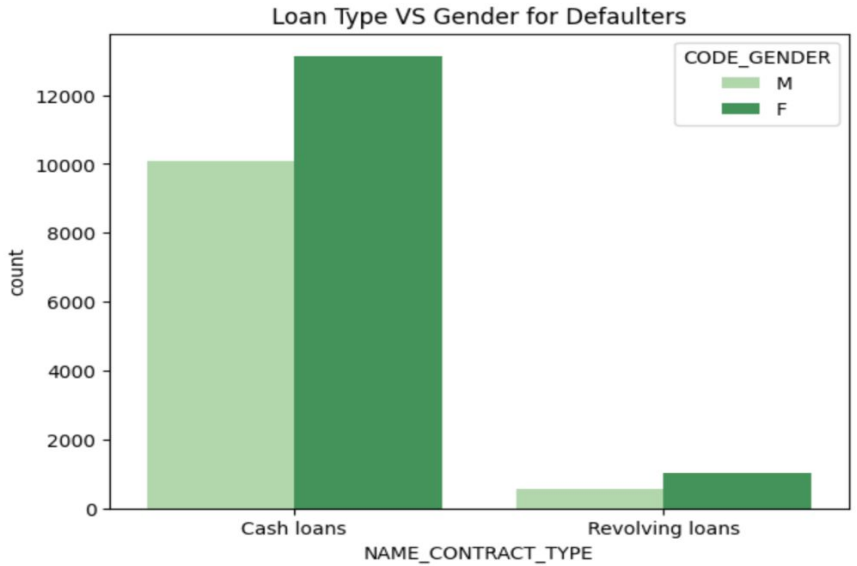




- Working class + Medium Income Group has higher number of Defaulters.

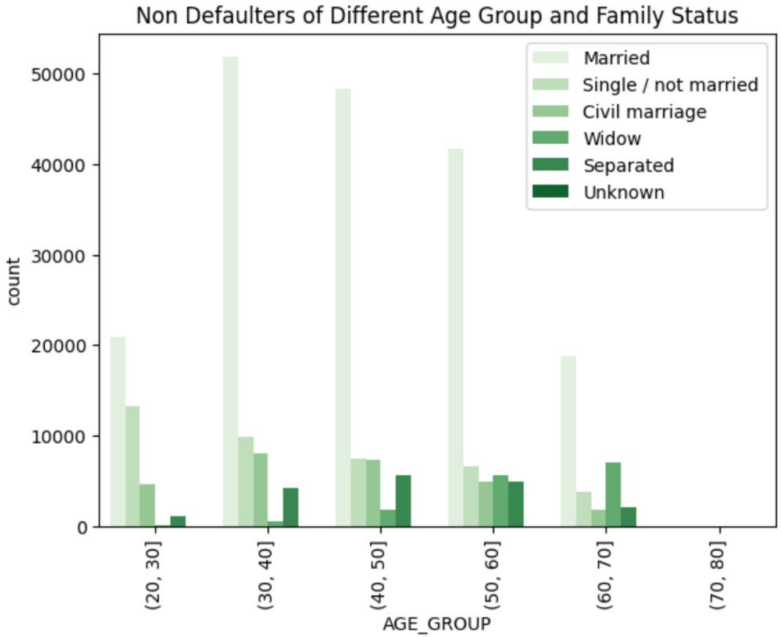
Checking Defaulted Type of loan with reference to CODE_GENDER

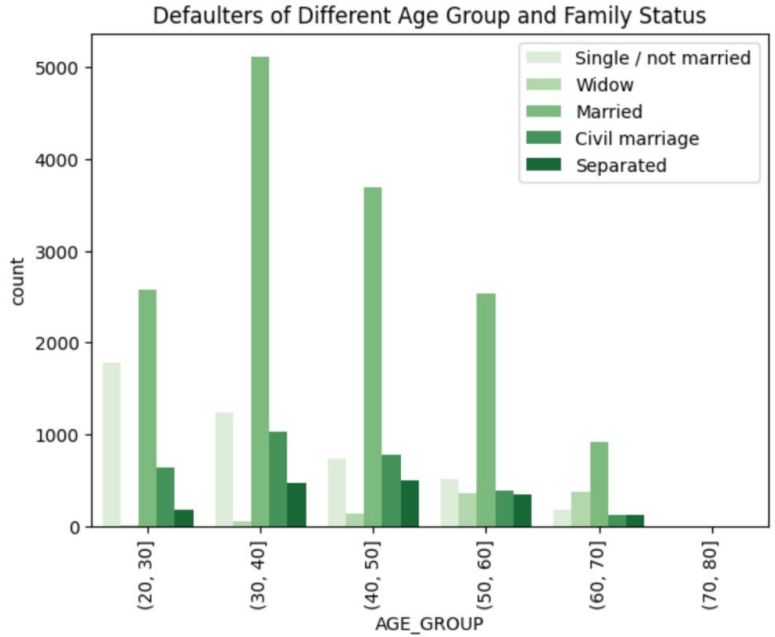




- It can be observed from above that MAles are more likely to Default on lOan Repayment than Females.
- From previous analysis we observed that Female have higher share as loan applicants.

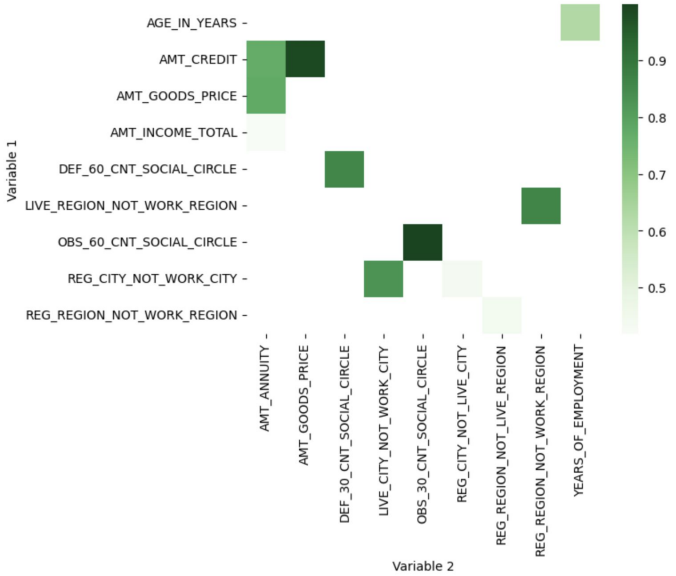
AGE_GROUP with NAME_FAMILY_STATUS as Non Defaulter and Defaulter



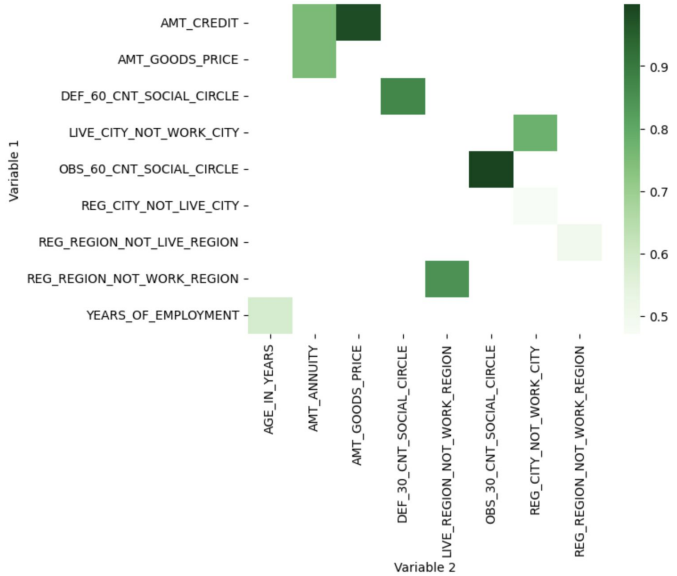


- Applicants from the age group 30-40 and 40-50 covers largest share in Defaulters group.

correlation of Non_Defaulters



correlation of Defaulters



- Above are the top Highly Correlated variable for Defaulters and Non_Defaulters.
- al these variables are selected above threshold correlation coefficient value 0.4.
- we can see OBS_60_CNT_SOCIAL_CIRCLE-OBS_30_CNT_SOCIAL_CIRCLE, AMT_CREDIT-AMT_GOODS_PRICE, DEF_60_CNT_SOCIAL_CIRCLE-DEF_30_CNT_SOCIAL_CIRCLE, REG REGION NOT WORK REGION-LIVE REGION NOT WORK REGION, LIVE_CITY_NOT_WORK_CITY-REG_CITY_NOT_WORK_CITY, AMT GOODS PRICE-AMT ANNUITY, AMT_CREDIT-AMT_ANNUITY,

Previous Application Data Variables

```
['SK ID PREV',
'SK ID CURR',
'NAME CONTRACT TYPE'
, 'AMT_ANNUITY',

'AMT APPLICATION',
'AMT CREDIT',
'AMT DOWN PAYMENT',
'AMT_GOODS_PRICE',

'WEEKDAY APPR_PROCES
S START',
'HOUR APPR_PROCESS_S
TART',

'FLAG LAST_APPL_PER_
CONTRACT',
'NFLAG_LAST_APPL_IN_
DAY',
       ……..

'DAYS FIRST DRAWING'
, 'DAYS FIRST DUE',
'DAYS LAST_DUE_1ST_V
ERSION',

'DAYS LAST DUE',
'DAYS TERMINATION',
'NFLAG INSURED_ON_AP
PROVAL']
```

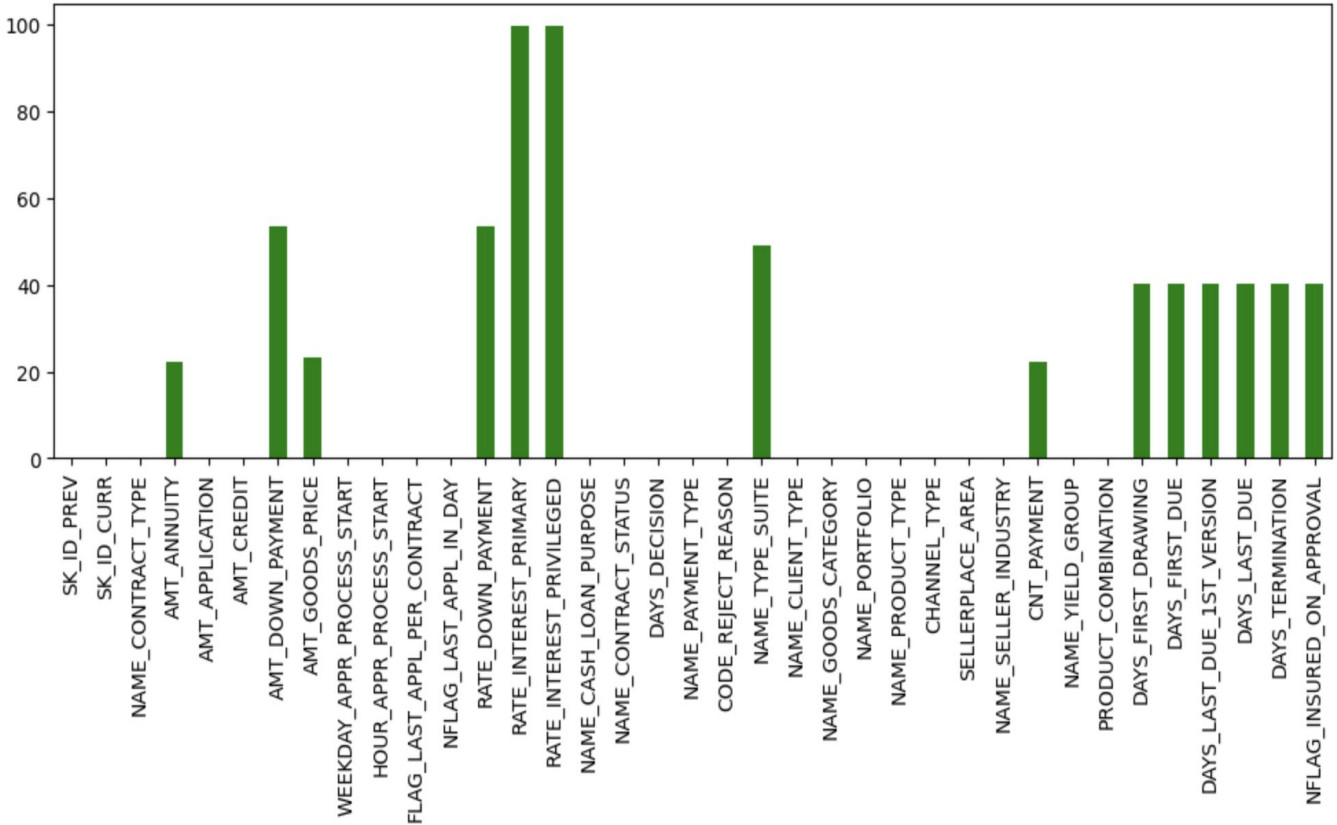
```
prev_app.head(10)
```

	SK_ID_PREV	SK_ID_CURR	NAME_CONTRACT_TYPE	AMT_ANNUITY	AMT_APPLICATION	AMT_CREDIT	AMT_DOWN_PAYMENT	AMT_GOODS_PRICE	WEEKDAY_APPR_PROCESS_START	HOUR_APPR_PROCESS_START	FLAG_LAST_APP
0	2030495	271877	Consumer loans	1730.430	17145.0	17145.0	0.0	17145.0	SATURDAY	15	
1	2802425	108129	Cash loans	25188.615	607500.0	679671.0	NaN	607500.0	THURSDAY	11	
2	2523466	122040	Cash loans	15060.735	112500.0	136444.5	NaN	112500.0	TUESDAY	11	
3	2819243	176158	Cash loans	47041.335	450000.0	470790.0	NaN	450000.0	MONDAY	7	
4	1784265	202054	Cash loans	31924.395	337500.0	404055.0	NaN	337500.0	THURSDAY	9	
5	1383531	199383	Cash loans	23703.930	315000.0	340573.5	NaN	315000.0	SATURDAY	8	
6	2315218	175704	Cash loans	NaN	0.0	0.0	NaN	NaN	TUESDAY	11	
7	1656711	296299	Cash loans	NaN	0.0	0.0	NaN	NaN	MONDAY	7	
8	2367563	342292	Cash loans	NaN	0.0	0.0	NaN	NaN	MONDAY	15	
9	2579447	334349	Cash loans	NaN	0.0	0.0	NaN	NaN	SATURDAY	15	

Missing Value Examination

These many columns having missing values greater than 20 %.

Shape:
(1670214- Rows , 37 - Columns)



Missing Value Examination

We deleted NaNs in AMT_ANNUITY ,AMT_GOODS_PRICE and NAME_SUIT_TYPE, as this dataset is much larger than Application_dataset.

We filtered out all columns having missing values greater than 20%

```
[357] col_20_miss=prev_app.columns[(prev_app.isnull().sum()*100/prev_app.shape[0])>20]
      col_20_miss

      Index(['AMT_DOWN_PAYMENT', 'RATE_DOWN_PAYMENT', 'RATE_INTEREST_PRIMARY',
             'RATE_INTEREST_PRIVILEGED', 'DAYS_FIRST_DRAWING', 'DAYS_FIRST_DUE',
             'DAYS_LAST_DUE_1ST_VERSION', 'DAYS_LAST_DUE', 'DAYS_TERMINATION',
             'NFLAG_INSURED_ON_APPROVAL'],
            dtype='object')
```

Now we will check if any of these columns can be included in our analysis.

We filter out the columns which will have NAME_CONTRACT_STATUS == Approved / Not Approved.

```
for i in col_20_miss:
  a=prev_app[(prev_app['NAME_CONTRACT_STATUS']!="Approved")][i].value_counts(dropna=True)
  print(i,a.shape)
```

```
AMT_DOWN_PAYMENT (6795,)
RATE_DOWN_PAYMENT (17648,)
RATE_INTEREST_PRIMARY (0,)
RATE_INTEREST_PRIVILEGED (0,)
DAYS_FIRST_DRAWING (0,)
DAYS_FIRST_DUE (0,)
DAYS_LAST_DUE_1ST_VERSION (0,)
DAYS_LAST_DUE (0,)
DAYS_TERMINATION (0,)
NFLAG_INSURED_ON_APPROVAL (0,)
```

Missing Value Examination

There is no such column where NAME_CONTRACT_STATUS is not Approved, so we will drop all of those columns.

Some additional columns were also dropped as they were not contributing in our analysis.
['HOUR_APPR_PROCESS_START','NFLAG_LAST_APPL_IN_DAY','DAYS_DECISION','SELLERPLACE_AREA']

Now our Dataset was left with shape (839428 , 23) with zero NaNs.

Object Type Variable

```
# Object variables

for i in prev_app.columns:
  if prev_app[i].dtypes=='object':
    print(prev_app[i].value_counts(normalize=True))
    print('\n\n')
```

```
Consumer loans     0.546538
Cash loans         0.386979
Revolving loans    0.066483
Name: NAME_CONTRACT_TYPE, dtype: float64



SATURDAY     0.153656
FRIDAY       0.149557
WEDNESDAY    0.147901
TUESDAY      0.147241
MONDAY       0.146559
THURSDAY     0.145114
SUNDAY       0.109973
Name: WEEKDAY_APPR_PROCESS_START, dtype: float64



Y    0.997592
N    0.002408
Name: FLAG_LAST_APPL_PER_CONTRACT, dtype: float64
```

```
XAP                                  0.613021
XNA                                  0.336678
Repairs                              0.017471
Other                                0.011104
Urgent needs                         0.005006
Buying a used car                    0.002231
Everyday expenses                    0.002066
Building a house or an annex         0.001970
Medicine                             0.001656
Payments on other loans              0.001502
Education                            0.001337
Buying a new car                     0.000909
Journey                              0.000904
Purchase of electronic equipment     0.000814
Buying a home                        0.000690
Wedding / gift / holiday             0.000653
Furniture                            0.000485
Car repairs                          0.000436
Buying a holiday home / land         0.000403
Business development                 0.000299
Gasification / water supply          0.000193
Buying a garage                      0.000101
Hobby                                0.000045
Money for a third person             0.000015
Refusal to name the goal             0.000011
Name: NAME_CASH_LOAN_PURPOSE, dtype: float64



Approved     0.791392
Refused      0.196773
Canceled     0.011834
Name: NAME_CONTRACT_STATUS, dtype: float64



Cash through the bank                        0.822317
XNA                                          0.170057
Non-cash from your account                   0.006739
Cashless from the account of the employer    0.000886
Name: NAME_PAYMENT_TYPE, dtype: float64
```

```
XAP          0.803223
HC           0.117879
LIMIT        0.037510
SCO          0.031220
SCOFR        0.004850
VERIF        0.003189
XNA          0.002107
SYSTEM       0.000021
Name: CODE_REJECT_REASON, dtype: float64



Unaccompanied      0.600422
Family             0.250467
Spouse, partner    0.078680
Children           0.036775
Other_B            0.020491
Other_A            0.010568
Group of people    0.002597
Name: NAME_TYPE_SUITE, dtype: float64



Repeater     0.683630
New          0.234339
Refreshed    0.081129
XNA          0.000902
Name: NAME_CLIENT_TYPE, dtype: float64
```

Object Type Variable

```
XNA                          0.461804
Mobile                       0.124208
Consumer Electronics         0.110989
Audio/Video                  0.092727
Computers                    0.086665
Furniture                    0.042166
Photo / Cinema Equipment     0.019249
Clothing and Accessories     0.018355
Construction Materials       0.017563
Auto Accessories             0.005106
Jewelry                      0.003762
Homewares                    0.003134
Vehicles                     0.002349
Sport and Leisure            0.002172
Office Appliances            0.002154
Other                        0.001949
Gardening                    0.001806
Medical Supplies             0.001723
Tourism                      0.000871
Medicine                     0.000592
Direct Sales                 0.000337
Fitness                      0.000112
Weapon                       0.000064
Additional Service           0.000057
Education                    0.000057
Insurance                    0.000027
House Construction           0.000001
Name: NAME_GOODS_CATEGORY, dtype: float64
```

```
POS      0.546538
Cash     0.386979
Cards    0.066483
Name: NAME_PORTFOLIO, dtype: float64
```


```
POS      0.546538
Cash     0.386979
Cards    0.066483
Name: NAME_PORTFOLIO, dtype: float64
```


```
XNA         0.546538
x-sell      0.338739
walk-in     0.114723
Name: NAME_PRODUCT_TYPE, dtype: float64
```

```
Credit and cash offices       0.337207
Country-wide                  0.327304
Stone                         0.169409
Regional / Local              0.091391
AP+ (Cash loan)               0.036793
Contact center                0.032470
Channel of corporate sales    0.005394
Car dealer                    0.000032
Name: CHANNEL_TYPE, dtype: float64
```

```
XNA                      0.412407
Consumer electronics     0.344217
Connectivity             0.142215
Furniture                0.044586
Construction             0.020298
Clothing                 0.018694
Industry                 0.011602
Auto technology          0.003037
Jewelry                  0.001743
MLM partners             0.000789
Tourism                  0.000413
Name: NAME_SELLER_INDUSTRY, dtype: float64
```

```
middle         0.301908
low_normal     0.277738
high           0.270512
low_action     0.083359
XNA            0.066483
Name: NAME_YIELD_GROUP, dtype: float64
```

```
POS household with interest       0.243007
Cash X-Sell: low                  0.125700
POS mobile with interest          0.119671
Cash X-Sell: middle               0.115126
POS household without interest    0.079051
POS industry with interest        0.073928
Cash Street: high                 0.043200
Cash X-Sell: high                 0.042531
Card X-Sell                       0.036886
Cash Street: low                  0.035082
Card Street                       0.029598
Cash Street: middle               0.025340
POS other with interest           0.014538
POS industry without interest     0.009231
POS mobile without interest       0.006049
POS others without interest       0.001061
Name: PRODUCT_COMBINATION, dtype: float64
```

Object Type Variable

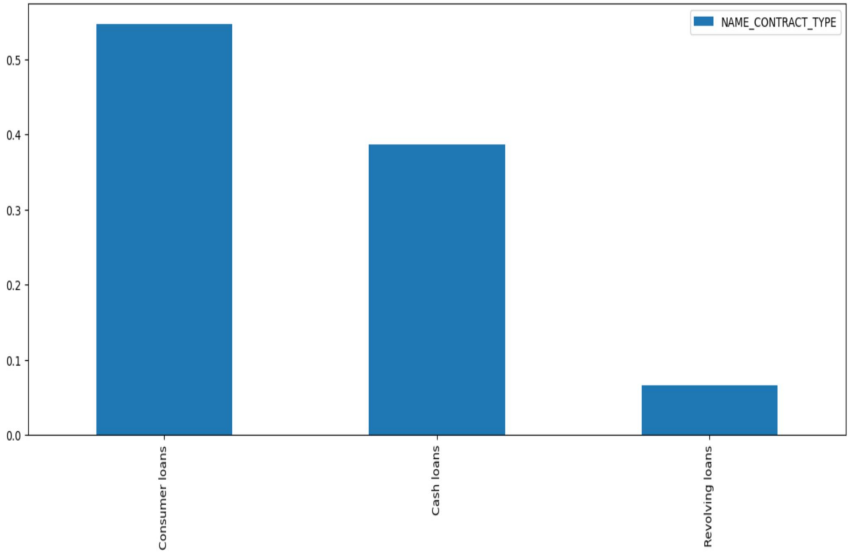
we ‘ve observed that many columns having values high amount of XNA, XAP, NNA , which indicate no value or equivalent to Nan. So we drop all these columns as well.

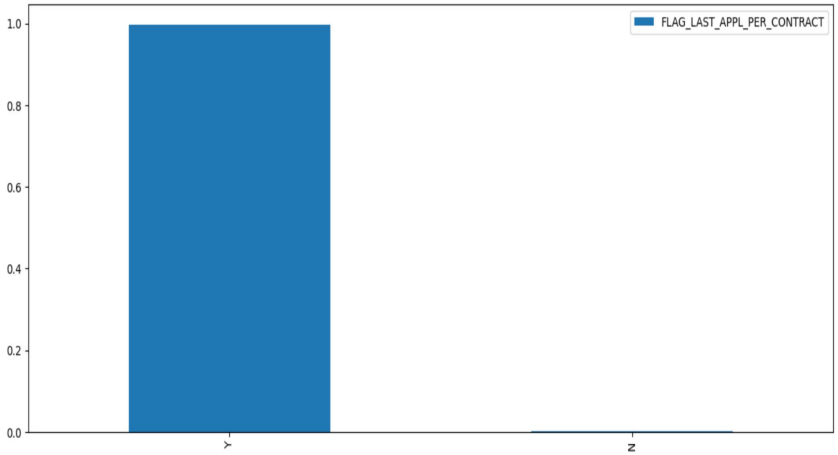
```
# Deleting columns that have XNA, XAP, NNA in high amount and columns
# that are not needed in our analysis.

collist=['WEEKDAY_APPR_PROCESS_START', 'NAME_PRODUCT_TYPE',
         'NAME_CASH_LOAN_PURPOSE', 'NAME_GOODS_CATEGORY','NAME_PAYMENT_TYPE'
         ,'CODE_REJECT_REASON']
prev_app.drop(axis=1, columns=collist, inplace=True)
prev_app.shape
```

```
(839428, 17)
```

Univariate Analysis

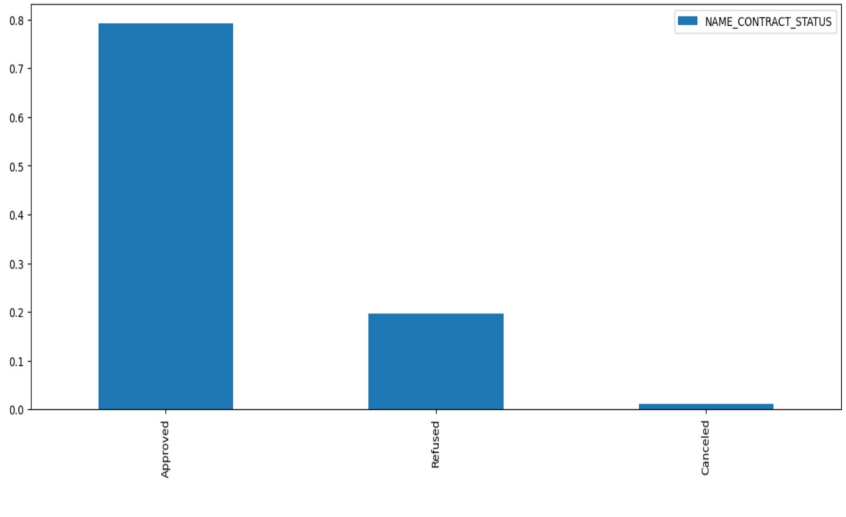
Categorical Nominal- all object type variables:

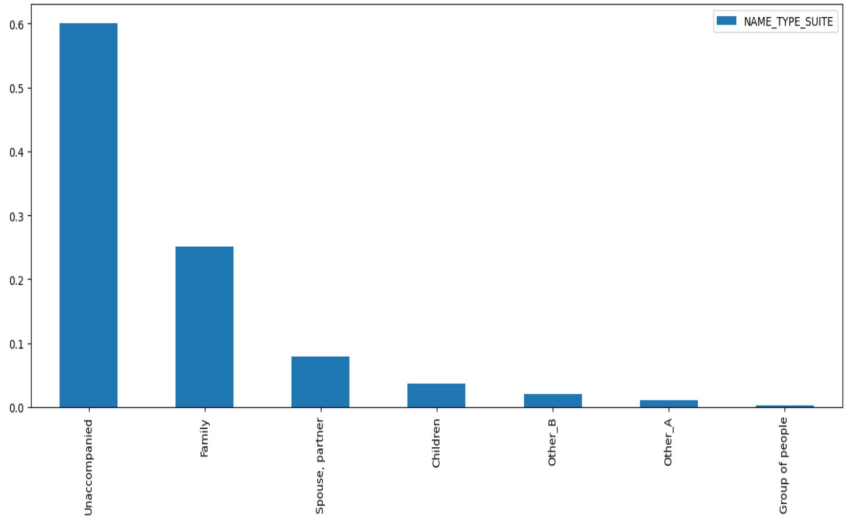




Univariate Analysis

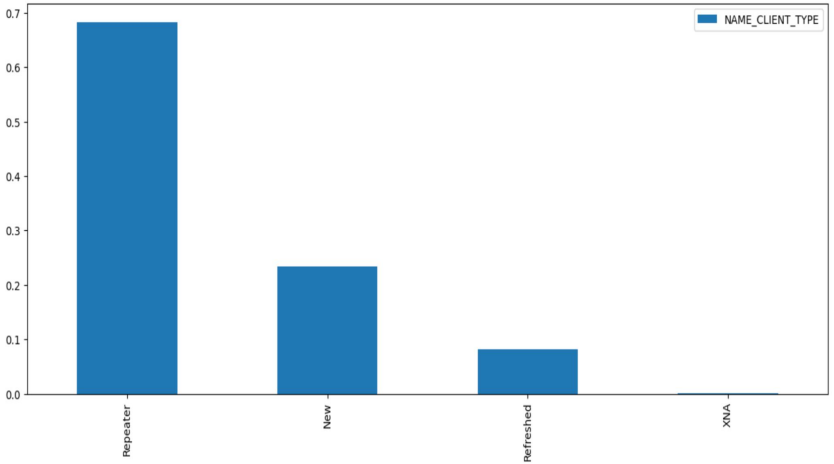
Categorical Nominal- all object type variables:

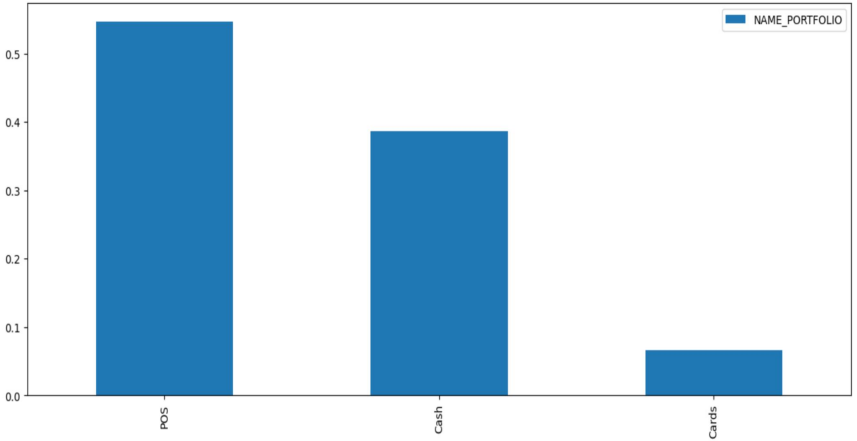




Univariate Analysis

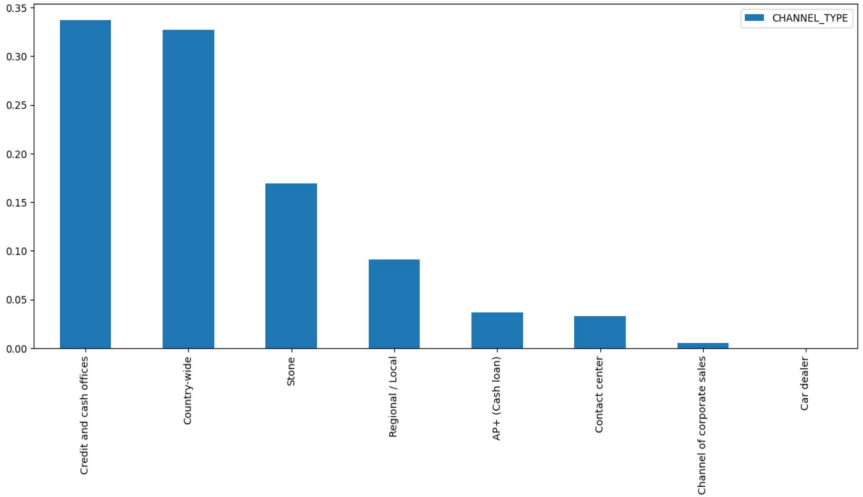
Categorical Nominal- all object type variables:

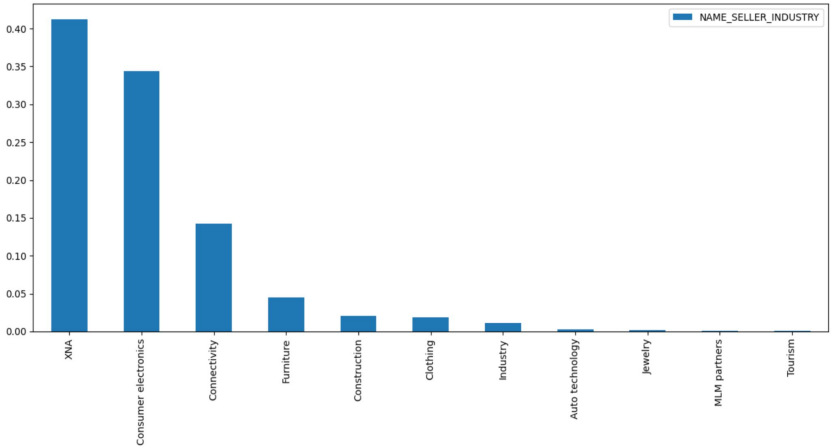




Univariate Analysis

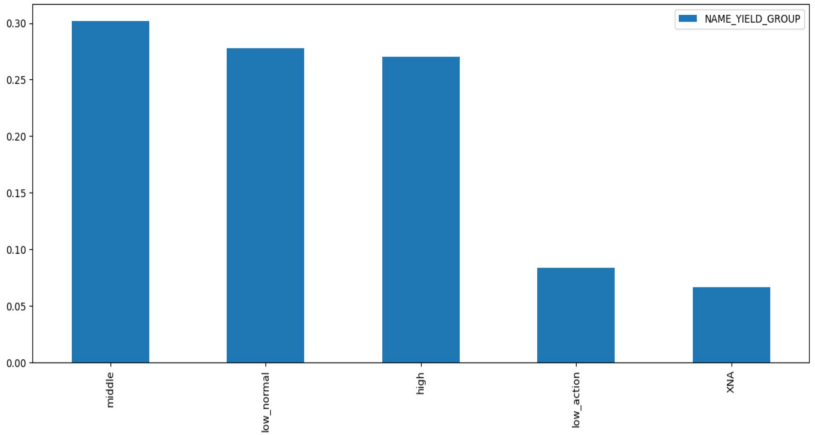
Categorical Nominal- all object type variables:

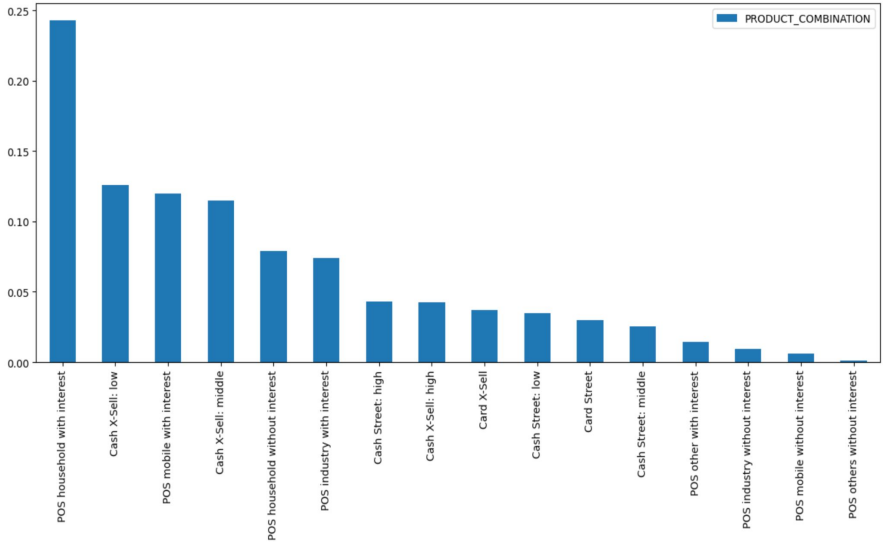




Univariate Analysis

Categorical Nominal- all object type variables:

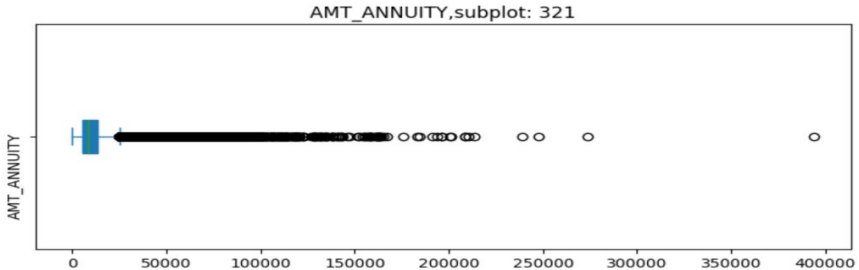


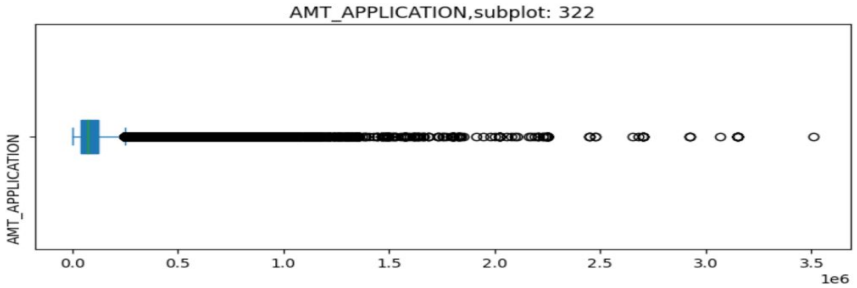


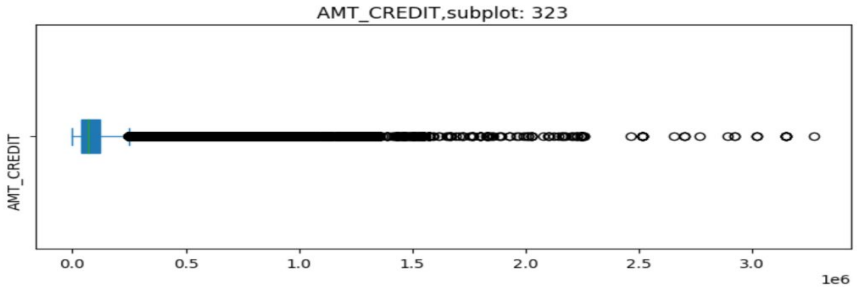
Inference :

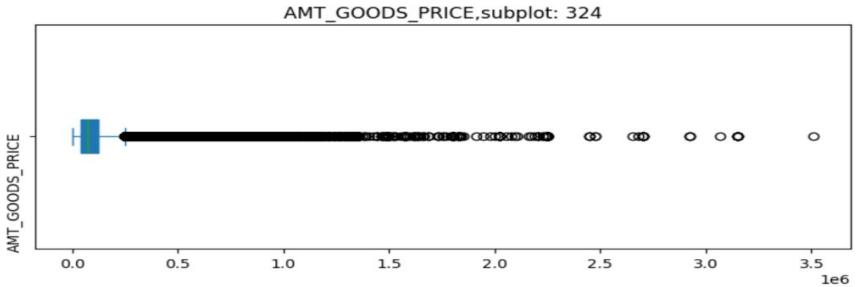
1. Where application data has only two types of value for loan types, this dataframe consists one additional loan type i.e. Consumer Loan which is 55%, Cash Loans are around 37% and revolving are 8%.
2. Approved are 79% , Refused are 19.6% and Cancelled are 1.4% approx.
3. in NAME_CLIENT_TYPE 67% applicants are repeaters.
4. From NAME_PORTFOLIO , 54.65% of users took loan to purchase something. whereas Cash loans are 38.69% and rest though cards.

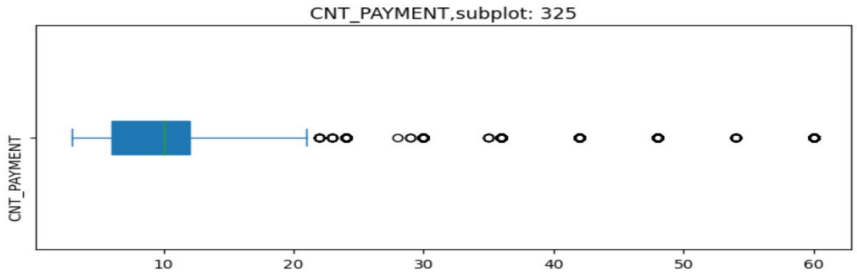
Outlier detection (Boxplot)



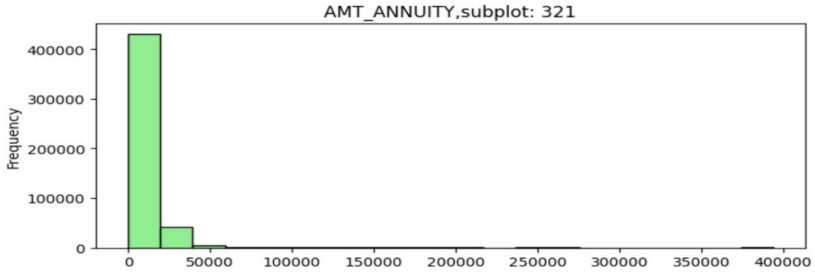


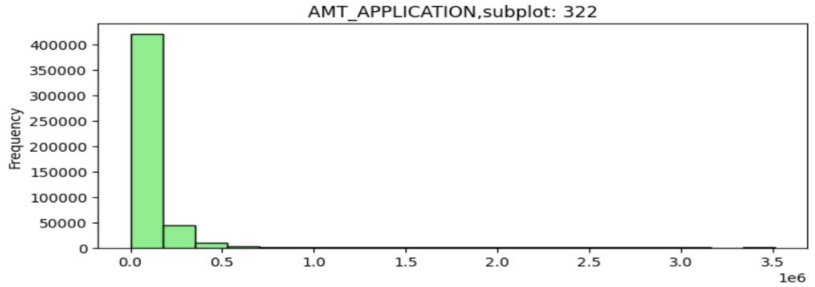


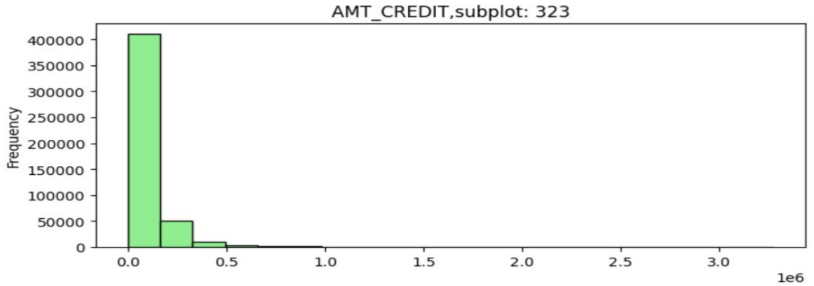


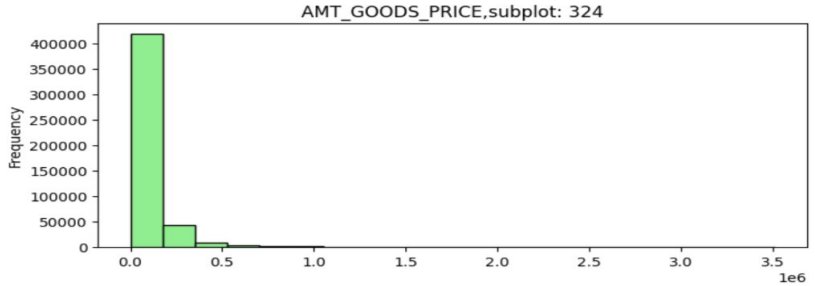


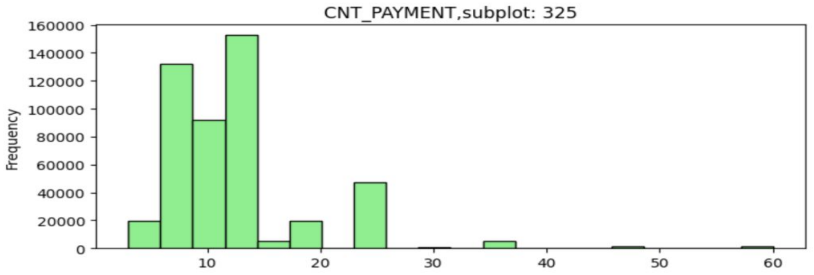
Outlier detection (Barplot)











Outlier detection (Barplot)

- By seeing both boxplot distribution and histplot distribution, we conclude that all of these continuous variable have high amount of Outliers.
- Though these points seems as outlier, but actually their distance from 97 percentile will be the deciding factor.
- We used zscore for outlier determination.

```
# we use zscore here to determine outliers above 97%.

for i in prev_app.columns:
    if prev_app[i].dtypes=="float64" or prev_app[i].dtypes=="float32":
        print( i, ":", end= " ")
        print(prev_app.iloc[np.where(np.abs(stat.zscore(prev_app[i]))>2))].shape, end=" ")
        print("Percentage of Outlier rows: ", round((prev_app.iloc[np.where
         (np.abs(stat.zscore(prev_app[i]))>2))].shape[0]/prev_app.shape[0]*100), 2))
```

```
AMT_ANNUITY : (18677, 16) Percentage of Outlier rows:  3.92
AMT_APPLICATION : (15667, 16) Percentage of Outlier rows:  3.29
AMT_CREDIT : (15374, 16) Percentage of Outlier rows:  3.23
AMT_GOODS_PRICE : (15667, 16) Percentage of Outlier rows:  3.29
CNT_PAYMENT : (8744, 16) Percentage of Outlier rows:  1.83
```

- These outliers can be treated either by dropping or by imputing median. As this metrics is less affected by outliers.

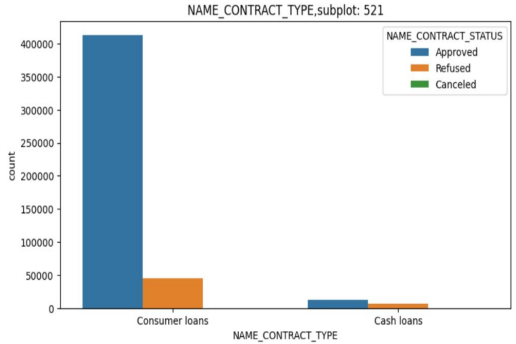
Bivariate Analysis

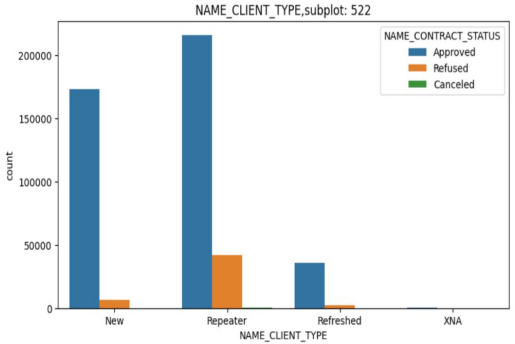
Performing bivariate analysis keeping 'NAME_CONTRACT_STATUS' in common , as this will let us know, which applications were approved, refused, cancelled. etc

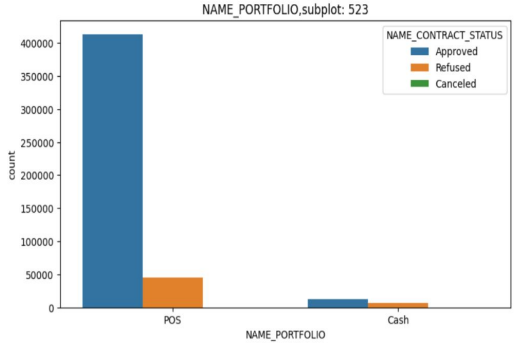
we will observe only over three columns 'NAME_CONTRACT_TYPE','NAME_CLIENT_TYPE','NAME_PORTFOLIO'

Inference:

1. Consumer loans are having highest rate of Approvals.
2. Refusal of loan is higher for CAsh loan than Consumer loans.
3. The company has higher number of repeating customers with approved as well as refused loans.
4. Higher Approval rate in POS shows consumer loans only.

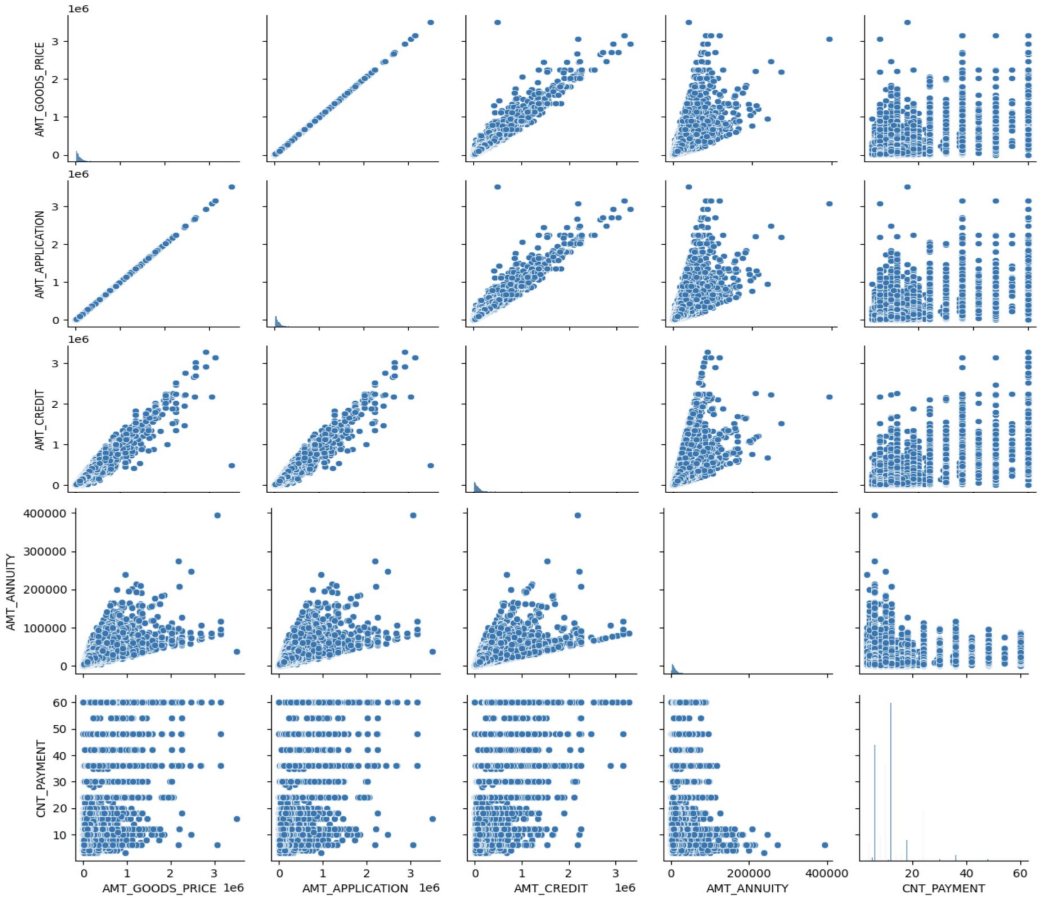






Correlation Analysis

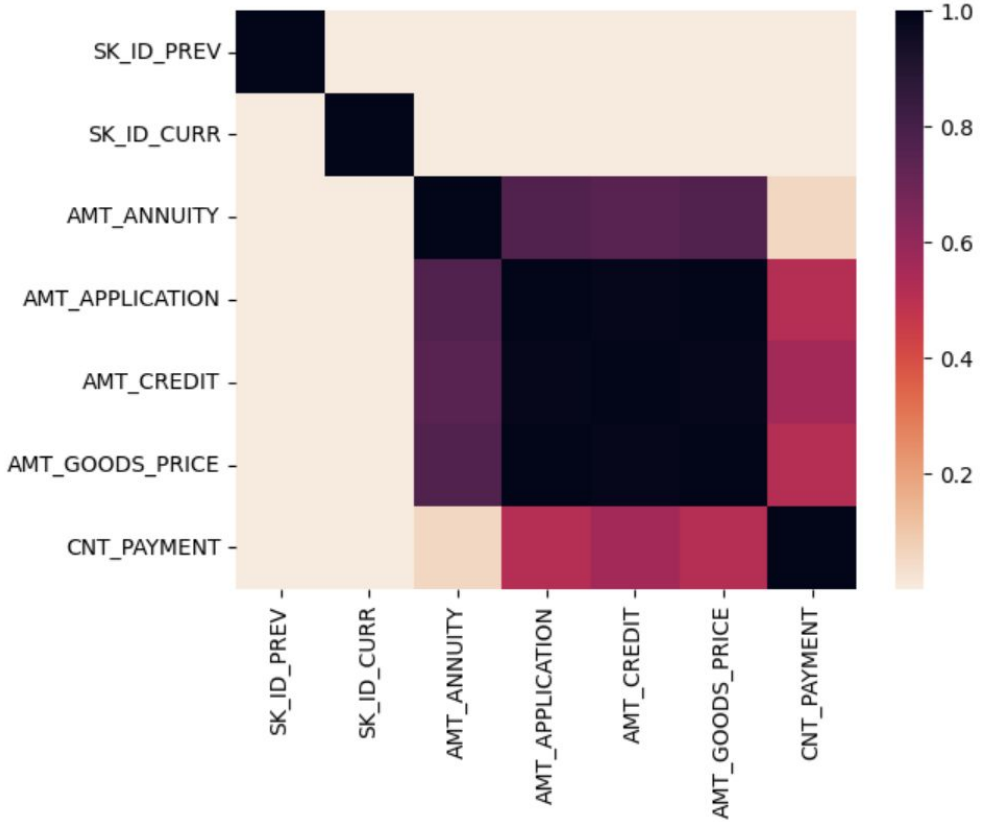
Pairplot analysis of columns appeared in correlation Dataframe That are 'AMT_GOODS_PRICE' ,'AMT_APPLICATION ','AMT_CREDIT' ,'AMT_ANNUITY' ,'CNT_PAYMENT'



Correlation Analysis

Inference :

1. It is observed that AMT_ANNUITY is equally correlated with AMT_GOODS_PRICE, AMT_APPLICATION,AMT_CREDIT.
2. CNT_PAYMENT doesn't show any significant correlation.
3. AMT_CREDIT is highly correlated with AMT_GOODS_PRICE, AMT_APPLICATION, and this is obvious,if goods price are higher than loan application and amount credit will automatically rise. we have already seen that most of the credit was taken to purchase something, ie. under consumer loan category.

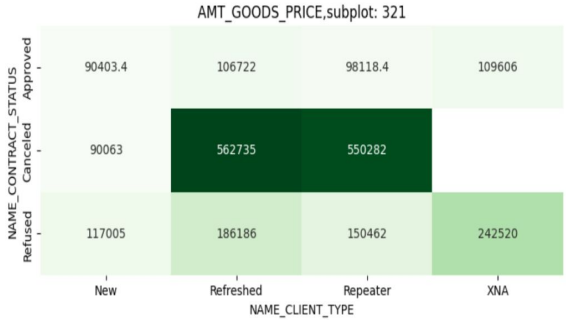


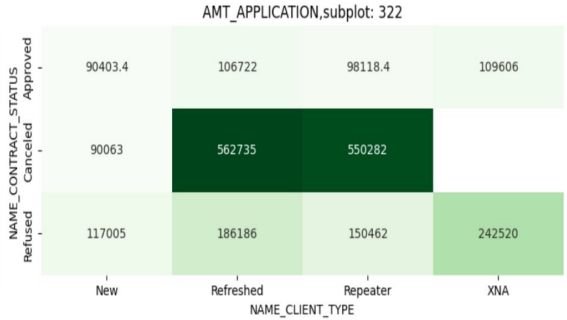
Multivariate Analysis

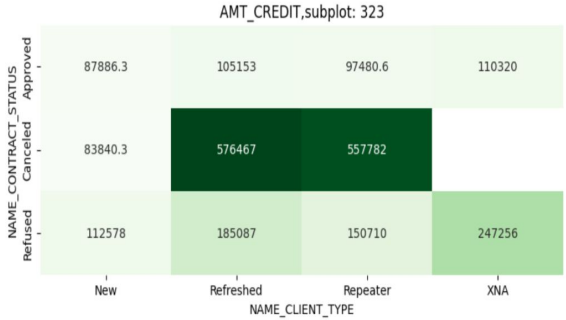
WE WILL Check status of Application ['Approved','Refused','Cancelled'] and client type 'NEW' or 'REPEATER' wrt all The highly Correlated columns That are ['AMT_GOODS_PRICE', 'AMT_APPLICATION','AMT_CREDIT']

Inference :

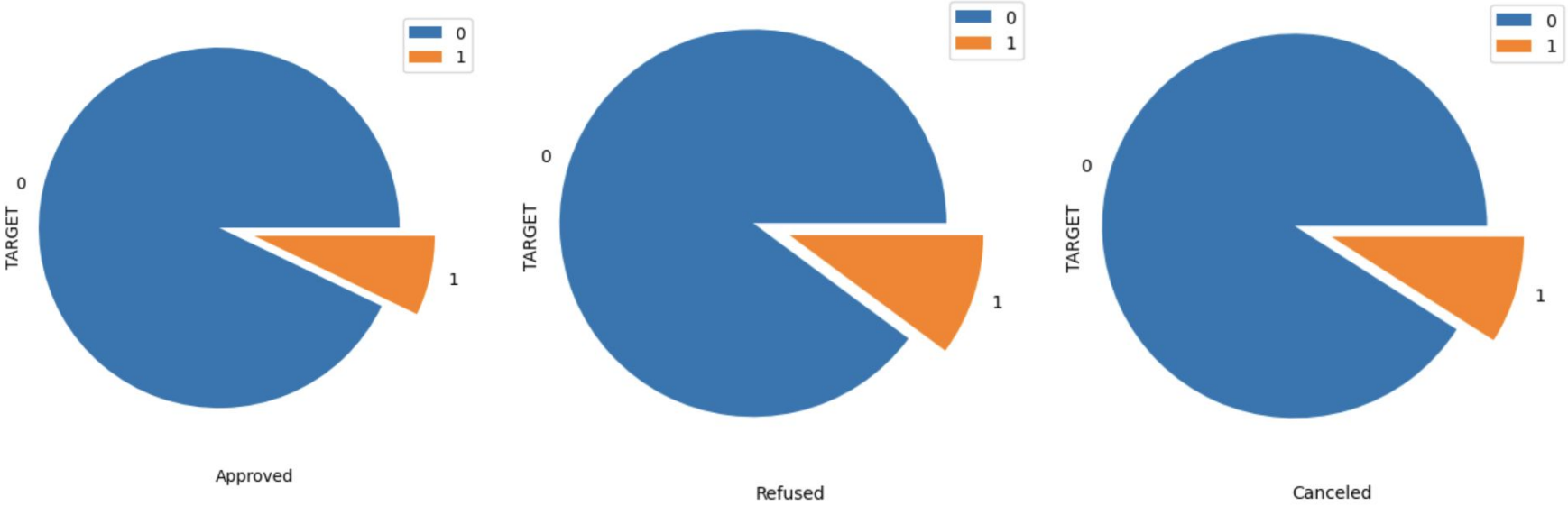
Ratio of Cancelled Applications is quite High.







Checking for Application status i.e 'Approved','Refused','Cancelled' , for DEFAULTERS AND NON_DEFAULTERS on Merged Data of Application and Prev_application.

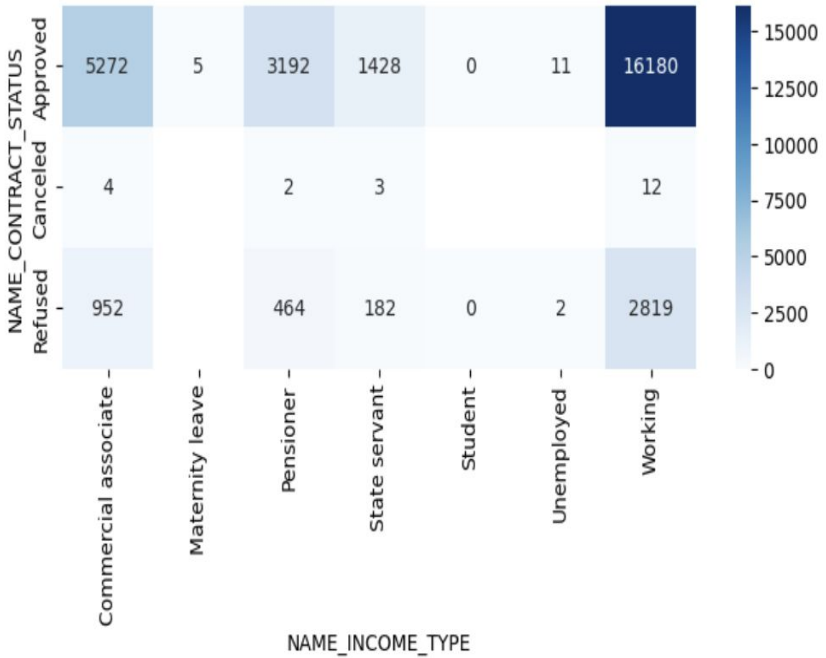


Inference :

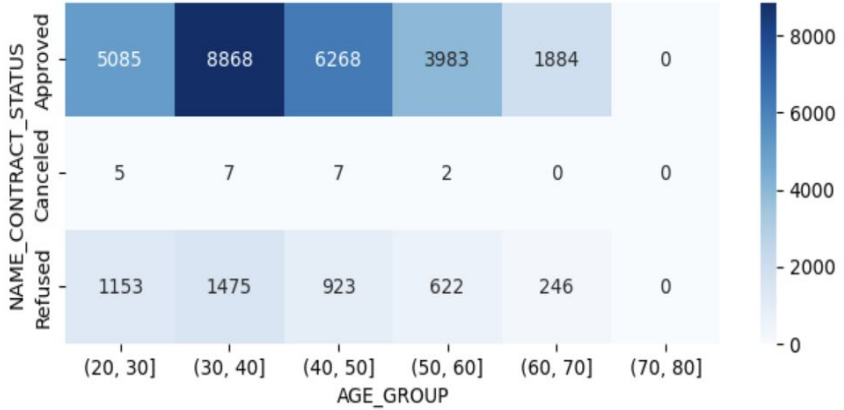
Approximately 7 % (26088 applications) of Approved Applications are of Defaulters.Approximately 9% Cancelled ,10% Refused, also have Defaulter applications.

This implies that Company have prevously refused those applications, or loan was cancelled by applicant, which are now Contributing in Default Cases.

NAME_CONTRACT_STATUS" Vs "NAME_INCOME_TYPE",aggregating on Target



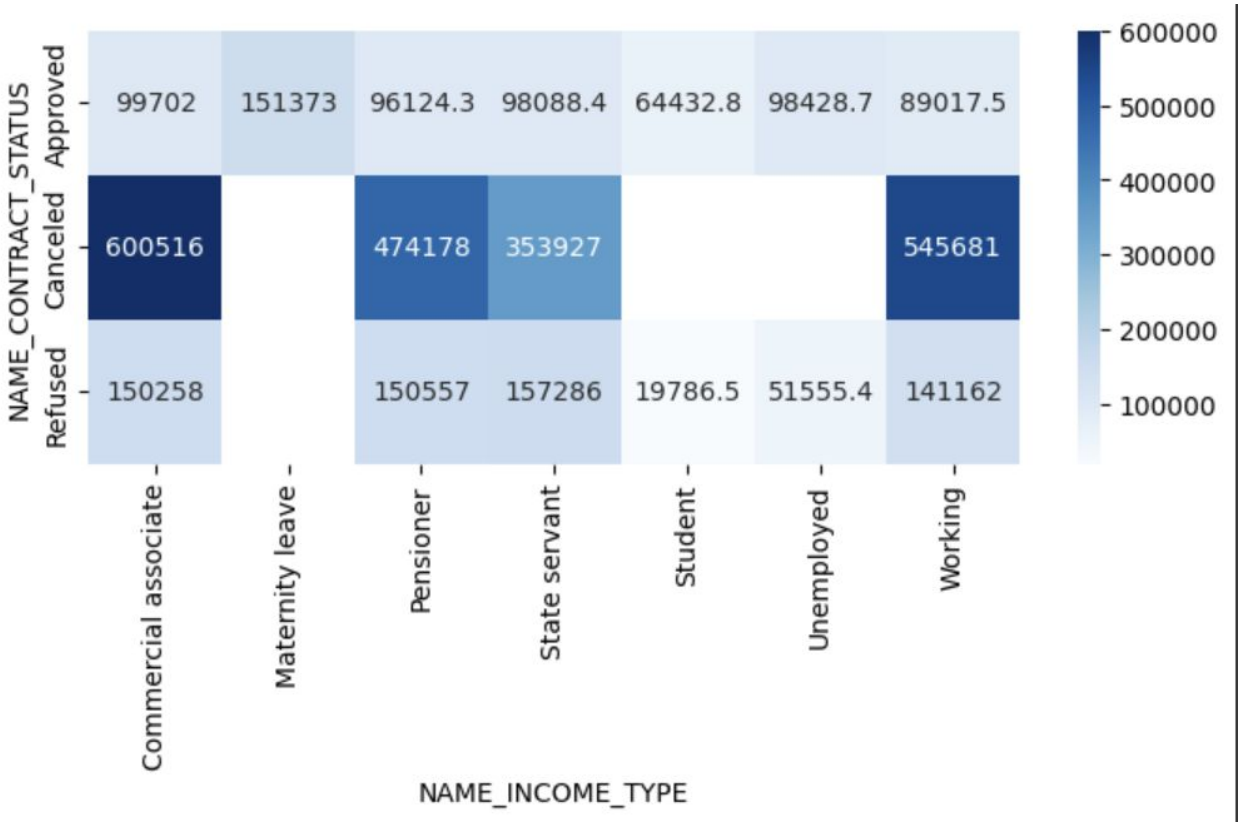
NAME_CONTRACT_STATUS" Vs "AGE_GROUP",aggregating on Target



NAME_CONTRACT_STATUS" Vs "NAME_INCOME_TYPE",aggregating on AMT_CREDIT

Inference :

- Highest Loan cancellation are for 'Commercial Associates' and 'working' peoples.
- pensioners and state servants are having significant amount of defaulters.
- Highest approval are for 'Maternity Leave' and 'Unemployed' applicants.



Summary

Characteristics of Defaulters:

- All the analysis conducted using application Dataset which was information about current loan applications.
- We explored the data and found many incompetencies in data , and treated them accordingly.
- We were supposed to find out the defaulting applicants, and our findings are as follows.

Cases of Defaulting Loan Applications.

The Variables listed below show Default cases on Approved Applications.

Default High

'INCOME_GROUP' - Medium income
'AGE_GROUP - 30-40, followed by 40-50
'NAME_INCOME_TYPE' - Working
'OCCUPATION_TYPE' - Labourers 32%
'ORGANIZATION_TYPE' - Business type 3
'OWN_CAR' - >30% don’t have car
'OWN_REALTY' - 70%(approx) don’t have own home