Eda-bank-loan-default-risk-analysis

By:- Gnaneshwari M

Table of Contents:

1. Introduction:

- This case study aims to give an idea of applying EDA in a real business scenario. In this case study, we will develop a basic understanding of risk analytics in banking and financial services and understand how data is used to minimise the risk of losing money while lending to customers.

Business Objective

- This case study aims to identify patterns which indicate if a client has difficulty paying their installments which may be used for taking actions such as denying the loan, reducing the amount of loan, lending (to risky applicants) at a higher interest rate, etc. This will ensure that the consumers capable of repaying the loan are not rejected. Identification of such applicants using EDA is the aim of this case study.
  In other words, the company wants to understand the driving factors (or driver variables) behind loan default, i.e. the variables which are strong indicators of default. The company can utilise this knowledge for its portfolio and risk assessment.

Business Understanding

The loan providing companies find it hard to give loans to the people due to their insufficient or non-existent credit history. Because of that, some consumers use it as their advantage by becoming a defaulter. Suppose you work for a consumer finance company which specialises in lending various types of loans to urban customers. You have to use EDA to analyse the patterns present in the data. This will ensure that the applicants capable of repaying the loan are not rejected.

When the company receives a loan application, the company has to decide for loan approval based on the applicant's profile. Two types of risks are associated with the bank's decision:

- If the applicant is likely to repay the loan, then not approving the loan results in a loss of business to the company
- If the applicant is not likely to repay the loan, i.e. he/she is likely to default, then approving the loan may lead to a financial loss for the company.

The data given below contains the information about the loan application at the time of applying for the loan. It contains two types of scenarios:

- **The client with payment difficulties:** he/she had late payment more than X days on at least one of the first Y instalments of the loan in our sample
- **All other cases:** All other cases when the payment is paid on time

When a client applies for a loan, there are four types of decisions that could be taken by the client/company):

1. **Approved:** The Company has approved loan Application
2. **Cancelled:** The client cancelled the application sometime during approval. Either the client changed her/his mind about the loan or in some cases due to a higher risk of the client he received worse pricing which he did not want.
3. **Refused:** The company had rejected the loan (because the client does not meet their requirements etc.)
4. **Unused offer:** Loan has been cancelled by the client but on different stages of the process.

2. Getting Jupyter Ready:

Import Python Libraries:

2.1 Import Python Libraries:

import matplotlib.pyplot as plt

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.style as style
import seaborn as sns
import itertools
%matplotlib inline
```

# setting up plot style

```
style.use('seaborn-v0_8-poster')
style.use('fivethirtyeight')
import warnings
warnings.filterwarnings('ignore')
```

2.2 Supress Warnings:

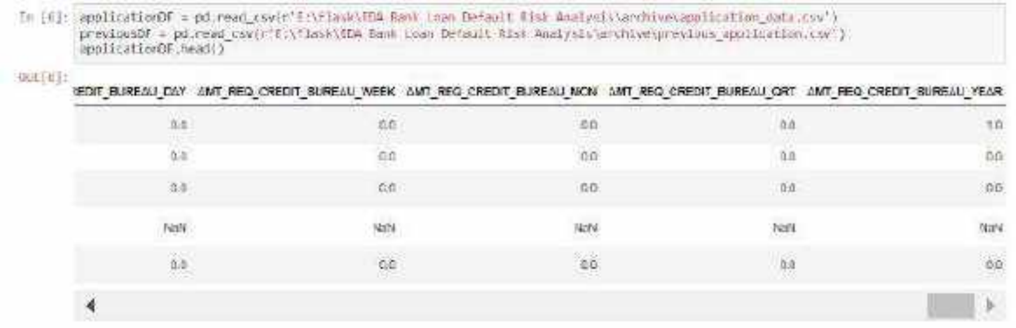
```
import warnings
warnings.filterwarnings('ignore')
```

2.3 Adjust Jupyer Views

```
pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)
pd.set_option('display.expand_frame_repr', False)
```

3. Reading & Understanding the data

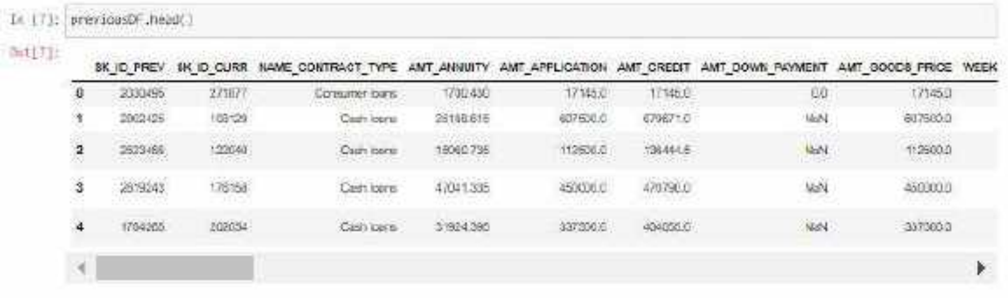
3.1 Importing the input files



```
In [6]: applicationDF = pd.read_csv(r'E:\Flask\EDA Bank Loan Default Risk Analysis\archive\application_data.csv')
        previousDF = pd.read_csv(r'E:\Flask\EDA Bank Loan Default Risk Analysis\archive\previous_application.csv')
        applicationDF.head()
```

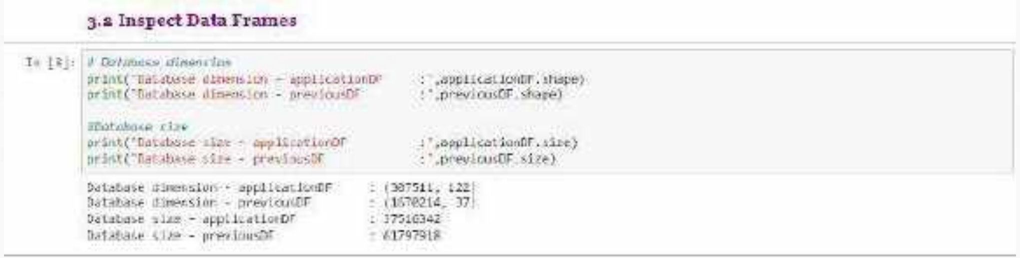
Out[6]:

EDIT_BUREAU_DAY	AMT_REQ_CREDIT_BUREAU_WEEK	AMT_REQ_CREDIT_BUREAU_MON	AMT_REQ_CREDIT_BUREAU_QRT	AMT_REQ_CREDIT_BUREAU_YEAR
0.0	0.0	0.0	0.0	1.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
NaN	NaN	NaN	NaN	NaN
0.0	0.0	0.0	0.0	0.0



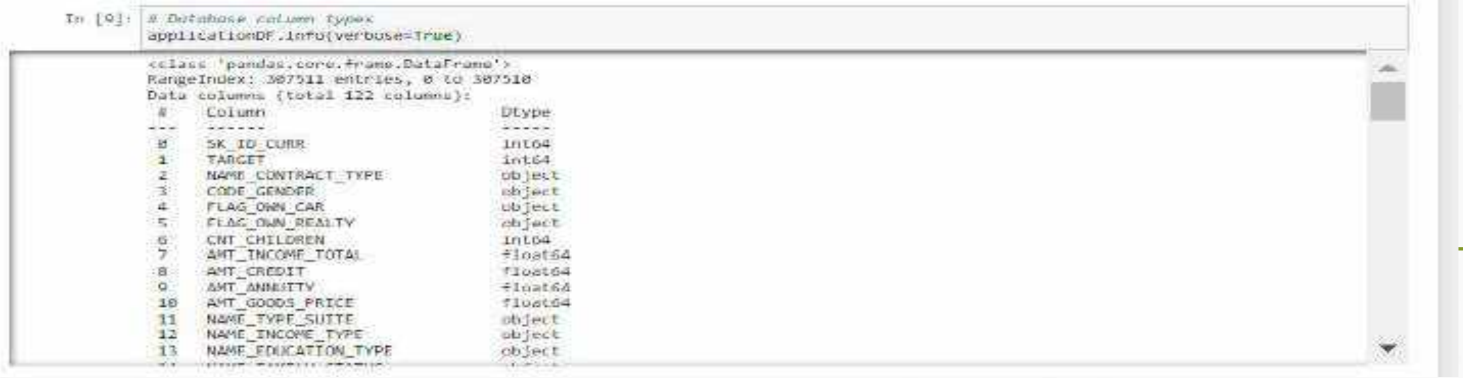
```
In [7]: previousDF.head()
Out[7]:
```

	SK_ID_PREV	SK_ID_CURR	NAME_CONTRACT_TYPE	AMT_ANNUITY	AMT_APPLICATION	AMT_CREDIT	AMT_DOWN_PAYMENT	AMT_GOODS_PRICE	WEEK
0	2030495	271877	Consumer loans	1730.430	17145.0	17145.0	0.0	17145.0	
1	2802425	108129	Cash loans	25188.615	607500.0	679671.0	NaN	607500.0	
2	2523466	122040	Cash loans	15060.735	112500.0	136444.5	NaN	112500.0	
3	2819243	176158	Cash loans	47041.335	450000.0	470790.0	NaN	450000.0	
4	1784265	202054	Cash loans	31924.395	337500.0	404055.0	NaN	337500.0	



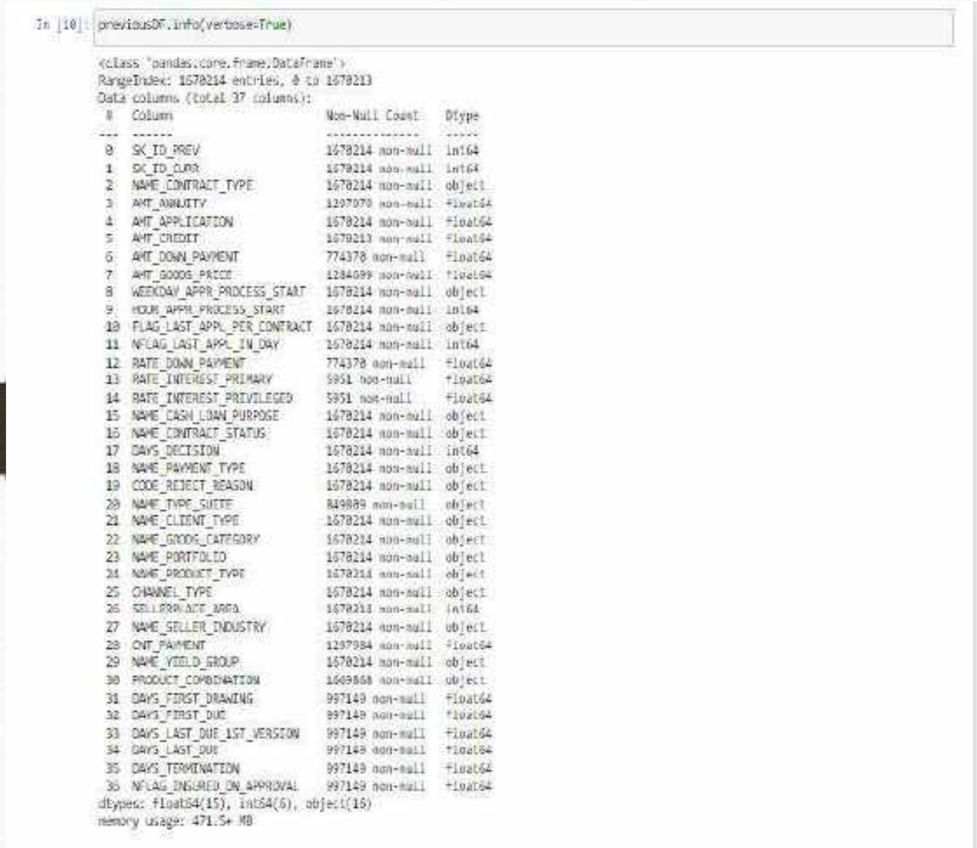
3.2 Inspect Data Frames

```
In [8]: # Database dimension
        print("Database dimension - applicationDF     :",applicationDF.shape)
        print("Database dimension - previousDF        :",previousDF.shape)

        #Database size
        print("Database size - applicationDF          :",applicationDF.size)
        print("Database size - previousDF             :",previousDF.size)

Database dimension - applicationDF     : (307511, 122
Database dimension - previousDF        : (1670214, 37
Database size - applicationDF          : 37516342
Database size - previousDF             : 61797918
```



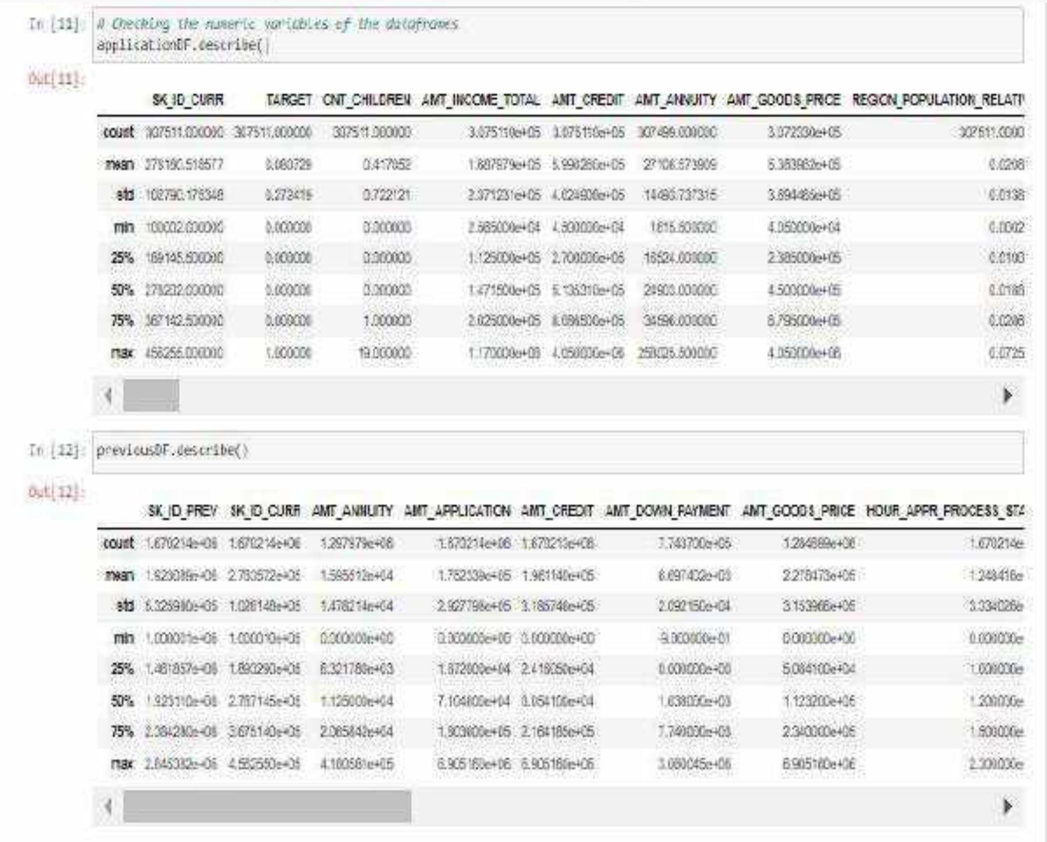
```
In [9]: # Database column types
        applicationDF.info(verbose=True)
```

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



```
In [10]: previousDF.info(verbose=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1670214 entries, 0 to 1670213
Data columns (total 37 columns):
 #   Column                       Non-Null Count    Dtype  
---  ------                       --------------    -----  
 0   SK_ID_PREV                   1670214 non-null  int64  
 1   SK_ID_CURR                   1670214 non-null  int64  
 2   NAME_CONTRACT_TYPE           1670214 non-null  object 
 3   AMT_ANNUITY                  1297979 non-null  float64
 4   AMT_APPLICATION              1670214 non-null  float64
 5   AMT_CREDIT                   1670213 non-null  float64
 6   AMT_DOWN_PAYMENT             774370 non-null   float64
 7   AMT_GOODS_PRICE              1284699 non-null  float64
 8   WEEKDAY_APPR_PROCESS_START   1670214 non-null  object 
 9   HOUR_APPR_PROCESS_START      1670214 non-null  int64  
 10  FLAG_LAST_APPL_PER_CONTRACT  1670214 non-null  object 
 11  NFLAG_LAST_APPL_IN_DAY       1670214 non-null  int64  
 12  RATE_DOWN_PAYMENT            774370 non-null   float64
 13  RATE_INTEREST_PRIMARY        5951 non-null     float64
 14  RATE_INTEREST_PRIVILEGED     5951 non-null     float64
 15  NAME_CASH_LOAN_PURPOSE       1670214 non-null  object 
 16  NAME_CONTRACT_STATUS         1670214 non-null  object 
 17  DAYS_DECISION                1670214 non-null  int64  
 18  NAME_PAYMENT_TYPE            1670214 non-null  object 
 19  CODE_REJECT_REASON           1670214 non-null  object 
 20  NAME_TYPE_SUITE              849809 non-null   object 
 21  NAME_CLIENT_TYPE             1670214 non-null  object 
 22  NAME_GOODS_CATEGORY          1670214 non-null  object 
 23  NAME_PORTFOLIO               1670214 non-null  object 
 24  NAME_PRODUCT_TYPE            1670214 non-null  object 
 25  CHANNEL_TYPE                 1670214 non-null  object 
 26  SELLERPLACE_AREA             1670214 non-null  int64  
 27  NAME_SELLER_INDUSTRY         1670214 non-null  object 
 28  CNT_PAYMENT                  1297984 non-null  float64
 29  NAME_YIELD_GROUP             1670214 non-null  object 
 30  PRODUCT_COMBINATION          1669868 non-null  object 
 31  DAYS_FIRST_DRAWING           997149 non-null   float64
 32  DAYS_FIRST_DUE               997149 non-null   float64
 33  DAYS_LAST_DUE_1ST_VERSION    997149 non-null   float64
 34  DAYS_LAST_DUE                997149 non-null   float64
 35  DAYS_TERMINATION             997149 non-null   float64
 36  NFLAG_INSURED_ON_APPROVAL    997149 non-null   float64
dtypes: float64(15), int64(6), object(16)
memory usage: 471.5+ MB
```



In [11]:

```python
# Checking the numeric variables of the dataframes
applicationDF.describe()
```

Out[11]:

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

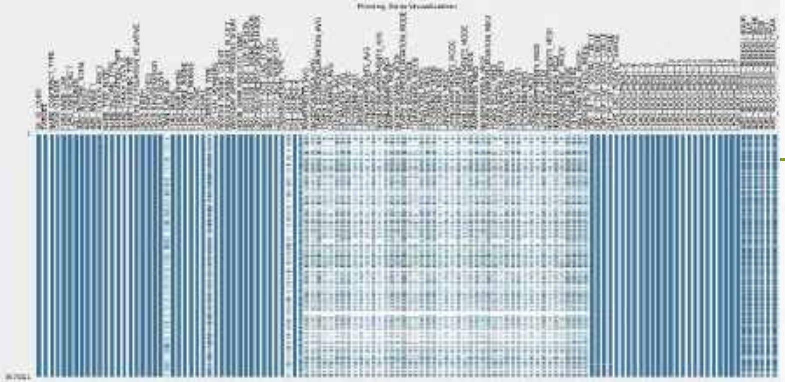
In [12]:

```python
previousDF.describe()
```

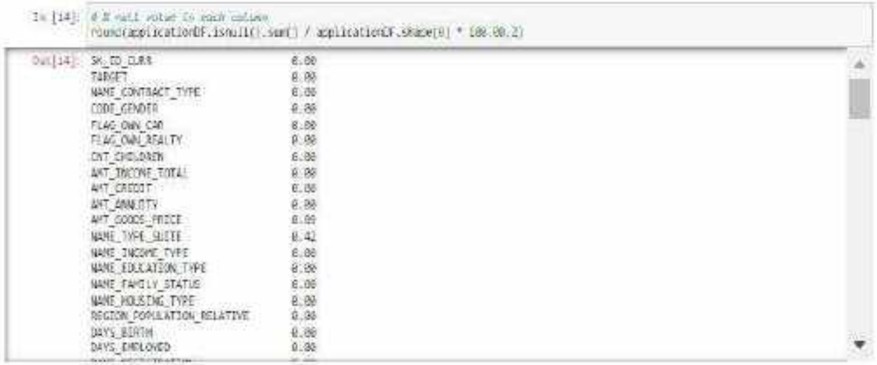
Out[12]:

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

4. Data Cleaning & Manipulation

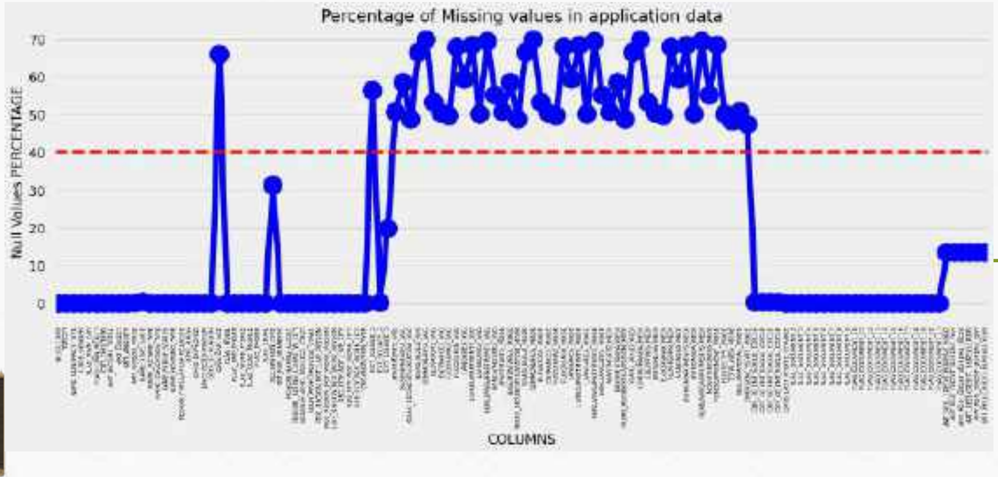


Insight:
Based on the above Matrix, it is evidednt that the dataset has many missing values. Let's check for each column what is the % of missing values

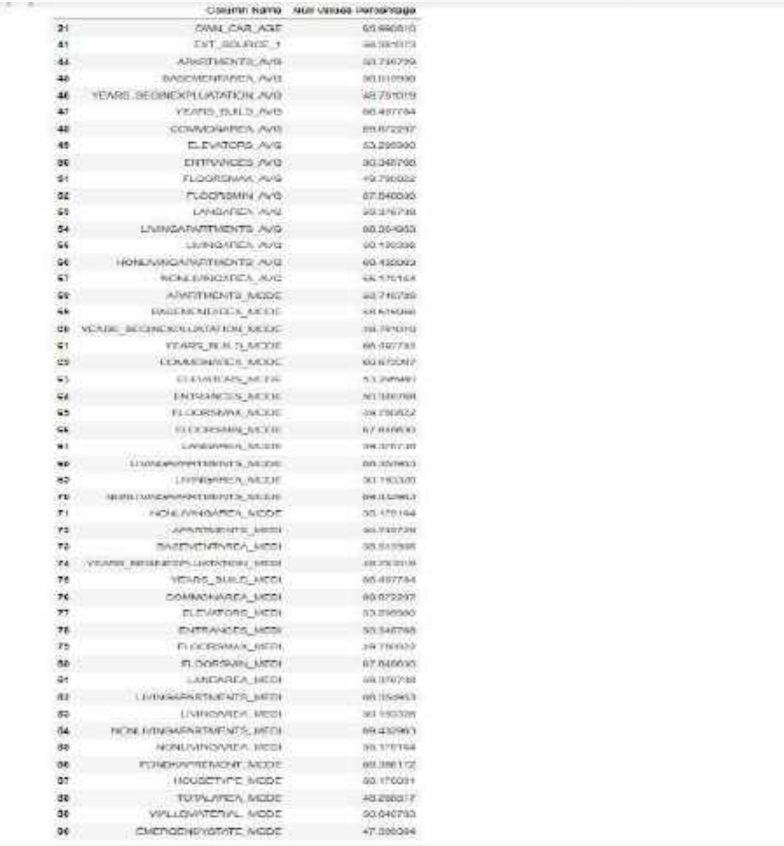


```
In [14]: # % null value in each column
         round(applicationDF.isnull().sum() / applicationDF.shape[0] * 100.00,2)

Out[14]: SK_ID_CURR                     0.00
         TARGET                         0.00
         NAME_CONTRACT_TYPE             0.00
         CODE_GENDER                    0.00
         FLAG_OWN_CAR                   0.00
         FLAG_OWN_REALTY                0.00
         CNT_CHILDREN                   0.00
         AMT_INCOME_TOTAL               0.00
         AMT_CREDIT                     0.00
         AMT_ANNUITY                    0.00
         AMT_GOODS_PRICE                0.09
         NAME_TYPE_SUITE                0.42
         NAME_INCOME_TYPE               0.00
         NAME_EDUCATION_TYPE            0.00
         NAME_FAMILY_STATUS             0.00
         NAME_HOUSING_TYPE              0.00
         REGION_POPULATION_RELATIVE     0.00
         DAYS_BIRTH                     0.00
         DAYS_EMPLOYED                  0.00
```

Insight:
There are many columns in applicationDF dataframe where missing value is more than 40%. Let's plot the columns vs missing value % with 40% being the cut-off marks



Insight:
From the plot we can see the columns in which percentage of null values more than 40% are marked above the red line and the columns which have less than 40 % null values below the red line. Let's check the columns which has more than 40% missing values

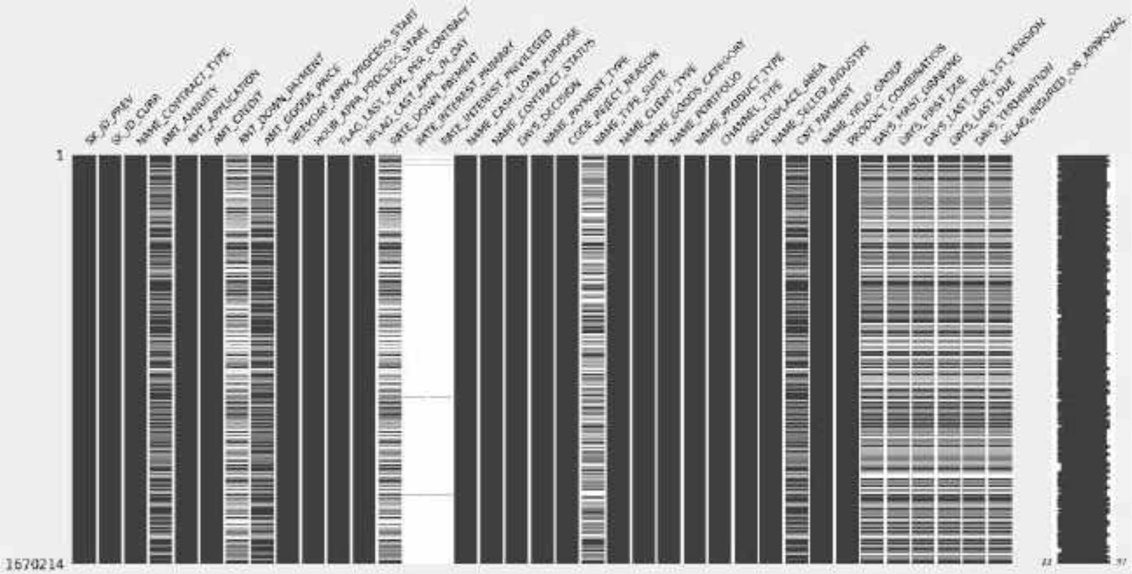


	Column Name	Null Values Percentage
21	OWN_CAR_AGE	65.990810
41	EXT_SOURCE_1	56.381073
44	APARTMENTS_AVG	50.749729
45	BASEMENTAREA_AVG	58.515956
46	YEARS_BEGINEXPLUATATION_AVG	48.781019
47	YEARS_BUILD_AVG	66.497784
48	COMMONAREA_AVG	69.872297
49	ELEVATORS_AVG	53.295980
50	ENTRANCES_AVG	50.348768
51	FLOORSMAX_AVG	49.760822
52	FLOORSMIN_AVG	67.848630
53	LANDAREA_AVG	59.376738
54	LIVINGAPARTMENTS_AVG	68.354953
55	LIVINGAREA_AVG	50.193326
56	NONLIVINGAPARTMENTS_AVG	69.432963
57	NONLIVINGAREA_AVG	55.179164
58	APARTMENTS_MODE	50.749729
59	BASEMENTAREA_MODE	58.515956
60	YEARS_BEGINEXPLUATATION_MODE	48.781019
61	YEARS_BUILD_MODE	66.497784
62	COMMONAREA_MODE	69.872297
63	ELEVATORS_MODE	53.295980
64	ENTRANCES_MODE	50.348768
65	FLOORSMAX_MODE	49.760822
66	FLOORSMIN_MODE	67.848630
67	LANDAREA_MODE	59.376738
68	LIVINGAPARTMENTS_MODE	68.354953
69	LIVINGAREA_MODE	50.193326
70	NONLIVINGAPARTMENTS_MODE	69.432963
71	NONLIVINGAREA_MODE	55.179164
72	APARTMENTS_MEDI	50.749729
73	BASEMENTAREA_MEDI	58.515956
74	YEARS_BEGINEXPLUATATION_MEDI	48.781019
75	YEARS_BUILD_MEDI	66.497784
76	COMMONAREA_MEDI	69.872297
77	ELEVATORS_MEDI	53.295980
78	ENTRANCES_MEDI	50.348768
79	FLOORSMAX_MEDI	49.760822
80	FLOORSMIN_MEDI	67.848630
81	LANDAREA_MEDI	59.376738
82	LIVINGAPARTMENTS_MEDI	68.354953
83	LIVINGAREA_MEDI	50.193326
84	NONLIVINGAPARTMENTS_MEDI	69.432963
85	NONLIVINGAREA_MEDI	55.179164
86	FONDKAPREMONT_MODE	68.386172
87	HOUSETYPE_MODE	50.176091
88	TOTALAREA_MODE	48.268517
89	WALLSMATERIAL_MODE	50.840783
90	EMERGENCYSTATE_MODE	47.398304

Insight:

Total of 49 columns are there which have more than 40% null values.Seems like most of the columns with high missing values are related to different area sizes on apartment owned/rented by the loan applicant

4.1.2 previousDF Missing Values

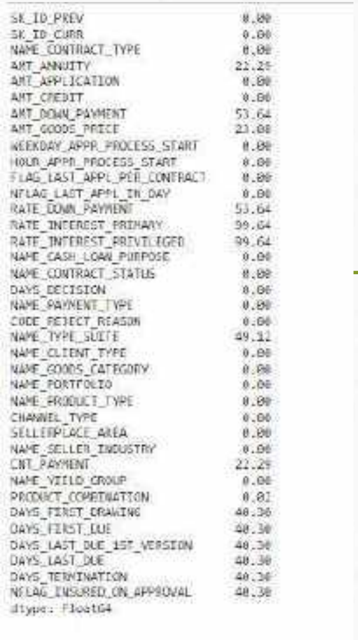


Insight:
Based on the above Matrix, it is evidednt that the dataset has many missing values. Let's check for each column what is the % of missing values

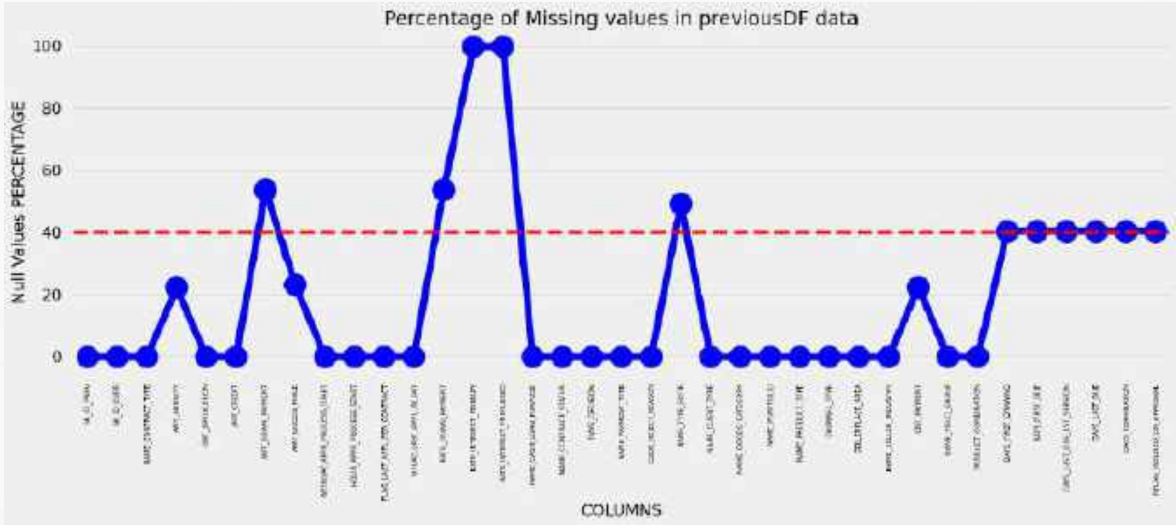
Insight:

There are many columns in previousDF dataframe where missing value is more than 40%. Let's plot the columns vs missing value % with 40% being the cut-off marks

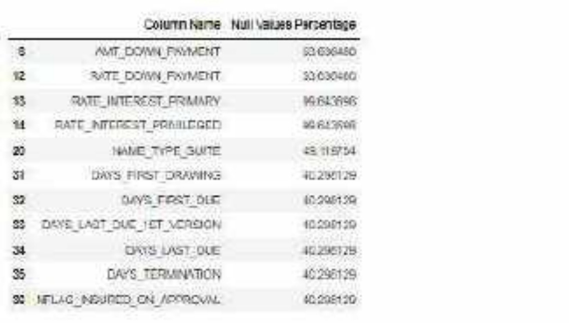
# checking the null value % of each column in previousDF dataframe



```
SK_ID_PREV                      0.00
SK_ID_CURR                      0.00
NAME_CONTRACT_TYPE              0.00
AMT_ANNUITY                    22.29
AMT_APPLICATION                 0.00
AMT_CREDIT                      0.00
AMT_DOWN_PAYMENT               53.64
AMT_GOODS_PRICE                23.08
WEEKDAY_APPR_PROCESS_START      0.00
HOUR_APPR_PROCESS_START         0.00
FLAG_LAST_APPL_PER_CONTRACT     0.00
NFLAG_LAST_APPL_IN_DAY          0.00
RATE_DOWN_PAYMENT              53.64
RATE_INTEREST_PRIMARY          99.64
RATE_INTEREST_PRIVILEGED       99.64
NAME_CASH_LOAN_PURPOSE          0.00
NAME_CONTRACT_STATUS            0.00
DAYS_DECISION                   0.00
NAME_PAYMENT_TYPE               0.00
CODE_REJECT_REASON              0.00
NAME_TYPE_SUITE                49.12
NAME_CLIENT_TYPE                0.00
NAME_GOODS_CATEGORY             0.00
NAME_PORTFOLIO                  0.00
NAME_PRODUCT_TYPE               0.00
CHANNEL_TYPE                    0.00
SELLERPLACE_AREA                0.00
NAME_SELLER_INDUSTRY            0.00
CNT_PAYMENT                    22.29
NAME_YIELD_GROUP                0.00
PRODUCT_COMBINATION             0.02
DAYS_FIRST_DRAWING             40.30
DAYS_FIRST_DUE                 40.30
DAYS_LAST_DUE_1ST_VERSION      40.30
DAYS_LAST_DUE                  40.30
DAYS_TERMINATION               40.30
NFLAG_INSURED_ON_APPROVAL      40.30
dtype: float64
```



- **Insight:**
  From the plot we can see the columns in which percentage of null values more than 40% are marked above the red line and the columns which have less than 40 % null values below the red line. Let's check the columns which has more than 40% missing values



	Column Name	Null Values Percentage
6	AMT_DOWN_PAYMENT	53.636480
12	RATE_DOWN_PAYMENT	53.636480
13	RATE_INTEREST_PRIMARY	99.643698
14	RATE_INTEREST_PRIVILEGED	99.643698
20	NAME_TYPE_SUITE	49.119754
31	DAYS_FIRST_DRAWING	40.298129
32	DAYS_FIRST_DUE	40.298129
33	DAYS_LAST_DUE_1ST_VERSION	40.298129
34	DAYS_LAST_DUE	40.298129
35	DAYS_TERMINATION	40.298129
36	NFLAG_INSURED_ON_APPROVAL	40.298129

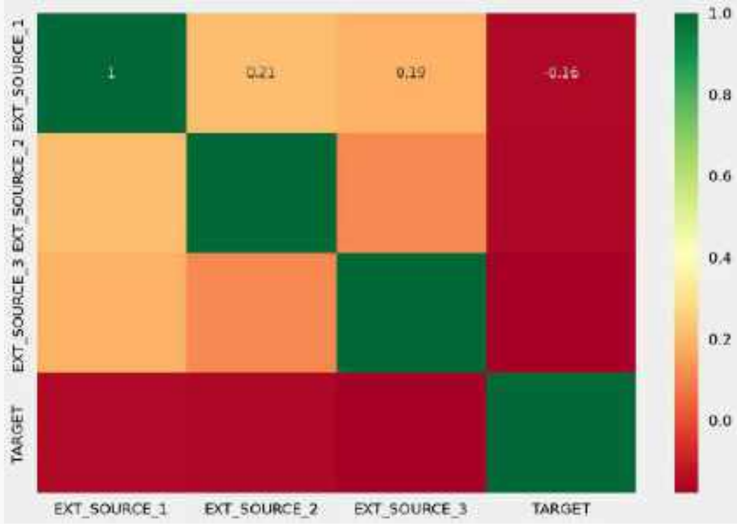
# more than or equal to 40% empty rows columns

Insight:
Total of 11 columns are there which have more than 40% null values. These columns can be deleted. Before deleting these columns, let's review if there are more columns which can be dropped or not[](http://)

4.2 Analyze & Delete Unnecessary Columns in applicationDF

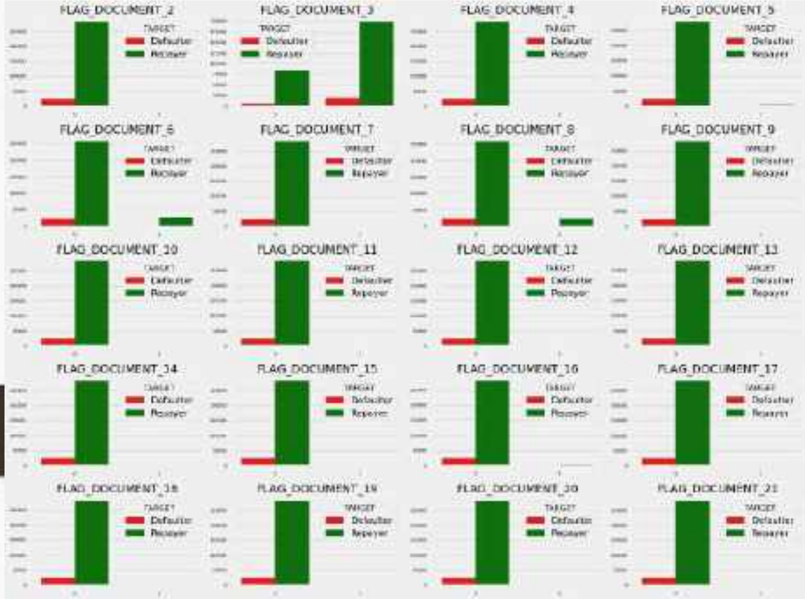
4.2.1 EXT_SOURCE_X

Insight:
Based on the above Heatmap, we can see there is almost no correlation between EXT_SOURCE_X columns and target column, thus we can drop these columns. EXT_SOURCE_1 has 56% null values, where as EXT_SOURCE_3 has close to 20% null values



# Checking correlation of EXT_SOURCE_X columns vs TARGET column

4.2.2 Flag Document

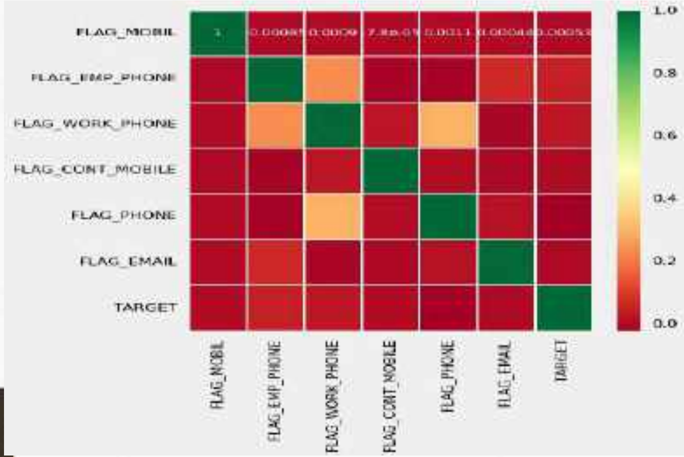


Insight:

The above graph shows that in most of the loan application cases, clients who applied for loans has not submitted FLAG_DOCUMENT_X except FLAG_DOCUMENT_3. Thus, Except for FLAG_DOCUMENT_3, we can delete rest of the columns. Data shows if borrower has submitted FLAG_DOCUMENT_3 then there is a less chance of defaulting the loan.

Contact Parameters

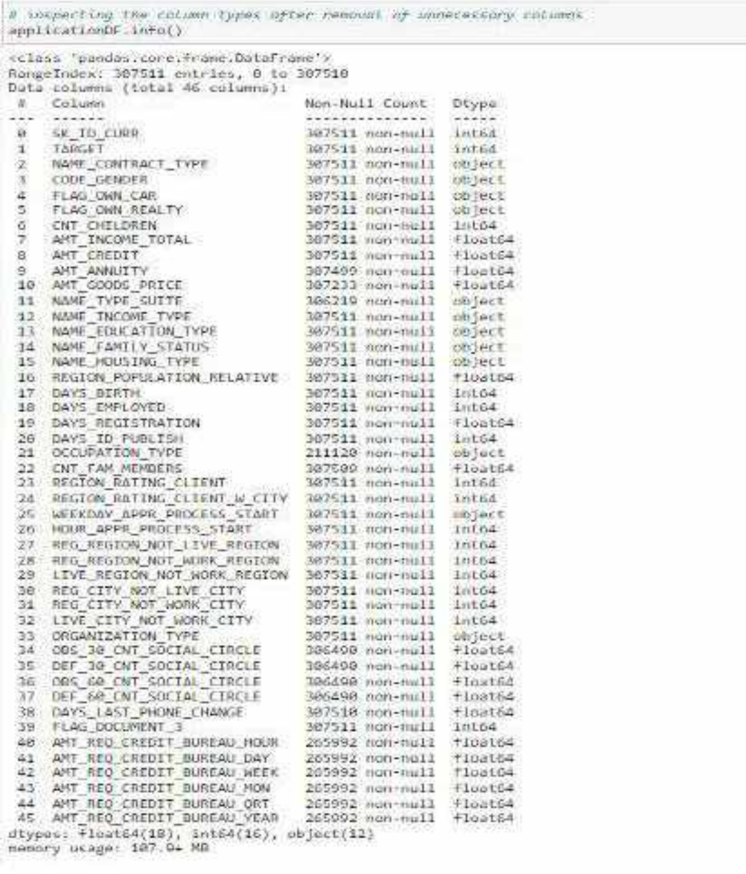
# checking is there is any correlation between mobile phone, work phone etc, email, Family members and Region rating



Insight:
There is no correlation between flags of mobile phone, email etc with loan repayment; thus these columns can be deleted

# including the 6 FLAG columns to be deleted

Insight:
Total 76 columns can be deleted from applicationDF



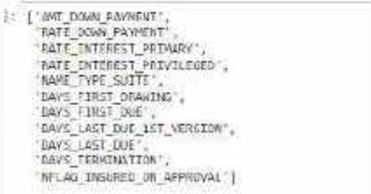
```
# inspecting the column types after removal of unnecessary columns
applicationDF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Data columns (total 46 columns):
 #   Column                        Non-Null Count   Dtype  
---  ------                        --------------   -----  
 0   SK_ID_CURR                    307511 non-null  int64  
 1   TARGET                        307511 non-null  int64  
 2   NAME_CONTRACT_TYPE            307511 non-null  object 
 3   CODE_GENDER                   307511 non-null  object 
 4   FLAG_OWN_CAR                  307511 non-null  object 
 5   FLAG_OWN_REALTY               307511 non-null  object 
 6   CNT_CHILDREN                  307511 non-null  int64  
 7   AMT_INCOME_TOTAL              307511 non-null  float64
 8   AMT_CREDIT                    307511 non-null  float64
 9   AMT_ANNUITY                   307499 non-null  float64
 10  AMT_GOODS_PRICE               307233 non-null  float64
 11  NAME_TYPE_SUITE               306219 non-null  object 
 12  NAME_INCOME_TYPE              307511 non-null  object 
 13  NAME_EDUCATION_TYPE           307511 non-null  object 
 14  NAME_FAMILY_STATUS            307511 non-null  object 
 15  NAME_HOUSING_TYPE             307511 non-null  object 
 16  REGION_POPULATION_RELATIVE    307511 non-null  float64
 17  DAYS_BIRTH                    307511 non-null  int64  
 18  DAYS_EMPLOYED                 307511 non-null  int64  
 19  DAYS_REGISTRATION             307511 non-null  float64
 20  DAYS_ID_PUBLISH               307511 non-null  int64  
 21  OCCUPATION_TYPE               211120 non-null  object 
 22  CNT_FAM_MEMBERS               307509 non-null  float64
 23  REGION_RATING_CLIENT          307511 non-null  int64  
 24  REGION_RATING_CLIENT_W_CITY   307511 non-null  int64  
 25  WEEKDAY_APPR_PROCESS_START    307511 non-null  object 
 26  HOUR_APPR_PROCESS_START       307511 non-null  int64  
 27  REG_REGION_NOT_LIVE_REGION    307511 non-null  int64  
 28  REG_REGION_NOT_WORK_REGION    307511 non-null  int64  
 29  LIVE_REGION_NOT_WORK_REGION   307511 non-null  int64  
 30  REG_CITY_NOT_LIVE_CITY        307511 non-null  int64  
 31  REG_CITY_NOT_WORK_CITY        307511 non-null  int64  
 32  LIVE_CITY_NOT_WORK_CITY       307511 non-null  int64  
 33  ORGANIZATION_TYPE             307511 non-null  object 
 34  OBS_30_CNT_SOCIAL_CIRCLE      306490 non-null  float64
 35  DEF_30_CNT_SOCIAL_CIRCLE      306490 non-null  float64
 36  OBS_60_CNT_SOCIAL_CIRCLE      306490 non-null  float64
 37  DEF_60_CNT_SOCIAL_CIRCLE      306490 non-null  float64
 38  DAYS_LAST_PHONE_CHANGE        307510 non-null  float64
 39  FLAG_DOCUMENT_3               307511 non-null  int64  
 40  AMT_REQ_CREDIT_BUREAU_HOUR    265992 non-null  float64
 41  AMT_REQ_CREDIT_BUREAU_DAY     265992 non-null  float64
 42  AMT_REQ_CREDIT_BUREAU_WEEK    265992 non-null  float64
 43  AMT_REQ_CREDIT_BUREAU_MON     265992 non-null  float64
 44  AMT_REQ_CREDIT_BUREAU_QRT     265992 non-null  float64
 45  AMT_REQ_CREDIT_BUREAU_YEAR    265992 non-null  float64
dtypes: float64(18), int64(16), object(12)
memory usage: 107.9+ MB
```

Insight:

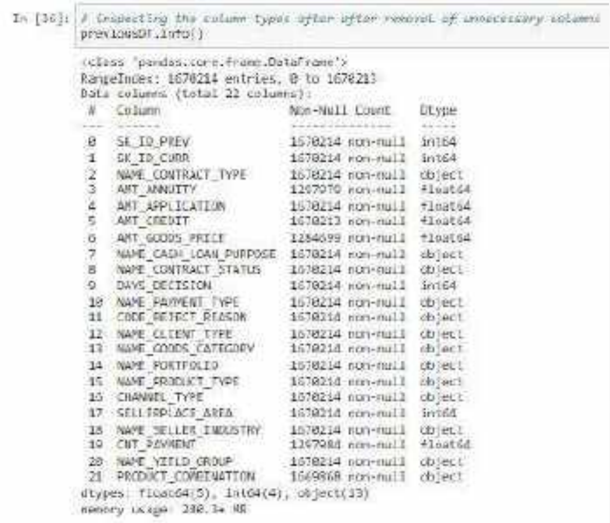
After deleting unnecessary columns, there are 46 columns remaining in applicationDF

4.3 Analyze & Delete Unnecessary Columns in previousDF



```
]: ['AMT_DOWN_PAYMENT',
    'RATE_DOWN_PAYMENT',
    'RATE_INTEREST_PRIMARY',
    'RATE_INTEREST_PRIVILEGED',
    'NAME_TYPE_SUITE',
    'DAYS_FIRST_DRAWING',
    'DAYS_FIRST_DUE',
    'DAYS_LAST_DUE_1ST_VERSION',
    'DAYS_LAST_DUE',
    'DAYS_TERMINATION',
    'NFLAG_INSURED_ON_APPROVAL']
```

# Getting the 11 columns which has more than 40% unknown



```
In [36]: # Inspecting the column types after after removal of unnecessary columns
         previousDF.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1670214 entries, 0 to 1670213
Data columns (total 22 columns):
 #   Column                  Non-Null Count    Dtype
---  ------                  --------------    -----
 0   SK_ID_PREV              1670214 non-null  int64
 1   SK_ID_CURR              1670214 non-null  int64
 2   NAME_CONTRACT_TYPE      1670214 non-null  object
 3   AMT_ANNUITY             1297979 non-null  float64
 4   AMT_APPLICATION         1670214 non-null  float64
 5   AMT_CREDIT              1670213 non-null  float64
 6   AMT_GOODS_PRICE         1284699 non-null  float64
 7   NAME_CASH_LOAN_PURPOSE  1670214 non-null  object
 8   NAME_CONTRACT_STATUS    1670214 non-null  object
 9   DAYS_DECISION           1670214 non-null  int64
 10  NAME_PAYMENT_TYPE       1670214 non-null  object
 11  CODE_REJECT_REASON      1670214 non-null  object
 12  NAME_CLIENT_TYPE        1670214 non-null  object
 13  NAME_GOODS_CATEGORY     1670214 non-null  object
 14  NAME_PORTFOLIO          1670214 non-null  object
 15  NAME_PRODUCT_TYPE       1670214 non-null  object
 16  CHANNEL_TYPE            1670214 non-null  object
 17  SELLERPLACE_AREA        1670214 non-null  int64
 18  NAME_SELLER_INDUSTRY    1670214 non-null  object
 19  CNT_PAYMENT             1297984 non-null  float64
 20  NAME_YIELD_GROUP        1670214 non-null  object
 21  PRODUCT_COMBINATION     1669868 non-null  object
dtypes: float64(5), int64(4), object(13)
memory usage: 280.3+ MB
```

Insight:
After deleting unnecessary columns, there are 22 columns remaining in applicationDF

4.4 Standardize Values

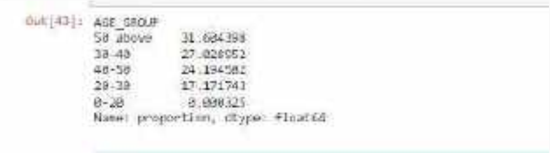
Strategy for applicationDF:

- Convert DAYS_DECISION,DAYS_EMPLOYED, DAYS_REGISTRATION,DAYS_ID_PUBLISH from negative to positive as days cannot be negative.
- Convert DAYS_BIRTH from negative to positive values and calculate age and create categorical bins columns
- Categorize the amount variables into bins
- Convert region rating column and few other columns to categorical

# Binning Numerical Columns to create a categorical column

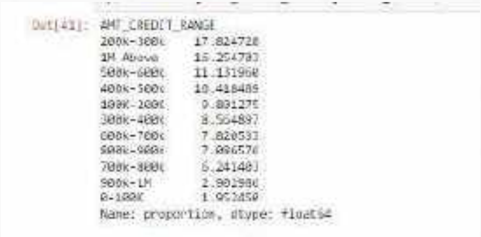
# Creating bins for income amount

# Converting Negative days to positive days



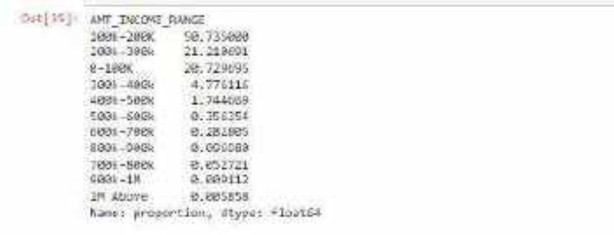
```
Out[43]: AGE_GROUP
         50 above     31.684398
         30-40        27.028952
         40-50        24.194582
         20-30        17.171743
         0-20          0.000325
         Name: proportion, dtype: float64
```

Insight:
More than 50% loan applicants have income amount in the range of 100K-200K. Almost 92% loan applicants have income less than 300K



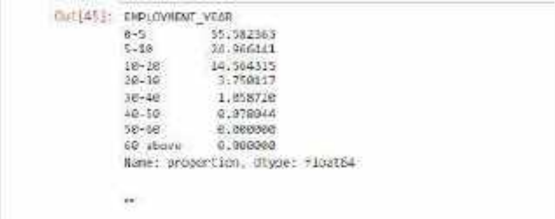
```
Out[41]: AMT_CREDIT_RANGE
         200k-300k    17.824728
         1M Above     16.254703
         500k-600k    11.131960
         400k-500k    10.418489
         100K-200K     9.801275
         300k-400k     8.564897
         600k-700k     7.820533
         800k-900k     7.086570
         700k-800k     6.241403
         900k-1M       2.902980
         0-100K        1.952450
         Name: proportion, dtype: float64
```

Insight:
31% loan applicants have age above 50 years. More than 55% of loan applicants have age over 40 years.



```
Out[35]: AMT_INCOME_RANGE
         100K-200K    50.735000
         200K-300K    21.210691
         0-100K       20.729095
         300K-400K     4.776116
         400K-500K     1.744009
         500K-600K     0.356354
         600K-700K     0.282005
         800K-900K     0.096088
         700K-800K     0.052721
         900K-1M       0.009112
         1M Above      0.005858
         Name: proportion, dtype: float64
```

Insight:
More Than 16% loan applicants have taken loan which amounts to more than 1M.



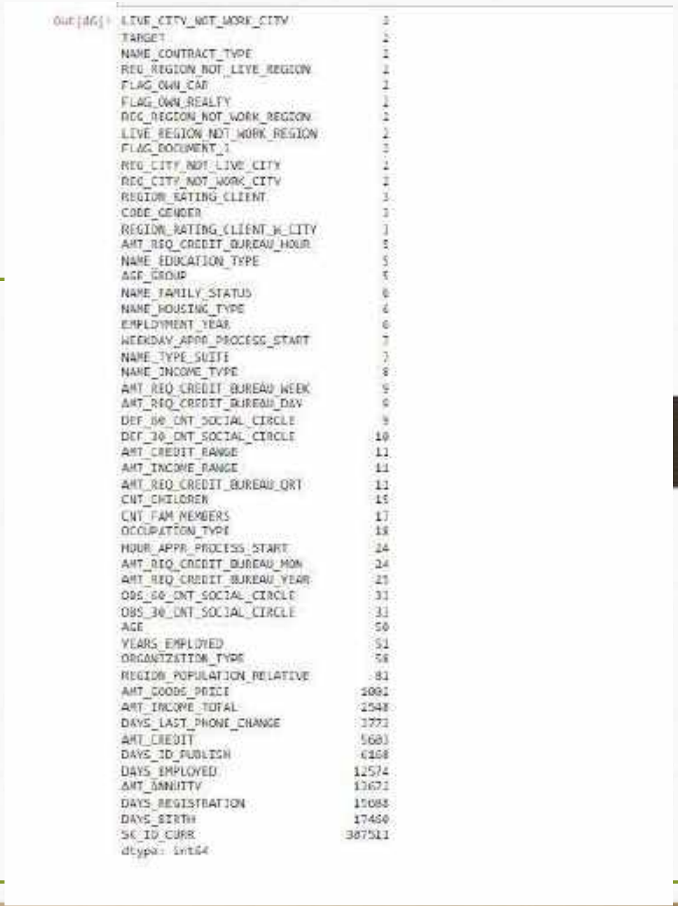
```
Out[45]: EMPLOYMENT_YEAR
         0-5         55.582363
         5-10        24.966141
         10-20       14.504315
         20-30        3.750117
         30-40        1.058720
         40-50        0.378044
         50-60        0.000000
         60 above     0.000000
         Name: proportion, dtype: float64
```

Insight:

More than 55% of the loan applicants have work experience within 0-5 years and almost 80% of them have less than 10 years of work experience

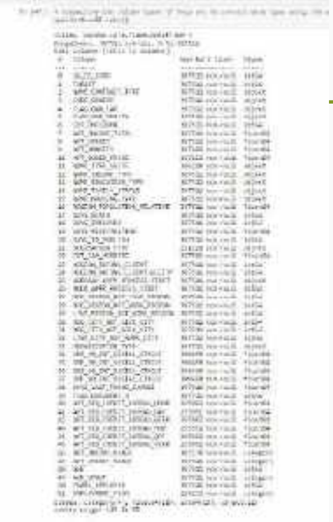
#Checking the number of unique values each column possess to identify categorical columns

applicationDF.nunique().sort_values()

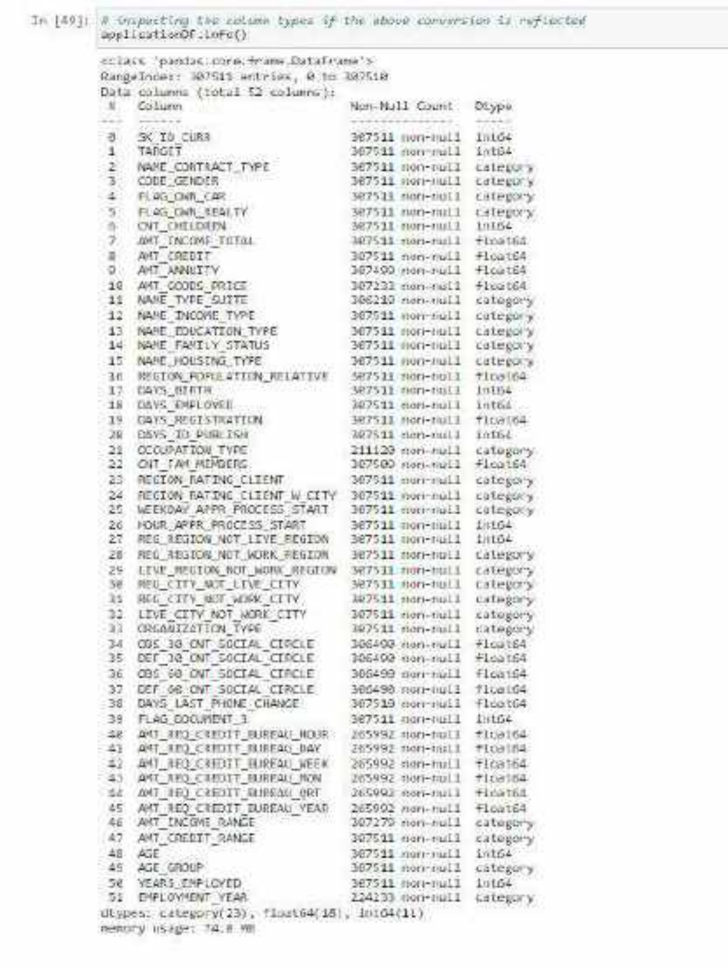


```
Out[46]: LIVE_CITY_NOT_WORK_CITY              2
         TARGET                               2
         NAME_CONTRACT_TYPE                   2
         REG_REGION_NOT_LIVE_REGION           2
         FLAG_OWN_CAR                         2
         FLAG_OWN_REALTY                      2
         REG_REGION_NOT_WORK_REGION           2
         LIVE_REGION_NOT_WORK_REGION          2
         FLAG_DOCUMENT_3                      2
         REG_CITY_NOT_LIVE_CITY               2
         REG_CITY_NOT_WORK_CITY               2
         REGION_RATING_CLIENT                 3
         CODE_GENDER                          3
         REGION_RATING_CLIENT_W_CITY          3
         AMT_REQ_CREDIT_BUREAU_HOUR           5
         NAME_EDUCATION_TYPE                  5
         AGE_GROUP                            5
         NAME_FAMILY_STATUS                   6
         NAME_HOUSING_TYPE                    6
         EMPLOYMENT_YEAR                      6
         WEEKDAY_APPR_PROCESS_START           7
         NAME_TYPE_SUITE                      7
         NAME_INCOME_TYPE                     8
         AMT_REQ_CREDIT_BUREAU_WEEK           9
         AMT_REQ_CREDIT_BUREAU_DAY            9
         DEF_60_CNT_SOCIAL_CIRCLE             9
         DEF_30_CNT_SOCIAL_CIRCLE            10
         AMT_CREDIT_RANGE                    11
         AMT_INCOME_RANGE                    11
         AMT_REQ_CREDIT_BUREAU_QRT           11
         CNT_CHILDREN                        15
         CNT_FAM_MEMBERS                     17
         OCCUPATION_TYPE                     18
         HOUR_APPR_PROCESS_START             24
         AMT_REQ_CREDIT_BUREAU_MON           24
         AMT_REQ_CREDIT_BUREAU_YEAR          25
         OBS_60_CNT_SOCIAL_CIRCLE            33
         OBS_30_CNT_SOCIAL_CIRCLE            33
         AGE                                 50
         YEARS_EMPLOYED                      51
         ORGANIZATION_TYPE                   58
         REGION_POPULATION_RELATIVE          81
         AMT_GOODS_PRICE                   1002
         AMT_INCOME_TOTAL                  2548
         DAYS_LAST_PHONE_CHANGE            3773
         AMT_CREDIT                        5603
         DAYS_ID_PUBLISH                   6168
         DAYS_EMPLOYED                    12574
         AMT_ANNUITY                      13672
         DAYS_REGISTRATION                15688
         DAYS_BIRTH                       17460
         SK_ID_CURR                      307511
         dtype: int64
```

4.5 Data Type Conversion



Insight:
Numeric columns are already in int64 and float64 format. Hence proceeding with other columns.

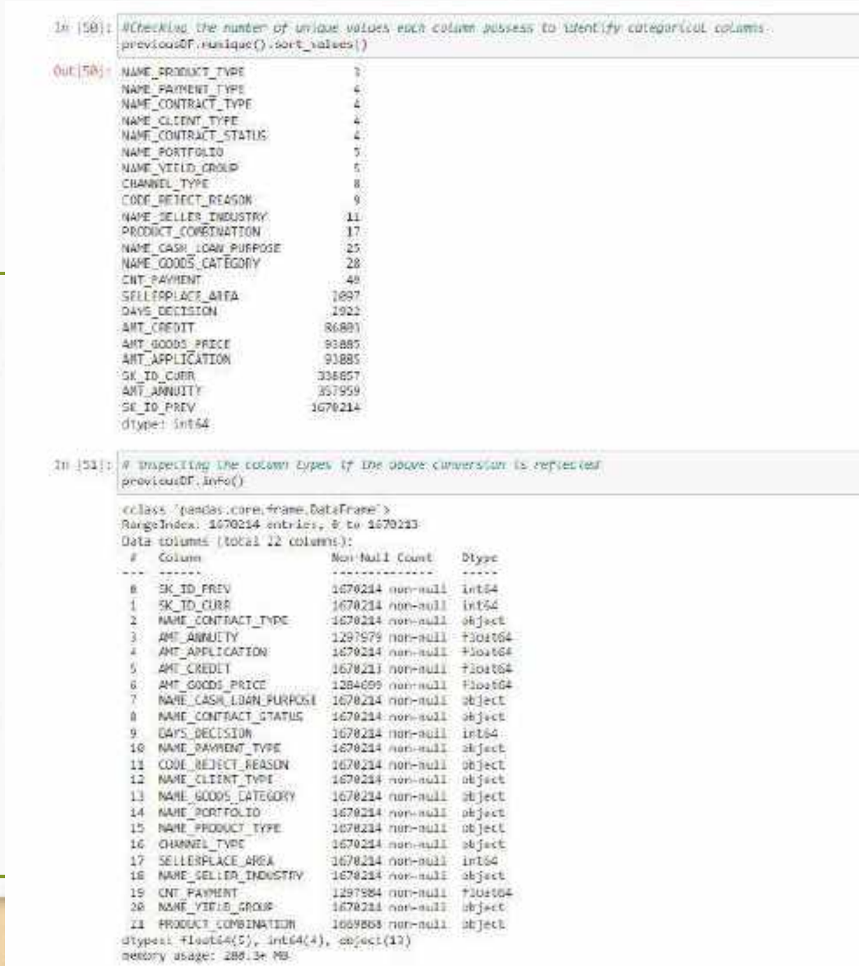


```
In [49]: # inspecting the column types if the above conversion is reflected
         applicationDF.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Data columns (total 52 columns):
 #   Column                       Non-Null Count   Dtype
---  ------                       --------------   -----
 0   SK_ID_CURR                   307511 non-null  int64
 1   TARGET                       307511 non-null  int64
 2   NAME_CONTRACT_TYPE           307511 non-null  category
 3   CODE_GENDER                  307511 non-null  category
 4   FLAG_OWN_CAR                 307511 non-null  category
 5   FLAG_OWN_REALTY              307511 non-null  category
 6   CNT_CHILDREN                 307511 non-null  int64
 7   AMT_INCOME_TOTAL             307511 non-null  float64
 8   AMT_CREDIT                   307511 non-null  float64
 9   AMT_ANNUITY                  307499 non-null  float64
 10  AMT_GOODS_PRICE              307233 non-null  float64
 11  NAME_TYPE_SUITE              306219 non-null  category
 12  NAME_INCOME_TYPE             307511 non-null  category
 13  NAME_EDUCATION_TYPE          307511 non-null  category
 14  NAME_FAMILY_STATUS           307511 non-null  category
 15  NAME_HOUSING_TYPE            307511 non-null  category
 16  REGION_POPULATION_RELATIVE   307511 non-null  float64
 17  DAYS_BIRTH                   307511 non-null  int64
 18  DAYS_EMPLOYED                307511 non-null  int64
 19  DAYS_REGISTRATION            307511 non-null  float64
 20  DAYS_ID_PUBLISH              307511 non-null  int64
 21  OCCUPATION_TYPE              211120 non-null  category
 22  CNT_FAM_MEMBERS              307509 non-null  float64
 23  REGION_RATING_CLIENT         307511 non-null  category
 24  REGION_RATING_CLIENT_W_CITY  307511 non-null  category
 25  WEEKDAY_APPR_PROCESS_START   307511 non-null  category
 26  HOUR_APPR_PROCESS_START      307511 non-null  int64
 27  REG_REGION_NOT_LIVE_REGION   307511 non-null  int64
 28  REG_REGION_NOT_WORK_REGION   307511 non-null  category
 29  LIVE_REGION_NOT_WORK_REGION  307511 non-null  category
 30  REG_CITY_NOT_LIVE_CITY       307511 non-null  category
 31  REG_CITY_NOT_WORK_CITY       307511 non-null  category
 32  LIVE_CITY_NOT_WORK_CITY      307511 non-null  category
 33  ORGANIZATION_TYPE            307511 non-null  category
 34  OBS_30_CNT_SOCIAL_CIRCLE     306490 non-null  float64
 35  DEF_30_CNT_SOCIAL_CIRCLE     306490 non-null  float64
 36  OBS_60_CNT_SOCIAL_CIRCLE     306490 non-null  float64
 37  DEF_60_CNT_SOCIAL_CIRCLE     306490 non-null  float64
 38  DAYS_LAST_PHONE_CHANGE       307510 non-null  float64
 39  FLAG_DOCUMENT_3              307511 non-null  int64
 40  AMT_REQ_CREDIT_BUREAU_HOUR   265992 non-null  float64
 41  AMT_REQ_CREDIT_BUREAU_DAY    265992 non-null  float64
 42  AMT_REQ_CREDIT_BUREAU_WEEK   265992 non-null  float64
 43  AMT_REQ_CREDIT_BUREAU_MON    265992 non-null  float64
 44  AMT_REQ_CREDIT_BUREAU_QRT    265992 non-null  float64
 45  AMT_REQ_CREDIT_BUREAU_YEAR   265992 non-null  float64
 46  AMT_INCOME_RANGE             307279 non-null  category
 47  AMT_CREDIT_RANGE             307511 non-null  category
 48  AGE                          307511 non-null  int64
 49  AGE_GROUP                    307511 non-null  category
 50  YEARS_EMPLOYED               307511 non-null  int64
 51  EMPLOYMENT_YEAR              224233 non-null  category
dtypes: category(23), float64(18), int64(11)
memory usage: 74.8 MB
```

4.4.2 Standardize Values for previousDF

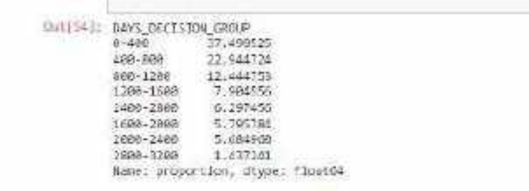
Strategy for previousDF:

- Convert DAYS_DECISION from negative to positive values and create categorical bins columns.
- Convert loan purpose and few other columns to categorical.



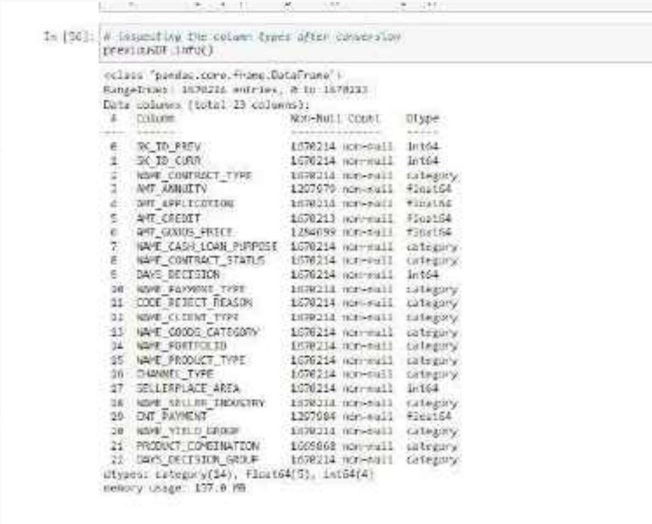
```
In [50]: #Checking the number of unique values each column possess to identify categorical columns
         previousDF.nunique().sort_values()

Out[50]: NAME_PRODUCT_TYPE              3
         NAME_PAYMENT_TYPE              4
         NAME_CONTRACT_TYPE             4
         NAME_CLIENT_TYPE               4
         NAME_CONTRACT_STATUS           4
         NAME_PORTFOLIO                 5
         NAME_YIELD_GROUP               5
         CHANNEL_TYPE                   8
         CODE_REJECT_REASON             9
         NAME_SELLER_INDUSTRY          11
         PRODUCT_COMBINATION           17
         NAME_CASH_LOAN_PURPOSE        25
         NAME_GOODS_CATEGORY           28
         CNT_PAYMENT                   49
         SELLERPLACE_AREA            2097
         DAYS_DECISION               2922
         AMT_CREDIT                 86803
         AMT_GOODS_PRICE            93885
         AMT_APPLICATION            93885
         SK_ID_CURR                338857
         AMT_ANNUITY               357959
         SK_ID_PREV               1670214
         dtype: int64
```

```
In [51]: # Inspecting the column types if the above conversion is reflected
         previousDF.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1670214 entries, 0 to 1670213
Data columns (total 22 columns):
 #   Column                  Non-Null Count    Dtype
---  ------                  --------------    -----
 0   SK_ID_PREV              1670214 non-null  int64
 1   SK_ID_CURR              1670214 non-null  int64
 2   NAME_CONTRACT_TYPE      1670214 non-null  object
 3   AMT_ANNUITY             1297979 non-null  float64
 4   AMT_APPLICATION         1670214 non-null  float64
 5   AMT_CREDIT              1670213 non-null  float64
 6   AMT_GOODS_PRICE         1284699 non-null  float64
 7   NAME_CASH_LOAN_PURPOSE  1670214 non-null  object
 8   NAME_CONTRACT_STATUS    1670214 non-null  object
 9   DAYS_DECISION           1670214 non-null  int64
 10  NAME_PAYMENT_TYPE       1670214 non-null  object
 11  CODE_REJECT_REASON      1670214 non-null  object
 12  NAME_CLIENT_TYPE        1670214 non-null  object
 13  NAME_GOODS_CATEGORY     1670214 non-null  object
 14  NAME_PORTFOLIO          1670214 non-null  object
 15  NAME_PRODUCT_TYPE       1670214 non-null  object
 16  CHANNEL_TYPE            1670214 non-null  object
 17  SELLERPLACE_AREA        1670214 non-null  int64
 18  NAME_SELLER_INDUSTRY    1670214 non-null  object
 19  CNT_PAYMENT             1297984 non-null  float64
 20  NAME_YIELD_GROUP        1670214 non-null  object
 21  PRODUCT_COMBINATION     1669868 non-null  object
dtypes: float64(5), int64(4), object(13)
memory usage: 280.3+ MB
```



```
Out[54]: DAYS_DECISION_GROUP
         0-400        37.499525
         400-800      22.944724
         800-1200     12.444753
         1200-1600     7.904556
         2400-2800     6.297456
         1600-2000     5.795788
         2000-2400     5.084900
         2800-3200     1.637141
         Name: proportion, dtype: float64
```

Insight:
Almost 37% loan applicatants have applied for a new loan within 0-400 days of previous loan decision



```
In [56]: # inspecting the column types after conversion
         previousDF.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1670214 entries, 0 to 1670213
Data columns (total 23 columns):
 #   Column                  Non-Null Count    Dtype
---  ------                  --------------    -----
 0   SK_ID_PREV              1670214 non-null  int64
 1   SK_ID_CURR              1670214 non-null  int64
 2   NAME_CONTRACT_TYPE      1670214 non-null  category
 3   AMT_ANNUITY             1297979 non-null  float64
 4   AMT_APPLICATION         1670214 non-null  float64
 5   AMT_CREDIT              1670213 non-null  float64
 6   AMT_GOODS_PRICE         1284699 non-null  float64
 7   NAME_CASH_LOAN_PURPOSE  1670214 non-null  category
 8   NAME_CONTRACT_STATUS    1670214 non-null  category
 9   DAYS_DECISION           1670214 non-null  int64
 10  NAME_PAYMENT_TYPE       1670214 non-null  category
 11  CODE_REJECT_REASON      1670214 non-null  category
 12  NAME_CLIENT_TYPE        1670214 non-null  category
 13  NAME_GOODS_CATEGORY     1670214 non-null  category
 14  NAME_PORTFOLIO          1670214 non-null  category
 15  NAME_PRODUCT_TYPE       1670214 non-null  category
 16  CHANNEL_TYPE            1670214 non-null  category
 17  SELLERPLACE_AREA        1670214 non-null  int64
 18  NAME_SELLER_INDUSTRY    1670214 non-null  category
 19  CNT_PAYMENT             1297984 non-null  float64
 20  NAME_YIELD_GROUP        1670214 non-null  category
 21  PRODUCT_COMBINATION     1669868 non-null  category
 22  DAYS_DECISION_GROUP     1670214 non-null  category
dtypes: category(14), float64(5), int64(4)
memory usage: 137.0 MB
```

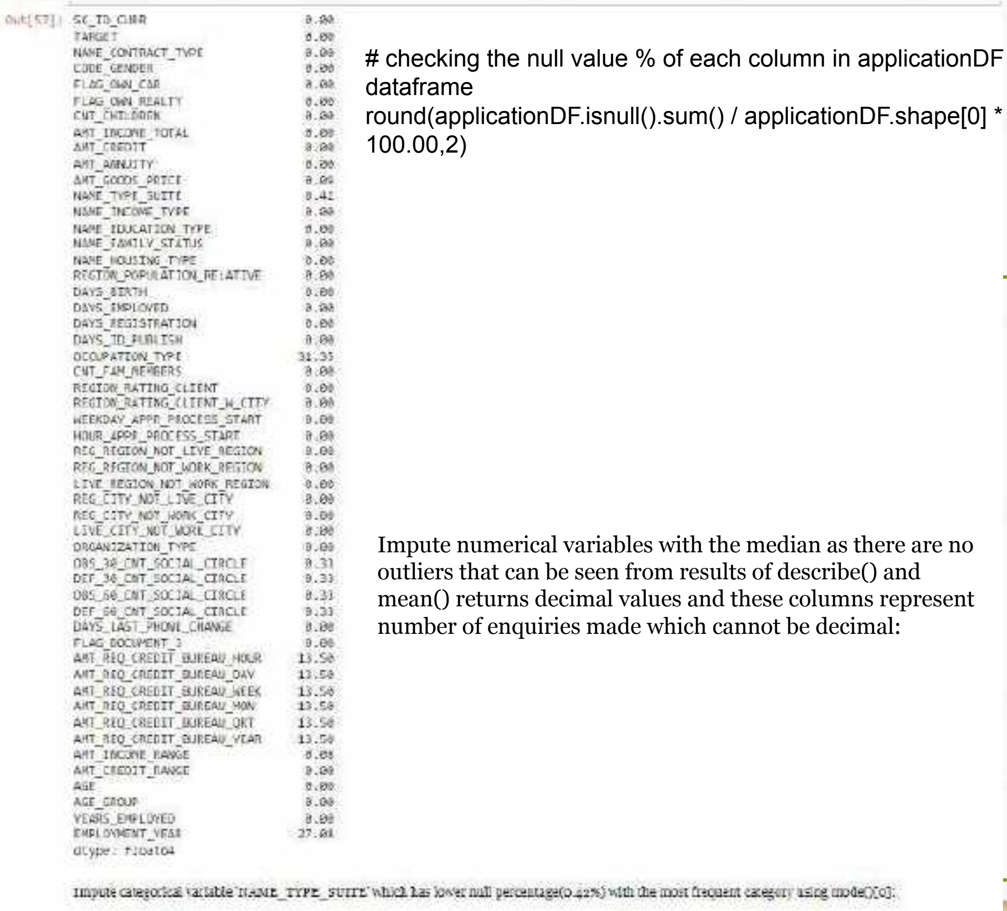
4.6 Null Value Data Imputation

4.6.1 Imputing Null Values in applicationDF

Strategy for applicationDF:

- To impute null values in categorical variables which has lower null percentage, mode() is used to impute the most frequent items.
- To impute null values in categorical variables which has higher null percentage, a new category is created.
- To impute null values in numerical variables which has lower null percentage, median() is used as
    - There are no outliers in the columns
    - Mean returned decimal values and median returned whole numbers and the columns were number of requests

# checking the null value % of each column in applicationDF dataframe
round(applicationDF.isnull().sum() / applicationDF.shape[0] * 100.00,2)

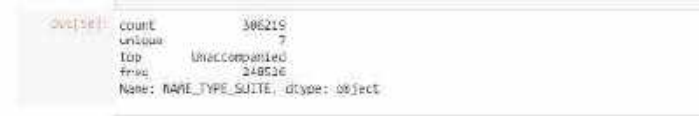


```
Out[57]: SK_ID_CURR                       0.00
         TARGET                           0.00
         NAME_CONTRACT_TYPE               0.00
         CODE_GENDER                      0.00
         FLAG_OWN_CAR                     0.00
         FLAG_OWN_REALTY                  0.00
         CNT_CHILDREN                     0.00
         AMT_INCOME_TOTAL                 0.00
         AMT_CREDIT                       0.00
         AMT_ANNUITY                      0.00
         AMT_GOODS_PRICE                  0.00
         NAME_TYPE_SUITE                  0.42
         NAME_INCOME_TYPE                 0.00
         NAME_EDUCATION_TYPE              0.00
         NAME_FAMILY_STATUS               0.00
         NAME_HOUSING_TYPE                0.00
         REGION_POPULATION_RELATIVE       0.00
         DAYS_BIRTH                       0.00
         DAYS_EMPLOYED                    0.00
         DAYS_REGISTRATION                0.00
         DAYS_ID_PUBLISH                  0.00
         OCCUPATION_TYPE                 31.35
         CNT_FAM_MEMBERS                  0.00
         REGION_RATING_CLIENT             0.00
         REGION_RATING_CLIENT_W_CITY      0.00
         WEEKDAY_APPR_PROCESS_START       0.00
         HOUR_APPR_PROCESS_START          0.00
         REG_REGION_NOT_LIVE_REGION       0.00
         REG_REGION_NOT_WORK_REGION       0.00
         LIVE_REGION_NOT_WORK_REGION      0.00
         REG_CITY_NOT_LIVE_CITY           0.00
         REG_CITY_NOT_WORK_CITY           0.00
         LIVE_CITY_NOT_WORK_CITY          0.00
         ORGANIZATION_TYPE                0.00
         OBS_30_CNT_SOCIAL_CIRCLE         0.33
         DEF_30_CNT_SOCIAL_CIRCLE         0.33
         OBS_60_CNT_SOCIAL_CIRCLE         0.33
         DEF_60_CNT_SOCIAL_CIRCLE         0.33
         DAYS_LAST_PHONE_CHANGE           0.00
         FLAG_DOCUMENT_3                  0.00
         AMT_REQ_CREDIT_BUREAU_HOUR      13.50
         AMT_REQ_CREDIT_BUREAU_DAY       13.50
         AMT_REQ_CREDIT_BUREAU_WEEK      13.50
         AMT_REQ_CREDIT_BUREAU_MON       13.50
         AMT_REQ_CREDIT_BUREAU_QRT       13.50
         AMT_REQ_CREDIT_BUREAU_YEAR      13.50
         AMT_INCOME_RANGE                 0.08
         AMT_CREDIT_RANGE                 0.00
         AGE                              0.00
         AGE_GROUP                        0.00
         YEARS_EMPLOYED                   0.00
         EMPLOYMENT_YEAR                 27.01
         dtype: float64
```

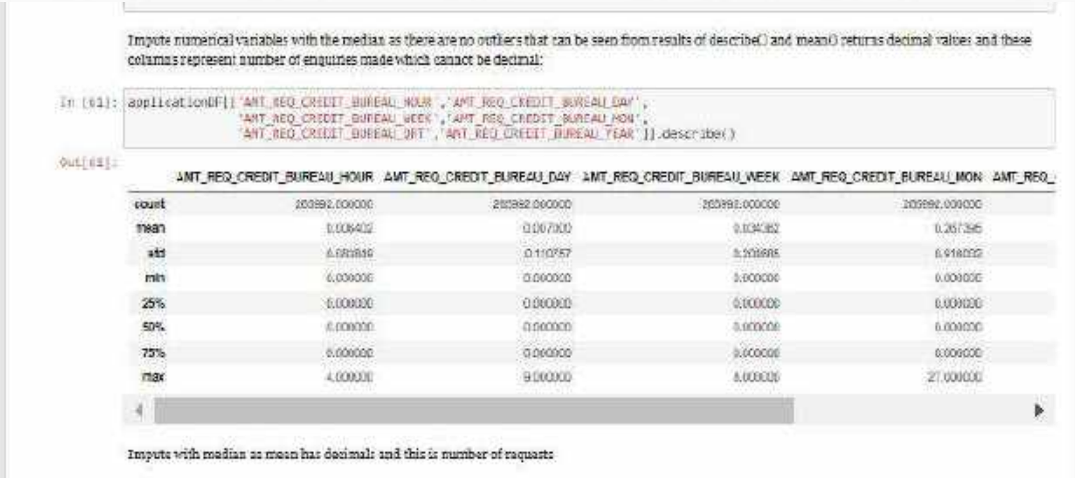
Impute numerical variables with the median as there are no outliers that can be seen from results of describe() and mean() returns decimal values and these columns represent number of enquiries made which cannot be decimal:

Impute categorical variable 'NAME_TYPE_SUITE' which has lower null percentage(0.42%) with the most frequent category using mode()[0]:

Impute categorical variable 'OCCUPATION_TYPE' which has higher null percentage(31.35%) with a new category as assigning to any existing category might influence the analysis



```
Out[58]: count              306219
         unique                  7
         top         Unaccompanied
         freq               248526
         Name: NAME_TYPE_SUITE, dtype: object
```



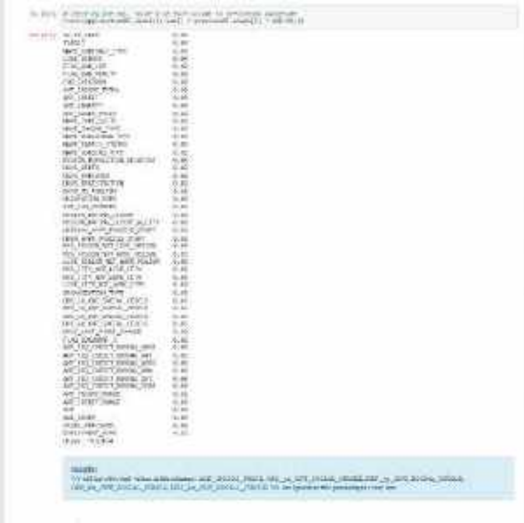
Impute numerical variables with the median as there are no outliers that can be seen from results of describe() and mean() returns decimal values and these columns represent number of enquiries made which cannot be decimal:

```
In [61]: applicationDF[['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']].describe()
```

Out[61]:

	AMT_REQ_CREDIT_BUREAU_HOUR	AMT_REQ_CREDIT_BUREAU_DAY	AMT_REQ_CREDIT_BUREAU_WEEK	AMT_REQ_CREDIT_BUREAU_MON	AMT_REQ_
count	265992.000000	265992.000000	265992.000000	265992.000000	
mean	0.006402	0.007000	0.034362	0.267395	
std	0.083849	0.110757	0.204685	0.916002	
min	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	0.000000	
75%	0.000000	0.000000	0.000000	0.000000	
max	4.000000	9.000000	8.000000	27.000000	

Impute with median as mean has decimals and this is number of requests

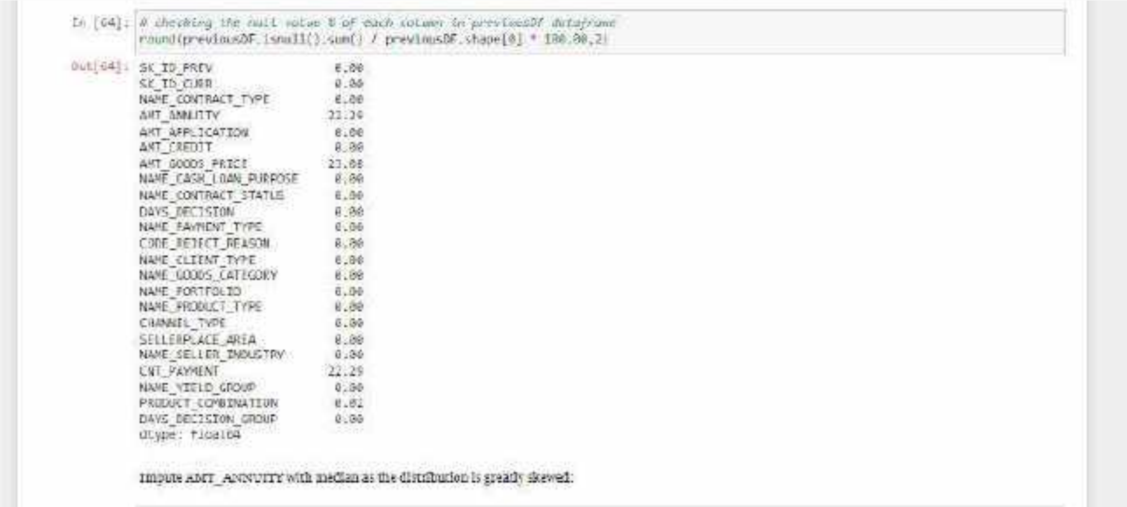


[illegible]

4.6.2 Imputing Null Values in previousDF

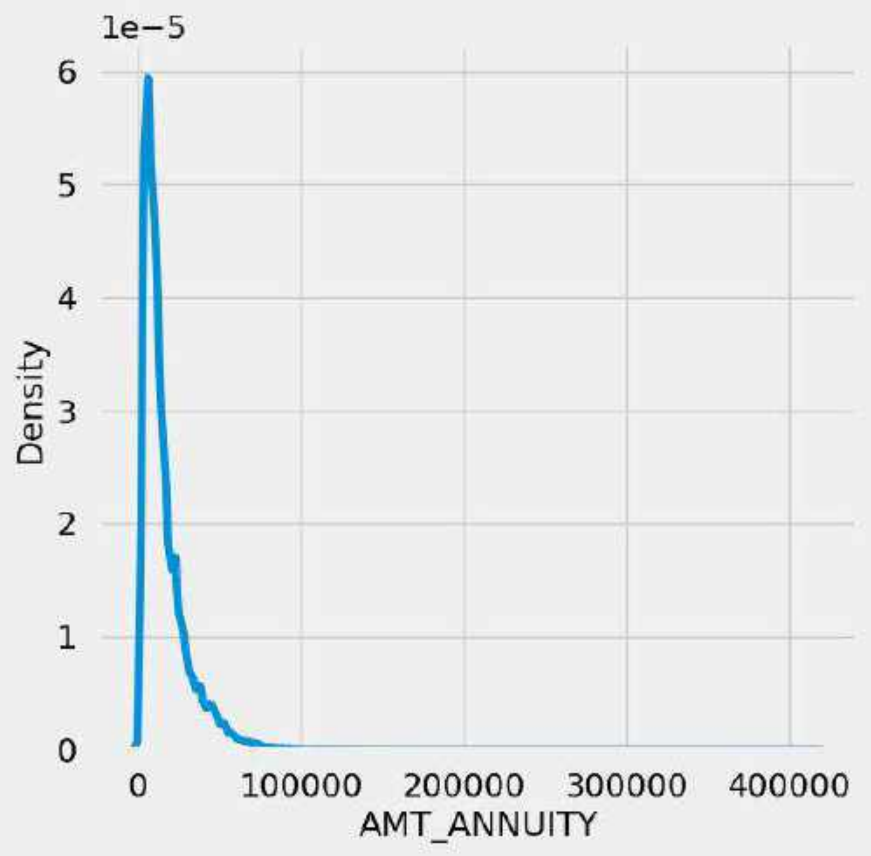
Strategy for applicationDF:

- To impute null values in numerical column, we analysed the loan status and assigned values.
- To impute null values in continuous variables, we plotted the distribution of the columns and used
  - median if the distribution is skewed
  - mode if the distribution pattern is preserved.



```
In [64]: # checking the null value % of each column in previousDF dataframe
         round(previousDF.isnull().sum() / previousDF.shape[0] * 100.00,2)

Out[64]: SK_ID_PREV                 0.00
         SK_ID_CURR                 0.00
         NAME_CONTRACT_TYPE         0.00
         AMT_ANNUITY               22.29
         AMT_APPLICATION            0.00
         AMT_CREDIT                 0.00
         AMT_GOODS_PRICE           23.08
         NAME_CASH_LOAN_PURPOSE     0.00
         NAME_CONTRACT_STATUS       0.00
         DAYS_DECISION              0.00
         NAME_PAYMENT_TYPE          0.00
         CODE_REJECT_REASON         0.00
         NAME_CLIENT_TYPE           0.00
         NAME_GOODS_CATEGORY        0.00
         NAME_PORTFOLIO             0.00
         NAME_PRODUCT_TYPE          0.00
         CHANNEL_TYPE               0.00
         SELLERPLACE_AREA           0.00
         NAME_SELLER_INDUSTRY       0.00
         CNT_PAYMENT               22.29
         NAME_YIELD_GROUP           0.00
         PRODUCT_COMBINATION        0.02
         DAYS_DECISION_GROUP        0.00
         dtype: float64
```

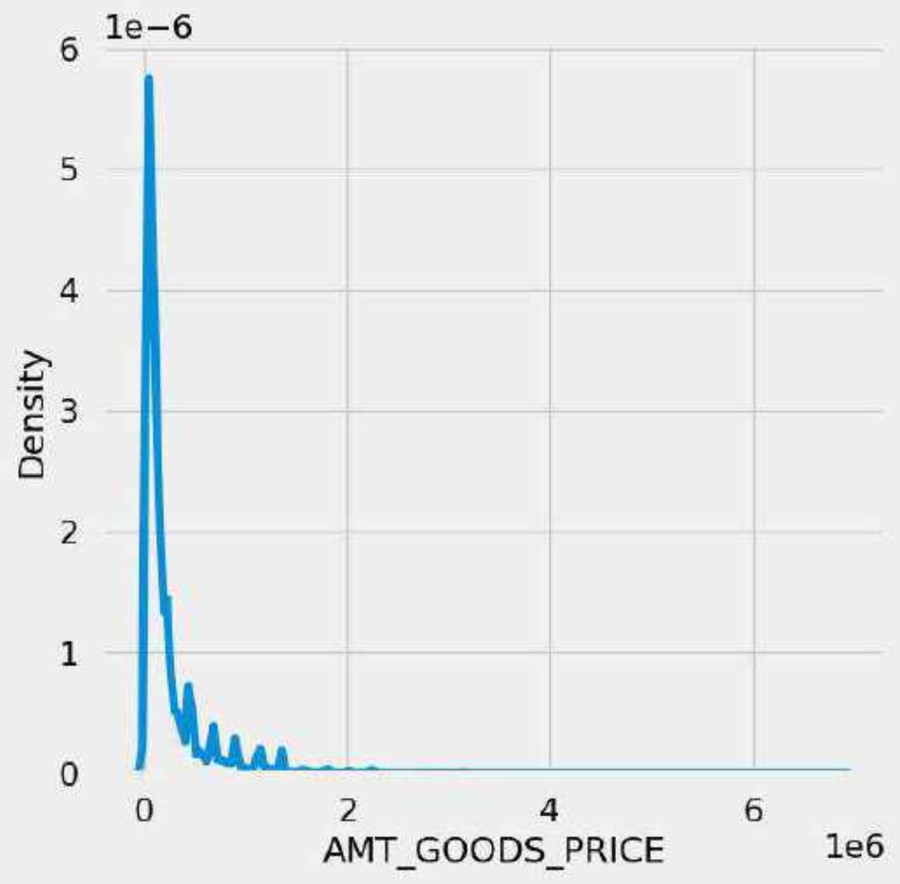
Impute AMT_ANNUITY with median as the distribution is greatly skewed:



Impute AMT_ANNUITY with median as the distribution is greatly skewed:

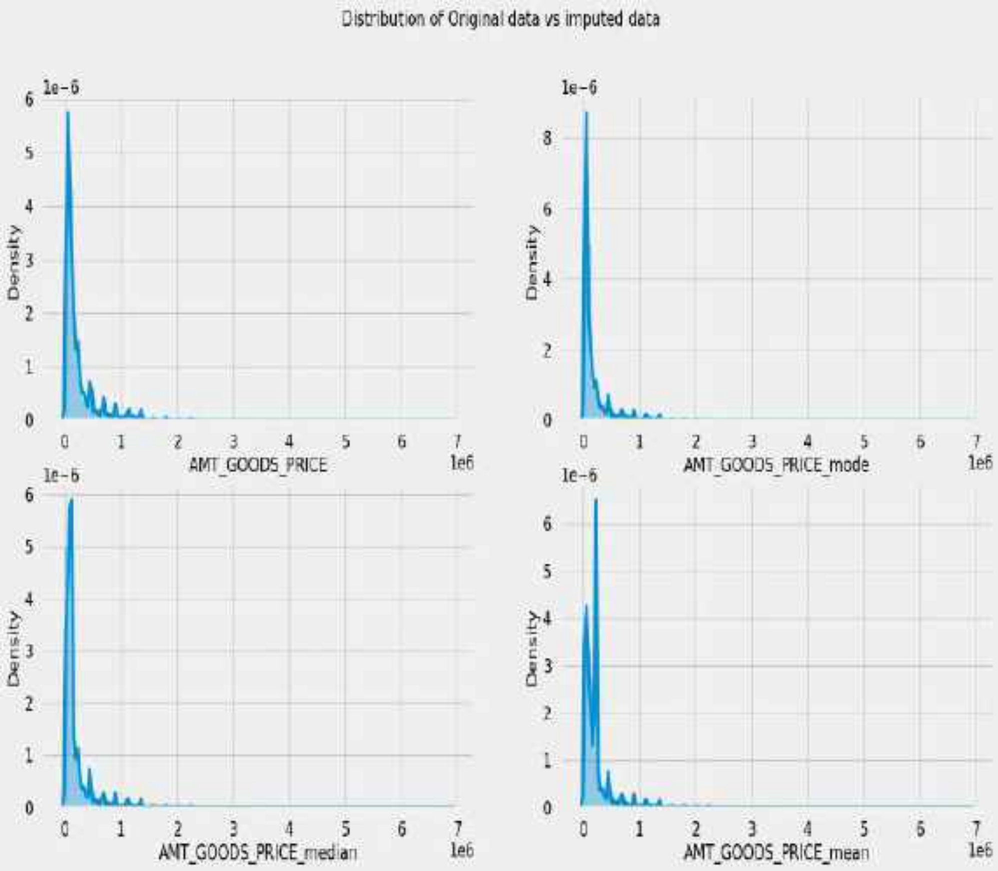
- **Insight:**
  There is a single peak at the left side of the distribution and it indicates the presence of outliers and hence imputing with mean would not be the right approach and hence imputing with median.

Impute AMT_GOODS_PRICE with mode as the distribution is closely similar:

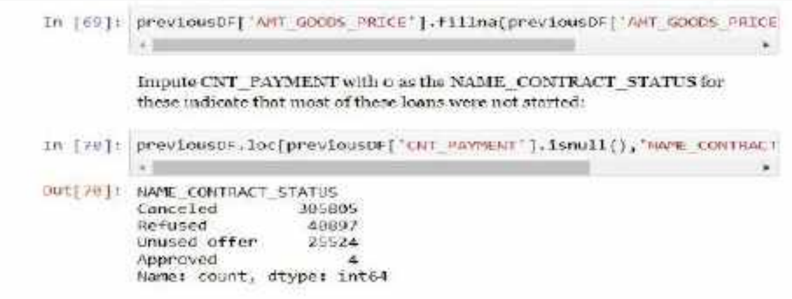


- There are several peaks along the distribution. Let's impute using the mode, mean and median and see if the distribution is still about the same.

Insight:
The original distribution is closer with the distribution of data imputed with mode in this case



There are several peaks along the distribution. Let's impute using the mode, mean and median and see if the distribution is still about the same.

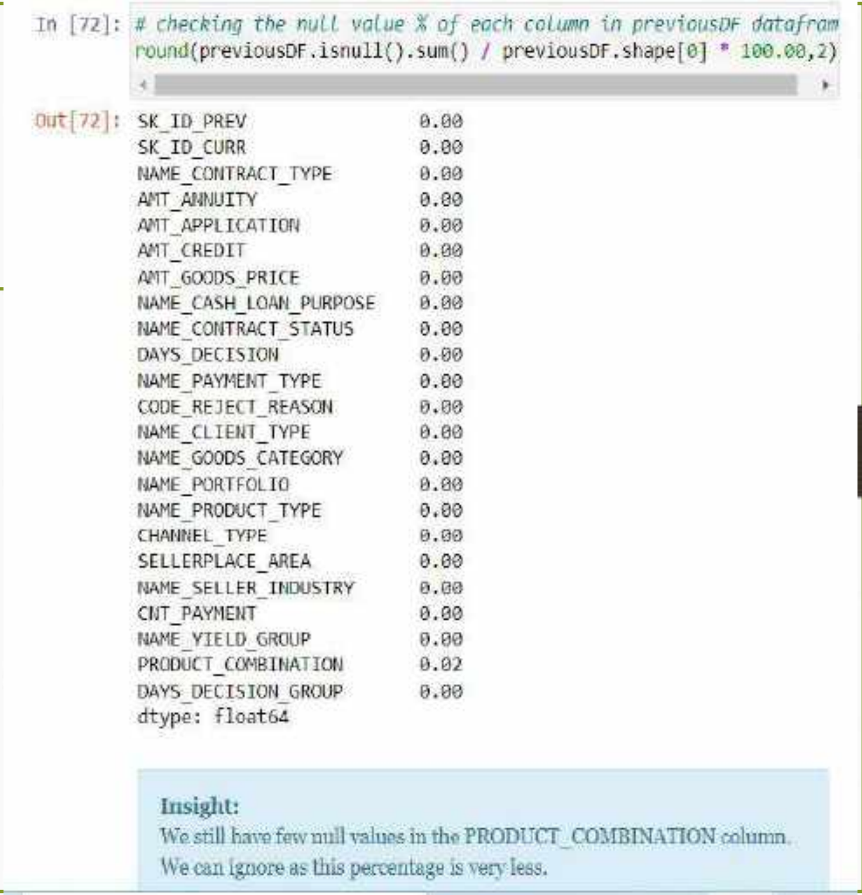


```
In [69]: previousDF['AMT_GOODS_PRICE'].fillna(previousDF['AMT_GOODS_PRICE
```

Impute CNT_PAYMENT with 0 as the NAME_CONTRACT_STATUS for these indicate that most of these loans were not started:

```
In [70]: previousDF.loc[previousDF['CNT_PAYMENT'].isnull(),'NAME_CONTRACT
```

```
Out[70]: NAME_CONTRACT_STATUS
         Canceled          305805
         Refused            40897
         Unused offer       25524
         Approved               4
         Name: count, dtype: int64
```



```
In [72]: # checking the null value % of each column in previousDF datafram
         round(previousDF.isnull().sum() / previousDF.shape[0] * 100.00,2)
```

```
Out[72]: SK_ID_PREV                 0.00
         SK_ID_CURR                 0.00
         NAME_CONTRACT_TYPE         0.00
         AMT_ANNUITY                0.00
         AMT_APPLICATION            0.00
         AMT_CREDIT                 0.00
         AMT_GOODS_PRICE            0.00
         NAME_CASH_LOAN_PURPOSE     0.00
         NAME_CONTRACT_STATUS       0.00
         DAYS_DECISION              0.00
         NAME_PAYMENT_TYPE          0.00
         CODE_REJECT_REASON         0.00
         NAME_CLIENT_TYPE           0.00
         NAME_GOODS_CATEGORY        0.00
         NAME_PORTFOLIO             0.00
         NAME_PRODUCT_TYPE          0.00
         CHANNEL_TYPE               0.00
         SELLERPLACE_AREA           0.00
         NAME_SELLER_INDUSTRY       0.00
         CNT_PAYMENT                0.00
         NAME_YIELD_GROUP           0.00
         PRODUCT_COMBINATION        0.02
         DAYS_DECISION_GROUP        0.00
         dtype: float64
```

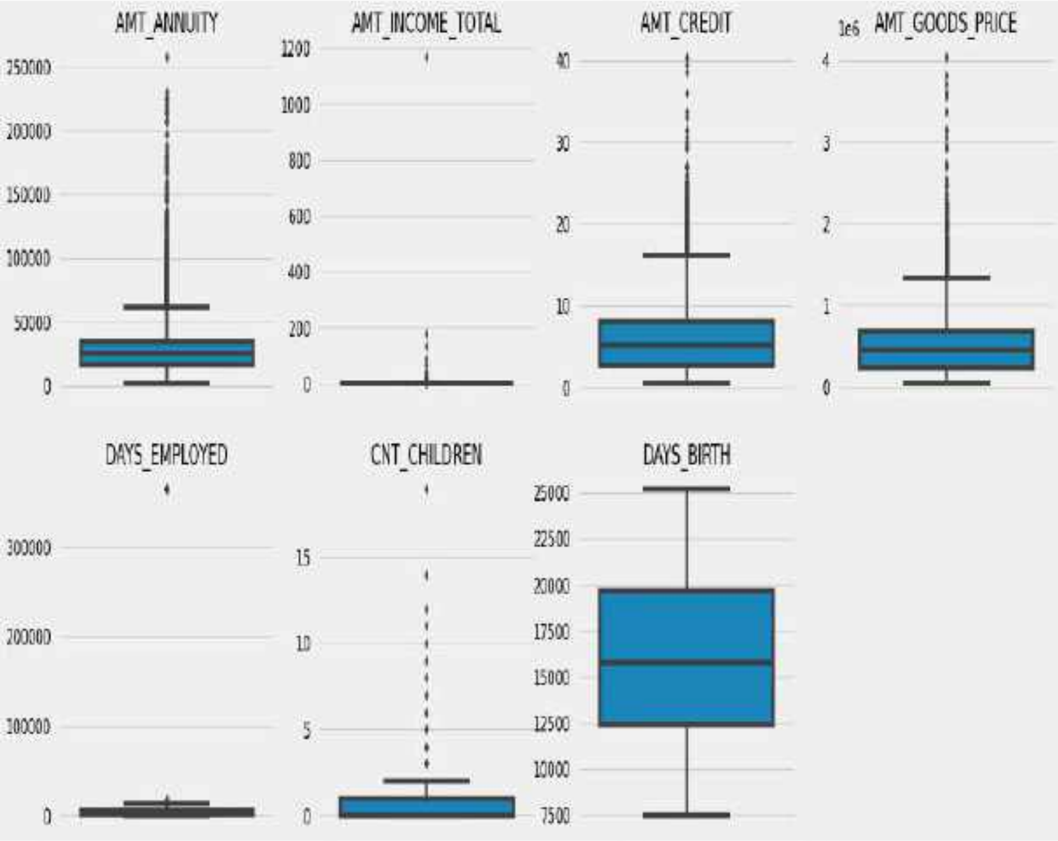
Insight:

We still have few null values in the PRODUCT_COMBINATION column. We can ignore as this percentage is very less.

4.7 Identifying the outliers

Finding outlier information in applicationDF

Finding outlier information in applicationDF

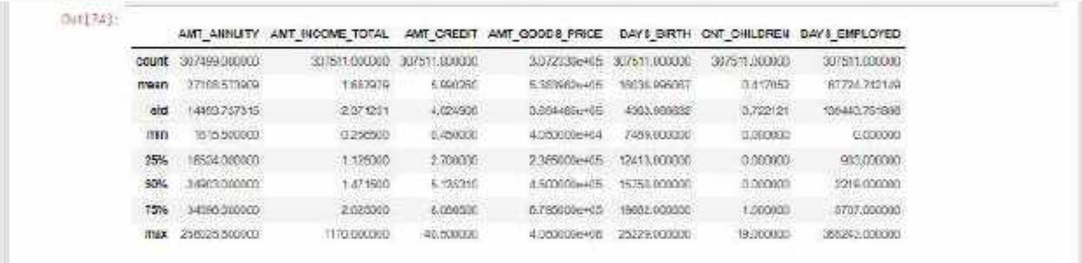


Insight:
It can be seen that in current application dataAMT_ANNUITY, AMT_CREDIT, AMT_GOODS_PRICE,CNT_CHILDREN have some number of outliers.

AMT_INCOME_TOTAL has huge number of outliers which indicate that few of the loan applicants have high income when compared to the others.

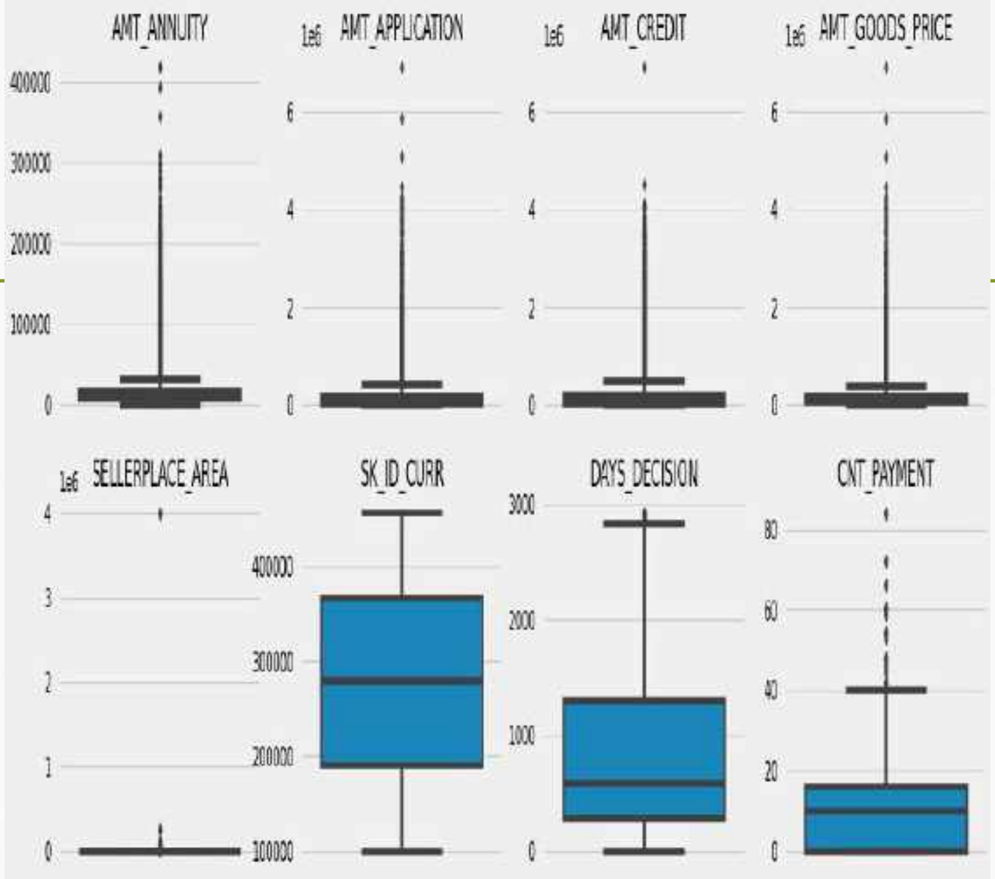
DAYS_BIRTH has no outliers which means the data available is reliable.

DAYS_EMPLOYED has outlier values around 350000(days) which is around 958 years which is impossible and hence this has to be incorrect entry.



```
Out[74]:
       AMT_ANNUITY   AMT_INCOME_TOTAL   AMT_CREDIT  AMT_GOODS_PRICE    DAYS_BIRTH  CNT_CHILDREN  DAYS_EMPLOYED
count  307499.000000    307511.000000 307511.000000   3.072330e+05 307511.000000 307511.000000  307511.000000
mean    27108.573909         1.687979      5.990260   5.383962e+05  16036.995067      0.417052   67724.742149
std     14493.737315         2.371231      4.024908   3.694465e+05   4363.988632      0.722121  139443.751806
min      1615.500000         0.256500      0.450000   4.050000e+04   7489.000000      0.000000       0.000000
25%     16524.000000         1.125000      2.700000   2.385000e+05  12413.000000      0.000000     933.000000
50%     24903.000000         1.471500      5.135310   4.500000e+05  15750.000000      0.000000    2219.000000
75%     34596.000000         2.025000      8.086500   6.795000e+05  19682.000000      1.000000    5707.000000
max    258025.500000      1170.000000     40.500000   4.050000e+06  25229.000000     19.000000  365243.000000
```

Finding outlier information in previousDF



- **Insight:** It can be seen that in previous application dataAMT_ANNUITY, AMT_APPLICATION, AMT_CREDIT, AMT_GOODS_PRICE, SELLERPLACE_AREA have huge number of outliers.
- CNT_PAYMENT has few outlier values.
- SK_ID_CURR is an ID column and hence no outliers.
- DAYS_DECISION has little number of outliers indicating that these previous applications decisions were taken long back.

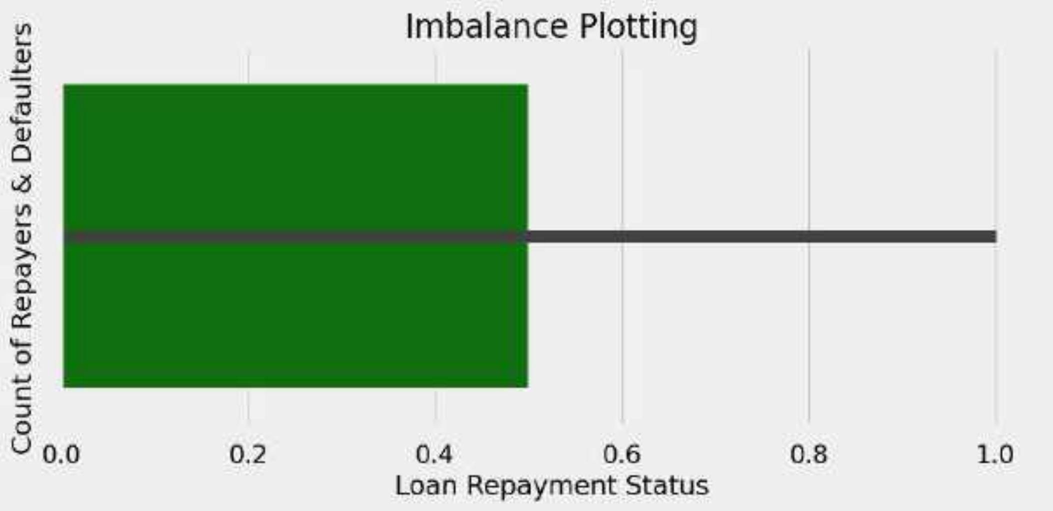
5. Data Analysis

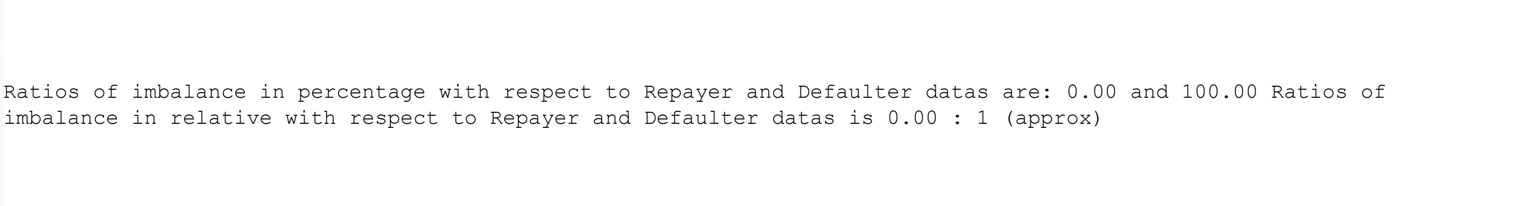
Strategy:

The data analysis flow has been planned in following way :

- Imbalance in Data
- Categorical Data Analysis
  - Categorical segmented Univariate Analysis
  - Categorical Bi/Multivariate analysis
- Numeric Data Analysis
  - Bi-furcation of databased based on TARGET data
  - Correlation Matrix
  - Numerical segmented Univariate Analysis
  - Numerical Bi/Multivariate analysis

5.1 Imbalance Analysis





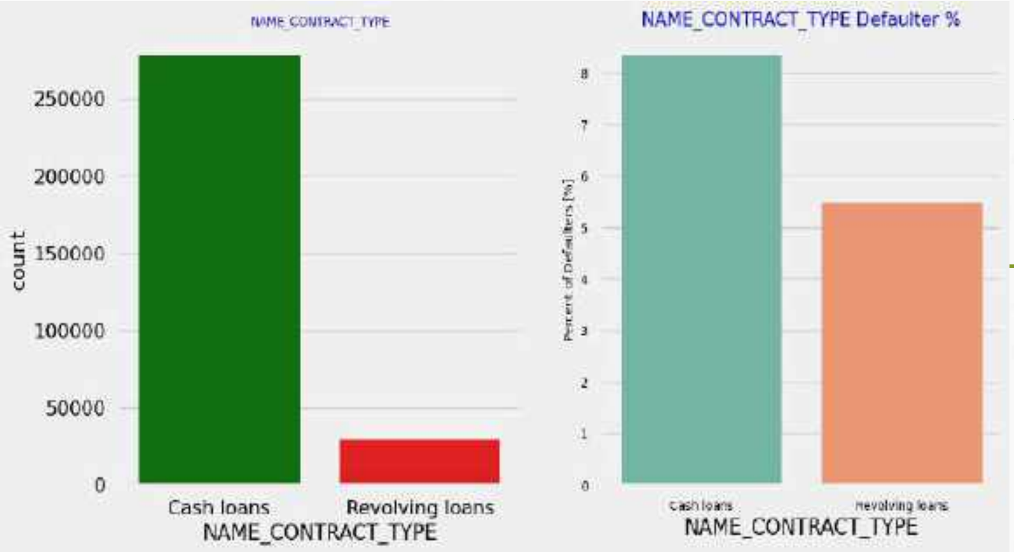
```
Ratios of imbalance in percentage with respect to Repayer and Defaulter datas are: 0.00 and 100.00 Ratios of
imbalance in relative with respect to Repayer and Defaulter datas is 0.00 : 1 (approx)
```

5.2 Plotting Functions

Following are the common functions customized to perform uniform anaysis that is called for all plots:

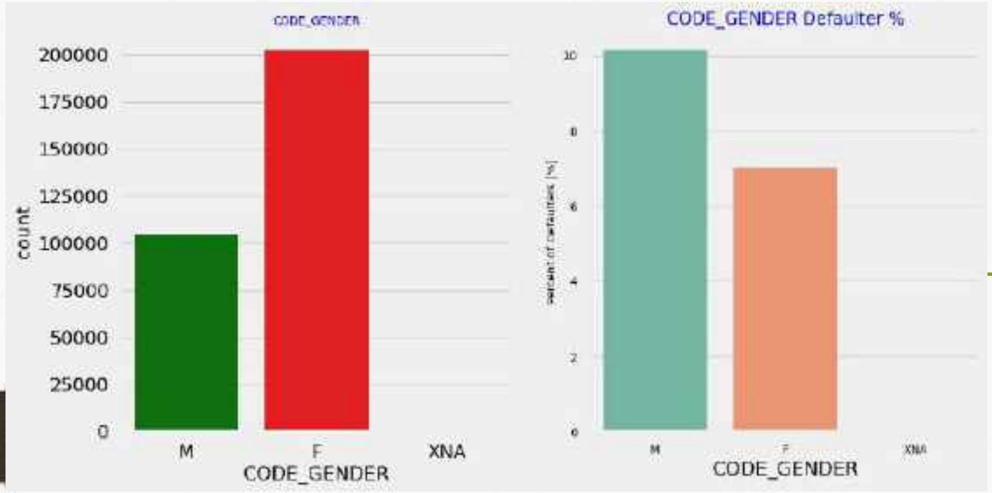
5.3 Categorical Variables Analysis

5.3.1 Segmented Univariate Analysis



```
# Checking the contract type based on loan repayment status
univariate_categorical('NAME_CONTRACT_TYPE', ylog=False, label_rotation=False,
horizontal_layout=True)
```

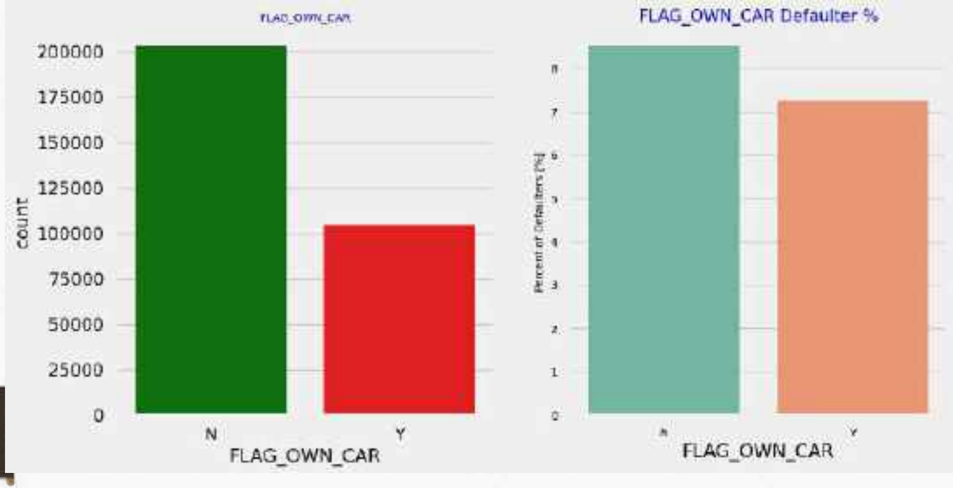
- **Inferences:**
  Contract type: Revolving loans are just a small fraction (10%) from the total number of loans; in the same time, a larger amount of Revolving loans, comparing with their frequency, are not repaid.



# Checking the type of Gender on loan repayment status
univariate_categorical('CODE_GENDER')

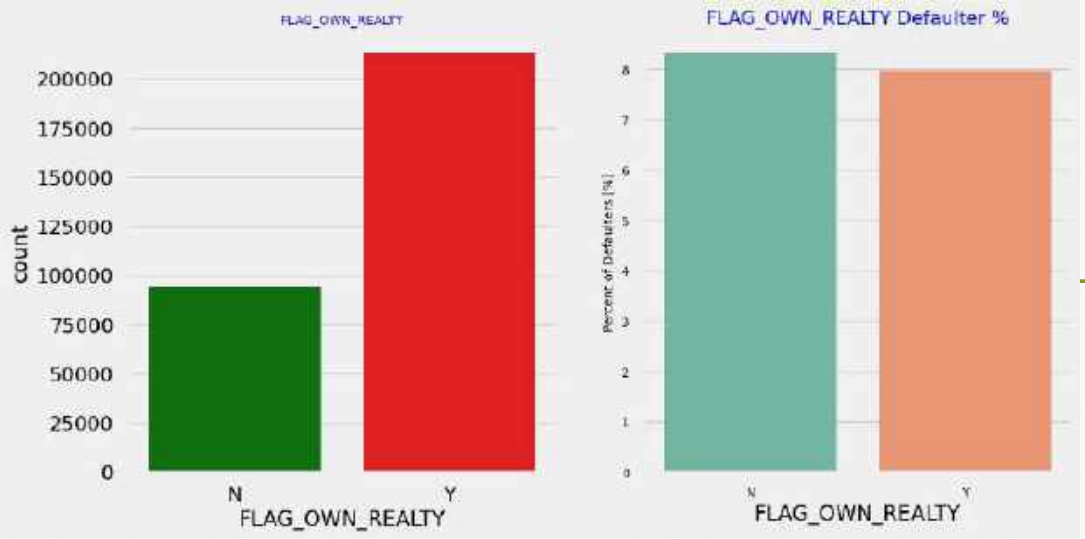
- **Inferences:**

  The number of female clients is almost double the number of male clients. Based on the percentage of defaulted credits, males have a higher chance of not returning their loans (~10%), comparing with women (~7%)



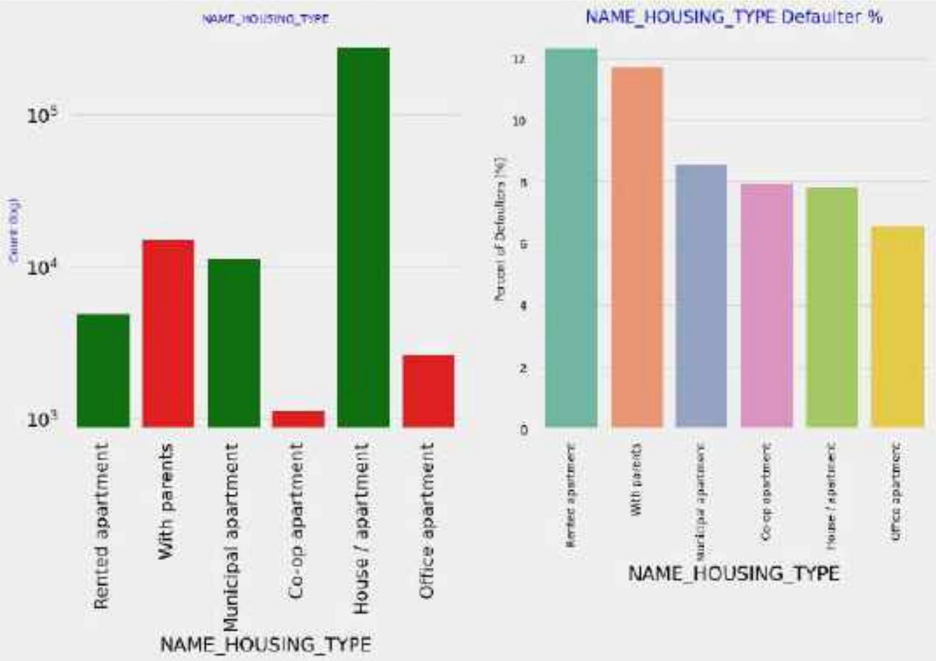
```
# Checking if owning a car is related to loan repayment status
univariate_categorical('FLAG_OWN_CAR')
```

- **Inferences:**
  Clients who own a car are half in number of the clients who dont own a car. But based on the percentage of deault, there is no correlation between owning a car and loan repayment as in both cases the default percentage is almost same.



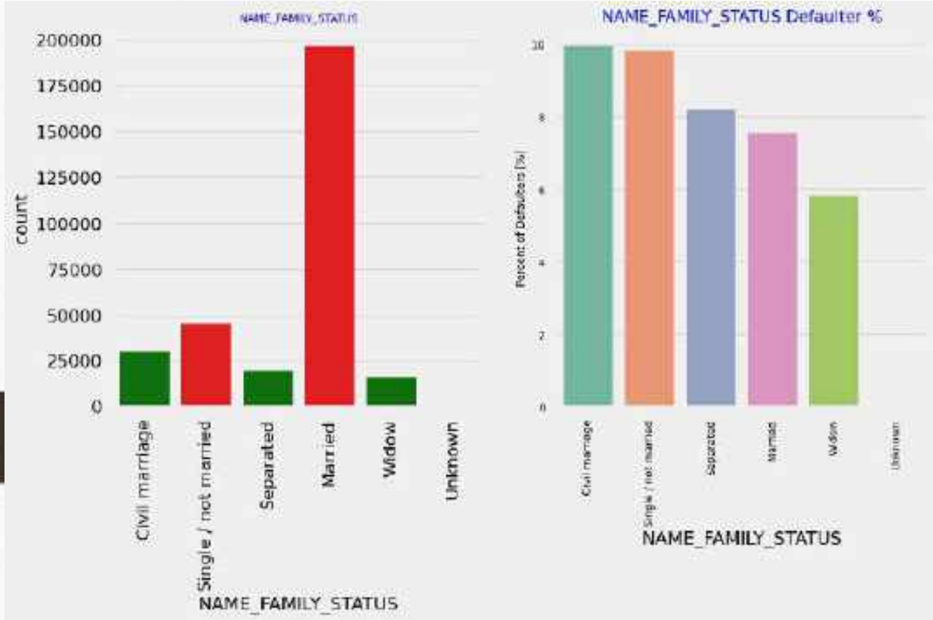
# Checking if owning a realty is related to loan repayment status
univariate_categorical('FLAG_OWN_REALTY')

- **Inferences:**
  The clients who own real estate are more than double of the ones that don't own. But the defaulting rate of both categories are around the same (~8%). Thus there is no correlation between owning a reality and defaulting the loan.



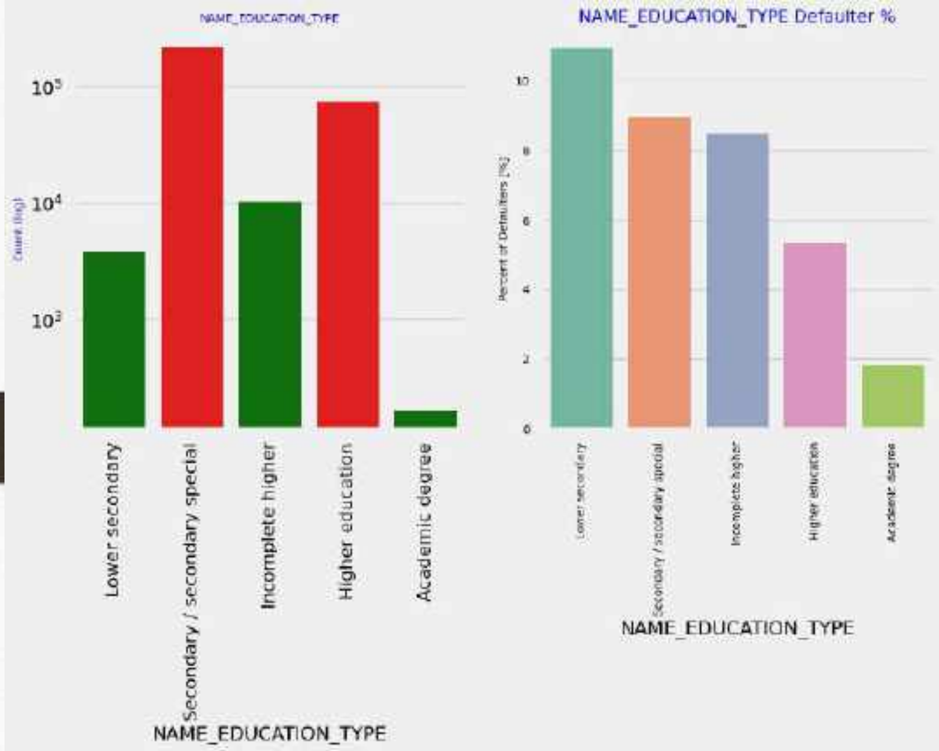
# Analyzing Housing Type based on loan repayment status univariate_categorical("NAME_HOUSING_TYPE",True,True,True)

- **Inferences:**Majority of people live in House/apartment
- People living in office apartments have lowest default rate
- People living with parents (~11.5%) and living in rented apartments(>12%) have higher probability of defaulting



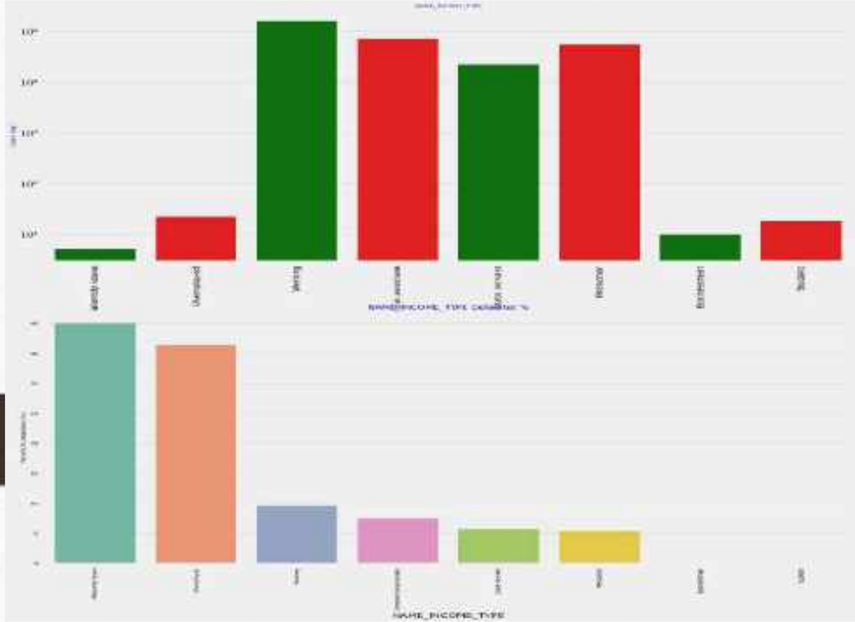
# Analyzing Family status based on loan repayment status
univariate_categorical("NAME_FAMILY_STATUS", False,True,True)

- **Inferences:**Most of the people who have taken loan are married, followed by Single/not married and civil marriage
- In terms of percentage of not repayment of loan, Civil marriage has the highest percent of not repayment (10%), with Widow the lowest (exception being Unknown).



# Analyzing Education Type based on loan repayment status univariate_categorical("NAME_EDUCATION_TYPE",True,True,True)

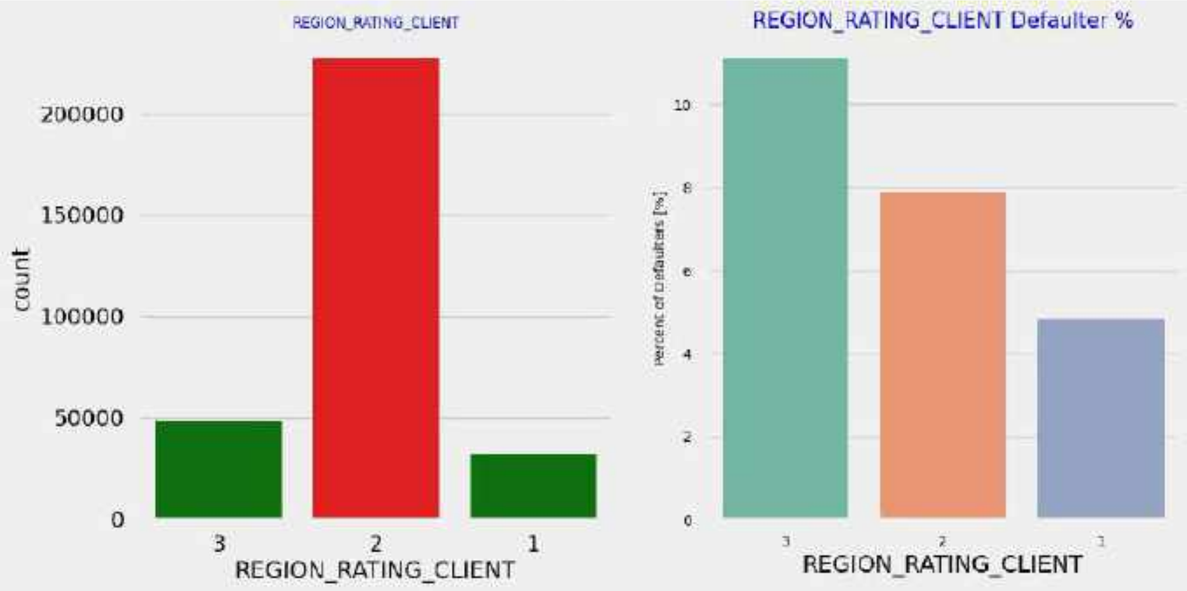
- **Inferences:**Majority of the clients have Secondary / secondary special education, followed by clients with Higher education. Only a very small number having an academic degree
- The Lower secondary category, although rare, have the largest rate of not returning the loan (11%). The people with Academic degree have less than 2% defaulting rate.



# Analyzing Income Type based on loan repayment status

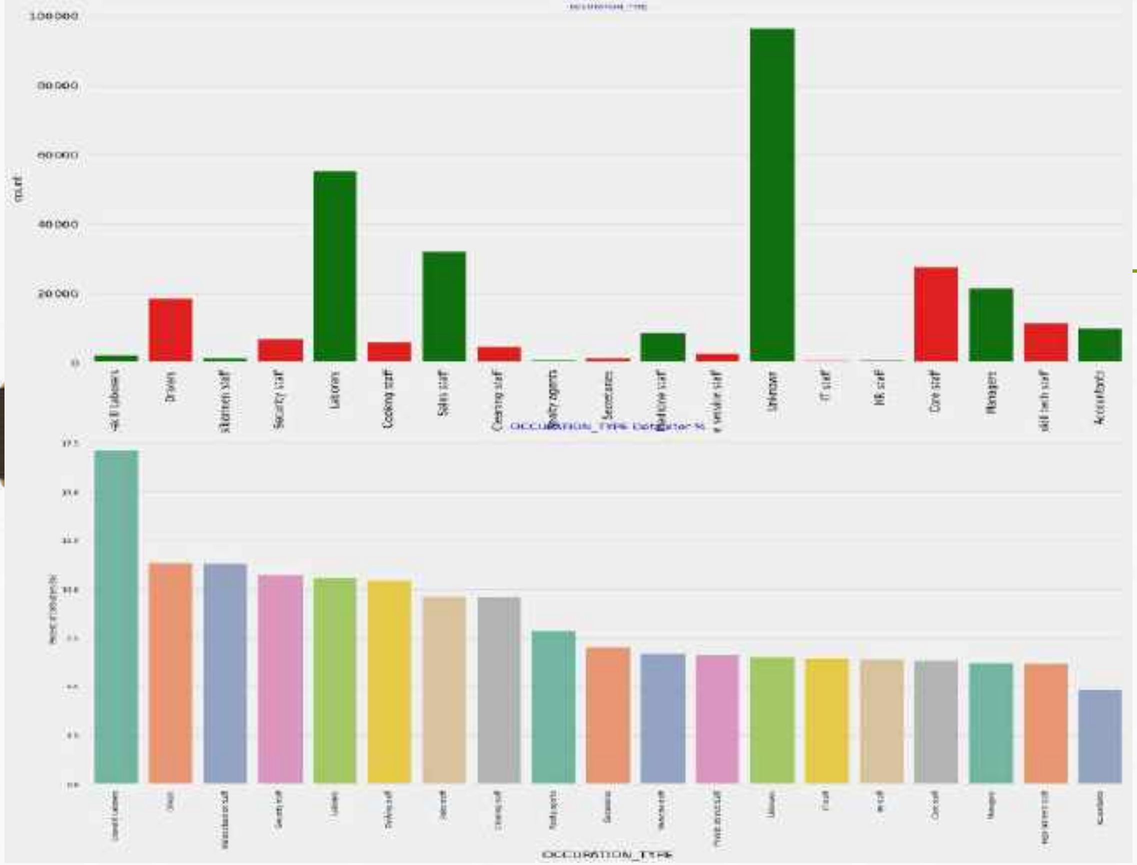
univariate_categorical("NAME_INCOME_TYPE",True,True,False)

- **Inferences:**Most of applicants for loans have income type as Working, followed by Commercial associate, Pensioner and State servant.
- The applicants with the type of income Maternity leave have almost 40% ratio of not returning loans, followed by Unemployed (37%). The rest of types of incomes are under the average of 10% for not returning loans.
- Student and Businessmen, though less in numbers do not have any default record. Thus these two category are **safest** for providing loan.



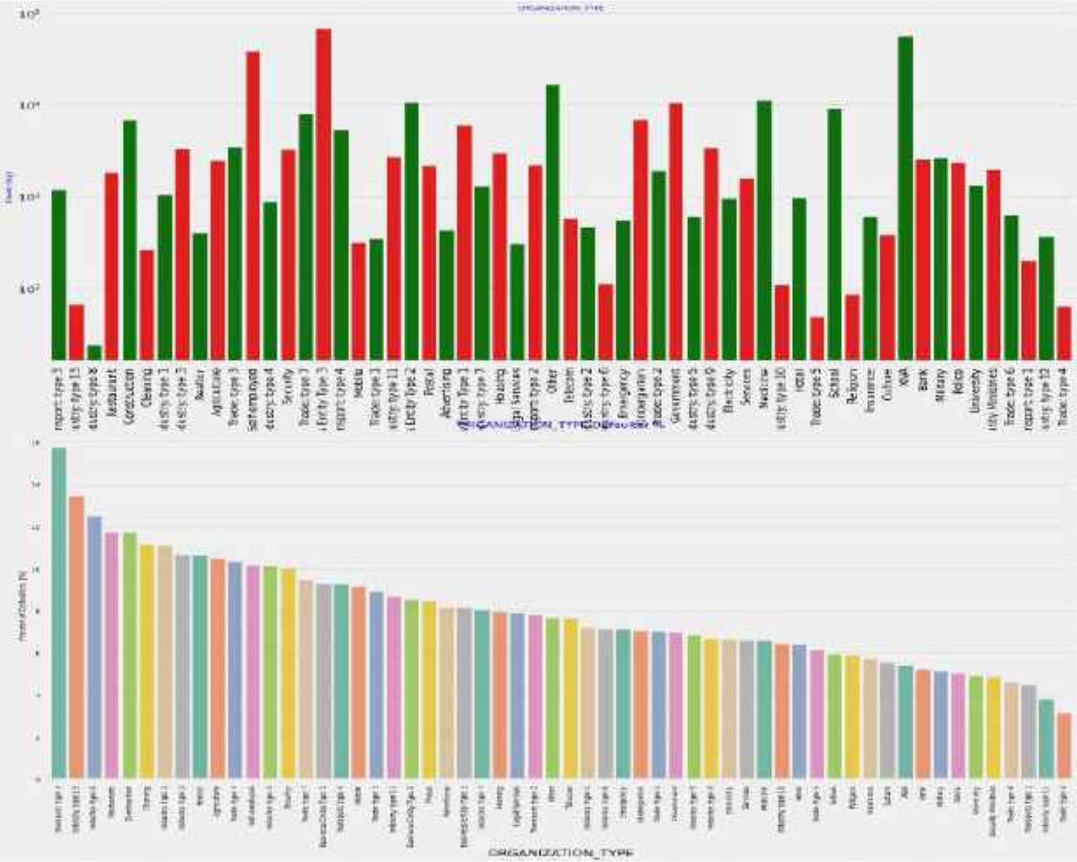
# Analyzing Region rating where applicant lives based on loan repayment status
univariate_categorical("REGION_RATING_CLIENT",False,False,True)

- **Inferences:**Most of the applicants are living in Region_Rating 2 place.
- Region Rating 3 has the highest default rate (11%)
- Applicant living in Region_Rating 1 has the lowest probability of defaulting, thus **safer** for approving loans

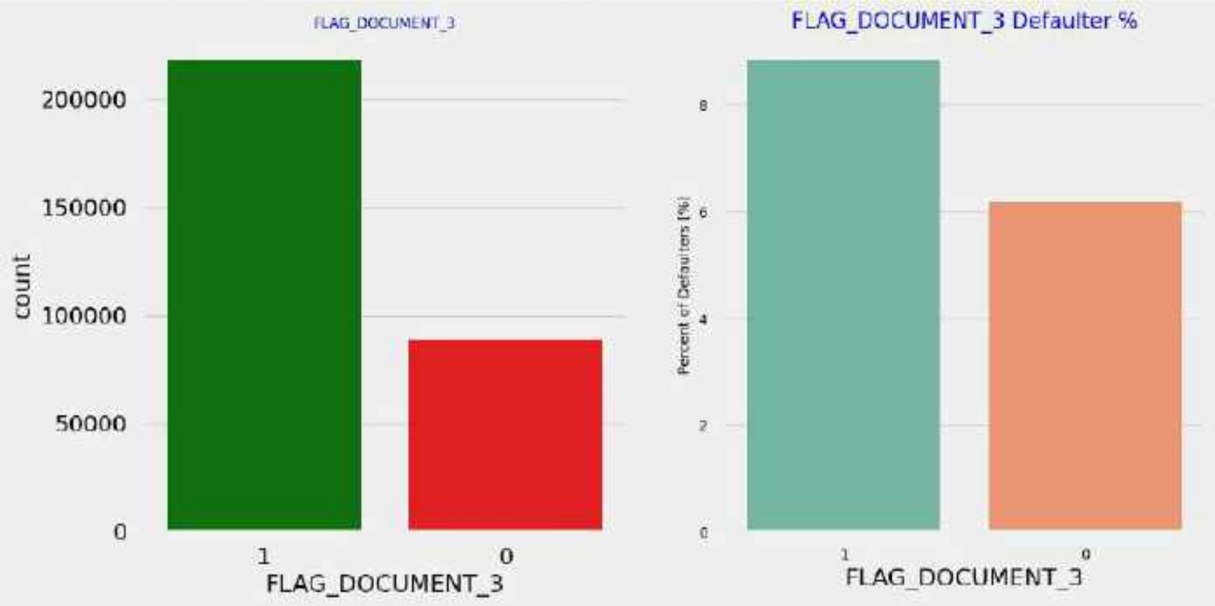


# Analyzing Occupation Type where applicant lives based on loan repayment status univariate_categorical("OCCUPATION_TYPE",False,True,False)

- **Inferences:**Most of the loans are taken by Laborers, followed by Sales staff. IT staff take the lowest amount of loans.
- The category with highest percent of not repaid loans are Low-skill Laborers (above 17%), followed by Drivers and Waiters/barmen staff, Security staff, Laborers and Cooking staff.

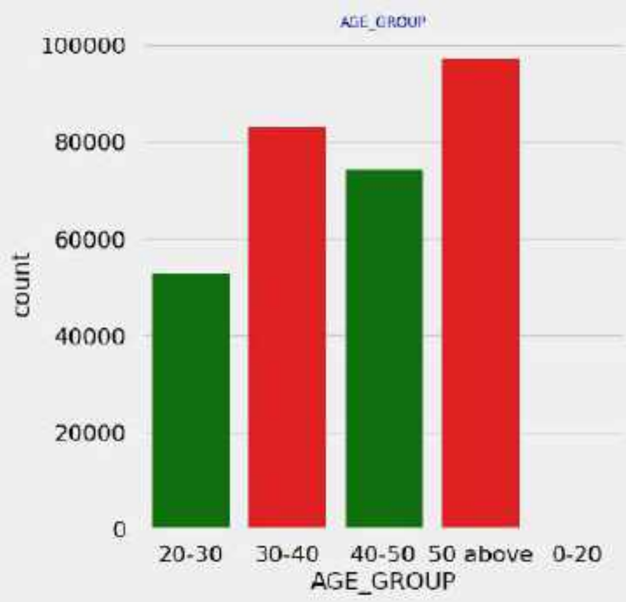


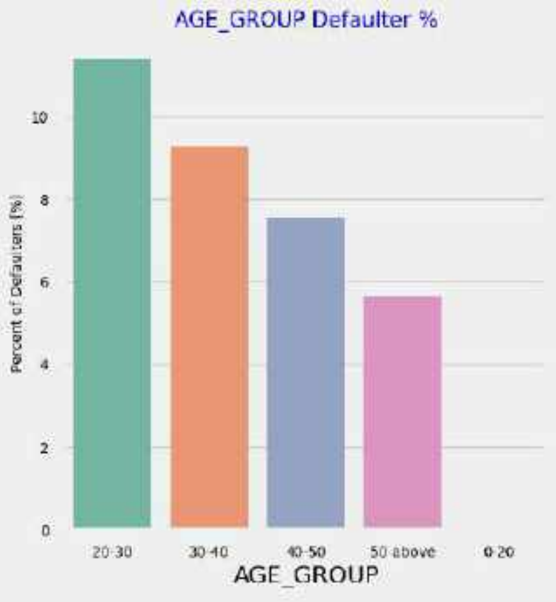
- **Inferences:**Organizations with highest percent of loans not repaid are Transport: type 3 (16%), Industry: type 13 (13.5%), Industry: type 8 (12.5%) and Restaurant (less than 12%). Self employed people have relative high defaulting rate, and thus should be avoided to be approved for loan or provide loan with higher interest rate to mitigate the risk of defaulting.
- Most of the people application for loan are from Business Entity Type 3
- For a very high number of applications, Organization type information is unavailable(XNA)
- It can be seen that following category of organization type has lesser defaulters thus safer for providing loans:Trade Type 4 and 5
- Industry type 8



```
# Analyzing Flag_Doc_3
submission status based on loan
repayment status
univariate_categorical("FLAG_D
OCUMENT_3",False,False,True)
```

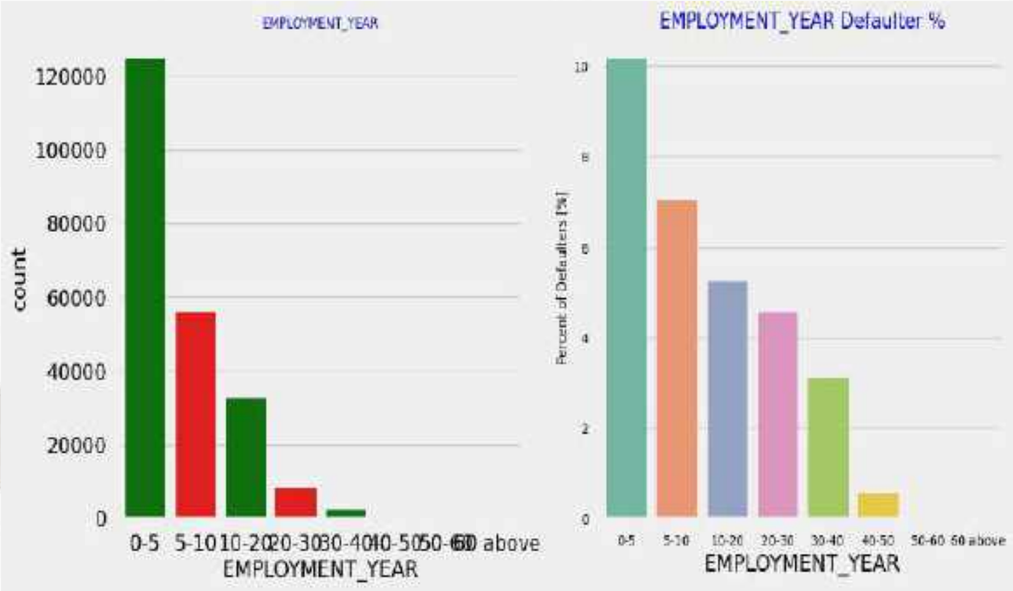
- **Inferences:**
  There is no significant correlation between repayers and defaulters in terms of submitting document 3 as we see even if applicants have submitted the document, they have defaulted a slightly more (~9%) than who have not submitted the document (6%)





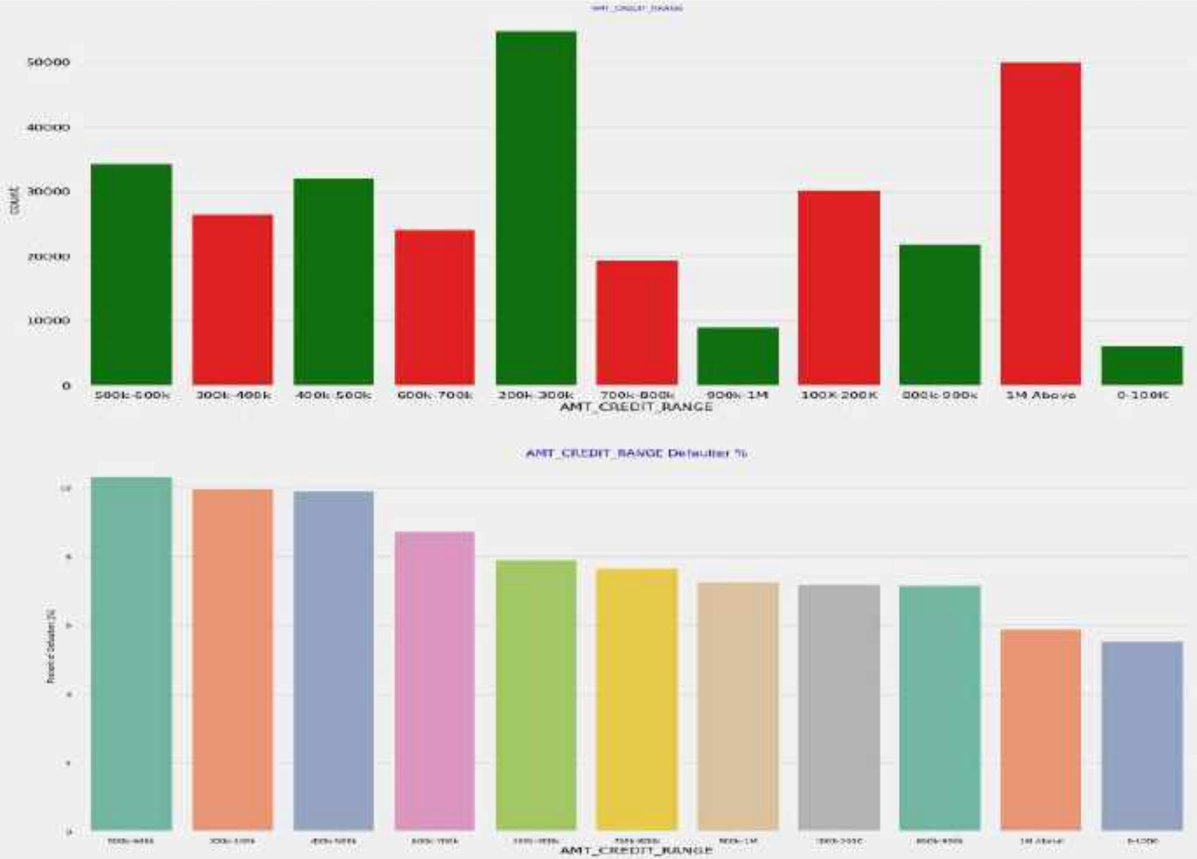
- **Inferences:**People in the age group range 20-40 have higher probability of defaulting
- People above age of 50 have low probability of defailting

```
# Analyzing Age Group based on loan repayment status
univariate_categorical("AGE_GROUP",False,False,True)
```



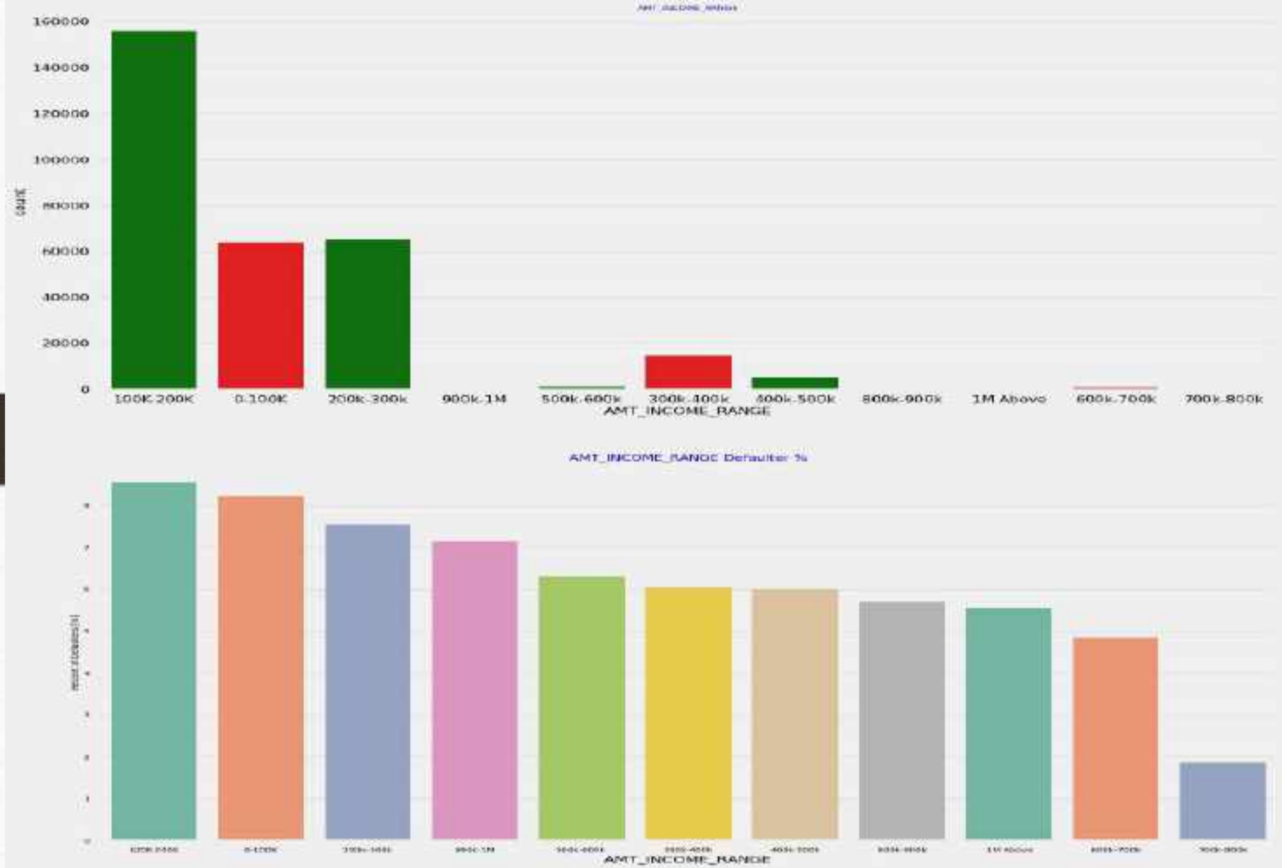
# Analyzing Employment_Year based on loan repayment status
univariate_categorical("EMPLOYMENT_YEAR",False,False,True)

- **Inferences:**Majority of the applicants have been employeed in between 0-5 years. The defaulting rating of this group is also the highest which is 10%
- With increase of employment year, defaulting rate is gradually decreasing with people having 40+ year experience having less than 1% default rate



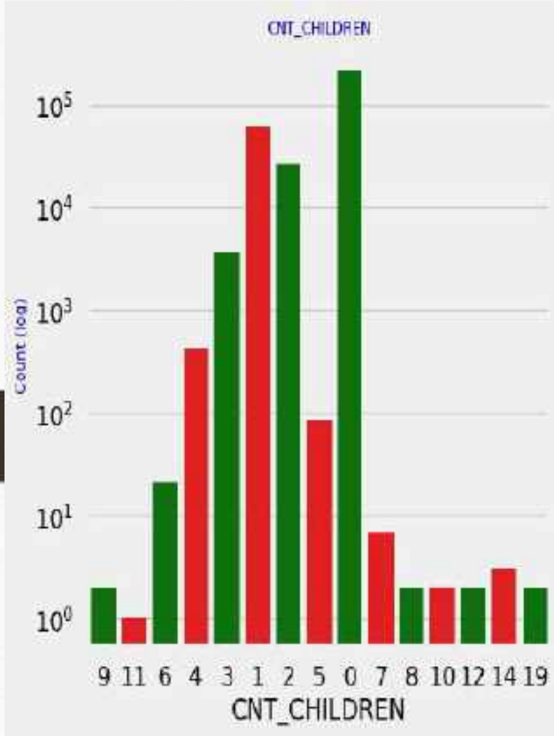
# Analyzing Amount_Credit based on loan repayment status univariate_categorical("AMT_CREDIT_RANGE",False,False,False)

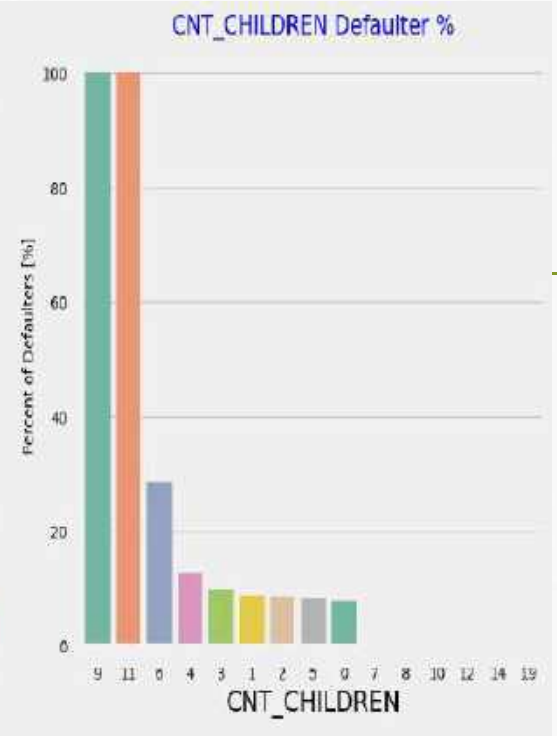
- **Inferences:**More than 80% of the loan provided are for amount less than 900,000
- People who get loan for 300-600k tend to default more than others.



# Analyzing Amount_Income Range based on loan repayment status univariate_categorical("AMT_INCOME_RANGE",False,False, False)

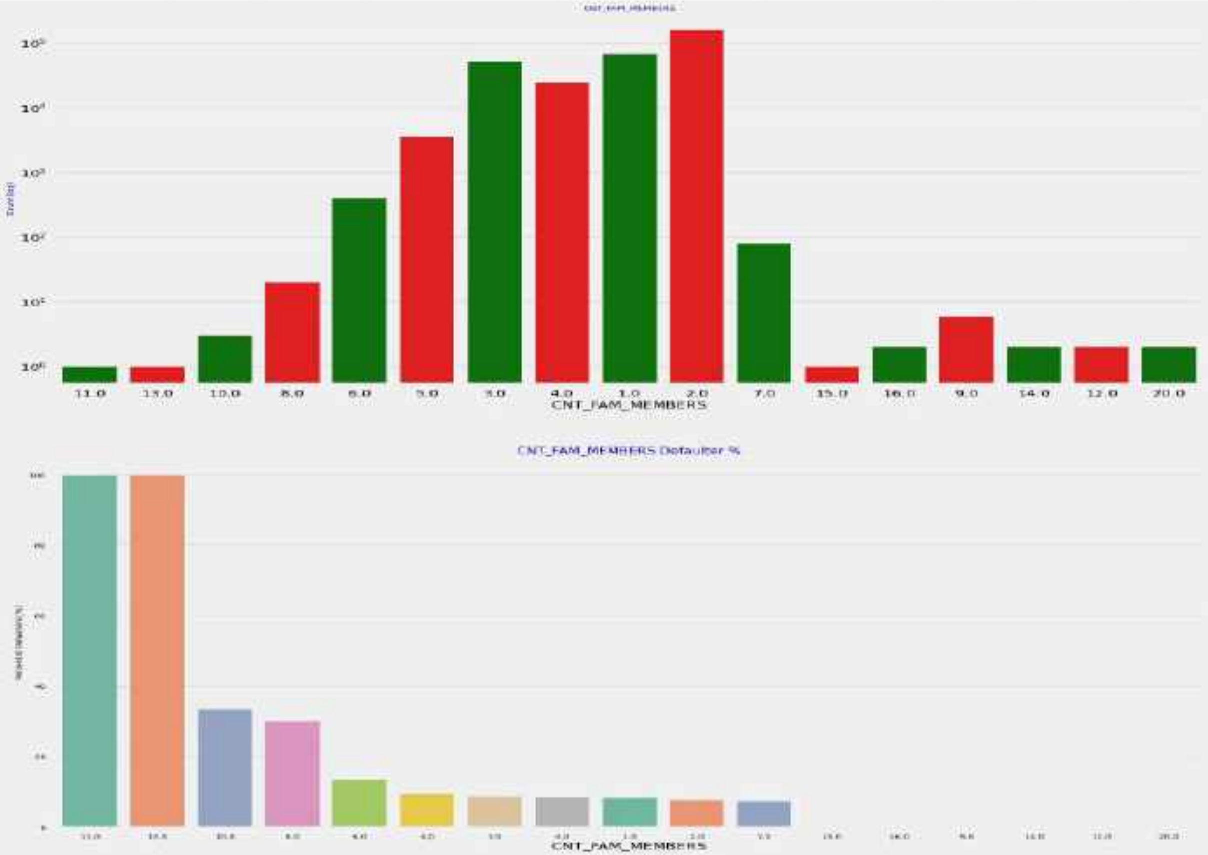
- **Inferences:**90% of the applications have Income total less than 300,000
- Application with Income less than 300,000 has high probability of defaulting
- Applicant with Income more than 700,000 are less likely to default





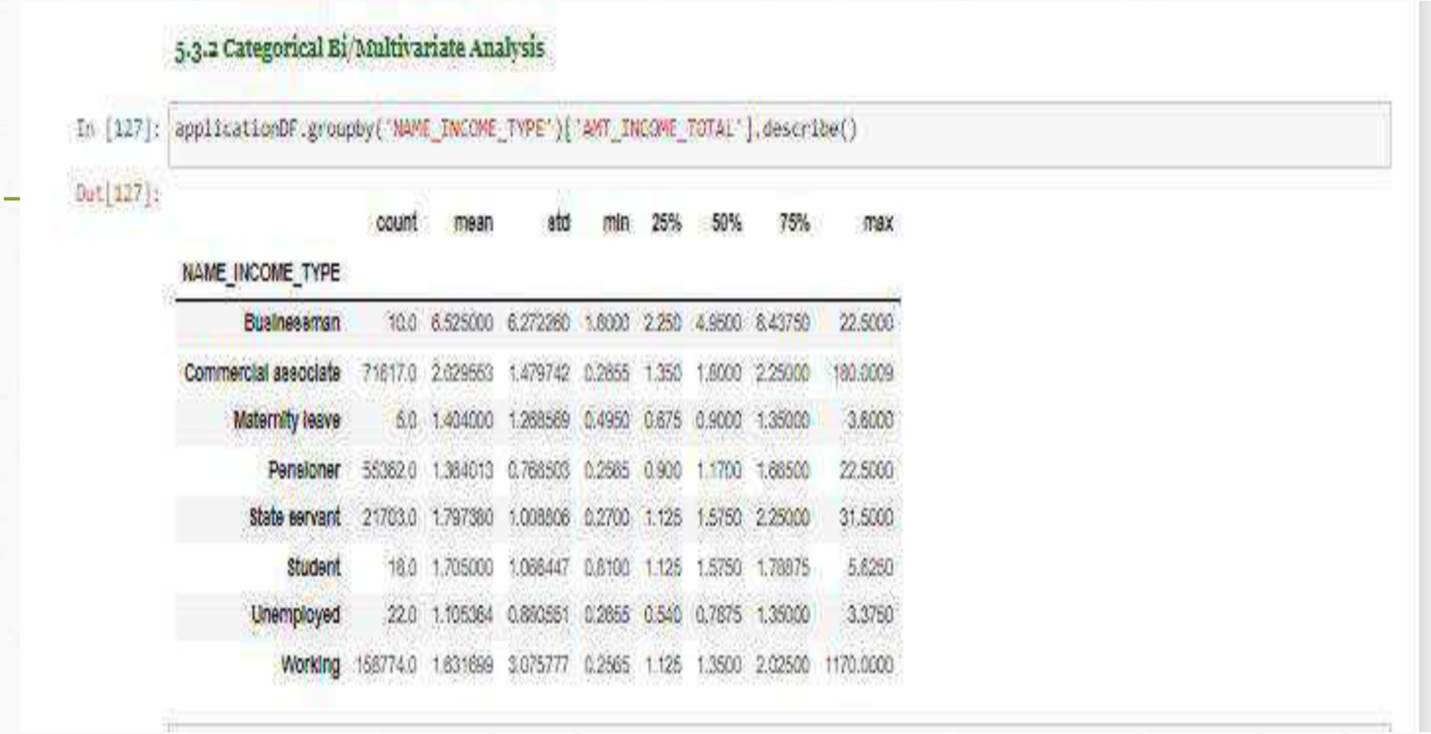
# Analyzing Number of children based on loan repayment status univariate_categorical("CNT_CHILDREN",True)

- **Inferences:**Most of the applicants do not have children
- Very few clients have more than 3 children.
- Client who have more than 4 children has a very high default rate with child count 9 and 11 showing 100% default rate



# Analyzing Number of family members based on loan repayment status univariate_categorical("CNT_FAM_MEMBERS",True, False, False)

- **Inferences:** Family member follows the same trend as children where having more family members increases the risk of defaulting



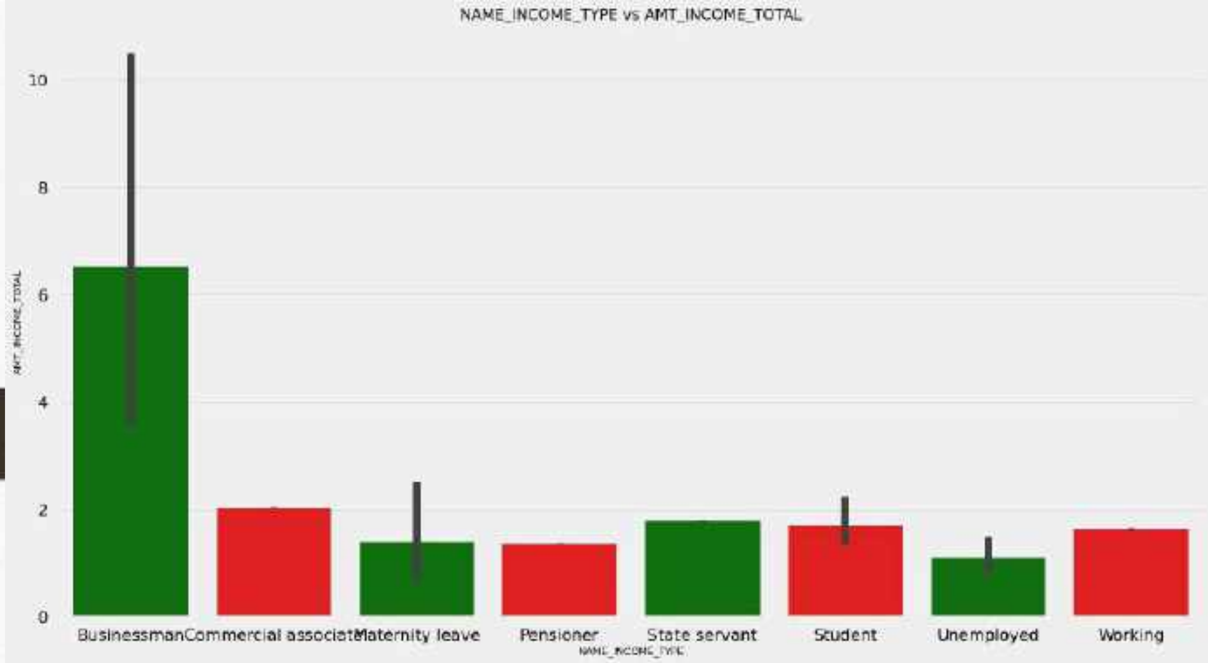
5.3.2 Categorical Bi/Multivariate Analysis

In [127]:

```
applicationDF.groupby('NAME_INCOME_TYPE')['AMT_INCOME_TOTAL'].describe()
```

Out[127]:

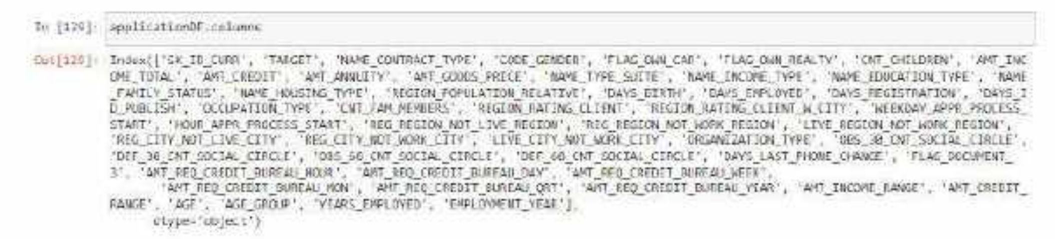
	count	mean	std	min	25%	50%	75%	max
NAME_INCOME_TYPE								
Businessman	10.0	6.525000	6.272260	1.8000	2.250	4.9500	8.43750	22.5000
Commercial associate	71617.0	2.029553	1.479742	0.2655	1.350	1.8000	2.25000	180.0009
Maternity leave	5.0	1.404000	1.268569	0.4950	0.675	0.9000	1.35000	3.6000
Pensioner	55362.0	1.364013	0.766503	0.2565	0.900	1.1700	1.66500	22.5000
State servant	21703.0	1.797380	1.008806	0.2700	1.125	1.5750	2.25000	31.5000
Student	18.0	1.705000	1.066447	0.8100	1.125	1.5750	1.78875	5.6250
Unemployed	22.0	1.105364	0.880551	0.2655	0.540	0.7875	1.35000	3.3750
Working	158774.0	1.631699	3.075777	0.2565	1.125	1.3500	2.02500	1170.0000



- **Inferences:**
  It can be seen that business man's income is the highest and the estimated range with default 95% confidence level seem to indicate that the income of a business man could be in the range of slightly close to 4 lakhs and slightly above 10 lakhs

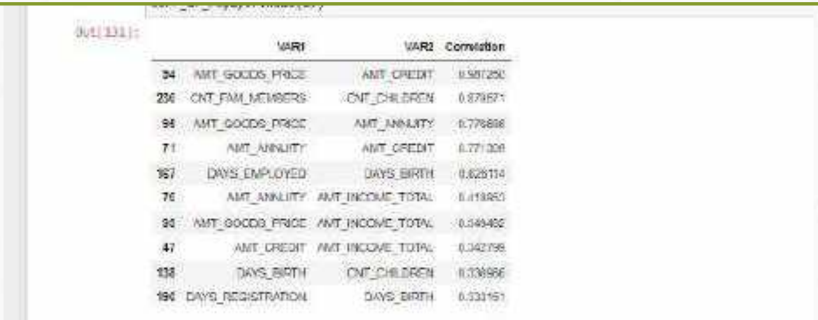
5.4 Numeric Variables Analysis

5.4.1 Bifurcating the applicationDF dataframe based on Target value 0 and 1 for correlation and other analysis



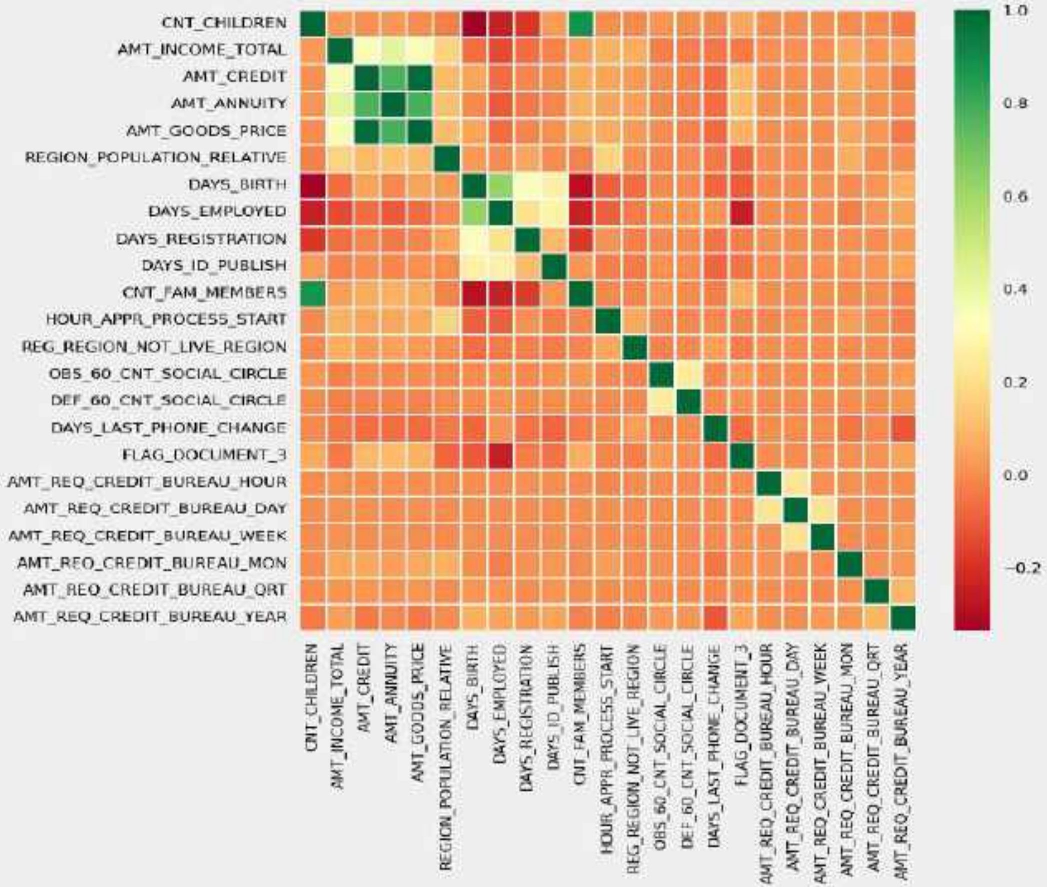
```
In [120]: applicationDF.columns

Out[120]: Index(['SK_ID_CURR', 'TARGET', 'NAME_CONTRACT_TYPE', 'CODE_GENDER', 'FLAG_OWN_CAR', 'FLAG_OWN_REALTY', 'CNT_CHILDREN', 'AMT_INC
          OME_TOTAL', 'AMT_CREDIT', 'AMT_ANNUITY', 'AMT_GOODS_PRICE', 'NAME_TYPE_SUITE', 'NAME_INCOME_TYPE', 'NAME_EDUCATION_TYPE', 'NAME
          _FAMILY_STATUS', 'NAME_HOUSING_TYPE', 'REGION_POPULATION_RELATIVE', 'DAYS_BIRTH', 'DAYS_EMPLOYED', 'DAYS_REGISTRATION', 'DAYS_I
          D_PUBLISH', 'OCCUPATION_TYPE', 'CNT_FAM_MEMBERS', 'REGION_RATING_CLIENT', 'REGION_RATING_CLIENT_W_CITY', 'WEEKDAY_APPR_PROCESS_
          START', 'HOUR_APPR_PROCESS_START', 'REG_REGION_NOT_LIVE_REGION', 'REG_REGION_NOT_WORK_REGION', 'LIVE_REGION_NOT_WORK_REGION',
          'REG_CITY_NOT_LIVE_CITY', 'REG_CITY_NOT_WORK_CITY', 'LIVE_CITY_NOT_WORK_CITY', 'ORGANIZATION_TYPE', 'OBS_30_CNT_SOCIAL_CIRCLE',
          'DEF_30_CNT_SOCIAL_CIRCLE', 'OBS_60_CNT_SOCIAL_CIRCLE', 'DEF_60_CNT_SOCIAL_CIRCLE', 'DAYS_LAST_PHONE_CHANGE', 'FLAG_DOCUMENT_
          3', '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', 'AMT_INCOME_RANGE', 'AMT_CREDIT_
          RANGE', 'AGE', 'AGE_GROUP', 'YEARS_EMPLOYED', 'EMPLOYMENT_YEAR'],
                dtype='object')
```

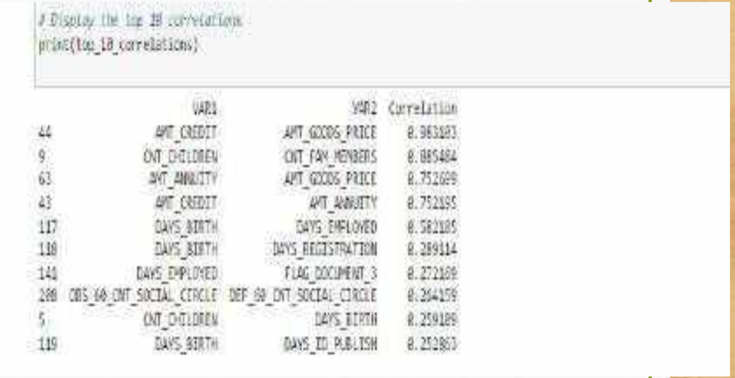
5.4.2 Correlation between numeric variable



Out[131]:

	VAR1	VAR2	Correlation
94	AMT_GOODS_PRICE	AMT_CREDIT	0.987250
236	CNT_FAM_MEMBERS	CNT_CHILDREN	0.879571
98	AMT_GOODS_PRICE	AMT_ANNUITY	0.776686
71	AMT_ANNUITY	AMT_CREDIT	0.771309
167	DAYS_EMPLOYED	DAYS_BIRTH	0.626114
76	AMT_ANNUITY	AMT_INCOME_TOTAL	0.418953
96	AMT_GOODS_PRICE	AMT_INCOME_TOTAL	0.349462
47	AMT_CREDIT	AMT_INCOME_TOTAL	0.342799
138	DAYS_BIRTH	CNT_CHILDREN	0.336966
190	DAYS_REGISTRATION	DAYS_BIRTH	0.331151





```
# Display the top 10 correlations
print(top_10_correlations)
```

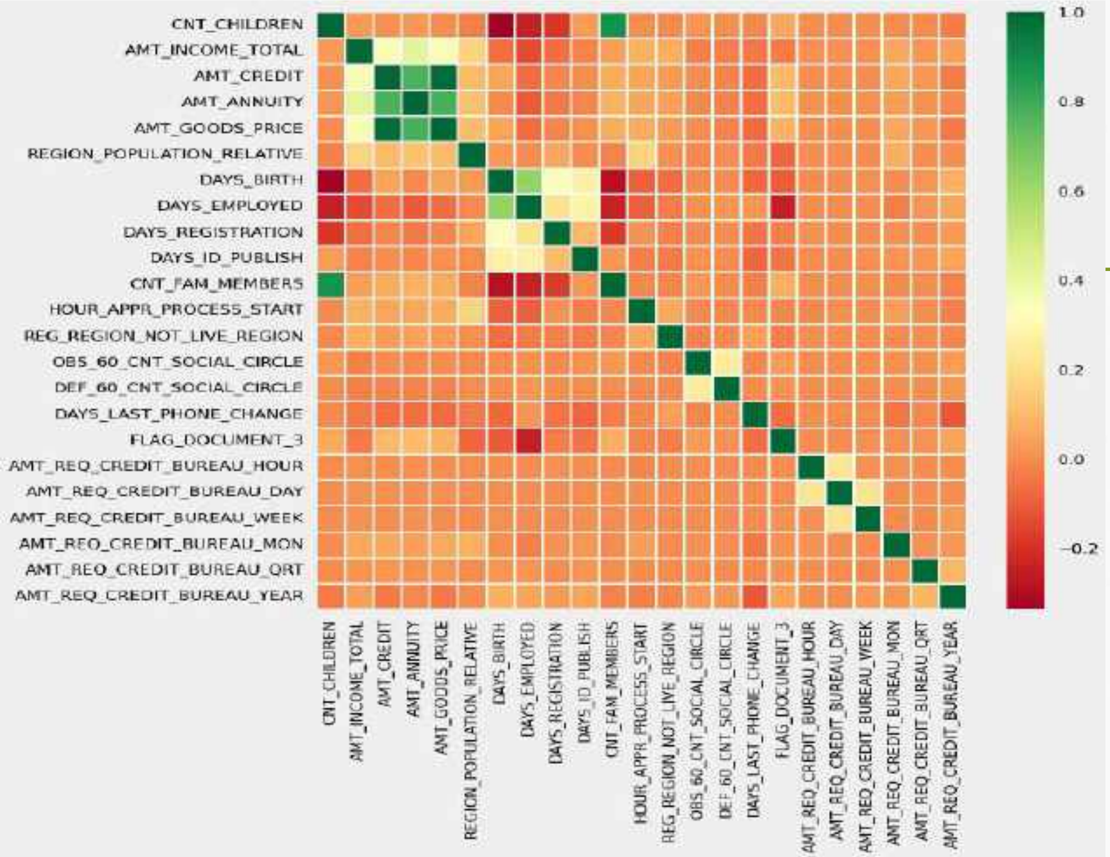
```
                          VAR1                      VAR2  Correlation
44                  AMT_CREDIT           AMT_GOODS_PRICE     0.983183
9                 CNT_CHILDREN           CNT_FAM_MEMBERS     0.885484
63                 AMT_ANNUITY           AMT_GOODS_PRICE     0.752699
43                  AMT_CREDIT               AMT_ANNUITY     0.752195
117                 DAYS_BIRTH             DAYS_EMPLOYED     0.582185
118                 DAYS_BIRTH         DAYS_REGISTRATION     0.289114
141              DAYS_EMPLOYED           FLAG_DOCUMENT_3     0.272169
288   OBS_60_CNT_SOCIAL_CIRCLE  DEF_60_CNT_SOCIAL_CIRCLE     0.264159
5                 CNT_CHILDREN                DAYS_BIRTH     0.259109
119                 DAYS_BIRTH           DAYS_ID_PUBLISH     0.252863
```

Inferences:

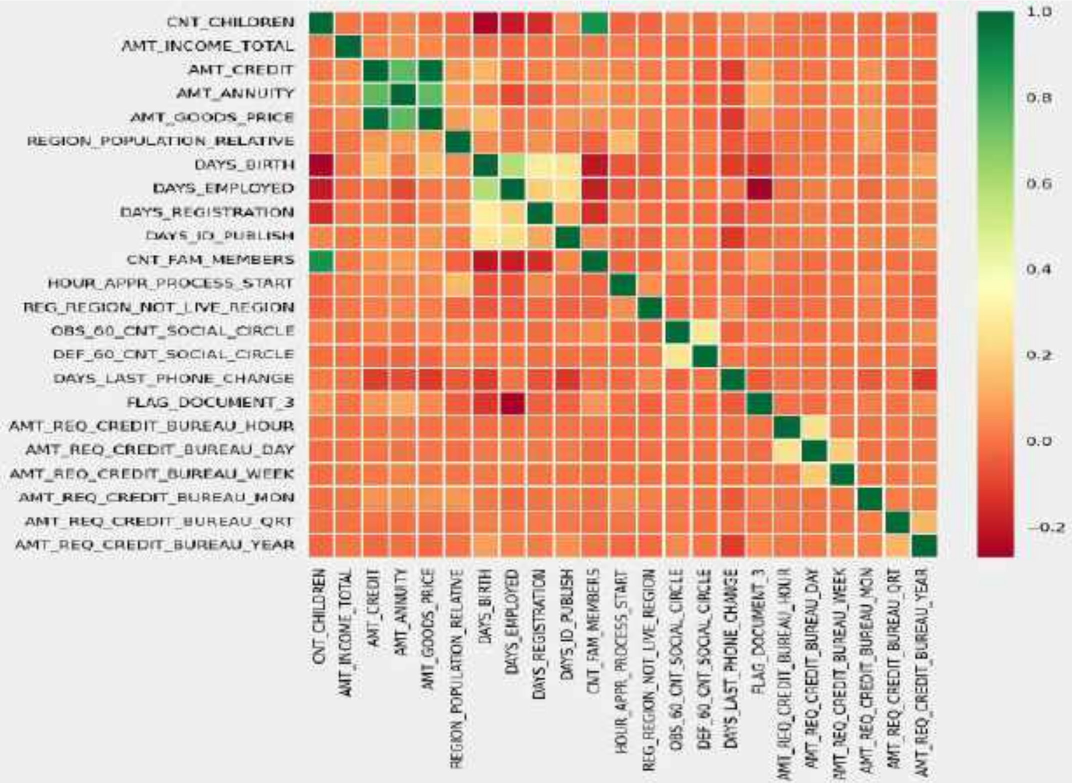
Correlating factors amongst repayers:
Credit amount is highly correlated with

- amount of goods price
- loan annuity
- total income

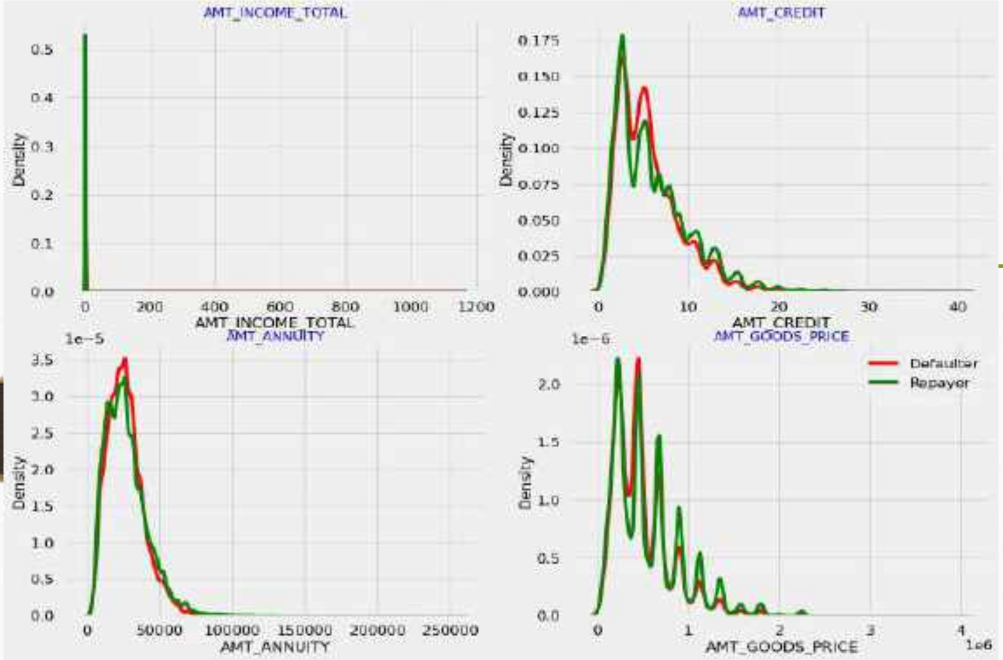
We can also see that repayers have high correlation in number of days employed.



- **Inferences:**
  Correlating factors amongst repayers: Credit amount is highly correlated withamount of goods price
- loan annuity
- total income
- We can also see that repayers have high correlation in number of days employed.

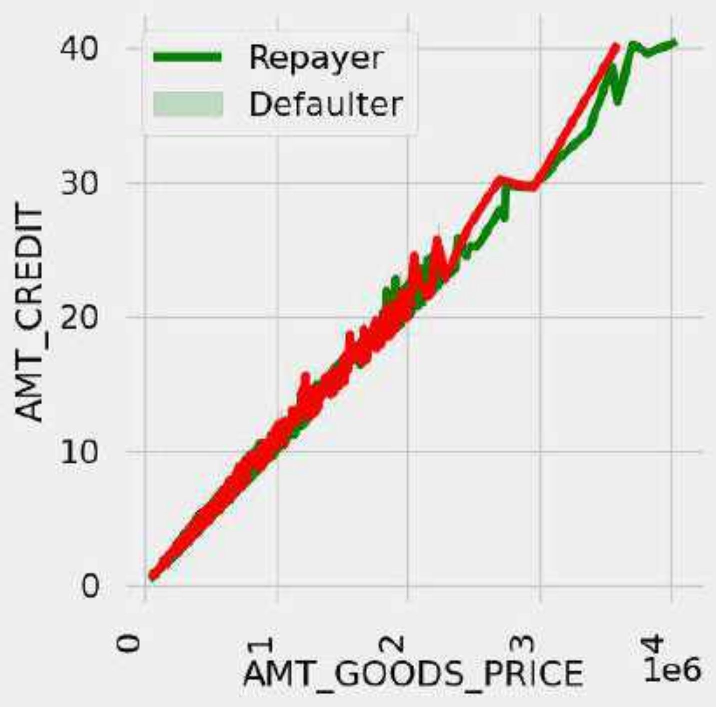


- **Inferences:**Credit amount is highly correlated with amount of goods price which is same as repayers.
- But the loan annuity correlation with credit amount has slightly reduced in defaulters(0.75) when compared to repayers(0.77)
- We can also see that repayers have high correlation in number of days employed(0.62) when compared to defaulters(0.58).
- There is a severe drop in the correlation between total income of the client and the credit amount(0.038) amongst defaulters whereas it is 0.342 among repayers.
- Days_birth and number of children correlation has reduced to 0.259 in defaulters when compared to 0.337 in repayers.
- There is a slight increase in defaulted to observed count in social circle among defaulters(0.264) when compared to repayers(0.254)



5.4.3 Numerical Univariate Analysis

- **Inferences:**Most no of loans are given for goods price below 10 lakhs
- Most people pay annuity below 50000 for the credit loan
- Credit amount of the loan is mostly less then 10 lakhs
- The repayers and defaulters distribution overlap in all the plots and hence we cannot use any of these variables in isolation to make a decision



5.4.4 Numerical Bivariate Analysis

Inferences: When amt_annuity >15000 amt_goods_price> 3M, there is a lesser chance of defaulters

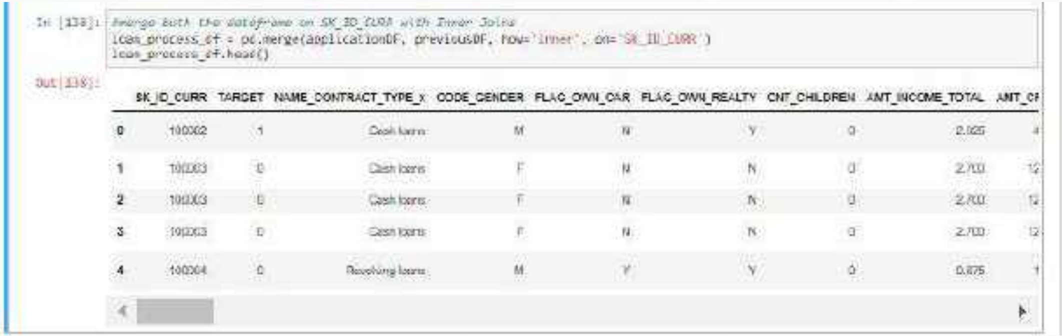
AMT_CREDIT and AMT_GOODS_PRICE are highly correlated as based on the scatterplot where most of the data are consolidated in form of a line

There are very less defaulters for AMT_CREDIT >3M

Inferences related to distribution plot has been already mentioned in previous distplot graphs inferences section

- **Inferences:**
  When the credit amount goes beyond 3M, there is an increase in defaulters.

6. Merged Dataframes Analysis

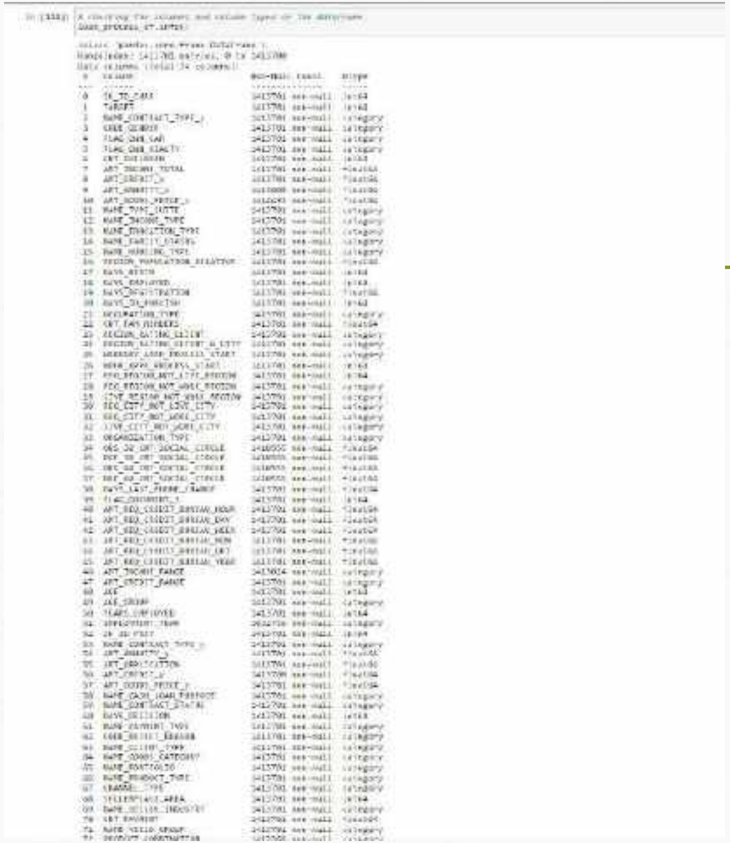


In [138]:

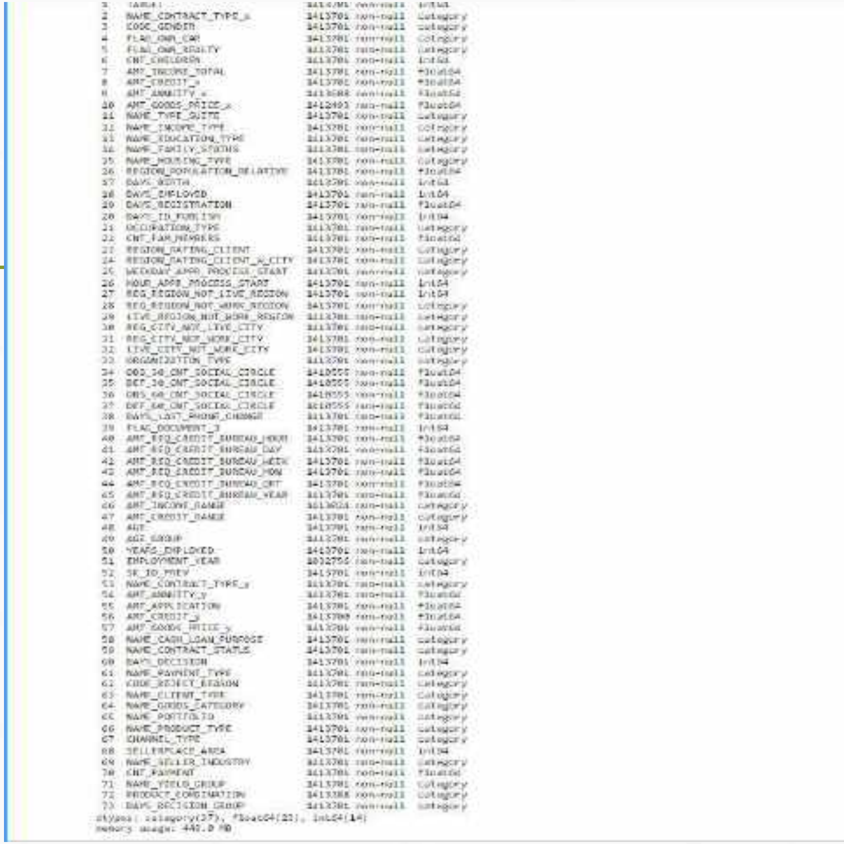
```
#merge both the dataframe on SK_ID_CURR with Inner Joins
loan_process_df = pd.merge(applicationDF, previousDF, how='inner', on='SK_ID_CURR')
loan_process_df.head()
```

Out[138]:

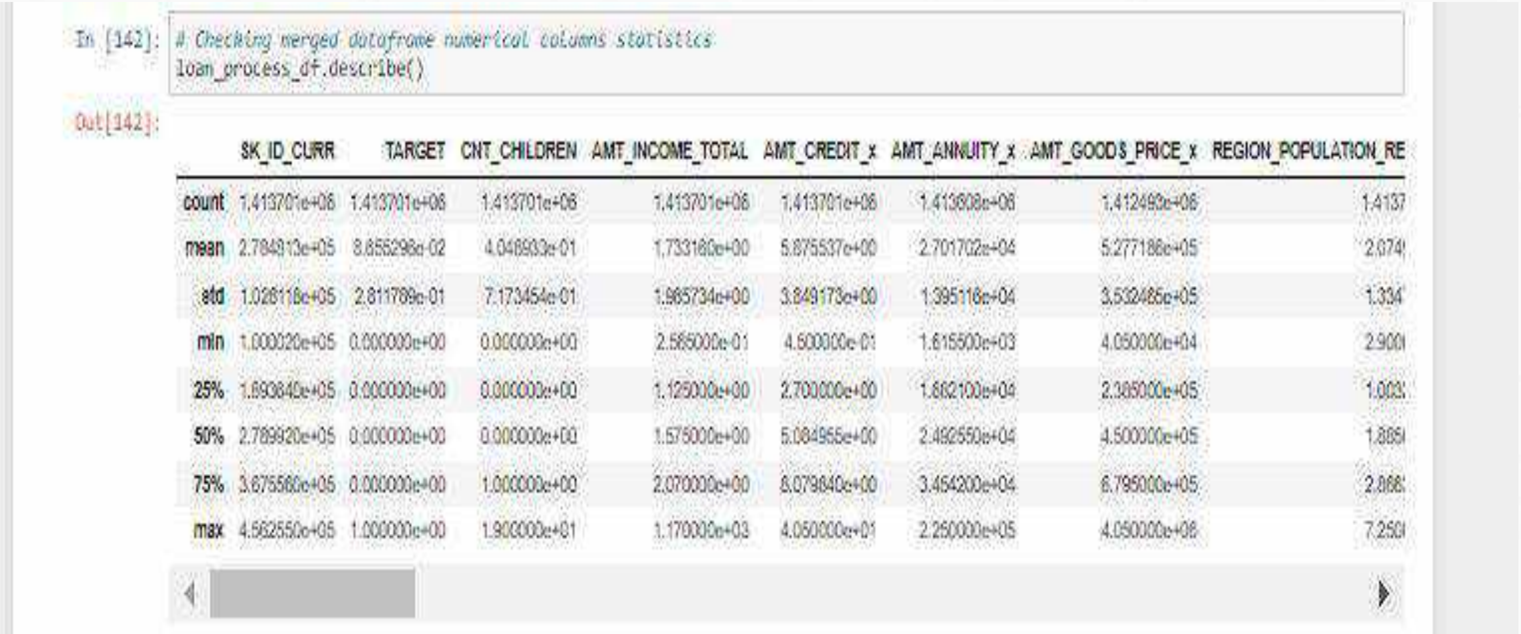
	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE_x	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CR
0	100002	1	Cash loans	M	N	Y	0	2.025	4
1	100003	0	Cash loans	F	N	N	0	2.700	12
2	100003	0	Cash loans	F	N	N	0	2.700	12
3	100003	0	Cash loans	F	N	N	0	2.700	12
4	100004	0	Revolving loans	M	Y	Y	0	0.675	1



```
In [111]: # checking for columns and column types in the dataframe
          loan_process_df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 1413701 entries, 0 to 1413700
Data columns (total 74 columns):
 #   Column                        Non-Null Count    Dtype
---  ------                        --------------    -----
 0   SK_ID_CURR                    1413701 non-null  int64
 1   TARGET                        1413701 non-null  int64
 2   NAME_CONTRACT_TYPE_x          1413701 non-null  category
 3   CODE_GENDER                   1413701 non-null  category
 4   FLAG_OWN_CAR                  1413701 non-null  category
 5   FLAG_OWN_REALTY               1413701 non-null  category
 6   CNT_CHILDREN                  1413701 non-null  int64
 7   AMT_INCOME_TOTAL              1413701 non-null  float64
 8   AMT_CREDIT_x                  1413701 non-null  float64
 9   AMT_ANNUITY_x                 1413608 non-null  float64
 10  AMT_GOODS_PRICE_x             1412493 non-null  float64
 11  NAME_TYPE_SUITE               1413701 non-null  category
 12  NAME_INCOME_TYPE              1413701 non-null  category
 13  NAME_EDUCATION_TYPE           1413701 non-null  category
 14  NAME_FAMILY_STATUS            1413701 non-null  category
 15  NAME_HOUSING_TYPE             1413701 non-null  category
 16  REGION_POPULATION_RELATIVE    1413701 non-null  float64
 17  DAYS_BIRTH                    1413701 non-null  int64
 18  DAYS_EMPLOYED                 1413701 non-null  int64
 19  DAYS_REGISTRATION             1413701 non-null  float64
 20  DAYS_ID_PUBLISH               1413701 non-null  int64
 21  OCCUPATION_TYPE               1413701 non-null  category
 22  CNT_FAM_MEMBERS               1413701 non-null  float64
 23  REGION_RATING_CLIENT          1413701 non-null  category
 24  REGION_RATING_CLIENT_W_CITY   1413701 non-null  category
 25  WEEKDAY_APPR_PROCESS_START    1413701 non-null  category
 26  HOUR_APPR_PROCESS_START       1413701 non-null  int64
 27  REG_REGION_NOT_LIVE_REGION    1413701 non-null  int64
 28  REG_REGION_NOT_WORK_REGION    1413701 non-null  category
 29  LIVE_REGION_NOT_WORK_REGION   1413701 non-null  category
 30  REG_CITY_NOT_LIVE_CITY        1413701 non-null  category
 31  REG_CITY_NOT_WORK_CITY        1413701 non-null  category
 32  LIVE_CITY_NOT_WORK_CITY       1413701 non-null  category
 33  ORGANIZATION_TYPE             1413701 non-null  category
 34  OBS_30_CNT_SOCIAL_CIRCLE      1410555 non-null  float64
 35  DEF_30_CNT_SOCIAL_CIRCLE      1410555 non-null  float64
 36  OBS_60_CNT_SOCIAL_CIRCLE      1410555 non-null  float64
 37  DEF_60_CNT_SOCIAL_CIRCLE      1410555 non-null  float64
 38  DAYS_LAST_PHONE_CHANGE        1413701 non-null  float64
 39  FLAG_DOCUMENT_3               1413701 non-null  int64
 40  AMT_REQ_CREDIT_BUREAU_HOUR    1413701 non-null  float64
 41  AMT_REQ_CREDIT_BUREAU_DAY     1413701 non-null  float64
 42  AMT_REQ_CREDIT_BUREAU_WEEK    1413701 non-null  float64
 43  AMT_REQ_CREDIT_BUREAU_MON     1413701 non-null  float64
 44  AMT_REQ_CREDIT_BUREAU_QRT     1413701 non-null  float64
 45  AMT_REQ_CREDIT_BUREAU_YEAR    1413701 non-null  float64
 46  AMT_INCOME_RANGE              1413024 non-null  category
 47  AMT_CREDIT_RANGE              1413701 non-null  category
 48  AGE                           1413701 non-null  int64
 49  AGE_GROUP                     1413701 non-null  category
 50  YEARS_EMPLOYED                1413701 non-null  int64
 51  EMPLOYMENT_YEAR               1032755 non-null  category
 52  SK_ID_PREV                    1413701 non-null  int64
 53  NAME_CONTRACT_TYPE_y          1413701 non-null  category
 54  AMT_ANNUITY_y                 1413701 non-null  float64
 55  AMT_APPLICATION               1413701 non-null  float64
 56  AMT_CREDIT_y                  1413700 non-null  float64
 57  AMT_GOODS_PRICE_y             1413701 non-null  float64
 58  NAME_CASH_LOAN_PURPOSE        1413701 non-null  category
 59  NAME_CONTRACT_STATUS          1413701 non-null  category
 60  DAYS_DECISION                 1413701 non-null  int64
 61  NAME_PAYMENT_TYPE             1413701 non-null  category
 62  CODE_REJECT_REASON            1413701 non-null  category
 63  NAME_CLIENT_TYPE              1413701 non-null  category
 64  NAME_GOODS_CATEGORY           1413701 non-null  category
 65  NAME_PORTFOLIO                1413701 non-null  category
 66  NAME_PRODUCT_TYPE             1413701 non-null  category
 67  CHANNEL_TYPE                  1413701 non-null  category
 68  SELLERPLACE_AREA              1413701 non-null  int64
 69  NAME_SELLER_INDUSTRY          1413701 non-null  category
 70  CNT_PAYMENT                   1413701 non-null  float64
 71  NAME_YIELD_GROUP              1413701 non-null  category
 72  PRODUCT_COMBINATION           1413388 non-null  category
```



```
 1   TARGET                        1413701 non-null  int64
 2   NAME_CONTRACT_TYPE_x          1413701 non-null  category
 3   CODE_GENDER                   1413701 non-null  category
 4   FLAG_OWN_CAR                  1413701 non-null  category
 5   FLAG_OWN_REALTY               1413701 non-null  category
 6   CNT_CHILDREN                  1413701 non-null  int64
 7   AMT_INCOME_TOTAL              1413701 non-null  float64
 8   AMT_CREDIT_x                  1413701 non-null  float64
 9   AMT_ANNUITY_x                 1413608 non-null  float64
 10  AMT_GOODS_PRICE_x             1412493 non-null  float64
 11  NAME_TYPE_SUITE               1413701 non-null  category
 12  NAME_INCOME_TYPE              1413701 non-null  category
 13  NAME_EDUCATION_TYPE           1413701 non-null  category
 14  NAME_FAMILY_STATUS            1413701 non-null  category
 15  NAME_HOUSING_TYPE             1413701 non-null  category
 16  REGION_POPULATION_RELATIVE    1413701 non-null  float64
 17  DAYS_BIRTH                    1413701 non-null  int64
 18  DAYS_EMPLOYED                 1413701 non-null  int64
 19  DAYS_REGISTRATION             1413701 non-null  float64
 20  DAYS_ID_PUBLISH               1413701 non-null  int64
 21  OCCUPATION_TYPE               1413701 non-null  category
 22  CNT_FAM_MEMBERS               1413701 non-null  float64
 23  REGION_RATING_CLIENT          1413701 non-null  category
 24  REGION_RATING_CLIENT_W_CITY   1413701 non-null  category
 25  WEEKDAY_APPR_PROCESS_START    1413701 non-null  category
 26  HOUR_APPR_PROCESS_START       1413701 non-null  int64
 27  REG_REGION_NOT_LIVE_REGION    1413701 non-null  int64
 28  REG_REGION_NOT_WORK_REGION    1413701 non-null  category
 29  LIVE_REGION_NOT_WORK_REGION   1413701 non-null  category
 30  REG_CITY_NOT_LIVE_CITY        1413701 non-null  category
 31  REG_CITY_NOT_WORK_CITY        1413701 non-null  category
 32  LIVE_CITY_NOT_WORK_CITY       1413701 non-null  category
 33  ORGANIZATION_TYPE             1413701 non-null  category
 34  OBS_30_CNT_SOCIAL_CIRCLE      1410555 non-null  float64
 35  DEF_30_CNT_SOCIAL_CIRCLE      1410555 non-null  float64
 36  OBS_60_CNT_SOCIAL_CIRCLE      1410555 non-null  float64
 37  DEF_60_CNT_SOCIAL_CIRCLE      1410555 non-null  float64
 38  DAYS_LAST_PHONE_CHANGE        1413701 non-null  float64
 39  FLAG_DOCUMENT_3               1413701 non-null  int64
 40  AMT_REQ_CREDIT_BUREAU_HOUR    1413701 non-null  float64
 41  AMT_REQ_CREDIT_BUREAU_DAY     1413701 non-null  float64
 42  AMT_REQ_CREDIT_BUREAU_WEEK    1413701 non-null  float64
 43  AMT_REQ_CREDIT_BUREAU_MON     1413701 non-null  float64
 44  AMT_REQ_CREDIT_BUREAU_QRT     1413701 non-null  float64
 45  AMT_REQ_CREDIT_BUREAU_YEAR    1413701 non-null  float64
 46  AMT_INCOME_RANGE              1413024 non-null  category
 47  AMT_CREDIT_RANGE              1413701 non-null  category
 48  AGE                           1413701 non-null  int64
 49  AGE_GROUP                     1413701 non-null  category
 50  YEARS_EMPLOYED                1413701 non-null  int64
 51  EMPLOYMENT_YEAR               1032755 non-null  category
 52  SK_ID_PREV                    1413701 non-null  int64
 53  NAME_CONTRACT_TYPE_y          1413701 non-null  category
 54  AMT_ANNUITY_y                 1413701 non-null  float64
 55  AMT_APPLICATION               1413701 non-null  float64
 56  AMT_CREDIT_y                  1413700 non-null  float64
 57  AMT_GOODS_PRICE_y             1413701 non-null  float64
 58  NAME_CASH_LOAN_PURPOSE        1413701 non-null  category
 59  NAME_CONTRACT_STATUS          1413701 non-null  category
 60  DAYS_DECISION                 1413701 non-null  int64
 61  NAME_PAYMENT_TYPE             1413701 non-null  category
 62  CODE_REJECT_REASON            1413701 non-null  category
 63  NAME_CLIENT_TYPE              1413701 non-null  category
 64  NAME_GOODS_CATEGORY           1413701 non-null  category
 65  NAME_PORTFOLIO                1413701 non-null  category
 66  NAME_PRODUCT_TYPE             1413701 non-null  category
 67  CHANNEL_TYPE                  1413701 non-null  category
 68  SELLERPLACE_AREA              1413701 non-null  int64
 69  NAME_SELLER_INDUSTRY          1413701 non-null  category
 70  CNT_PAYMENT                   1413701 non-null  float64
 71  NAME_YIELD_GROUP              1413701 non-null  category
 72  PRODUCT_COMBINATION           1413388 non-null  category
 73  DAYS_DECISION_GROUP           1413701 non-null  category
dtypes: category(37), float64(23), int64(14)
memory usage: 440.0 MB
```

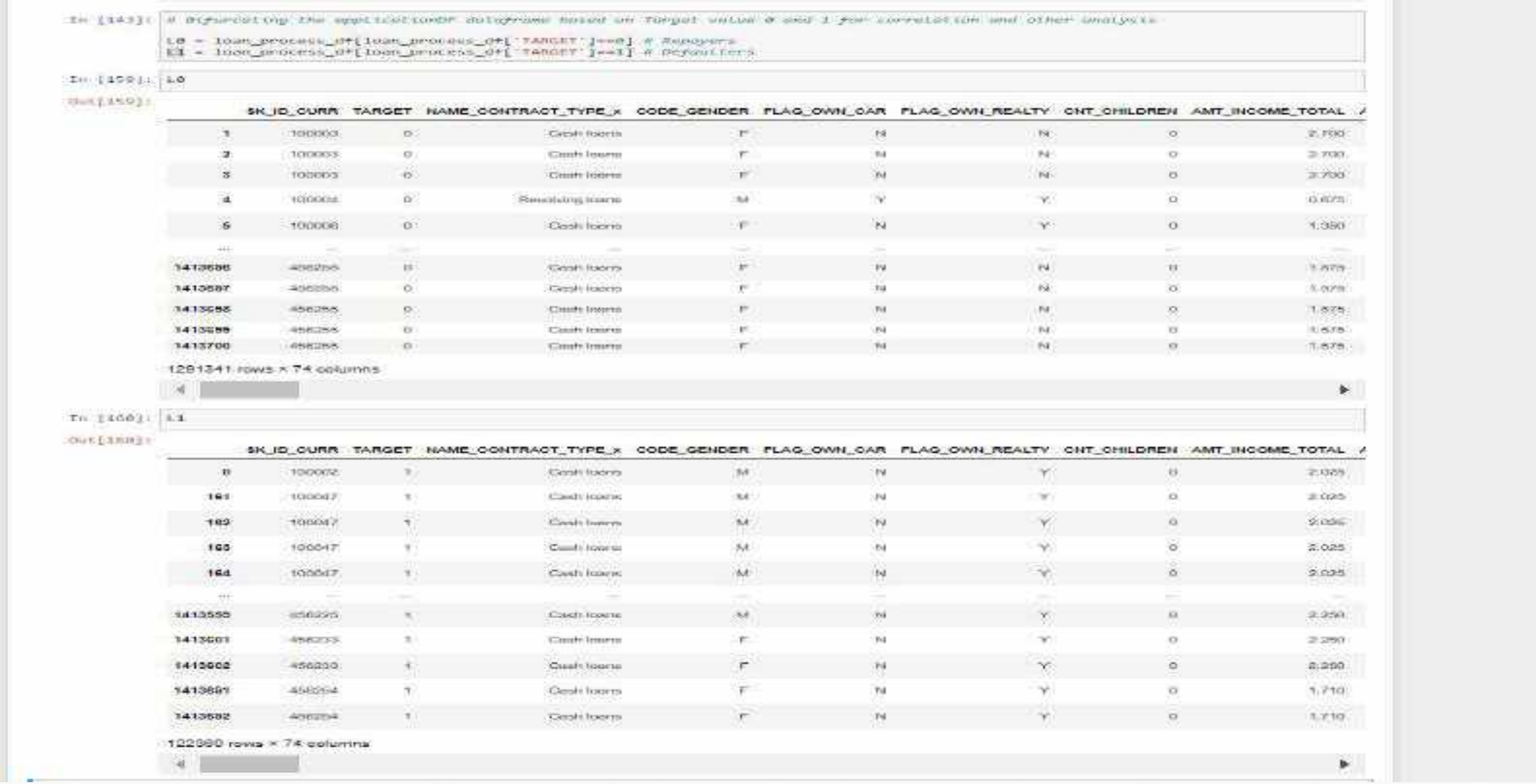


In [142]:

```
# Checking merged dataframe numerical columns statistics
loan_process_df.describe()
```

Out[142]:

	SK_ID_CURR	TARGET	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT_x	AMT_ANNUITY_x	AMT_GOODS_PRICE_x	REGION_POPULATION_RE
count	1.413701e+06	1.413701e+06	1.413701e+06	1.413701e+06	1.413701e+06	1.413608e+06	1.412493e+06	1.4137
mean	2.784913e+05	8.855296e-02	4.048933e-01	1.733160e+00	5.875537e+00	2.701702e+04	5.277186e+05	2.074
std	1.028118e+05	2.811789e-01	7.173454e-01	1.985734e+00	3.849173e+00	1.395116e+04	3.532485e+05	1.334
min	1.000020e+05	0.000000e+00	0.000000e+00	2.565000e-01	4.500000e-01	1.615500e+03	4.050000e+04	2.900
25%	1.893840e+05	0.000000e+00	0.000000e+00	1.125000e+00	2.700000e+00	1.662100e+04	2.385000e+05	1.003
50%	2.789920e+05	0.000000e+00	0.000000e+00	1.575000e+00	5.084955e+00	2.492550e+04	4.500000e+05	1.885
75%	3.675560e+05	0.000000e+00	1.000000e+00	2.070000e+00	8.079840e+00	3.454200e+04	6.795000e+05	2.868
max	4.562550e+05	1.000000e+00	1.900000e+01	1.170000e+03	4.050000e+01	2.250000e+05	4.050000e+06	7.250



```
In [143]: # Bifurcating the applicationDF dataframe based on Target value 0 and 1 for correlation and other analysis

          L0 = loan_process_df[loan_process_df['TARGET']==0] # Repayers
          L1 = loan_process_df[loan_process_df['TARGET']==1] # Defaulters
```

```
In [159]: L0
```

Out[159]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE_x	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL
1	100003	0	Cash loans	F	N	N	0	2.700
2	100003	0	Cash loans	F	N	N	0	2.700
3	100003	0	Cash loans	F	N	N	0	2.700
4	100004	0	Revolving loans	M	Y	Y	0	0.675
5	100006	0	Cash loans	F	N	Y	0	1.350
...	...	...	...	...	...	...	...	...
1413696	456255	0	Cash loans	F	N	N	0	1.575
1413697	456255	0	Cash loans	F	N	N	0	1.575
1413698	456255	0	Cash loans	F	N	N	0	1.575
1413699	456255	0	Cash loans	F	N	N	0	1.575
1413700	456255	0	Cash loans	F	N	N	0	1.575

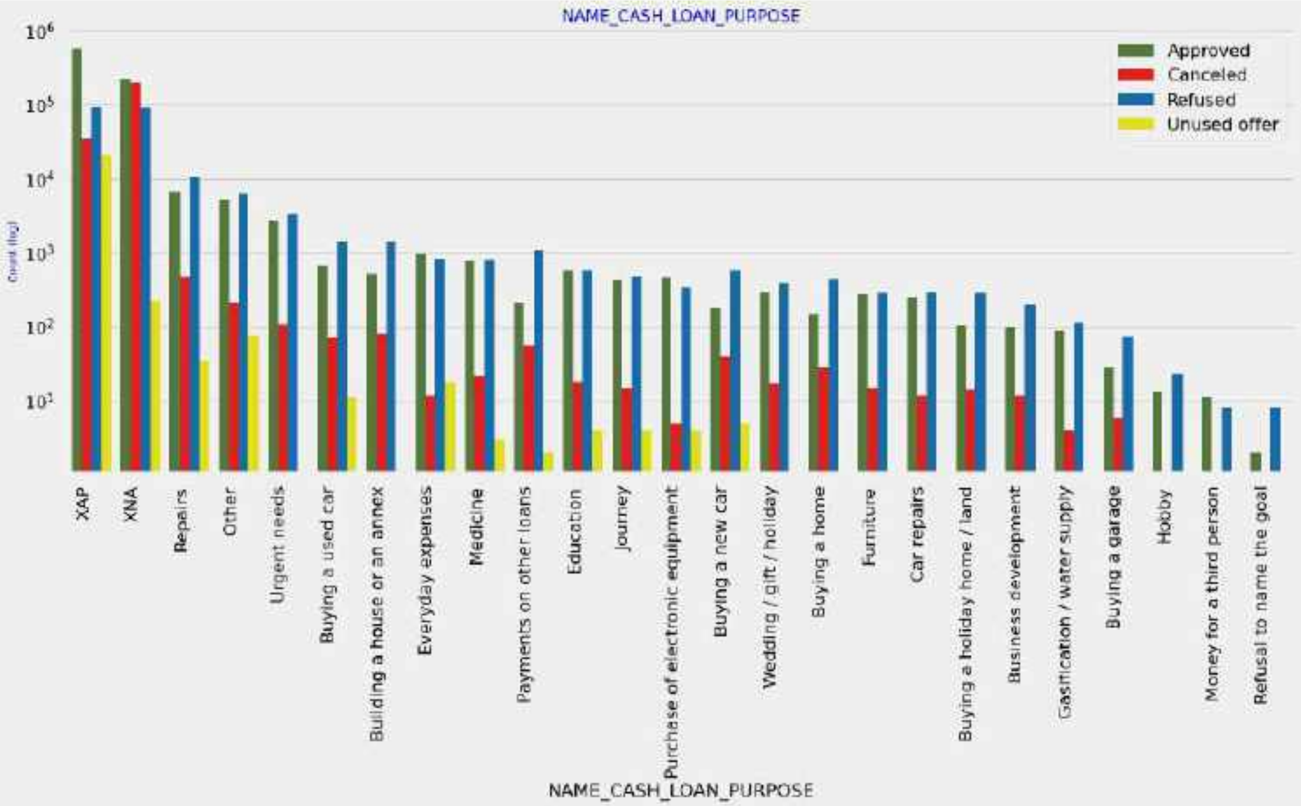
1291341 rows × 74 columns

```
In [160]: L1
```

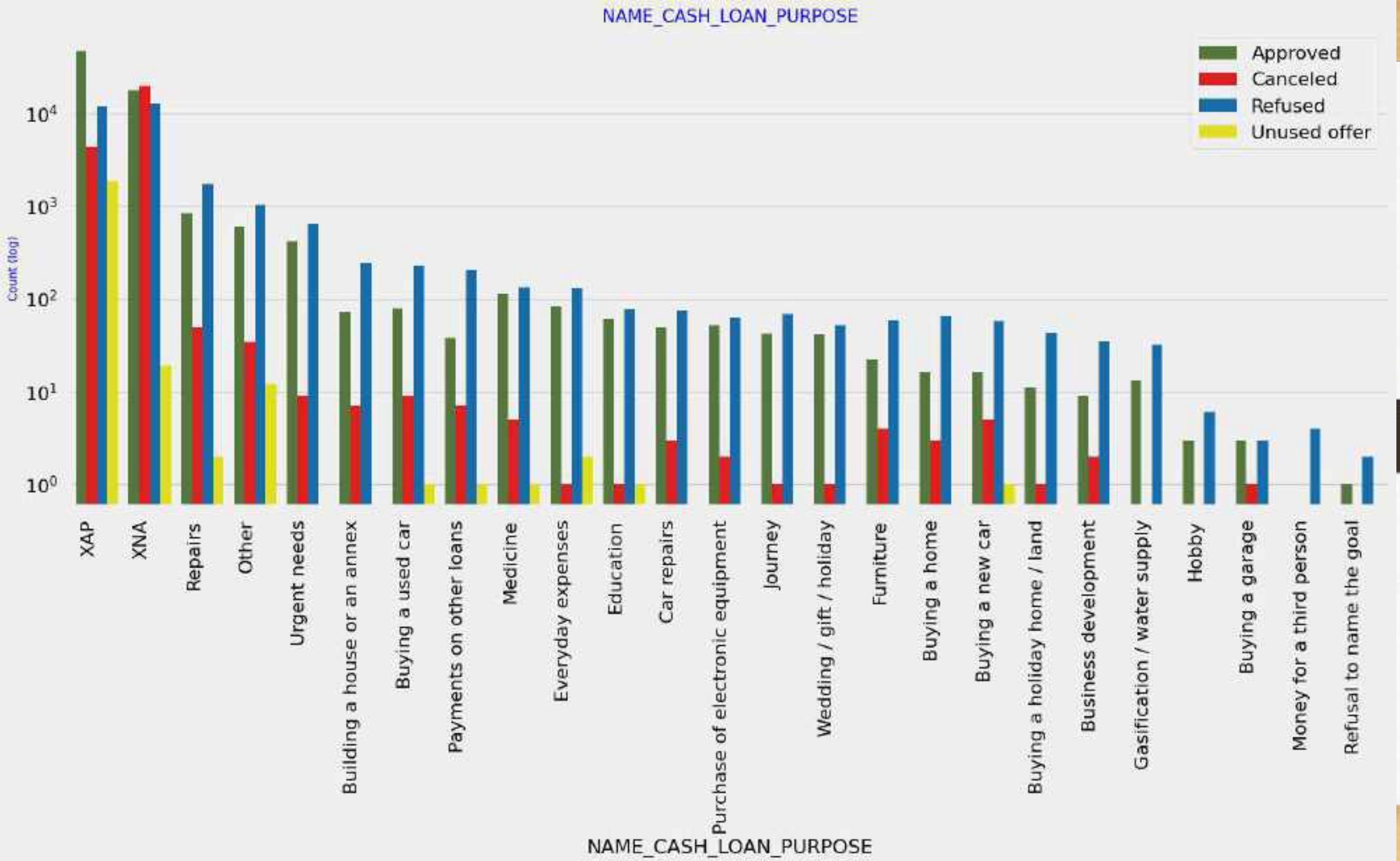
Out[160]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE_x	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL
0	100002	1	Cash loans	M	N	Y	0	2.025
161	100047	1	Cash loans	M	N	Y	0	2.025
162	100047	1	Cash loans	M	N	Y	0	2.025
163	100047	1	Cash loans	M	N	Y	0	2.025
164	100047	1	Cash loans	M	N	Y	0	2.025
...	...	...	...	...	...	...	...	...
1413555	456225	1	Cash loans	M	N	Y	0	2.250
1413601	456233	1	Cash loans	F	N	Y	0	2.250
1413602	456233	1	Cash loans	F	N	Y	0	2.250
1413691	456254	1	Cash loans	F	N	Y	0	1.710
1413692	456254	1	Cash loans	F	N	Y	0	1.710

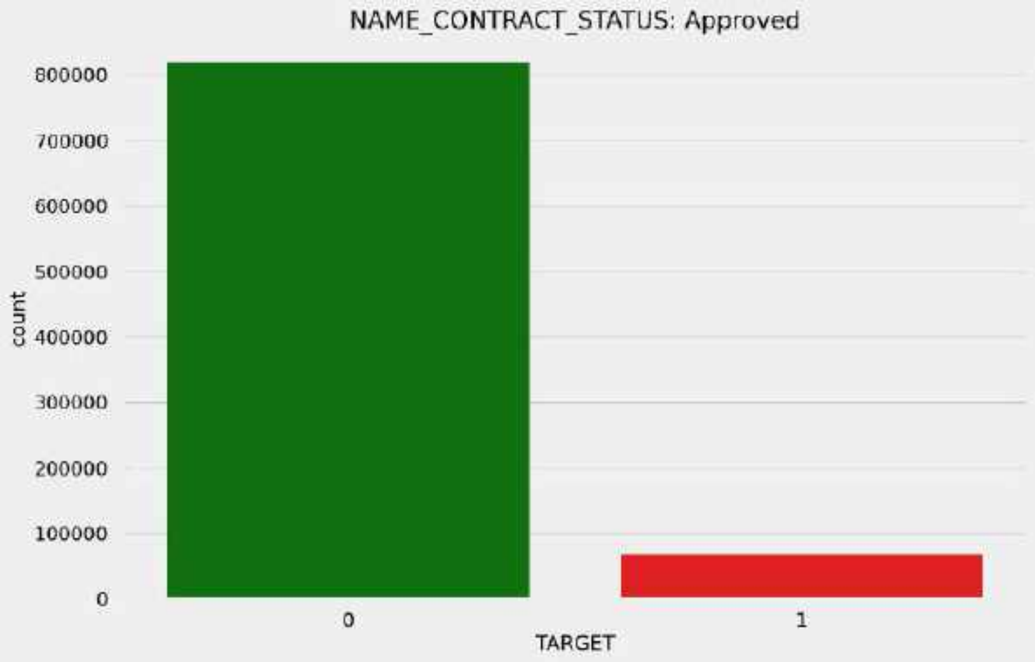
122980 rows × 74 columns

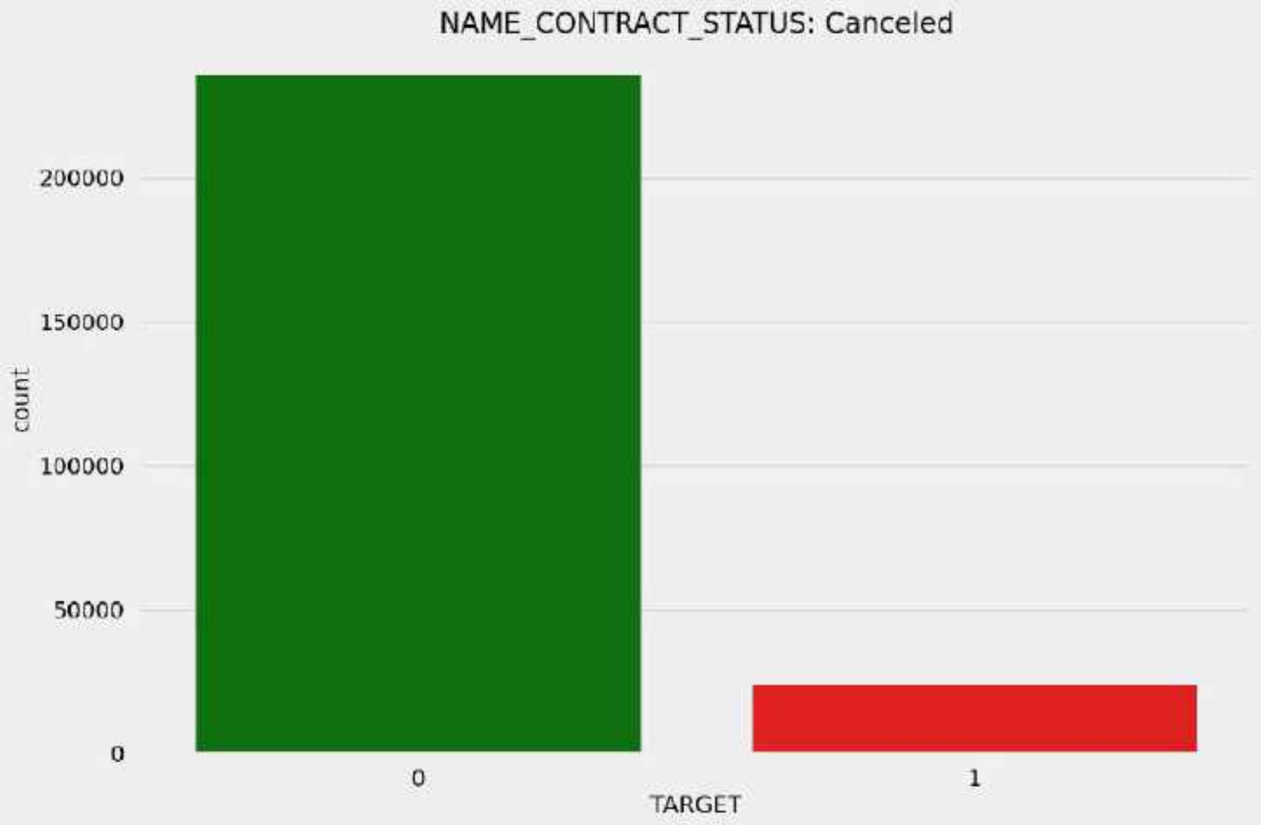


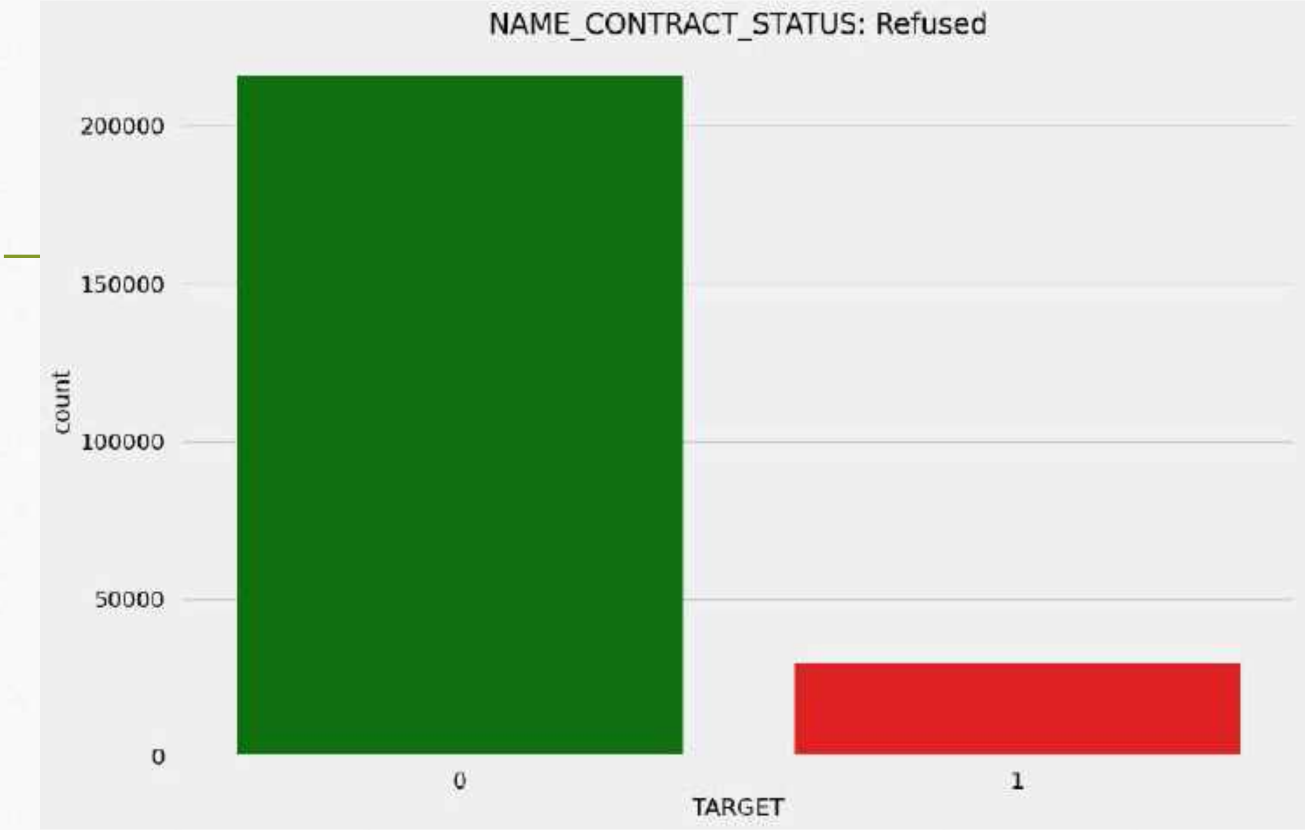
Plotting Contract Status vs purpose of the loan:

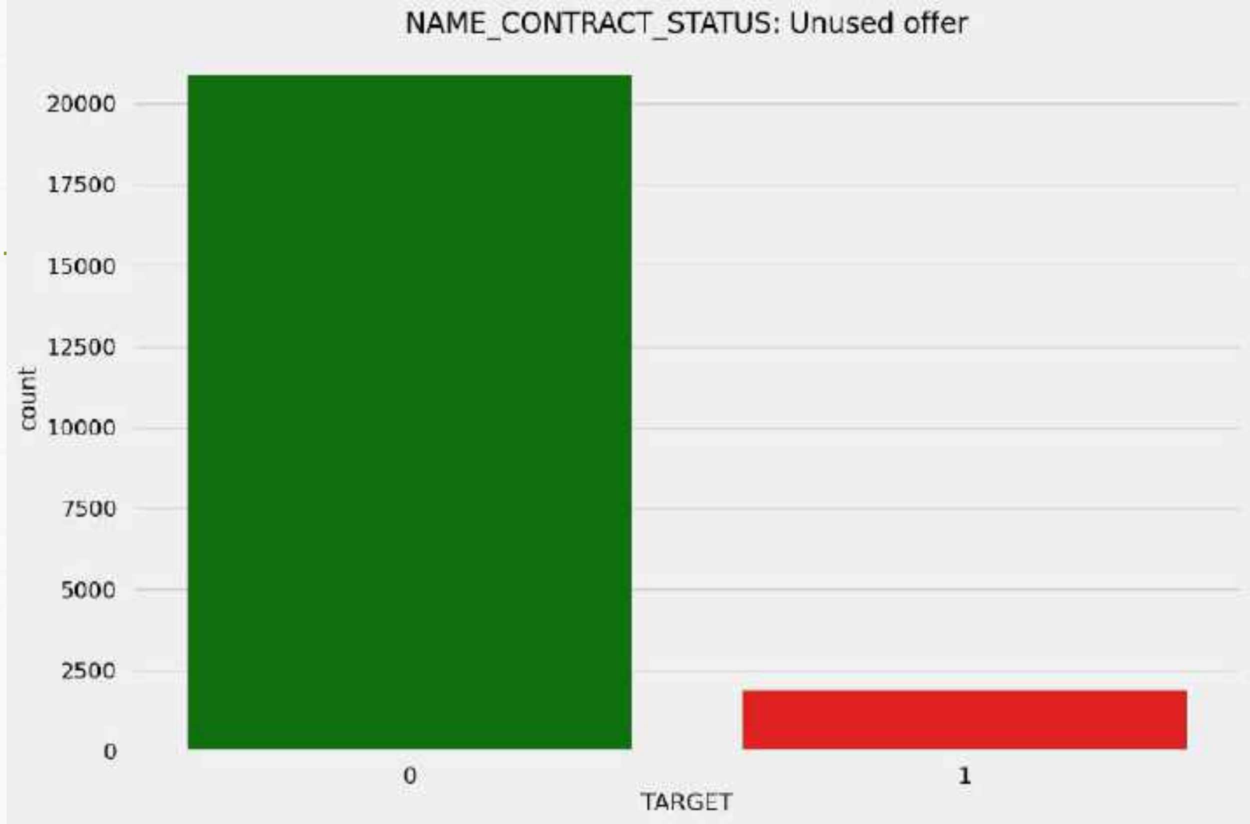

NAME_CASH_LOAN_PURPOSE
Approved
Canceled
Refused
Unused offer
Count (log)
10^0
10^1
10^2
10^3
10^4
XAP
XNA
Repairs
Other
Urgent needs
Building a house or an annex
Buying a used car
Payments on other loans
Medicine
Everyday expenses
Education
Car repairs
Purchase of electronic equipment
Journey
Wedding / gift / holiday
Furniture
Buying a home
Buying a new car
Buying a holiday home / land
Business development
Gasification / water supply
Hobby
Buying a garage
Money for a third person
Refusal to name the goal
NAME_CASH_LOAN_PURPOSE

- **Inferences:**Loan purpose has high number of unknown values (XAP, XNA)
- Loan taken for the purpose of Repairs seems to have highest default rate
- A very high number application have been rejected by bank or refused by client which has purpose as repair or other. This shows that purpose repair is taken as high risk by bank and either they are rejected or bank offers very high loan interest rate which is not feasible by the clients, thus they refuse the loan.

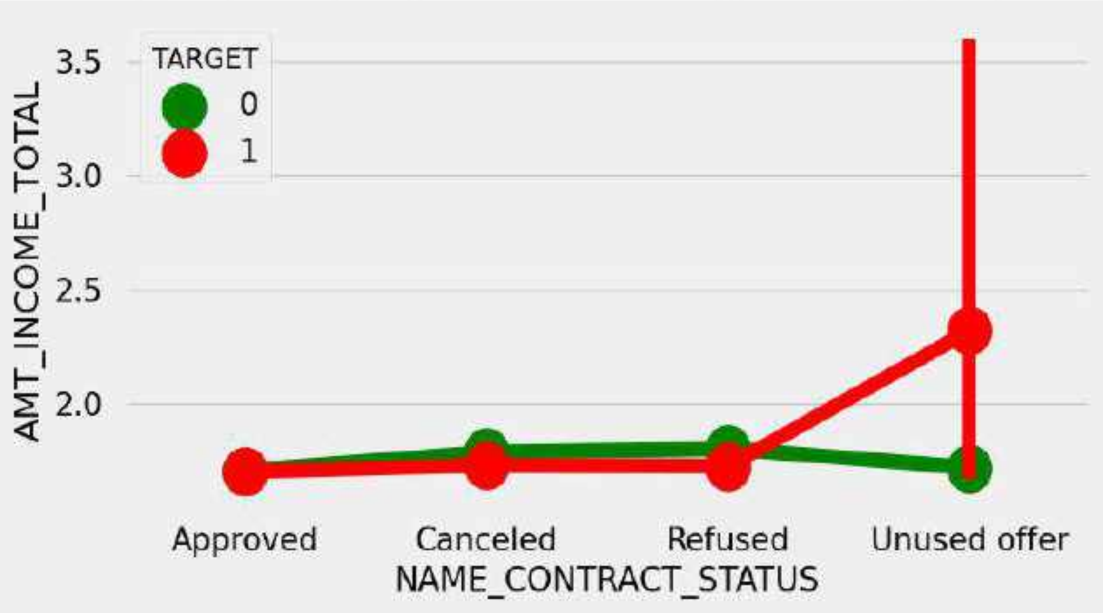

NAME_CONTRACT_STATUS: Approved
count
800000
700000
600000
500000
400000
300000
200000
100000
0
0
1
TARGET


NAME_CONTRACT_STATUS: Canceled
count
200000
150000
100000
50000
0
0
1
TARGET


NAME_CONTRACT_STATUS: Refused
count
200000
150000
100000
50000
0
0
1
TARGET


NAME_CONTRACT_STATUS: Unused offer
count
20000
17500
15000
12500
10000
7500
5000
2500
0
0
1
TARGET

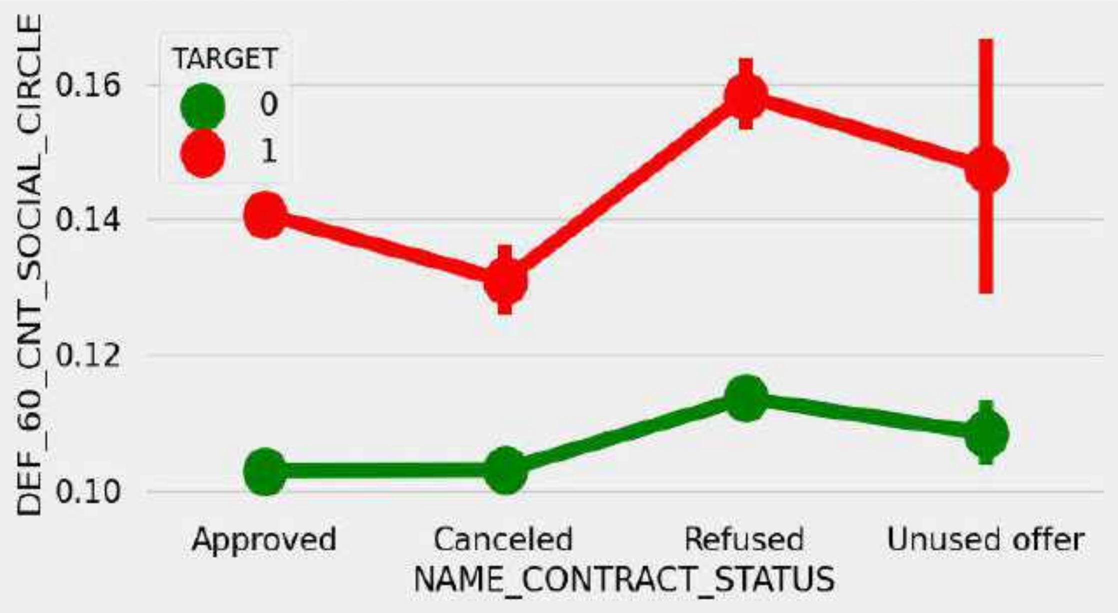
- **Inferences:**90% of the previously cancelled client have actually repayed the loan. Revisiting the interest rates would increase business opoortunity for these clients
- 88% of the clients who have been previously refused a loan has payed back the loan in current case.
- Refual reason should be recorded for further analysis as these clients would turn into potential repaying customer.



```
# plotting the relationship between income total and contact status
merged_pointplot("NAME_CONTRACT_STATUS",'AMT_INCOME_TOTAL')
```

Inferences:

The point plot show that the people who have not used offer earlier have defaulted even when there average income is higher than others



```
# plotting the relationship between people who defaulted in last 60 days being in client's social circle and contact status
merged_pointplot("NAME_CONTRACT_STATUS",'DEF_60_CNT_SOCIAL_CIRCLE')
```

Inferences:
Clients who have average of 0.13 or higher DEF_60_CNT_SOCIAL_CIRCLE score tend to default more and hence client's social circle has to be analysed before providing the loan.

- **7. Conclusions**
- After analysing the datasets, there are few attributes of a client with which the bank would be able to identify if they will repay the loan or not. The analysis is consised as below with the contributing factors and categorization:

1. **Decisive Factor whether an applicant will be Repayer:**NAME_EDUCATION_TYPE: Academic degree has less defaults.
2. NAME_INCOME_TYPE: Student and Businessmen have no defaults.
3. REGION_RATING_CLIENT: RATING 1 is safer.
4. ORGANIZATION_TYPE: Clients with Trade Type 4 and 5 and Industry type 8 have defaulted less than 3%
5. DAYS_BIRTH: People above age of 50 have low probability of defaulting
6. DAYS_EMPLOYED: Clients with 40+ year experience having less than 1% default rate
7. AMT_INCOME_TOTAL:Applicant with Income more than 700,000 are less likely to default
8. NAME_CASH_LOAN_PURPOSE: Loans bought for Hobby, Buying garage are being repayed mostly.
9. CNT_CHILDREN: People with zero to two children tend to repay the loans.

1. **Decisive Factor whether an applicant will be Defaulter:**CODE_GENDER: Men are at relatively higher default rate
2. NAME_FAMILY_STATUS : People who have civil marriage or who are single default a lot.
3. NAME_EDUCATION_TYPE: People with Lower Secondary & Secondary education
4. NAME_INCOME_TYPE: Clients who are either at Maternity leave OR Unemployed default a lot.
5. REGION_RATING_CLIENT: People who live in Rating 3 has highest defaults.

---

6. OCCUPATION_TYPE: Avoid Low-skill Laborers, Drivers and Waiters/barmen staff, Security staff, Laborers and Cooking staff as the default rate is huge.
7. ORGANIZATION_TYPE: Organizations with highest percent of loans not repaid are Transport: type 3 (16%), Industry: type 13 (13.5%), Industry: type 8 (12.5%) and Restaurant (less than 12%).
8. Self-employed people have relative high defaulting rate, and thus should be avoided to be approved for loan or provide loan with higher interest rate to mitigate the risk of defaulting.
9. DAYS_BIRTH: Avoid young people who are in age group of 20-40 as they have higher probability of defaulting
10. DAYS_EMPLOYED: People who have less than 5 years of employment have high default rate.
11. CNT_CHILDREN & CNT_FAM_MEMBERS: Client who have children equal to or more than 9 default 100% and hence their applications are to be rejected.
12. AMT_GOODS_PRICE: When the credit amount goes beyond 3M, there is an increase in defaulters.

- The following attributes indicate that people from these category tend to default but then due to the number of people and the amount of loan, the bank could provide loan with higher interest to mitigate any default risk thus preventing business loss:

1. NAME_HOUSING_TYPE: High number of loan applications are from the category of people who live in Rented apartments & living with parents and hence offering the loan would mitigate the loss if any of those default.
2. AMT_CREDIT: People who get loan for 300-600k tend to default more than others and hence having higher interest specifically for this credit range would be ideal.
3. AMT_INCOME: Since 90% of the applications have Income total less than 300,000 and they have high probability of defaulting, they could be offered loan with higher interest compared to other income category.
4. CNT_CHILDREN & CNT_FAM_MEMBERS: Clients who have 4 to 8 children has a very high default rate and hence higher interest should be imposed on their loans.
5. NAME_CASH_LOAN_PURPOSE: Loan taken for the purpose of Repairs seems to have highest default rate. A very high number applications have been rejected by bank or refused by client in previous applications as well which has purpose as repair or other. This shows that purpose repair is taken as high risk by bank and either they are rejected, or bank offers very high loan interest rate which is not feasible by the clients, thus they refuse the loan. The same approach could be followed in future as well.

- **Other suggestions:**90% of the previously cancelled client have actually repayed the loan. Record the reason for cancellation which might help the bank to determine and negotiate terms with these repaying customers in future for increase business opportunity.
- 88% of the clients who were refused by bank for loan earlier have now turned into a repaying client. Hence documenting the reason for rejection could mitigate the business loss and these clients could be contacted for further loans.