Bank Default risk analysis

Import Python Libraries:

```
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
```

Supress Warnings:

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

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)
```

Reading & Understanding the data

Importing the input files

```
# Input data files are available in the read-only "../input/"
directory
# For example, running this (by clicking run or pressing Shift+Enter)
will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

applicationDF = pd.read_csv(r"C:\Users\91799\OneDrive\Desktop\
projects\Bank defaulters risk analysis\application_data.csv")
previousDF = pd.read_csv(r"C:\Users\91799\OneDrive\Desktop\projects\
Bank defaulters risk analysis\previous_application.csv")
applicationDF.head()
```

```
   SK_ID_CURR  TARGET NAME_CONTRACT_TYPE CODE_GENDER FLAG_OWN_CAR
FLAG_OWN_REALTY  CNT_CHILDREN  AMT_INCOME_TOTAL  AMT_CREDIT
AMT_ANNUITY  AMT_GOODS_PRICE NAME_TYPE_SUITE NAME_INCOME_TYPE
NAME_EDUCATION_TYPE    NAME_FAMILY_STATUS  NAME_HOUSING_TYPE
REGION_POPULATION_RELATIVE  DAYS_BIRTH  DAYS_EMPLOYED
DAYS_REGISTRATION  DAYS_ID_PUBLISH  OWN_CAR_AGE  FLAG_MOBIL
FLAG_EMP_PHONE  FLAG_WORK_PHONE  FLAG_CONT_MOBILE  FLAG_PHONE
FLAG_EMAIL OCCUPATION_TYPE  CNT_FAM_MEMBERS  REGION_RATING_CLIENT
REGION_RATING_CLIENT_W_CITY WEEKDAY_APPR_PROCESS_START
HOUR_APPR_PROCESS_START  REG_REGION_NOT_LIVE_REGION
REG_REGION_NOT_WORK_REGION  LIVE_REGION_NOT_WORK_REGION
REG_CITY_NOT_LIVE_CITY  REG_CITY_NOT_WORK_CITY
LIVE_CITY_NOT_WORK_CITY       ORGANIZATION_TYPE  EXT_SOURCE_1
EXT_SOURCE_2  EXT_SOURCE_3  APARTMENTS_AVG  BASEMENTAREA_AVG
YEARS_BEGINEXPLUATATION_AVG  YEARS_BUILD_AVG  COMMONAREA_AVG
ELEVATORS_AVG  ENTRANCES_AVG  FLOORSMAX_AVG  FLOORSMIN_AVG
LANDAREA_AVG  LIVINGAPARTMENTS_AVG  LIVINGAREA_AVG
NONLIVINGAPARTMENTS_AVG  NONLIVINGAREA_AVG  APARTMENTS_MODE
BASEMENTAREA_MODE  YEARS_BEGINEXPLUATATION_MODE  YEARS_BUILD_MODE
COMMONAREA_MODE  ELEVATORS_MODE  ENTRANCES_MODE  FLOORSMAX_MODE
FLOORSMIN_MODE  LANDAREA_MODE  LIVINGAPARTMENTS_MODE  LIVINGAREA_MODE
NONLIVINGAPARTMENTS_MODE  NONLIVINGAREA_MODE  APARTMENTS_MEDI
BASEMENTAREA_MEDI  YEARS_BEGINEXPLUATATION_MEDI  YEARS_BUILD_MEDI
COMMONAREA_MEDI  ELEVATORS_MEDI  ENTRANCES_MEDI  FLOORSMAX_MEDI
FLOORSMIN_MEDI  LANDAREA_MEDI  LIVINGAPARTMENTS_MEDI  LIVINGAREA_MEDI
NONLIVINGAPARTMENTS_MEDI  NONLIVINGAREA_MEDI FONDKAPREMONT_MODE
HOUSETYPE_MODE  TOTALAREA_MODE WALLSMATERIAL_MODE EMERGENCYSTATE_MODE
OBS_30_CNT_SOCIAL_CIRCLE  DEF_30_CNT_SOCIAL_CIRCLE
OBS_60_CNT_SOCIAL_CIRCLE  DEF_60_CNT_SOCIAL_CIRCLE
DAYS_LAST_PHONE_CHANGE  FLAG_DOCUMENT_2  FLAG_DOCUMENT_3
FLAG_DOCUMENT_4  FLAG_DOCUMENT_5  FLAG_DOCUMENT_6  FLAG_DOCUMENT_7
FLAG_DOCUMENT_8  FLAG_DOCUMENT_9  FLAG_DOCUMENT_10  FLAG_DOCUMENT_11
FLAG_DOCUMENT_12  FLAG_DOCUMENT_13  FLAG_DOCUMENT_14  FLAG_DOCUMENT_15
FLAG_DOCUMENT_16  FLAG_DOCUMENT_17  FLAG_DOCUMENT_18  FLAG_DOCUMENT_19
FLAG_DOCUMENT_20  FLAG_DOCUMENT_21  AMT_REQ_CREDIT_BUREAU_HOUR
AMT_REQ_CREDIT_BUREAU_DAY  AMT_REQ_CREDIT_BUREAU_WEEK
AMT_REQ_CREDIT_BUREAU_MON  AMT_REQ_CREDIT_BUREAU_QRT
AMT_REQ_CREDIT_BUREAU_YEAR
0      100002       1         Cash loans           M            N
Y             0          202500.0    406597.5      24700.5
351000.0   Unaccompanied          Working  Secondary / secondary
special  Single / not married  House / apartment
0.018801       -9461           -637            -3648.0            -
2120          NaN           1               1                0
1           1           0        Laborers              1.0
2                           2                  WEDNESDAY
10                          0                          0
0                       0                       0
0  Business Entity Type 3      0.083037      0.262949      0.139376
0.0247            0.0369                        0.9722           0.6192
```

```
0.0143           0.00         0.0690         0.0833         0.1250
0.0369               0.0202          0.0190                    0.0000
0.0000           0.0252             0.0383
0.9722            0.6341           0.0144          0.0000
0.0690          0.0833          0.1250         0.0377
0.022           0.0198                         0.0                  0.0
0.0250             0.0369                         0.9722
0.6243           0.0144            0.00          0.0690
0.0833          0.1250         0.0375                0.0205
0.0193                      0.0000                 0.00   reg oper
account  block of flats          0.0149       Stone, brick
No                        2.0                         2.0
2.0                        2.0                 -1134.0                 0
1                0                0                0                 0
0                0                  0                 0
0                 0                 0                 0
0                 0                 0                 0
0                 0                          0.0
0.0                          0.0                          0.0
0.0                          1.0
1      100003       0         Cash loans           F             N
N             0          270000.0   1293502.5      35698.5
1129500.0          Family    State servant               Higher
education               Married  House / apartment
0.003541      -16765          -1188            -1186.0              -
291          NaN           1               1                0
1          1           0      Core staff              2.0
1                           1                      MONDAY
11                          0                           0
0                        0                        0
0                  School      0.311267      0.622246           NaN
0.0959            0.0529                         0.9851           0.7960
0.0605           0.08         0.0345         0.2917         0.3333
0.0130               0.0773          0.0549                    0.0039
0.0098           0.0924             0.0538
0.9851            0.8040           0.0497          0.0806
0.0345          0.2917          0.3333         0.0128
0.079           0.0554                         0.0                  0.0
0.0968             0.0529                         0.9851
0.7987           0.0608            0.08          0.0345
0.2917          0.3333         0.0132                0.0787
0.0558                      0.0039                 0.01   reg oper
account  block of flats          0.0714              Block
No                        1.0                         0.0
1.0                        0.0                  -828.0                 0
1                0                0                0                 0
0                0                  0                 0
0                 0                 0                 0
0                 0                 0                 0
0                 0                          0.0
```

```
0.0                          0.0                          0.0
0.0                          0.0
2      100004       0    Revolving loans           M            Y
Y             0           67500.0    135000.0       6750.0
135000.0   Unaccompanied          Working  Secondary / secondary
special  Single / not married  House / apartment
0.010032      -19046           -225            -4260.0            -
2531         26.0           1               1                1
1           1           0        Laborers              1.0
2                            2                      MONDAY
9                           0                           0
0                       0                         0
0               Government           NaN      0.555912      0.729567
NaN               NaN                            NaN              NaN
NaN             NaN            NaN             NaN             NaN
NaN                    NaN             NaN                          NaN
NaN               NaN                NaN                             NaN
NaN               NaN              NaN              NaN              NaN
NaN             NaN                       NaN               NaN
NaN                 NaN               NaN                NaN
NaN               NaN              NaN              NaN               NaN
NaN              NaN             NaN                    NaN
NaN                         NaN                   NaN                  NaN
NaN                  NaN                 NaN                  NaN
0.0                         0.0                          0.0
0.0                  -815.0                 0                 0
0                0                 0                 0                   0
0                  0                  0                  0
0                  0                  0                  0
0                  0                  0                  0
0                         0.0                           0.0
0.0                        0.0                           0.0
0.0
3      100006       0         Cash loans           F            N
Y             0          135000.0    312682.5      29686.5
297000.0   Unaccompanied          Working  Secondary / secondary
special        Civil marriage  House / apartment
0.008019      -19005          -3039            -9833.0            -
2437          NaN           1               1                0
1           0           0        Laborers              2.0
2                            2                  WEDNESDAY
17                           0                           0
0                       0                         0
0  Business Entity Type 3           NaN      0.650442           NaN
NaN               NaN                            NaN              NaN
NaN             NaN            NaN             NaN             NaN
NaN                    NaN             NaN                          NaN
NaN               NaN                NaN                             NaN
NaN               NaN              NaN              NaN              NaN
NaN             NaN                       NaN               NaN
```

```
NaN                 NaN              NaN                NaN
NaN               NaN              NaN              NaN              NaN
NaN             NaN            NaN                    NaN
NaN                       NaN                  NaN                NaN
NaN             NaN                NaN                NaN
2.0                       0.0                        2.0
0.0                  -617.0               0                1
0                0                0                0                0
0                 0                 0                 0
0                 0                 0                 0
0                 0                 0                 0
0                         NaN                        NaN
NaN                        NaN                        NaN
NaN
4      100007       0         Cash loans          M            N
Y             0          121500.0    513000.0      21865.5
513000.0   Unaccompanied          Working  Secondary / secondary
special  Single / not married  House / apartment
0.028663      -19932          -3038           -4311.0           -
3458          NaN           1               1                0
1           0           0      Core staff             1.0
2                            2                   THURSDAY
11                           0                            0
0                           0                       1
1                Religion           NaN      0.322738           NaN
NaN               NaN                        NaN              NaN
NaN            NaN            NaN            NaN            NaN
NaN                     NaN             NaN                        NaN
NaN              NaN                NaN                           NaN
NaN              NaN             NaN             NaN              NaN
NaN            NaN                    NaN                NaN
NaN                 NaN              NaN                NaN
NaN               NaN              NaN              NaN              NaN
NaN             NaN            NaN                    NaN
NaN                       NaN                  NaN                NaN
NaN             NaN                NaN                NaN
0.0                       0.0                        0.0
0.0                 -1106.0               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                        0.0
0.0
```

```
previousDF.head()
```

```
   SK_ID_PREV  SK_ID_CURR NAME_CONTRACT_TYPE  AMT_ANNUITY
AMT_APPLICATION  AMT_CREDIT  AMT_DOWN_PAYMENT  AMT_GOODS_PRICE
```

```
WEEKDAY_APPR_PROCESS_START  HOUR_APPR_PROCESS_START
FLAG_LAST_APPL_PER_CONTRACT  NFLAG_LAST_APPL_IN_DAY  RATE_DOWN_PAYMENT
RATE_INTEREST_PRIMARY  RATE_INTEREST_PRIVILEGED NAME_CASH_LOAN_PURPOSE
NAME_CONTRACT_STATUS  DAYS_DECISION      NAME_PAYMENT_TYPE
CODE_REJECT_REASON  NAME_TYPE_SUITE NAME_CLIENT_TYPE
NAME_GOODS_CATEGORY NAME_PORTFOLIO NAME_PRODUCT_TYPE
CHANNEL_TYPE  SELLERPLACE_AREA NAME_SELLER_INDUSTRY  CNT_PAYMENT
NAME_YIELD_GROUP       PRODUCT_COMBINATION  DAYS_FIRST_DRAWING
DAYS_FIRST_DUE  DAYS_LAST_DUE_1ST_VERSION  DAYS_LAST_DUE
DAYS_TERMINATION  NFLAG_INSURED_ON_APPROVAL
0     2030495      271877     Consumer loans     1730.430
17145.0     17145.0              0.0          17145.0
SATURDAY                         15                              Y
1                0.0               0.182832                   0.867336
XAP             Approved            -73  Cash through the bank
XAP              NaN         Repeater              Mobile
POS               XNA             Country-wide                 35
Connectivity         12.0           middle  POS mobile with interest
365243.0           -42.0                         300.0          -42.0
-37.0                         0.0
1     2802425      108129         Cash loans    25188.615
607500.0    679671.0              NaN         607500.0
THURSDAY                        11                              Y
1                NaN                    NaN                         NaN
XNA             Approved           -164                    XNA
XAP    Unaccompanied         Repeater                 XNA
Cash            x-sell           Contact center                -1
XNA         36.0       low_action          Cash X-Sell: low
365243.0          -134.0                       916.0       365243.0
365243.0                          1.0
2     2523466      122040         Cash loans    15060.735
112500.0    136444.5              NaN         112500.0
TUESDAY                         11                              Y
1                NaN                    NaN                         NaN
XNA             Approved           -301  Cash through the bank
XAP  Spouse, partner         Repeater                 XNA
Cash            x-sell  Credit and cash offices                -1
XNA         12.0             high         Cash X-Sell: high
365243.0          -271.0                         59.0       365243.0
365243.0                          1.0
3     2819243      176158         Cash loans    47041.335
450000.0    470790.0              NaN         450000.0
MONDAY                          7                              Y
1                NaN                    NaN                         NaN
XNA             Approved           -512  Cash through the bank
XAP              NaN         Repeater                 XNA
Cash            x-sell  Credit and cash offices                -1
XNA         12.0           middle       Cash X-Sell: middle
365243.0          -482.0                      -152.0         -182.0
```

```
-177.0                          1.0
4     1784265      202054         Cash loans    31924.395
337500.0    404055.0               NaN         337500.0
THURSDAY                          9                              Y
1                NaN                     NaN                         NaN
Repairs              Refused           -781  Cash through the bank
HC               NaN         Repeater                 XNA
Cash           walk-in  Credit and cash offices                  -1
XNA         24.0             high         Cash Street: high
NaN              NaN                          NaN            NaN
NaN                          NaN
```

Inspect Data Frames

```
# 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        : (1048575, 37)
Database size - applicationDF          : 37516342
Database size - previousDF             : 38797275
```

```
# 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
 14   NAME_FAMILY_STATUS            object
```

```
15   NAME_HOUSING_TYPE             object
16   REGION_POPULATION_RELATIVE    float64
17   DAYS_BIRTH                    int64
18   DAYS_EMPLOYED                 int64
19   DAYS_REGISTRATION             float64
20   DAYS_ID_PUBLISH               int64
21   OWN_CAR_AGE                   float64
22   FLAG_MOBIL                    int64
23   FLAG_EMP_PHONE                int64
24   FLAG_WORK_PHONE               int64
25   FLAG_CONT_MOBILE              int64
26   FLAG_PHONE                    int64
27   FLAG_EMAIL                    int64
28   OCCUPATION_TYPE               object
29   CNT_FAM_MEMBERS               float64
30   REGION_RATING_CLIENT          int64
31   REGION_RATING_CLIENT_W_CITY   int64
32   WEEKDAY_APPR_PROCESS_START    object
33   HOUR_APPR_PROCESS_START       int64
34   REG_REGION_NOT_LIVE_REGION    int64
35   REG_REGION_NOT_WORK_REGION    int64
36   LIVE_REGION_NOT_WORK_REGION   int64
37   REG_CITY_NOT_LIVE_CITY        int64
38   REG_CITY_NOT_WORK_CITY        int64
39   LIVE_CITY_NOT_WORK_CITY       int64
40   ORGANIZATION_TYPE             object
41   EXT_SOURCE_1                  float64
42   EXT_SOURCE_2                  float64
43   EXT_SOURCE_3                  float64
44   APARTMENTS_AVG                float64
45   BASEMENTAREA_AVG              float64
46   YEARS_BEGINEXPLUATATION_AVG   float64
47   YEARS_BUILD_AVG               float64
48   COMMONAREA_AVG                float64
49   ELEVATORS_AVG                 float64
50   ENTRANCES_AVG                 float64
51   FLOORSMAX_AVG                 float64
52   FLOORSMIN_AVG                 float64
53   LANDAREA_AVG                  float64
54   LIVINGAPARTMENTS_AVG          float64
55   LIVINGAREA_AVG                float64
56   NONLIVINGAPARTMENTS_AVG       float64
57   NONLIVINGAREA_AVG             float64
58   APARTMENTS_MODE               float64
59   BASEMENTAREA_MODE             float64
60   YEARS_BEGINEXPLUATATION_MODE  float64
61   YEARS_BUILD_MODE              float64
62   COMMONAREA_MODE               float64
63   ELEVATORS_MODE                float64
```

```
64   ENTRANCES_MODE                float64
65   FLOORSMAX_MODE                float64
66   FLOORSMIN_MODE                float64
67   LANDAREA_MODE                 float64
68   LIVINGAPARTMENTS_MODE         float64
69   LIVINGAREA_MODE               float64
70   NONLIVINGAPARTMENTS_MODE      float64
71   NONLIVINGAREA_MODE            float64
72   APARTMENTS_MEDI               float64
73   BASEMENTAREA_MEDI             float64
74   YEARS_BEGINEXPLUATATION_MEDI  float64
75   YEARS_BUILD_MEDI              float64
76   COMMONAREA_MEDI               float64
77   ELEVATORS_MEDI                float64
78   ENTRANCES_MEDI                float64
79   FLOORSMAX_MEDI                float64
80   FLOORSMIN_MEDI                float64
81   LANDAREA_MEDI                 float64
82   LIVINGAPARTMENTS_MEDI         float64
83   LIVINGAREA_MEDI               float64
84   NONLIVINGAPARTMENTS_MEDI      float64
85   NONLIVINGAREA_MEDI            float64
86   FONDKAPREMONT_MODE            object
87   HOUSETYPE_MODE                object
88   TOTALAREA_MODE                float64
89   WALLSMATERIAL_MODE            object
90   EMERGENCYSTATE_MODE           object
91   OBS_30_CNT_SOCIAL_CIRCLE      float64
92   DEF_30_CNT_SOCIAL_CIRCLE      float64
93   OBS_60_CNT_SOCIAL_CIRCLE      float64
94   DEF_60_CNT_SOCIAL_CIRCLE      float64
95   DAYS_LAST_PHONE_CHANGE        float64
96   FLAG_DOCUMENT_2               int64
97   FLAG_DOCUMENT_3               int64
98   FLAG_DOCUMENT_4               int64
99   FLAG_DOCUMENT_5               int64
100  FLAG_DOCUMENT_6               int64
101  FLAG_DOCUMENT_7               int64
102  FLAG_DOCUMENT_8               int64
103  FLAG_DOCUMENT_9               int64
104  FLAG_DOCUMENT_10              int64
105  FLAG_DOCUMENT_11              int64
106  FLAG_DOCUMENT_12              int64
107  FLAG_DOCUMENT_13              int64
108  FLAG_DOCUMENT_14              int64
109  FLAG_DOCUMENT_15              int64
110  FLAG_DOCUMENT_16              int64
111  FLAG_DOCUMENT_17              int64
112  FLAG_DOCUMENT_18              int64
```

```
 113  FLAG_DOCUMENT_19              int64
 114  FLAG_DOCUMENT_20              int64
 115  FLAG_DOCUMENT_21              int64
 116  AMT_REQ_CREDIT_BUREAU_HOUR    float64
 117  AMT_REQ_CREDIT_BUREAU_DAY     float64
 118  AMT_REQ_CREDIT_BUREAU_WEEK    float64
 119  AMT_REQ_CREDIT_BUREAU_MON     float64
 120  AMT_REQ_CREDIT_BUREAU_QRT     float64
 121  AMT_REQ_CREDIT_BUREAU_YEAR    float64
dtypes: float64(65), int64(41), object(16)
memory usage: 286.2+ MB
```

```
previousDF.info(verbose=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1048575 entries, 0 to 1048574
Data columns (total 37 columns):
 #   Column                         Non-Null Count    Dtype
---  ------                         --------------    -----
 0   SK_ID_PREV                     1048575 non-null  int64
 1   SK_ID_CURR                     1048575 non-null  int64
 2   NAME_CONTRACT_TYPE             1048575 non-null  object
 3   AMT_ANNUITY                    815566 non-null   float64
 4   AMT_APPLICATION                1048575 non-null  float64
 5   AMT_CREDIT                     1048575 non-null  float64
 6   AMT_DOWN_PAYMENT               489179 non-null   float64
 7   AMT_GOODS_PRICE                807610 non-null   float64
 8   WEEKDAY_APPR_PROCESS_START     1048575 non-null  object
 9   HOUR_APPR_PROCESS_START        1048575 non-null  int64
 10  FLAG_LAST_APPL_PER_CONTRACT    1048575 non-null  object
 11  NFLAG_LAST_APPL_IN_DAY         1048575 non-null  int64
 12  RATE_DOWN_PAYMENT              489179 non-null   float64
 13  RATE_INTEREST_PRIMARY          3721 non-null     float64
 14  RATE_INTEREST_PRIVILEGED       3721 non-null     float64
 15  NAME_CASH_LOAN_PURPOSE         1048575 non-null  object
 16  NAME_CONTRACT_STATUS           1048575 non-null  object
 17  DAYS_DECISION                  1048575 non-null  int64
 18  NAME_PAYMENT_TYPE              1048575 non-null  object
 19  CODE_REJECT_REASON             1048575 non-null  object
 20  NAME_TYPE_SUITE                533435 non-null   object
 21  NAME_CLIENT_TYPE               1048575 non-null  object
 22  NAME_GOODS_CATEGORY            1048575 non-null  object
 23  NAME_PORTFOLIO                 1048575 non-null  object
 24  NAME_PRODUCT_TYPE              1048575 non-null  object
 25  CHANNEL_TYPE                   1048575 non-null  object
 26  SELLERPLACE_AREA               1048575 non-null  int64
 27  NAME_SELLER_INDUSTRY           1048575 non-null  object
 28  CNT_PAYMENT                    815569 non-null   float64
 29  NAME_YIELD_GROUP               1048575 non-null  object
 30  PRODUCT_COMBINATION            1048351 non-null  object
```

```
 31  DAYS_FIRST_DRAWING           627867 non-null   float64
 32  DAYS_FIRST_DUE               627867 non-null   float64
 33  DAYS_LAST_DUE_1ST_VERSION    627867 non-null   float64
 34  DAYS_LAST_DUE                627867 non-null   float64
 35  DAYS_TERMINATION             627867 non-null   float64
 36  NFLAG_INSURED_ON_APPROVAL    627867 non-null   float64
dtypes: float64(15), int64(6), object(16)
memory usage: 296.0+ MB
```

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

```
          SK_ID_CURR         TARGET   CNT_CHILDREN  AMT_INCOME_TOTAL
AMT_CREDIT    AMT_ANNUITY  AMT_GOODS_PRICE  REGION_POPULATION_RELATIVE
DAYS_BIRTH  DAYS_EMPLOYED  DAYS_REGISTRATION  DAYS_ID_PUBLISH
OWN_CAR_AGE     FLAG_MOBIL  FLAG_EMP_PHONE  FLAG_WORK_PHONE
FLAG_CONT_MOBILE     FLAG_PHONE     FLAG_EMAIL  CNT_FAM_MEMBERS
REGION_RATING_CLIENT  REGION_RATING_CLIENT_W_CITY
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   EXT_SOURCE_1  EXT_SOURCE_2   EXT_SOURCE_3
APARTMENTS_AVG  BASEMENTAREA_AVG  YEARS_BEGINEXPLUATATION_AVG
YEARS_BUILD_AVG  COMMONAREA_AVG  ELEVATORS_AVG  ENTRANCES_AVG
FLOORSMAX_AVG  FLOORSMIN_AVG   LANDAREA_AVG  LIVINGAPARTMENTS_AVG
LIVINGAREA_AVG  NONLIVINGAPARTMENTS_AVG  NONLIVINGAREA_AVG
APARTMENTS_MODE  BASEMENTAREA_MODE  YEARS_BEGINEXPLUATATION_MODE
YEARS_BUILD_MODE  COMMONAREA_MODE  ELEVATORS_MODE  ENTRANCES_MODE
FLOORSMAX_MODE  FLOORSMIN_MODE  LANDAREA_MODE  LIVINGAPARTMENTS_MODE
LIVINGAREA_MODE  NONLIVINGAPARTMENTS_MODE  NONLIVINGAREA_MODE
APARTMENTS_MEDI  BASEMENTAREA_MEDI  YEARS_BEGINEXPLUATATION_MEDI
YEARS_BUILD_MEDI  COMMONAREA_MEDI  ELEVATORS_MEDI  ENTRANCES_MEDI
FLOORSMAX_MEDI  FLOORSMIN_MEDI  LANDAREA_MEDI  LIVINGAPARTMENTS_MEDI
LIVINGAREA_MEDI  NONLIVINGAPARTMENTS_MEDI  NONLIVINGAREA_MEDI
TOTALAREA_MODE  OBS_30_CNT_SOCIAL_CIRCLE  DEF_30_CNT_SOCIAL_CIRCLE
OBS_60_CNT_SOCIAL_CIRCLE  DEF_60_CNT_SOCIAL_CIRCLE
DAYS_LAST_PHONE_CHANGE  FLAG_DOCUMENT_2  FLAG_DOCUMENT_3
FLAG_DOCUMENT_4  FLAG_DOCUMENT_5  FLAG_DOCUMENT_6  FLAG_DOCUMENT_7
FLAG_DOCUMENT_8  FLAG_DOCUMENT_9  FLAG_DOCUMENT_10  FLAG_DOCUMENT_11
FLAG_DOCUMENT_12  FLAG_DOCUMENT_13  FLAG_DOCUMENT_14  FLAG_DOCUMENT_15
FLAG_DOCUMENT_16  FLAG_DOCUMENT_17  FLAG_DOCUMENT_18  FLAG_DOCUMENT_19
FLAG_DOCUMENT_20  FLAG_DOCUMENT_21  AMT_REQ_CREDIT_BUREAU_HOUR
AMT_REQ_CREDIT_BUREAU_DAY  AMT_REQ_CREDIT_BUREAU_WEEK
AMT_REQ_CREDIT_BUREAU_MON  AMT_REQ_CREDIT_BUREAU_QRT
AMT_REQ_CREDIT_BUREAU_YEAR
count  307511.000000  307511.000000  307511.000000      3.075110e+05
3.075110e+05  307499.000000     3.072330e+05
307511.000000  307511.000000  307511.000000      307511.000000
307511.000000  104582.000000  307511.000000   307511.000000
307511.000000     307511.000000  307511.000000  307511.000000
```

```
307509.000000         307511.000000                307511.000000
307511.000000              307511.000000              307511.000000
307511.000000           307511.000000           307511.000000
307511.000000  134133.000000  3.068510e+05  246546.000000
151450.00000     127568.000000                157504.000000
103023.000000    92646.000000  143620.000000  152683.000000
154491.000000   98869.000000  124921.000000          97312.000000
153161.000000             93997.000000      137829.000000
151450.000000      127568.000000                 157504.000000
103023.000000     92646.000000   143620.000000   152683.000000
154491.000000    98869.000000  124921.000000           97312.000000
153161.000000             93997.000000       137829.000000
151450.000000      127568.000000                 157504.000000
103023.000000     92646.000000   143620.000000   152683.000000
154491.000000    98869.000000  124921.000000           97312.000000
153161.000000             93997.000000       137829.000000
159080.000000            306490.000000             306490.000000
306490.000000            306490.000000           307510.000000
307511.000000    307511.000000    307511.000000    307511.000000
307511.000000    307511.000000    307511.000000    307511.000000
307511.000000     307511.000000     307511.000000     307511.000000
307511.000000      307511.00000     307511.000000     307511.000000
307511.000000     307511.000000     307511.000000     307511.000000
265992.000000              265992.000000               265992.000000
265992.000000              265992.000000               265992.000000
mean   278180.518577       0.080729       0.417052      1.687979e+05
5.990260e+05   27108.573909     5.383962e+05
0.020868  -16036.995067   63815.045904       -4986.120328     -
2994.202373      12.061091       0.999997        0.819889
0.199368          0.998133       0.281066       0.056720
2.152665              2.052463                     2.031521
12.063419                   0.015144                    0.050769
0.040659                0.078173                0.230454
0.179555       0.502130  5.143927e-01       0.510853         0.11744
0.088442                    0.977735         0.752471        0.044621
0.078942       0.149725       0.226282       0.231894       0.066333
0.100775        0.107399                 0.008809           0.028358
0.114231           0.087543                      0.977065
0.759637         0.042553        0.074490        0.145193
0.222315        0.228058       0.064958               0.105645
0.105975                  0.008076            0.027022
0.117850           0.087955                      0.977752
0.755746         0.044595        0.078078        0.149213
0.225897        0.231625       0.067169               0.101954
0.108607                  0.008651            0.028236        0.102547
1.422245                  0.143421                  1.405292
0.100049             -962.858788         0.000042         0.710023
0.000081         0.015115         0.088055         0.000192
0.081376         0.003896          0.000023          0.003912
```

```
0.000007          0.003525          0.002936           0.00121
0.009928          0.000267          0.008130          0.000595
0.000507          0.000335                  0.006402
0.007000                  0.034362                  0.267395
0.265474                  1.899974
std    102790.175348       0.272419       0.722121      2.371231e+05
4.024908e+05   14493.737315     3.694465e+05
0.013831    4363.988632  141275.766519        3522.886321
1509.450419      11.944812       0.001803        0.384280
0.399526          0.043164       0.449521       0.231307
0.910682              0.509034                  0.502737
3.265832                  0.122126                  0.219526
0.197499                0.268444                0.421124
0.383817       0.211062  1.910602e-01       0.194844         0.10824
0.082438                  0.059223         0.113280        0.076036
0.134576       0.100049       0.144641       0.161380       0.081184
0.092576        0.110565                0.047732           0.069523
0.107936           0.084307                     0.064575
0.110111         0.074445        0.132256        0.100977
0.143709        0.161160       0.081750               0.097880
0.111845                0.046276            0.070254
0.109076           0.082179                     0.059897
0.112066         0.076144        0.134467        0.100368
0.145067        0.161934       0.082167               0.093642
0.112260                0.047415            0.070166        0.107462
2.400989                0.446698                 2.379803
0.362291              826.808487         0.006502         0.453752
0.009016         0.122010         0.283376         0.013850
0.273412         0.062295          0.004771          0.062424
0.002550          0.059268          0.054110           0.03476
0.099144          0.016327          0.089798          0.024387
0.022518          0.018299                  0.083849
0.110757                  0.204685                  0.916002
0.794056                  1.869295
min    100002.000000       0.000000       0.000000      2.565000e+04
4.500000e+04    1615.500000     4.050000e+04
0.000290  -25229.000000  -17912.000000      -24672.000000     -
7197.000000       0.000000       0.000000        0.000000
0.000000          0.000000       0.000000       0.000000
1.000000              1.000000                  1.000000
0.000000                  0.000000                  0.000000
0.000000                0.000000                0.000000
0.000000       0.014568  8.170000e-08       0.000527         0.00000
0.000000                  0.000000         0.000000        0.000000
0.000000       0.000000       0.000000       0.000000       0.000000
0.000000        0.000000                0.000000           0.000000
0.000000           0.000000                     0.000000
0.000000         0.000000        0.000000        0.000000
0.000000        0.000000       0.000000               0.000000
```

```
0.000000                  0.000000            0.000000
0.000000           0.000000                   0.000000
0.000000         0.000000        0.000000        0.000000
0.000000        0.000000       0.000000               0.000000
0.000000                  0.000000            0.000000        0.000000
0.000000                  0.000000                  0.000000
0.000000            -4292.000000         0.000000         0.000000
0.000000         0.000000         0.000000         0.000000
0.000000         0.000000          0.000000          0.000000
0.000000          0.000000          0.000000           0.00000
0.000000          0.000000          0.000000          0.000000
0.000000          0.000000                  0.000000
0.000000                    0.000000                   0.000000
0.000000                    0.000000
25%    189145.500000       0.000000       0.000000      1.125000e+05
2.700000e+05   16524.000000     2.385000e+05
0.010006  -19682.000000   -2760.000000       -7479.500000     -
4299.000000       5.000000       1.000000        1.000000
0.000000          1.000000       0.000000       0.000000
2.000000              2.000000                   2.000000
10.000000                   0.000000                   0.000000
0.000000               0.000000               0.000000
0.000000       0.334007  3.924574e-01       0.370650         0.05770
0.044200                  0.976700         0.687200        0.007800
0.000000       0.069000       0.166700       0.083300       0.018700
0.050400        0.045300                0.000000          0.000000
0.052500           0.040700                       0.976700
0.699400         0.007200        0.000000        0.069000
0.166700        0.083300       0.016600               0.054200
0.042700                  0.000000            0.000000
0.058300           0.043700                       0.976700
0.691400         0.007900        0.000000        0.069000
0.166700        0.083300       0.018700               0.051300
0.045700                  0.000000            0.000000        0.041200
0.000000                  0.000000                  0.000000
0.000000            -1570.000000         0.000000         0.000000
0.000000         0.000000         0.000000         0.000000
0.000000         0.000000          0.000000          0.000000
0.000000          0.000000          0.000000           0.00000
0.000000          0.000000          0.000000          0.000000
0.000000          0.000000                  0.000000
0.000000                    0.000000                   0.000000
0.000000                    0.000000
50%    278202.000000       0.000000       0.000000      1.471500e+05
5.135310e+05   24903.000000     4.500000e+05
0.018850  -15750.000000   -1213.000000       -4504.000000     -
3254.000000       9.000000       1.000000        1.000000
0.000000          1.000000       0.000000       0.000000
2.000000              2.000000                   2.000000
```

```
12.000000                    0.000000                    0.000000
0.000000                0.000000                0.000000
0.000000       0.505998  5.659614e-01       0.535276         0.08760
0.076300                    0.981600         0.755200        0.021100
0.000000       0.137900       0.166700       0.208300       0.048100
0.075600        0.074500                0.000000           0.003600
0.084000           0.074600                    0.981600
0.764800         0.019000        0.000000        0.137900
0.166700        0.208300       0.045800               0.077100
0.073100                0.000000            0.001100
0.086400           0.075800                    0.981600
0.758500         0.020800        0.000000        0.137900
0.166700        0.208300       0.048700               0.076100
0.074900                0.000000            0.003100        0.068800
0.000000                0.000000                 0.000000
0.000000            -757.000000         0.000000         1.000000
0.000000         0.000000         0.000000         0.000000
0.000000         0.000000          0.000000          0.000000
0.000000          0.000000          0.000000           0.00000
0.000000          0.000000          0.000000          0.000000
0.000000          0.000000                    0.000000
0.000000                    0.000000                    0.000000
0.000000                    1.000000
75%    367142.500000       0.000000       1.000000      2.025000e+05
8.086500e+05   34596.000000     6.795000e+05
0.028663  -12413.000000    -289.000000       -2010.000000     -
1720.000000      15.000000       1.000000        1.000000
0.000000          1.000000       1.000000       0.000000
3.000000              2.000000                    2.000000
14.000000                   0.000000                    0.000000
0.000000                0.000000                0.000000
0.000000       0.675053  6.636171e-01       0.669057         0.14850
0.112200                    0.986600         0.823200        0.051500
0.120000       0.206900       0.333300       0.375000       0.085600
0.121000        0.129900                0.003900           0.027700
0.143900           0.112400                    0.986600
0.823600         0.049000        0.120800        0.206900
0.333300        0.375000       0.084100               0.131300
0.125200                0.003900            0.023100
0.148900           0.111600                    0.986600
0.825600         0.051300        0.120000        0.206900
0.333300        0.375000       0.086800               0.123100
0.130300                0.003900            0.026600        0.127600
2.000000                0.000000                 2.000000
0.000000            -274.000000         0.000000         1.000000
0.000000         0.000000         0.000000         0.000000
0.000000          0.000000          0.000000          0.000000
0.000000          0.000000          0.000000           0.00000
0.000000          0.000000          0.000000          0.000000
```

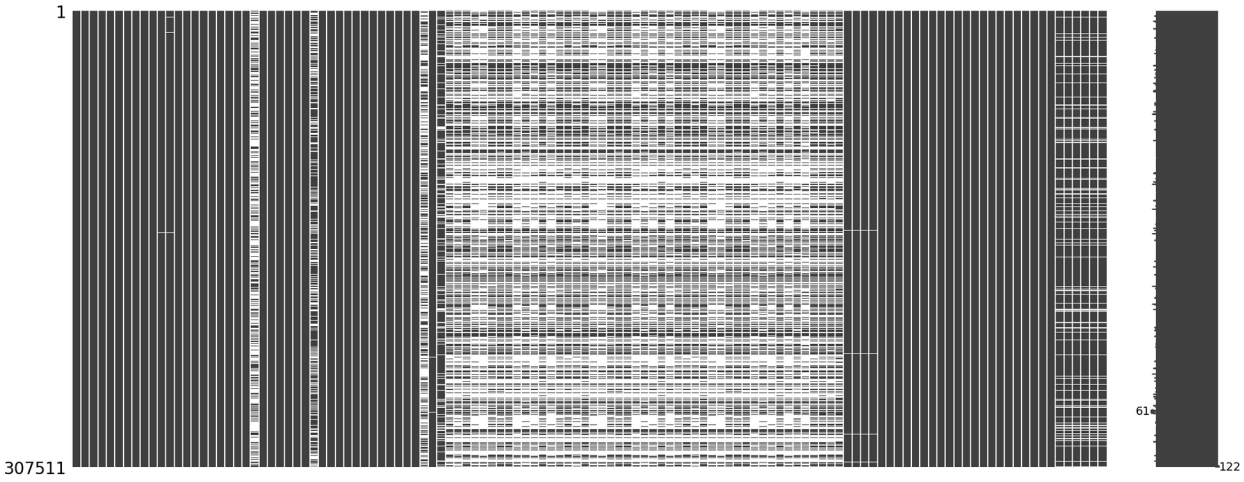
```
0.000000          0.000000                      0.000000
0.000000                    0.000000                    0.000000
0.000000                    3.000000
max    456255.000000       1.000000      19.000000      1.170000e+08
4.050000e+06  258025.500000     4.050000e+06
0.072508   -7489.000000  365243.000000           0.000000
0.000000      91.000000       1.000000        1.000000
1.000000          1.000000       1.000000       1.000000
20.000000             3.000000                     3.000000
23.000000                   1.000000                    1.000000
1.000000                1.000000                1.000000
1.000000       0.962693  8.549997e-01       0.896010         1.00000
1.000000                    1.000000         1.000000        1.000000
1.000000       1.000000       1.000000       1.000000       1.000000
1.000000        1.000000                1.000000           1.000000
1.000000           1.000000                      1.000000
1.000000         1.000000        1.000000        1.000000
1.000000        1.000000       1.000000               1.000000
1.000000                  1.000000            1.000000
1.000000           1.000000                     1.000000
1.000000         1.000000        1.000000        1.000000
1.000000        1.000000       1.000000               1.000000
1.000000                  1.000000            1.000000        1.000000
348.000000                 34.000000                344.000000
24.000000                0.000000         1.000000         1.000000
1.000000         1.000000         1.000000         1.000000
1.000000         1.000000          1.000000          1.000000
1.000000          1.000000          1.000000           1.00000
1.000000          1.000000          1.000000          1.000000
1.000000          1.000000                    4.000000
9.000000                   8.000000                   27.000000
261.000000                  25.000000

previousDF.describe()

         SK_ID_PREV    SK_ID_CURR    AMT_ANNUITY  AMT_APPLICATION
AMT_CREDIT  AMT_DOWN_PAYMENT  AMT_GOODS_PRICE  HOUR_APPR_PROCESS_START
NFLAG_LAST_APPL_IN_DAY  RATE_DOWN_PAYMENT  RATE_INTEREST_PRIMARY
RATE_INTEREST_PRIVILEGED  DAYS_DECISION  SELLERPLACE_AREA
CNT_PAYMENT  DAYS_FIRST_DRAWING  DAYS_FIRST_DUE
DAYS_LAST_DUE_1ST_VERSION  DAYS_LAST_DUE  DAYS_TERMINATION
NFLAG_INSURED_ON_APPROVAL
count  1.048575e+06  1.048575e+06  815566.000000     1.048575e+06
1.048575e+06      4.891790e+05     8.076100e+05
1.048575e+06            1.048575e+06      489179.000000
3721.000000              3721.000000   1.048575e+06      1.048575e+06
815569.000000       627867.000000   627867.000000
627867.000000  627867.000000     627867.000000
627867.000000
mean   1.922775e+06  2.784367e+05   15891.265151     1.742698e+05
```

```
1.950000e+05      6.700778e+03     2.262892e+05
1.248486e+01            9.964123e-01           0.079619
0.187177                  0.774922  -8.820381e+02      3.183904e+02
15.995639       342387.346201    13833.802031
33614.930898   76591.061435      81985.701661
0.331530
std    5.329366e+05  1.028569e+05   14745.557438     2.910789e+05
3.169407e+05      2.078570e+04     3.134490e+05
3.333140e+00            5.979011e-02           0.107882
0.083343                  0.099514   7.792649e+02      7.996734e+03
14.508109        88595.441587    72460.126454
106643.960780  149653.053854     153298.887247
0.470764
min    1.000001e+06  1.000010e+05       0.000000     0.000000e+00
0.000000e+00     -9.000000e-01     0.000000e+00
0.000000e+00            0.000000e+00          -0.000014
0.034781                  0.373150  -2.922000e+03     -1.000000e+00
0.000000        -2921.000000    -2892.000000                -
2801.000000   -2889.000000      -2874.000000
0.000000
25%    1.460642e+06  1.893860e+05    6301.350000     1.890000e+04
2.427750e+04      0.000000e+00     5.058000e+04
1.000000e+01            1.000000e+00           0.000000
0.160716                  0.715645  -1.303000e+03     -1.000000e+00
6.000000       365243.000000    -1626.000000                -
1241.000000   -1313.000000      -1269.000000
0.000000
50%    1.923419e+06  2.788100e+05   11250.000000     7.081650e+04
8.025300e+04      1.624500e+03     1.115116e+05
1.200000e+01            1.000000e+00           0.051062
0.189122                  0.835095  -5.830000e+02      4.000000e+00
12.000000       365243.000000     -830.000000                -
361.000000    -537.000000       -498.000000                  0.000000
75%    2.384448e+06  3.677445e+05   20523.003750     1.800000e+05
2.152395e+05      7.749000e+03     2.295000e+05
1.500000e+01            1.000000e+00           0.108909
0.193330                  0.852537  -2.810000e+02      8.500000e+01
24.000000       365243.000000     -410.000000
128.000000     -74.000000        -44.000000                  1.000000
max    2.845382e+06  4.562550e+05  418058.145000     6.905160e+06
6.905160e+06      2.150100e+06     6.905160e+06
2.300000e+01            1.000000e+00           0.989740
1.000000                  1.000000  -2.000000e+00      4.000000e+06
84.000000       365243.000000   365243.000000
365243.000000  365243.000000     365243.000000
1.000000
```

Data Cleaning & Manipulation

Null Value Calculation

```
import missingno as mn
mn.matrix(applicationDF)

<Axes: >
```



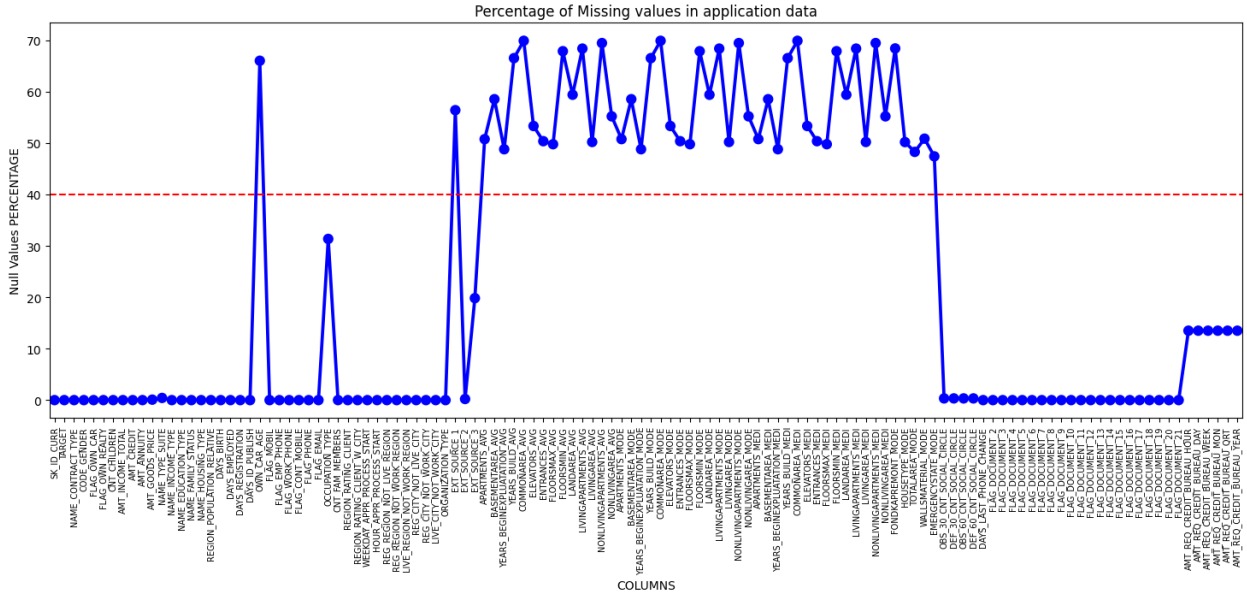

```
# % null value in each column
round(applicationDF.isnull().sum() / applicationDF.shape[0] *
100.00,2)

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
DAYS_REGISTRATION               0.00
DAYS_ID_PUBLISH                 0.00
```

```
OWN_CAR_AGE                   65.99
FLAG_MOBIL                     0.00
FLAG_EMP_PHONE                 0.00
FLAG_WORK_PHONE                0.00
FLAG_CONT_MOBILE               0.00
FLAG_PHONE                     0.00
FLAG_EMAIL                     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
EXT_SOURCE_1                  56.38
EXT_SOURCE_2                   0.21
EXT_SOURCE_3                  19.83
APARTMENTS_AVG                50.75
BASEMENTAREA_AVG              58.52
YEARS_BEGINEXPLUATATION_AVG   48.78
YEARS_BUILD_AVG               66.50
COMMONAREA_AVG                69.87
ELEVATORS_AVG                 53.30
ENTRANCES_AVG                 50.35
FLOORSMAX_AVG                 49.76
FLOORSMIN_AVG                 67.85
LANDAREA_AVG                  59.38
LIVINGAPARTMENTS_AVG          68.35
LIVINGAREA_AVG                50.19
NONLIVINGAPARTMENTS_AVG       69.43
NONLIVINGAREA_AVG             55.18
APARTMENTS_MODE               50.75
BASEMENTAREA_MODE             58.52
YEARS_BEGINEXPLUATATION_MODE  48.78
YEARS_BUILD_MODE              66.50
COMMONAREA_MODE               69.87
ELEVATORS_MODE                53.30
ENTRANCES_MODE                50.35
FLOORSMAX_MODE                49.76
FLOORSMIN_MODE                67.85
LANDAREA_MODE                 59.38
LIVINGAPARTMENTS_MODE         68.35
LIVINGAREA_MODE               50.19
```

```
NONLIVINGAPARTMENTS_MODE         69.43
NONLIVINGAREA_MODE               55.18
APARTMENTS_MEDI                  50.75
BASEMENTAREA_MEDI                58.52
YEARS_BEGINEXPLUATATION_MEDI     48.78
YEARS_BUILD_MEDI                 66.50
COMMONAREA_MEDI                  69.87
ELEVATORS_MEDI                   53.30
ENTRANCES_MEDI                   50.35
FLOORSMAX_MEDI                   49.76
FLOORSMIN_MEDI                   67.85
LANDAREA_MEDI                    59.38
LIVINGAPARTMENTS_MEDI            68.35
LIVINGAREA_MEDI                  50.19
NONLIVINGAPARTMENTS_MEDI         69.43
NONLIVINGAREA_MEDI               55.18
FONDKAPREMONT_MODE               68.39
HOUSETYPE_MODE                   50.18
TOTALAREA_MODE                   48.27
WALLSMATERIAL_MODE               50.84
EMERGENCYSTATE_MODE              47.40
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_2                   0.00
FLAG_DOCUMENT_3                   0.00
FLAG_DOCUMENT_4                   0.00
FLAG_DOCUMENT_5                   0.00
FLAG_DOCUMENT_6                   0.00
FLAG_DOCUMENT_7                   0.00
FLAG_DOCUMENT_8                   0.00
FLAG_DOCUMENT_9                   0.00
FLAG_DOCUMENT_10                  0.00
FLAG_DOCUMENT_11                  0.00
FLAG_DOCUMENT_12                  0.00
FLAG_DOCUMENT_13                  0.00
FLAG_DOCUMENT_14                  0.00
FLAG_DOCUMENT_15                  0.00
FLAG_DOCUMENT_16                  0.00
FLAG_DOCUMENT_17                  0.00
FLAG_DOCUMENT_18                  0.00
FLAG_DOCUMENT_19                  0.00
FLAG_DOCUMENT_20                  0.00
FLAG_DOCUMENT_21                  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
dtype: float64
```

```
null_applicationDF =
pd.DataFrame((applicationDF.isnull().sum())*100/applicationDF.shape[0]
).reset_index()
null_applicationDF.columns = ['Column Name', 'Null Values Percentage']
fig = plt.figure(figsize=(18,6))
ax = sns.pointplot(x="Column Name",y="Null Values
Percentage",data=null_applicationDF,color='blue')
plt.xticks(rotation =90,fontsize =7)
ax.axhline(40, ls='--',color='red')
plt.title("Percentage of Missing values in application data")
plt.ylabel("Null Values PERCENTAGE")
plt.xlabel("COLUMNS")
plt.show()
```




```
# more than or equal to 40% empty rows columns
nullcol_40_application = null_applicationDF[null_applicationDF["Null
Values Percentage"]>=40]
nullcol_40_application
```

	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
```

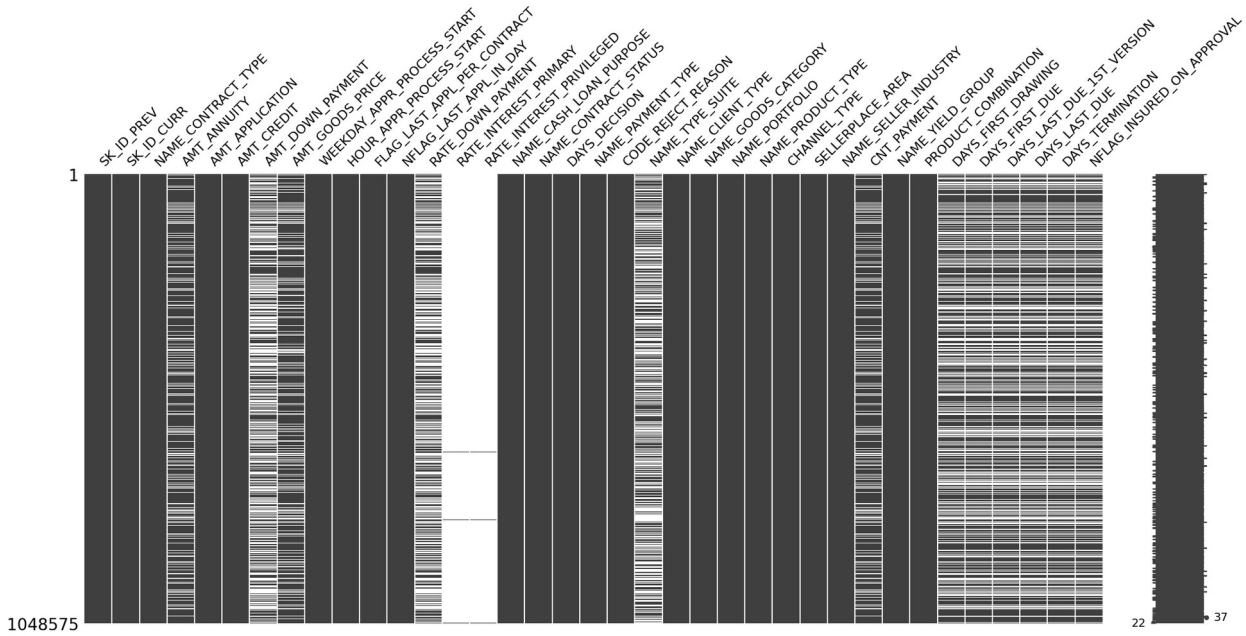
```
# How many columns have more than or euqal to 40% null values ?
len(nullcol_40_application)
```

```
49
```

previousDF Missing Values

```
mn.matrix(previousDF)
```

```
<Axes: >
```

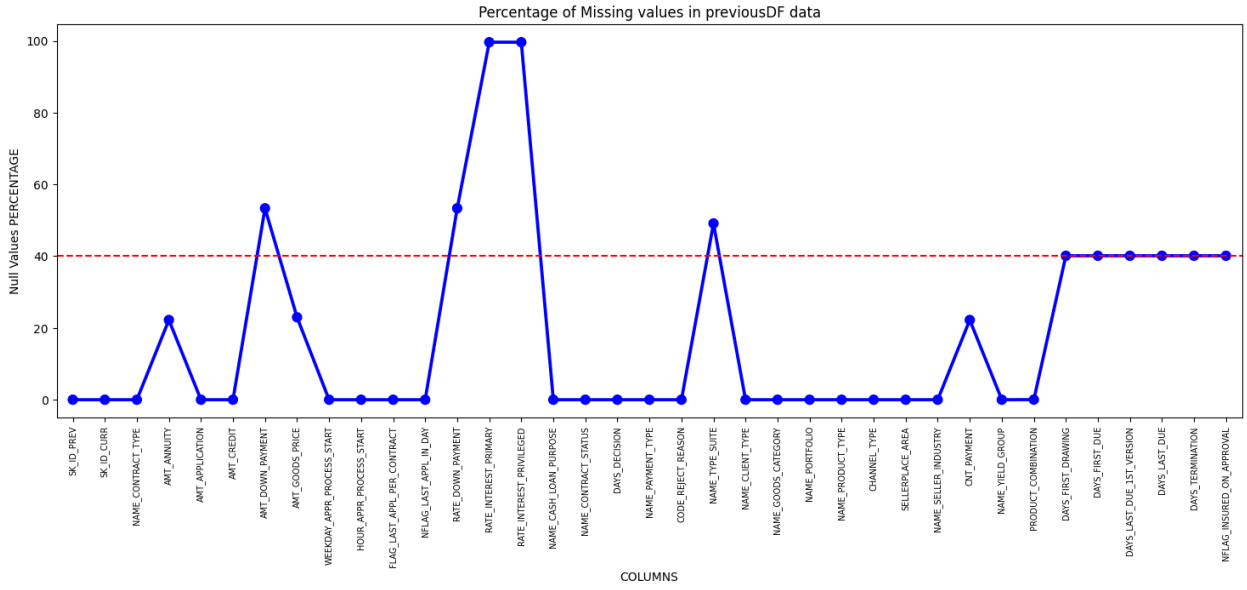



```
# checking the null value % of each column in previousDF dataframe
round(previousDF.isnull().sum() / previousDF.shape[0] * 100.00,2)
```

```
SK_ID_PREV                     0.00
SK_ID_CURR                     0.00
NAME_CONTRACT_TYPE             0.00
AMT_ANNUITY                   22.22
AMT_APPLICATION                0.00
AMT_CREDIT                     0.00
AMT_DOWN_PAYMENT              53.35
AMT_GOODS_PRICE               22.98
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             53.35
RATE_INTEREST_PRIMARY         99.65
RATE_INTEREST_PRIVILEGED      99.65
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               49.13
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                   22.22
NAME_YIELD_GROUP               0.00
PRODUCT_COMBINATION            0.02
DAYS_FIRST_DRAWING            40.12
DAYS_FIRST_DUE                40.12
DAYS_LAST_DUE_1ST_VERSION     40.12
DAYS_LAST_DUE                 40.12
DAYS_TERMINATION              40.12
NFLAG_INSURED_ON_APPROVAL     40.12
dtype: float64
```

```
null_previousDF =
pd.DataFrame((previousDF.isnull().sum())*100/previousDF.shape[0]).rese
t_index()
null_previousDF.columns = ['Column Name', 'Null Values Percentage']
fig = plt.figure(figsize=(18,6))
ax = sns.pointplot(x="Column Name",y="Null Values
Percentage",data=null_previousDF,color ='blue')
plt.xticks(rotation =90,fontsize =7)
ax.axhline(40, ls='--',color='red')
plt.title("Percentage of Missing values in previousDF data")
plt.ylabel("Null Values PERCENTAGE")
plt.xlabel("COLUMNS")
plt.show()
```



```
# more than or equal to 40% empty rows columns
nullcol_40_previous = null_previousDF[null_previousDF["Null Values
Percentage"]>=40]
nullcol_40_previous
```

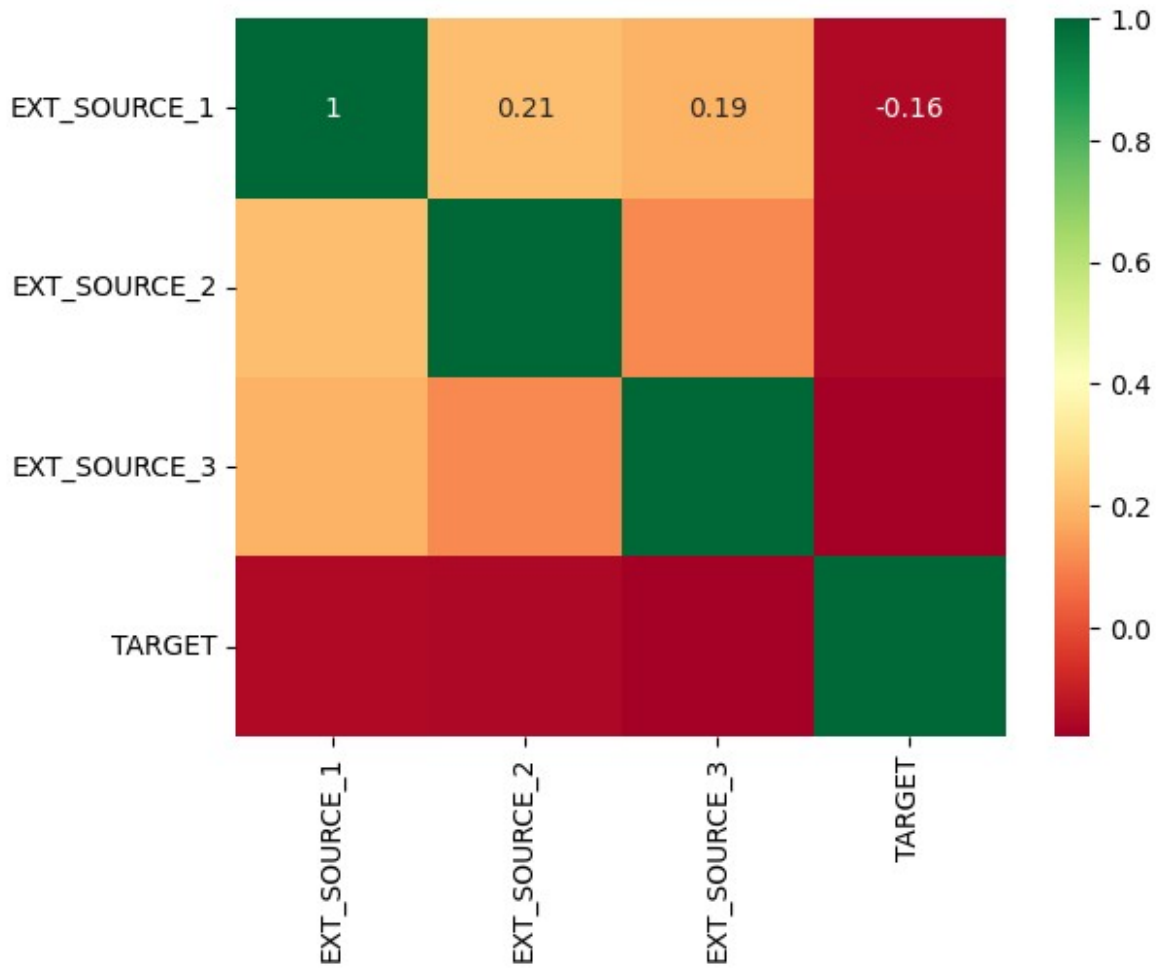
	Column Name	Null Values Percentage
6	AMT_DOWN_PAYMENT	53.348211
12	RATE_DOWN_PAYMENT	53.348211
13	RATE_INTEREST_PRIMARY	99.645137
14	RATE_INTEREST_PRIVILEGED	99.645137
20	NAME_TYPE_SUITE	49.127626
31	DAYS_FIRST_DRAWING	40.121880
32	DAYS_FIRST_DUE	40.121880
33	DAYS_LAST_DUE_1ST_VERSION	40.121880
34	DAYS_LAST_DUE	40.121880
35	DAYS_TERMINATION	40.121880
36	NFLAG_INSURED_ON_APPROVAL	40.121880

```
# How many columns have more than or euqal to 40% null values ?
len(nullcol_40_previous)
```

```
11
```

Analyze & Delete Unnecessary Columns in applicationDF

```
# Checking correlation of EXT_SOURCE_X columns vs TARGET column
Source =
applicationDF[["EXT_SOURCE_1","EXT_SOURCE_2","EXT_SOURCE_3","TARGET"]]
source_corr = Source.corr()
ax = sns.heatmap(source_corr,
            xticklabels=source_corr.columns,
            yticklabels=source_corr.columns,
            annot = True,
            cmap ="RdYlGn")
```




```
# create a list of columns that needs to be dropped including the
columns with >40% null values
Unwanted_application = nullcol_40_application["Column Name"].tolist()+
['EXT_SOURCE_2','EXT_SOURCE_3']
# as EXT_SOURCE_1 column is already included in nullcol_40_application

len(Unwanted_application)
```

```
51
```

Flag Document

```
import itertools
import matplotlib.pyplot as plt
import seaborn as sns

# Assuming `applicationDF` is your dataframe
col_Doc = ['FLAG_DOCUMENT_2', 'FLAG_DOCUMENT_3', 'FLAG_DOCUMENT_4',
'FLAG_DOCUMENT_5', 'FLAG_DOCUMENT_6',
           'FLAG_DOCUMENT_7', 'FLAG_DOCUMENT_8', 'FLAG_DOCUMENT_9',
```

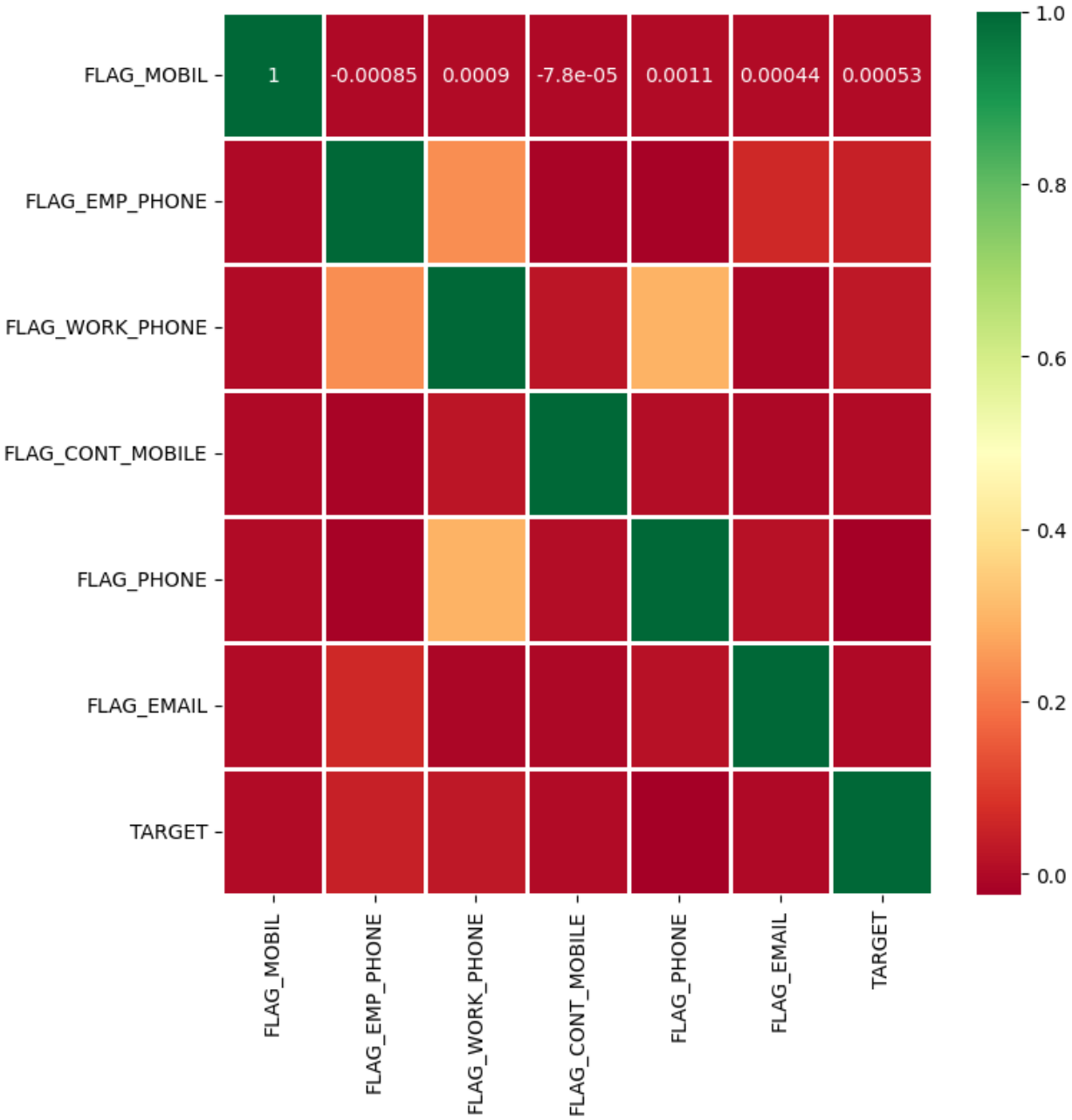
```
'FLAG_DOCUMENT_10', 'FLAG_DOCUMENT_11',
           'FLAG_DOCUMENT_12', 'FLAG_DOCUMENT_13', 'FLAG_DOCUMENT_14',
'FLAG_DOCUMENT_15', 'FLAG_DOCUMENT_16',
           'FLAG_DOCUMENT_17', 'FLAG_DOCUMENT_18', 'FLAG_DOCUMENT_19',
'FLAG_DOCUMENT_20', 'FLAG_DOCUMENT_21']
df_flag = applicationDF[col_Doc + ["TARGET"]]

length = len(col_Doc)

df_flag["TARGET"] = df_flag["TARGET"].replace({1: "Defaulter", 0:
"Repayer"})

fig = plt.figure(figsize=(21, 24))

for i, j in itertools.zip_longest(col_Doc, range(length)):
    plt.subplot(5, 4, j + 1)
    ax = sns.countplot(x=df_flag[i], hue=df_flag["TARGET"],
palette=["r", "g"])
    plt.yticks(fontsize=8)
    plt.xlabel("")
    plt.ylabel("")
    plt.title(i)

plt.tight_layout()
plt.show()
```


```
# Including the flag documents for dropping the Document columns
col_Doc.remove('FLAG_DOCUMENT_3')
Unwanted_application = Unwanted_application + col_Doc
len(Unwanted_application)
```

```
70
```

Contact Parameters

```
# checking is there is any correlation between mobile phone, work
phone etc, email, Family members and Region rating
contact_col = ['FLAG_MOBIL', 'FLAG_EMP_PHONE', 'FLAG_WORK_PHONE',
'FLAG_CONT_MOBILE',
       'FLAG_PHONE', 'FLAG_EMAIL','TARGET']
Contact_corr = applicationDF[contact_col].corr()
fig = plt.figure(figsize=(8,8))
ax = sns.heatmap(Contact_corr,
            xticklabels=Contact_corr.columns,
            yticklabels=Contact_corr.columns,
            annot = True,
            cmap ="RdYlGn",
            linewidth=1)
```




```
# including the 6 FLAG columns to be deleted
contact_col.remove('TARGET')
Unwanted_application = Unwanted_application + contact_col
len(Unwanted_application)
```

```
76
```

```
# Dropping the unnecessary columns from applicationDF
applicationDF.drop(labels=Unwanted_application, axis=1, inplace=True,
errors='ignore')
```

```
# Inspecting the dataframe after removal of unnecessary columns
applicationDF.shape
```

```
(307511, 46)
```

```
# 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
```

Analyze & Delete Unnecessary Columns in previousDF

```
# Getting the 11 columns which has more than 40% unknown
Unwanted_previous = nullcol_40_previous["Column Name"].tolist()
Unwanted_previous
```

```
['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']
```

```
# Listing down columns which are not needed
Unnecessary_previous =
['WEEKDAY_APPR_PROCESS_START','HOUR_APPR_PROCESS_START',

'FLAG_LAST_APPL_PER_CONTRACT','NFLAG_LAST_APPL_IN_DAY']

Unwanted_previous = Unwanted_previous + Unnecessary_previous
len(Unwanted_previous)
```

```
15
```

```
# Dropping the unnecessary columns from previous
previousDF.drop(labels=Unwanted_previous,axis=1,inplace=True)
# Inspecting the dataframe after removal of unnecessary columns
previousDF.shape
```

```
(1048575, 22)
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1048575 entries, 0 to 1048574
Data columns (total 22 columns):
 #   Column                  Non-Null Count    Dtype
---  ------                  --------------    -----
 0   SK_ID_PREV              1048575 non-null  int64
 1   SK_ID_CURR              1048575 non-null  int64
 2   NAME_CONTRACT_TYPE      1048575 non-null  object
 3   AMT_ANNUITY             815566 non-null   float64
 4   AMT_APPLICATION         1048575 non-null  float64
 5   AMT_CREDIT              1048575 non-null  float64
 6   AMT_GOODS_PRICE         807610 non-null   float64
 7   NAME_CASH_LOAN_PURPOSE  1048575 non-null  object
 8   NAME_CONTRACT_STATUS    1048575 non-null  object
 9   DAYS_DECISION           1048575 non-null  int64
 10  NAME_PAYMENT_TYPE       1048575 non-null  object
 11  CODE_REJECT_REASON      1048575 non-null  object
 12  NAME_CLIENT_TYPE        1048575 non-null  object
 13  NAME_GOODS_CATEGORY     1048575 non-null  object
 14  NAME_PORTFOLIO          1048575 non-null  object
 15  NAME_PRODUCT_TYPE       1048575 non-null  object
 16  CHANNEL_TYPE            1048575 non-null  object
 17  SELLERPLACE_AREA        1048575 non-null  int64
 18  NAME_SELLER_INDUSTRY    1048575 non-null  object
 19  CNT_PAYMENT             815569 non-null   float64
 20  NAME_YIELD_GROUP        1048575 non-null  object
 21  PRODUCT_COMBINATION     1048351 non-null  object
dtypes: float64(5), int64(4), object(13)
memory usage: 176.0+ MB
```

Standardize Values

```
# Converting Negative days to positive days

date_col =
['DAYS_BIRTH','DAYS_EMPLOYED','DAYS_REGISTRATION','DAYS_ID_PUBLISH']

for col in date_col:
    applicationDF[col] = abs(applicationDF[col])

# Binning Numerical Columns to create a categorical column

# Creating bins for income amount
applicationDF['AMT_INCOME_TOTAL']=applicationDF['AMT_INCOME_TOTAL']/
100000

bins = [0,1,2,3,4,5,6,7,8,9,10,11]
slot = ['0-100K','100K-200K', '200k-300k','300k-400k','400k-
500k','500k-600k','600k-700k','700k-800k','800k-900k','900k-1M', '1M
Above']
```

```
applicationDF['AMT_INCOME_RANGE']=pd.cut(applicationDF['AMT_INCOME_TOT
AL'],bins,labels=slot)

applicationDF['AMT_INCOME_RANGE'].value_counts(normalize=True)*100
```

```
AMT_INCOME_RANGE
100K-200K    50.735000
200k-300k    21.210691
0-100K       20.729695
300k-400k     4.776116
400k-500k     1.744669
500k-600k     0.356354
600k-700k     0.282805
800k-900k     0.096980
700k-800k     0.052721
900k-1M       0.009112
1M Above      0.005858
Name: proportion, dtype: float64
```

```
# Creating bins for Credit amount
applicationDF['AMT_CREDIT']=applicationDF['AMT_CREDIT']/100000

bins = [0,1,2,3,4,5,6,7,8,9,10,100]
slots = ['0-100K','100K-200K', '200k-300k','300k-400k','400k-
500k','500k-600k','600k-700k','700k-800k',
       '800k-900k','900k-1M', '1M Above']

applicationDF['AMT_CREDIT_RANGE']=pd.cut(applicationDF['AMT_CREDIT'],b
ins=bins,labels=slots)

#checking the binning of data and % of data in each category
applicationDF['AMT_CREDIT_RANGE'].value_counts(normalize=True)*100
```

```
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     7.086576
700k-800k     6.241403
900k-1M       2.902986
0-100K        1.952450
Name: proportion, dtype: float64
```

```
# Creating bins for Age
applicationDF['AGE'] = applicationDF['DAYS_BIRTH'] // 365
bins = [0,20,30,40,50,100]
```

```
slots = ['0-20','20-30','30-40','40-50','50 above']

applicationDF['AGE_GROUP']=pd.cut(applicationDF['AGE'],bins=bins,label
s=slots)

#checking the binning of data and % of data in each category
applicationDF['AGE_GROUP'].value_counts(normalize=True)*100
```

```
AGE_GROUP
50 above    31.604398
30-40       27.028952
40-50       24.194582
20-30       17.171743
0-20         0.000325
Name: proportion, dtype: float64
```

```
#Creating bins for Employement Time
applicationDF['YEARS_EMPLOYED'] = applicationDF['DAYS_EMPLOYED'] //
365
bins = [0,5,10,20,30,40,50,60,150]
slots = ['0-5','5-10','10-20','20-30','30-40','40-50','50-60','60
above']

applicationDF['EMPLOYMENT_YEAR']=pd.cut(applicationDF['YEARS_EMPLOYED'
],bins=bins,labels=slots)

#checking the binning of data and % of data in each category
applicationDF['EMPLOYMENT_YEAR'].value_counts(normalize=True)*100
```

```
EMPLOYMENT_YEAR
0-5         55.582363
5-10        24.966441
10-20       14.564315
20-30        3.750117
30-40        1.058720
40-50        0.078044
50-60        0.000000
60 above     0.000000
Name: proportion, dtype: float64
```

```
#Checking the number of unique values each column possess to identify
categorical columns
applicationDF.nunique().sort_values()
```

```
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
```

Data Type Conversion

```
# inspecting the column types if they are in correct data type using
the above result.
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  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
 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(4), float64(18), int64(18), object(12)
memory usage: 113.8+ MB
```

```
#Conversion of Object and Numerical columns to Categorical Columns
categorical_columns =
['NAME_CONTRACT_TYPE','CODE_GENDER','NAME_TYPE_SUITE','NAME_INCOME_TYP
E','NAME_EDUCATION_TYPE',

'NAME_FAMILY_STATUS','NAME_HOUSING_TYPE','OCCUPATION_TYPE','WEEKDAY_AP
PR_PROCESS_START',

'ORGANIZATION_TYPE','FLAG_OWN_CAR','FLAG_OWN_REALTY','LIVE_CITY_NOT_WO
RK_CITY',

'REG_CITY_NOT_LIVE_CITY','REG_CITY_NOT_WORK_CITY','REG_REGION_NOT_WORK
_REGION',

'LIVE_REGION_NOT_WORK_REGION','REGION_RATING_CLIENT','WEEKDAY_APPR_PRO
CESS_START',
                       'REGION_RATING_CLIENT_W_CITY'
                      ]
for col in categorical_columns:
    applicationDF[col] =pd.Categorical(applicationDF[col])

# 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
```

Standardize Values for previousDF

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

```
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                    48
SELLERPLACE_AREA             2023
DAYS_DECISION                2921
AMT_CREDIT                  74637
AMT_GOODS_PRICE             75635
AMT_APPLICATION             75635
AMT_ANNUITY                282291
SK_ID_CURR                 305828
SK_ID_PREV                1048575
dtype: int64
```

```
# inspecting the column types if the above conversion is reflected
previousDF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1048575 entries, 0 to 1048574
Data columns (total 22 columns):
 #   Column                  Non-Null Count    Dtype
---  ------                  --------------    -----
 0   SK_ID_PREV              1048575 non-null  int64
 1   SK_ID_CURR              1048575 non-null  int64
 2   NAME_CONTRACT_TYPE      1048575 non-null  object
 3   AMT_ANNUITY             815566 non-null   float64
 4   AMT_APPLICATION         1048575 non-null  float64
 5   AMT_CREDIT              1048575 non-null  float64
 6   AMT_GOODS_PRICE         807610 non-null   float64
 7   NAME_CASH_LOAN_PURPOSE  1048575 non-null  object
 8   NAME_CONTRACT_STATUS    1048575 non-null  object
 9   DAYS_DECISION           1048575 non-null  int64
 10  NAME_PAYMENT_TYPE       1048575 non-null  object
 11  CODE_REJECT_REASON      1048575 non-null  object
 12  NAME_CLIENT_TYPE        1048575 non-null  object
 13  NAME_GOODS_CATEGORY     1048575 non-null  object
```

```
 14  NAME_PORTFOLIO           1048575 non-null  object
 15  NAME_PRODUCT_TYPE        1048575 non-null  object
 16  CHANNEL_TYPE             1048575 non-null  object
 17  SELLERPLACE_AREA         1048575 non-null  int64
 18  NAME_SELLER_INDUSTRY     1048575 non-null  object
 19  CNT_PAYMENT              815569 non-null   float64
 20  NAME_YIELD_GROUP         1048575 non-null  object
 21  PRODUCT_COMBINATION      1048351 non-null  object
dtypes: float64(5), int64(4), object(13)
memory usage: 176.0+ MB
```

```
#Converting negative days to positive days
previousDF['DAYS_DECISION'] = abs(previousDF['DAYS_DECISION'])

#age group calculation e.g. 388 will be grouped as 300-400
previousDF['DAYS_DECISION_GROUP'] = (previousDF['DAYS_DECISION']-
(previousDF['DAYS_DECISION'] % 400)).astype(str)+'-'+
((previousDF['DAYS_DECISION'] - (previousDF['DAYS_DECISION'] % 400)) +
(previousDF['DAYS_DECISION'] % 400) + (400 -
(previousDF['DAYS_DECISION'] % 400))).astype(str)

previousDF['DAYS_DECISION_GROUP'].value_counts(normalize=True)*100
```

```
DAYS_DECISION_GROUP
0-400        37.392747
400-800      22.969840
800-1200     12.444508
1200-1600     7.932766
2400-2800     6.323820
1600-2000     5.829435
2000-2400     5.672556
2800-3200     1.434328
Name: proportion, dtype: float64
```

```
#Converting Categorical columns from Object to categorical
Catgorical_col_p =
['NAME_CASH_LOAN_PURPOSE','NAME_CONTRACT_STATUS','NAME_PAYMENT_TYPE',

'CODE_REJECT_REASON','NAME_CLIENT_TYPE','NAME_GOODS_CATEGORY','NAME_PO
RTFOLIO',

'NAME_PRODUCT_TYPE','CHANNEL_TYPE','NAME_SELLER_INDUSTRY','NAME_YIELD_
GROUP','PRODUCT_COMBINATION',
                    'NAME_CONTRACT_TYPE','DAYS_DECISION_GROUP']

for col in Catgorical_col_p:
    previousDF[col] =pd.Categorical(previousDF[col])

# inspecting the column types after conversion
previousDF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1048575 entries, 0 to 1048574
Data columns (total 23 columns):
 #   Column                  Non-Null Count    Dtype
---  ------                  --------------    -----
 0   SK_ID_PREV              1048575 non-null  int64
 1   SK_ID_CURR              1048575 non-null  int64
 2   NAME_CONTRACT_TYPE      1048575 non-null  category
 3   AMT_ANNUITY             815566 non-null   float64
 4   AMT_APPLICATION         1048575 non-null  float64
 5   AMT_CREDIT              1048575 non-null  float64
 6   AMT_GOODS_PRICE         807610 non-null   float64
 7   NAME_CASH_LOAN_PURPOSE  1048575 non-null  category
 8   NAME_CONTRACT_STATUS    1048575 non-null  category
 9   DAYS_DECISION           1048575 non-null  int64
 10  NAME_PAYMENT_TYPE       1048575 non-null  category
 11  CODE_REJECT_REASON      1048575 non-null  category
 12  NAME_CLIENT_TYPE        1048575 non-null  category
 13  NAME_GOODS_CATEGORY     1048575 non-null  category
 14  NAME_PORTFOLIO          1048575 non-null  category
 15  NAME_PRODUCT_TYPE       1048575 non-null  category
 16  CHANNEL_TYPE            1048575 non-null  category
 17  SELLERPLACE_AREA        1048575 non-null  int64
 18  NAME_SELLER_INDUSTRY    1048575 non-null  category
 19  CNT_PAYMENT             815569 non-null   float64
 20  NAME_YIELD_GROUP        1048575 non-null  category
 21  PRODUCT_COMBINATION     1048351 non-null  category
 22  DAYS_DECISION_GROUP     1048575 non-null  category
dtypes: category(14), float64(5), int64(4)
memory usage: 86.0 MB
```

Null Value Data Imputation

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

```
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
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                27.08
dtype: float64
```

```
applicationDF['NAME_TYPE_SUITE'].describe()
```

```
count            306219
unique                7
top       Unaccompanied
freq             248526
Name: NAME_TYPE_SUITE, dtype: object
```

```
applicationDF['NAME_TYPE_SUITE'].fillna((applicationDF['NAME_TYPE_SUIT
E'].mode()[0]),inplace = True)

applicationDF['OCCUPATION_TYPE'] =
applicationDF['OCCUPATION_TYPE'].cat.add_categories('Unknown')
applicationDF['OCCUPATION_TYPE'].fillna('Unknown', inplace =True)

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()
```

```
       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
count               265992.000000              265992.000000
265992.000000               265992.000000              265992.000000
265992.000000
mean                     0.006402                   0.007000
0.034362                   0.267395                   0.265474
1.899974
std                      0.083849                   0.110757
0.204685                   0.916002                   0.794056
1.869295
min                      0.000000                   0.000000
0.000000                   0.000000                   0.000000
0.000000
25%                      0.000000                   0.000000
0.000000                   0.000000                   0.000000
0.000000
50%                      0.000000                   0.000000
0.000000                   0.000000                   0.000000
1.000000
75%                      0.000000                   0.000000
0.000000                   0.000000                   0.000000
3.000000
max                      4.000000                   9.000000
8.000000                  27.000000                 261.000000
25.000000
```

```
amount = ['AMT_REQ_CREDIT_BUREAU_HOUR',
'AMT_REQ_CREDIT_BUREAU_DAY','AMT_REQ_CREDIT_BUREAU_WEEK','AMT_REQ_CRED
IT_BUREAU_MON',
         'AMT_REQ_CREDIT_BUREAU_QRT','AMT_REQ_CREDIT_BUREAU_YEAR']

for col in amount:
    applicationDF[col].fillna(applicationDF[col].median(),inplace =
True)
```

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

```
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                0.03
NAME_TYPE_SUITE                0.00
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                0.00
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       0.10
DEF_30_CNT_SOCIAL_CIRCLE       0.10
OBS_60_CNT_SOCIAL_CIRCLE       0.10
DEF_60_CNT_SOCIAL_CIRCLE       0.10
DAYS_LAST_PHONE_CHANGE         0.00
FLAG_DOCUMENT_3                0.00
AMT_REQ_CREDIT_BUREAU_HOUR     0.00
AMT_REQ_CREDIT_BUREAU_DAY      0.00
AMT_REQ_CREDIT_BUREAU_WEEK     0.00
AMT_REQ_CREDIT_BUREAU_MON      0.00
AMT_REQ_CREDIT_BUREAU_QRT      0.00
AMT_REQ_CREDIT_BUREAU_YEAR     0.00
AMT_INCOME_RANGE               0.02
```

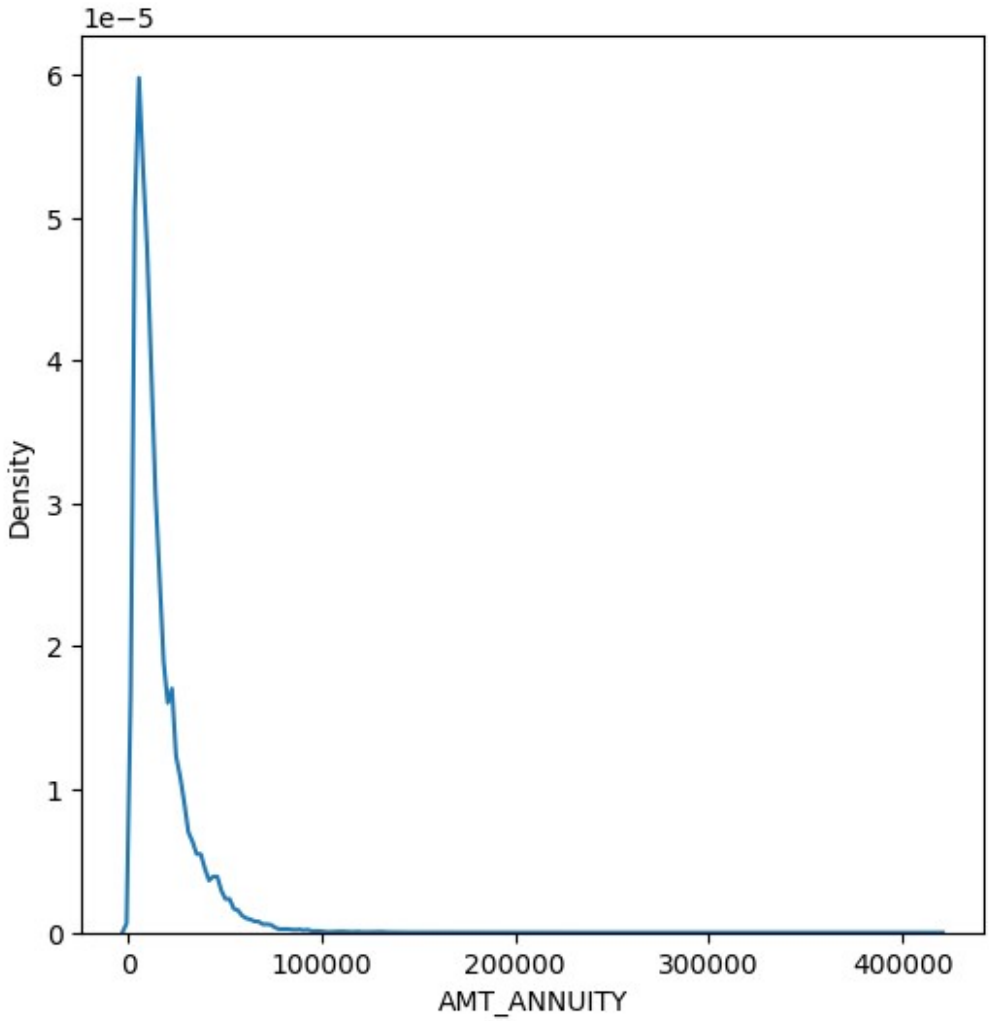
```
AMT_CREDIT_RANGE              0.00
AGE                           0.00
AGE_GROUP                     0.00
YEARS_EMPLOYED                0.00
EMPLOYMENT_YEAR               7.94
dtype: float64
```

Imputing Null Values in previousDF

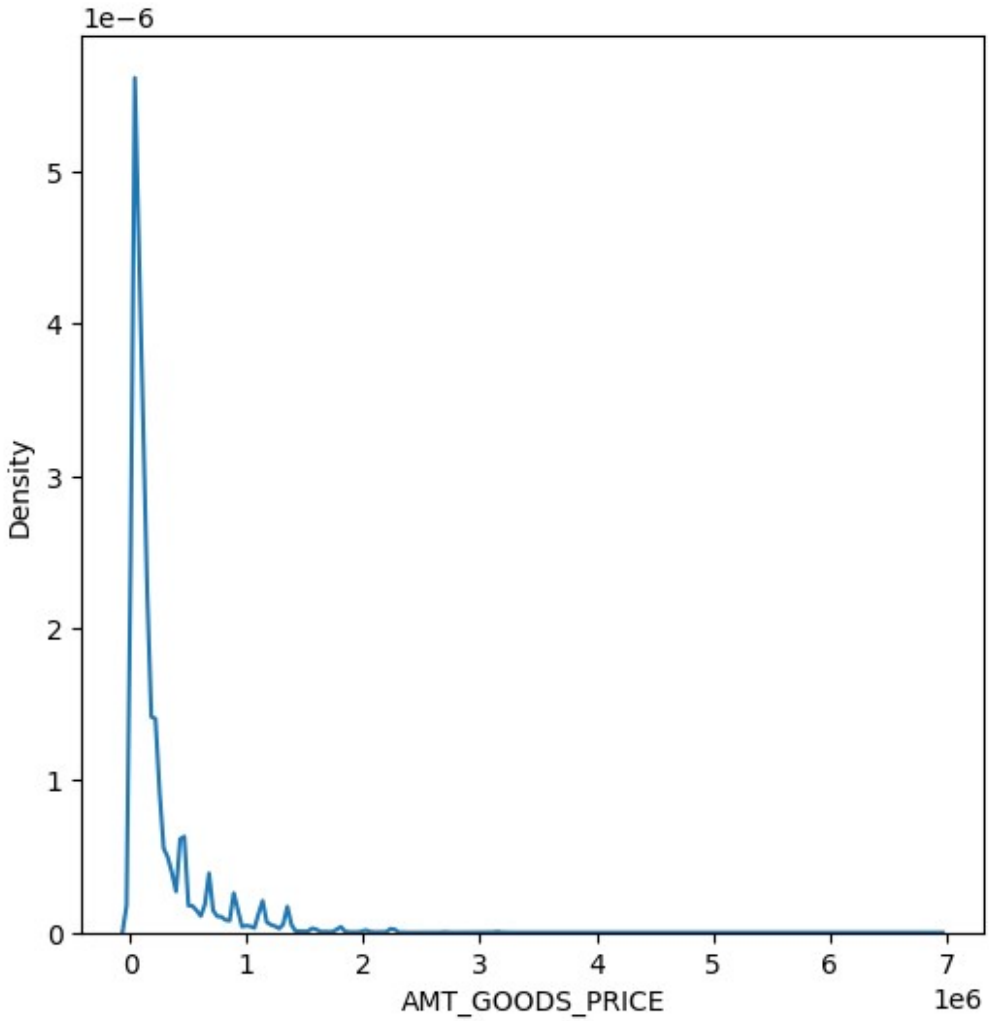
```
# checking the null value % of each column in previousDF dataframe
round(previousDF.isnull().sum() / previousDF.shape[0] * 100.00,2)
```

```
SK_ID_PREV                   0.00
SK_ID_CURR                   0.00
NAME_CONTRACT_TYPE           0.00
AMT_ANNUITY                 22.22
AMT_APPLICATION              0.00
AMT_CREDIT                   0.00
AMT_GOODS_PRICE             22.98
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                 22.22
NAME_YIELD_GROUP             0.00
PRODUCT_COMBINATION          0.02
DAYS_DECISION_GROUP          0.00
dtype: float64
```

```
plt.figure(figsize=(6,6))
sns.kdeplot(previousDF['AMT_ANNUITY'])
plt.show()
```

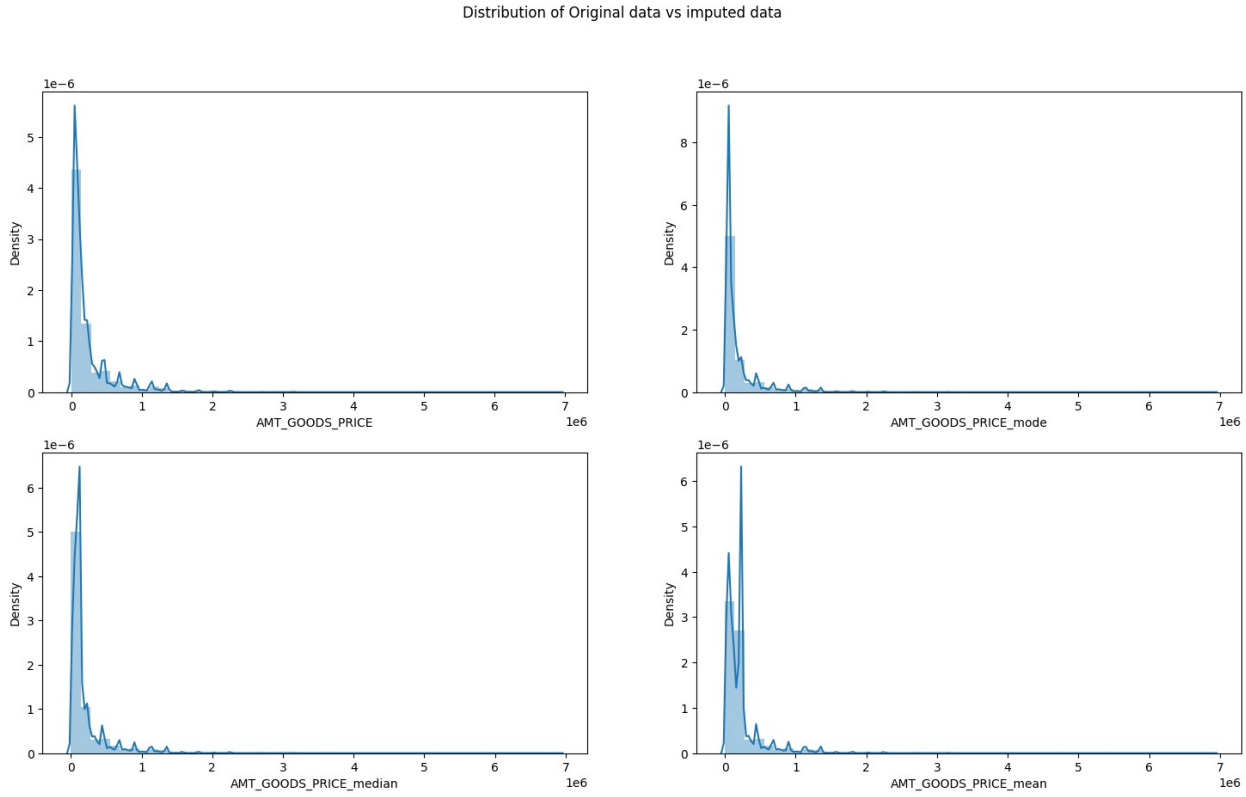



```
previousDF['AMT_ANNUITY'].fillna(previousDF['AMT_ANNUITY'].median(),in
place = True)

plt.figure(figsize=(6,6))
sns.kdeplot(previousDF['AMT_GOODS_PRICE']
[pd.notnull(previousDF['AMT_GOODS_PRICE'])])
plt.show()
```




```
statsDF = pd.DataFrame() # new dataframe with columns imputed with
mode, median and mean
statsDF['AMT_GOODS_PRICE_mode'] =
previousDF['AMT_GOODS_PRICE'].fillna(previousDF['AMT_GOODS_PRICE'].mod
e()[0])
statsDF['AMT_GOODS_PRICE_median'] =
previousDF['AMT_GOODS_PRICE'].fillna(previousDF['AMT_GOODS_PRICE'].med
ian())
statsDF['AMT_GOODS_PRICE_mean'] =
previousDF['AMT_GOODS_PRICE'].fillna(previousDF['AMT_GOODS_PRICE'].mea
n())

cols = ['AMT_GOODS_PRICE_mode',
'AMT_GOODS_PRICE_median','AMT_GOODS_PRICE_mean']

plt.figure(figsize=(18,10))
plt.suptitle('Distribution of Original data vs imputed data')
plt.subplot(221)
```

```
sns.distplot(previousDF['AMT_GOODS_PRICE']
[pd.notnull(previousDF['AMT_GOODS_PRICE'])]);
for i in enumerate(cols):
    plt.subplot(2,2,i[0]+2)
    sns.distplot(statsDF[i[1]])
```




```
previousDF['AMT_GOODS_PRICE'].fillna(previousDF['AMT_GOODS_PRICE'].mod
e()[0], inplace=True)

previousDF.loc[previousDF['CNT_PAYMENT'].isnull(),'NAME_CONTRACT_STATU
S'].value_counts()
```

```
NAME_CONTRACT_STATUS
Canceled        191156
Refused          25655
Unused offer     16192
Approved             3
Name: count, dtype: int64
```

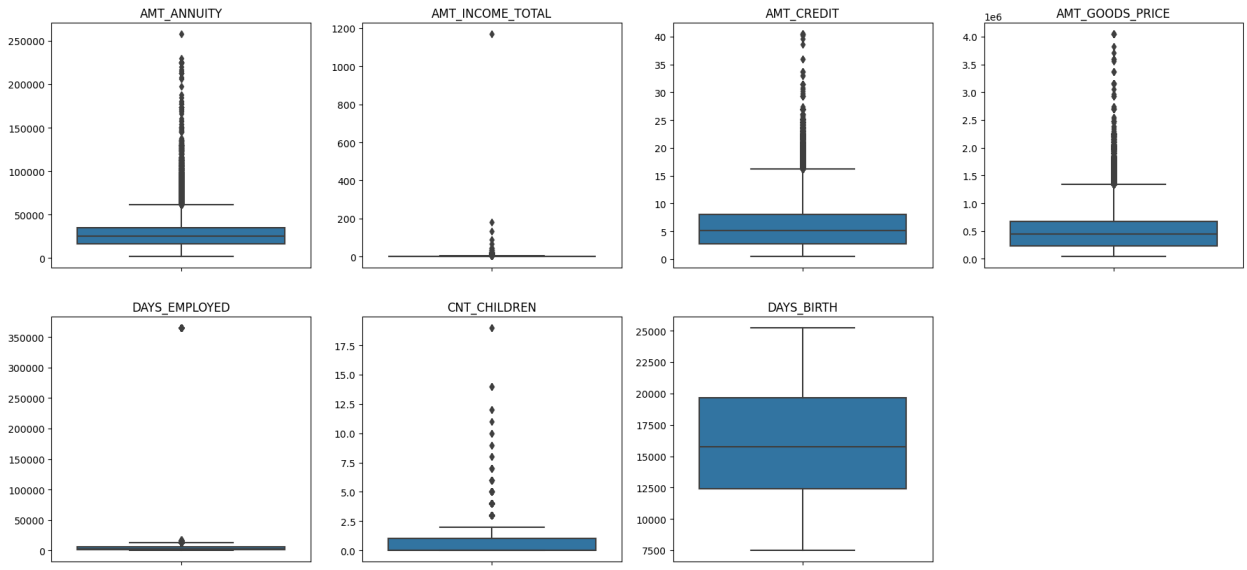
```
previousDF['CNT_PAYMENT'].fillna(0,inplace = True)

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

```
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
```

Identifying the outliers

```
plt.figure(figsize=(22,10))

app_outlier_col_1 =
['AMT_ANNUITY','AMT_INCOME_TOTAL','AMT_CREDIT','AMT_GOODS_PRICE','DAYS
_EMPLOYED']
app_outlier_col_2 = ['CNT_CHILDREN','DAYS_BIRTH']
for i in enumerate(app_outlier_col_1):
    plt.subplot(2,4,i[0]+1)
    sns.boxplot(y=applicationDF[i[1]])
    plt.title(i[1])
    plt.ylabel("")

for i in enumerate(app_outlier_col_2):
    plt.subplot(2,4,i[0]+6)
    sns.boxplot(y=applicationDF[i[1]])
    plt.title(i[1])
    plt.ylabel("")
```

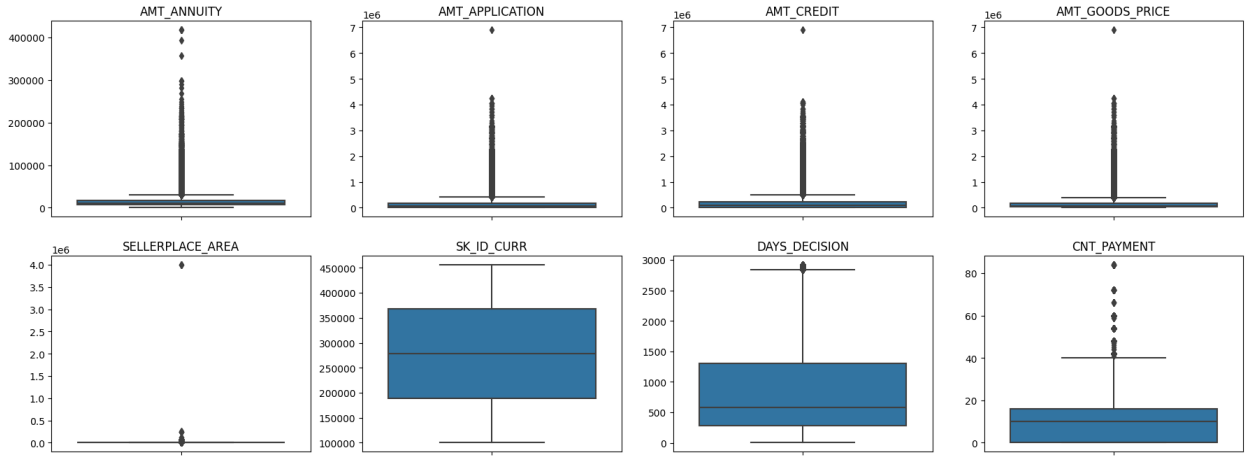



```
applicationDF[['AMT_ANNUITY', 'AMT_INCOME_TOTAL', 'AMT_CREDIT',
'AMT_GOODS_PRICE',
'DAYS_BIRTH','CNT_CHILDREN','DAYS_EMPLOYED']].describe()
```

```
         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
```

```
plt.figure(figsize=(22,8))

prev_outlier_col_1 =
['AMT_ANNUITY','AMT_APPLICATION','AMT_CREDIT','AMT_GOODS_PRICE','SELLE
RPLACE_AREA']
prev_outlier_col_2 = ['SK_ID_CURR','DAYS_DECISION','CNT_PAYMENT']
for i in enumerate(prev_outlier_col_1):
    plt.subplot(2,4,i[0]+1)
    sns.boxplot(y=previousDF[i[1]])
```

```
    plt.title(i[1])
    plt.ylabel("")

for i in enumerate(prev_outlier_col_2):
    plt.subplot(2,4,i[0]+6)
    sns.boxplot(y=previousDF[i[1]])
    plt.title(i[1])
    plt.ylabel("")
```



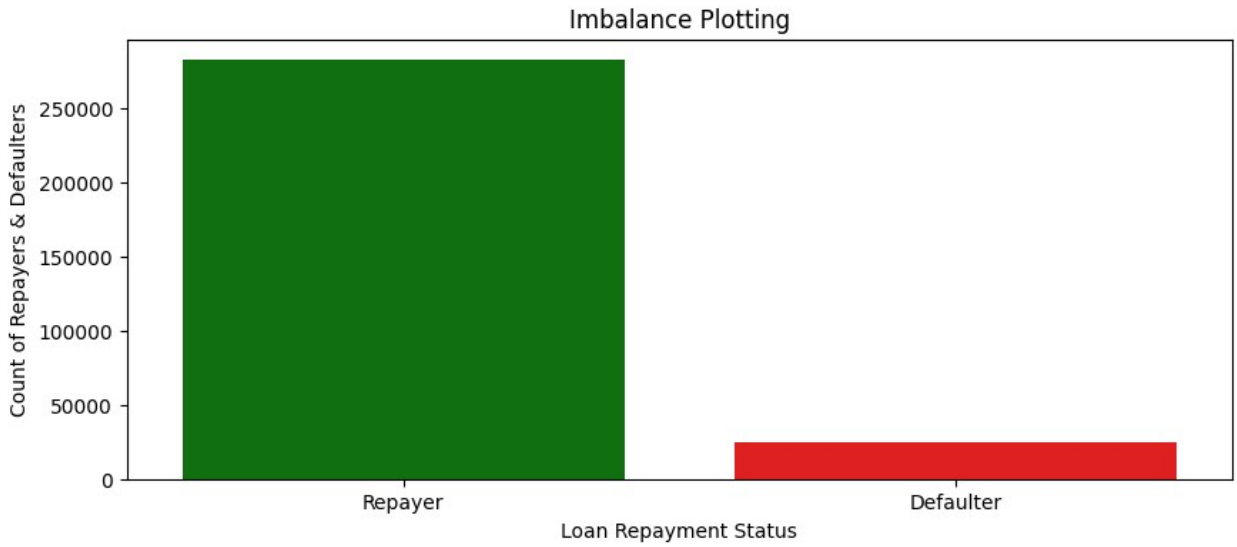

```
previousDF[['AMT_ANNUITY', 'AMT_APPLICATION', 'AMT_CREDIT',
'AMT_GOODS_PRICE',
'SELLERPLACE_AREA','CNT_PAYMENT','DAYS_DECISION']].describe()
```

```
        AMT_ANNUITY  AMT_APPLICATION    AMT_CREDIT  AMT_GOODS_PRICE
SELLERPLACE_AREA   CNT_PAYMENT  DAYS_DECISION
count  1.048575e+06     1.048575e+06  1.048575e+06     1.048575e+06
1.048575e+06  1.048575e+06   1.048575e+06
mean   1.485991e+04     1.742698e+05  1.950000e+05     1.846285e+05
3.183904e+02  1.244121e+01   8.820381e+02
std    1.314679e+04     2.910789e+05  3.169407e+05     2.854630e+05
7.996734e+03  1.441992e+01   7.792649e+02
min    0.000000e+00     0.000000e+00  0.000000e+00     0.000000e+00
-1.000000e+00  0.000000e+00   2.000000e+00
25%    7.506765e+03     1.890000e+04  2.427750e+04     4.500000e+04
-1.000000e+00  0.000000e+00   2.810000e+02
50%    1.125000e+04     7.081650e+04  8.025300e+04     7.082100e+04
4.000000e+00  1.000000e+01   5.830000e+02
75%    1.673721e+04     1.800000e+05  2.152395e+05     1.800000e+05
8.500000e+01  1.600000e+01   1.303000e+03
max    4.180581e+05     6.905160e+06  6.905160e+06     6.905160e+06
4.000000e+06  8.400000e+01   2.922000e+03
```

Data Analysis

Imbalance Analysis

```
# Calculate the imbalance
Imbalance = applicationDF["TARGET"].value_counts().reset_index()
Imbalance.columns = ['Loan Repayment Status', 'Count']
Imbalance['Loan Repayment Status'] = Imbalance['Loan Repayment
Status'].replace({0: 'Repayer', 1: 'Defaulter'})

# Plot the imbalance
plt.figure(figsize=(10, 4))
sns.barplot(x='Loan Repayment Status', y='Count', data=Imbalance,
palette=['g', 'r'])
plt.xlabel("Loan Repayment Status")
plt.ylabel("Count of Repayers & Defaulters")
plt.title("Imbalance Plotting")
plt.show()
```




```
count_0 = Imbalance.iloc[0]["Count"]
count_1 = Imbalance.iloc[1]["Count"]
count_0_perc = round(count_0 / (count_0 + count_1) * 100, 2)
count_1_perc = round(count_1 / (count_0 + count_1) * 100, 2)

print('Ratios of imbalance in percentage with respect to Repayer and
Defaulter data are: %.2f%% and %.2f%%' % (count_0_perc, count_1_perc))
print('Ratios of imbalance in relative with respect to Repayer and
Defaulter data is %.2f : 1 (approx)' % (count_0 / count_1))
```

```
Ratios of imbalance in percentage with respect to Repayer and
Defaulter data are: 91.93% and 8.07%
```

```
Ratios of imbalance in relative with respect to Repayer and Defaulter
data is 11.39 : 1 (approx)
```

Plotting Functions

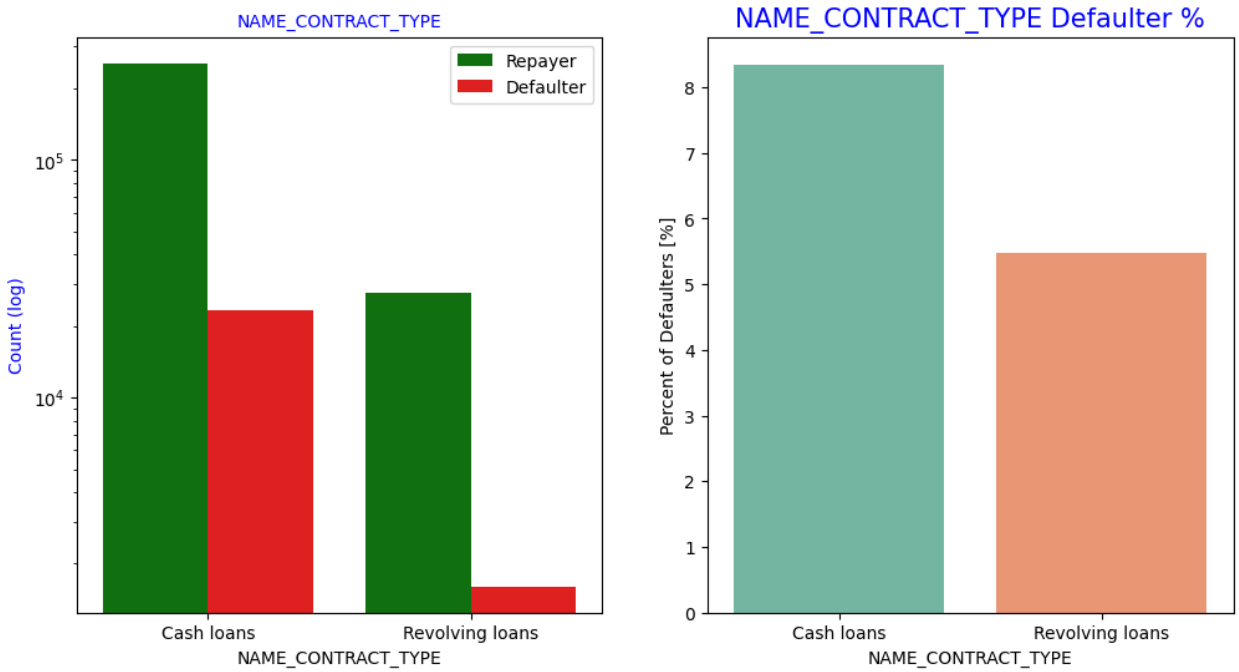
```
# function for plotting repetitive countplots in univariate
categorical analysis on applicationDF
# This function will create two subplots:
# 1. Count plot of categorical column w.r.t TARGET;
# 2. Percentage of defaulters within column

def
univariate_categorical(feature,ylog=False,label_rotation=False,horizon
tal_layout=True):
    temp = applicationDF[feature].value_counts()
    df1 = pd.DataFrame({feature: temp.index,'Number of contracts':
temp.values})

    # Calculate the percentage of target=1 per category value
    cat_perc = applicationDF[[feature,
'TARGET']].groupby([feature],as_index=False).mean()
    cat_perc["TARGET"] = cat_perc["TARGET"]*100
    cat_perc.sort_values(by='TARGET', ascending=False, inplace=True)

    if(horizontal_layout):
        fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12,6))
    else:
        fig, (ax1, ax2) = plt.subplots(nrows=2, figsize=(20,24))

    # 1. Subplot 1: Count plot of categorical column
    # sns.set_palette("Set2")
    s = sns.countplot(ax=ax1,
                      x = feature,
                      data=applicationDF,
                      hue ="TARGET",
                      order=cat_perc[feature],
                      palette=['g','r'])

    # Define common styling
    ax1.set_title(feature, fontdict={'fontsize' : 10, 'fontweight' :
3, 'color' : 'Blue'})
    ax1.legend(['Repayer','Defaulter'])

    # If the plot is not readable, use the log scale.
    if ylog:
        ax1.set_yscale('log')
        ax1.set_ylabel("Count (log)",fontdict={'fontsize' : 10,
'fontweight' : 3, 'color' : 'Blue'})
```

```
    if(label_rotation):
        s.set_xticklabels(s.get_xticklabels(),rotation=90)

    # 2. Subplot 2: Percentage of defaulters within the categorical
column
    s = sns.barplot(ax=ax2,
                    x = feature,
                    y='TARGET',
                    order=cat_perc[feature],
                    data=cat_perc,
                    palette='Set2')

    if(label_rotation):
        s.set_xticklabels(s.get_xticklabels(),rotation=90)
    plt.ylabel('Percent of Defaulters [%]', fontsize=10)
    plt.tick_params(axis='both', which='major', labelsize=10)
    ax2.set_title(feature + " Defaulter %", fontdict={'fontsize' : 15,
'fontweight' : 5, 'color' : 'Blue'})

    plt.show();

# function for plotting repetitive countplots in bivariate categorical
analysis

def bivariate_bar(x,y,df,hue,figsize):

    plt.figure(figsize=figsize)
    sns.barplot(x=x,
                  y=y,
                  data=df,
                  hue=hue,
                  palette =['g','r'])

    # Defining aesthetics of Labels and Title of the plot using style
dictionaries
    plt.xlabel(x,fontdict={'fontsize' : 10, 'fontweight' : 3,
'color' : 'Blue'})
    plt.ylabel(y,fontdict={'fontsize' : 10, 'fontweight' : 3,
'color' : 'Blue'})
    plt.title(col, fontdict={'fontsize' : 15, 'fontweight' : 5,
'color' : 'Blue'})
    plt.xticks(rotation=90, ha='right')
    plt.legend(labels = ['Repayer','Defaulter'])
    plt.show()

# function for plotting repetitive rel plots in bivaritae numerical
analysis on applicationDF

def bivariate_rel(x,y,data, hue, kind, palette, legend,figsize):
```

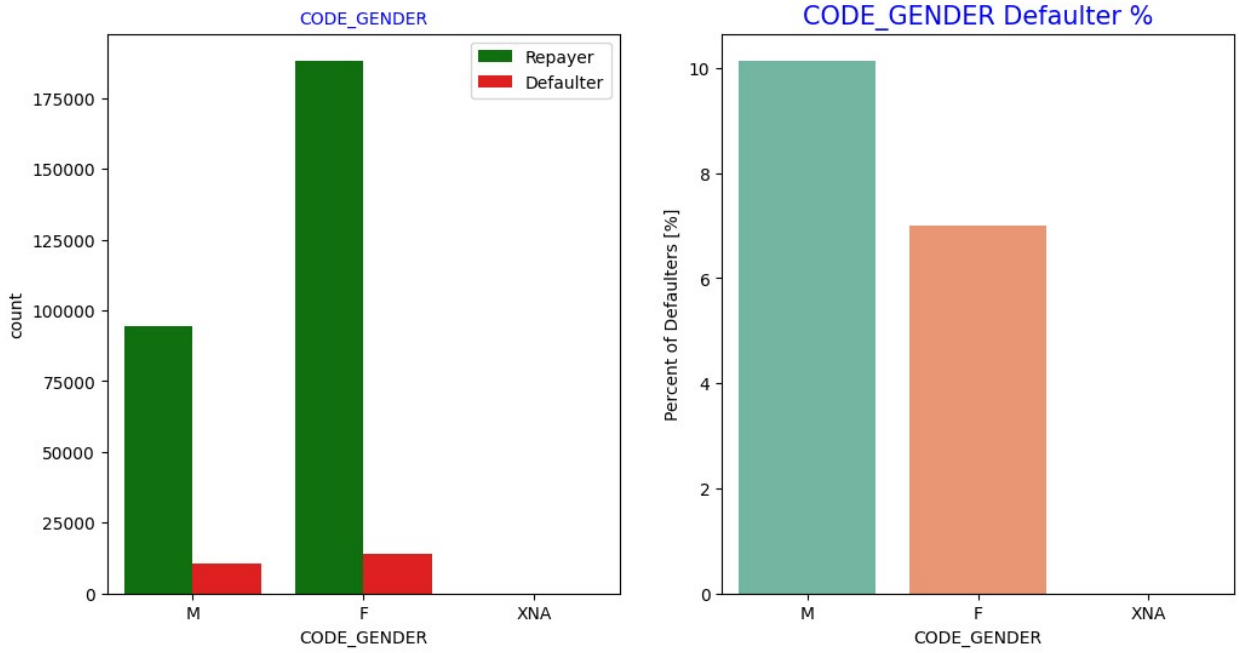
```
    plt.figure(figsize=figsize)
    sns.relplot(x=x,
                y=y,
                data=applicationDF,
                hue="TARGET",
                kind=kind,
                palette = ['g','r'],
                legend = False)
    plt.legend(['Repayer','Defaulter'])
    plt.xticks(rotation=90, ha='right')
    plt.show()

#function for plotting repetitive countplots in univariate categorical
analysis on the merged df

def univariate_merged(col,df,hue,palette,ylog,figsize):
    plt.figure(figsize=figsize)
    ax=sns.countplot(x=col,
                  data=df,
                  hue= hue,
                  palette= palette,
                  order=df[col].value_counts().index)


    if ylog:
        plt.yscale('log')
        plt.ylabel("Count (log)",fontdict={'fontsize' : 10,
'fontweight' : 3, 'color' : 'Blue'})
    else:
        plt.ylabel("Count",fontdict={'fontsize' : 10, 'fontweight' :
3, 'color' : 'Blue'})

    plt.title(col , fontdict={'fontsize' : 15, 'fontweight' : 5,
'color' : 'Blue'})
    plt.legend(loc = "upper right")
    plt.xticks(rotation=90, ha='right')

    plt.show()

# Function to plot point plots on merged dataframe

def merged_pointplot(x,y):
    plt.figure(figsize=(8,4))
    sns.pointplot(x=x,
                  y=y,
                  hue="TARGET",
                  data=loan_process_df,
                  palette =['g','r'])
   # plt.legend(['Repayer','Defaulter'])
```

Categorical Variables Analysis

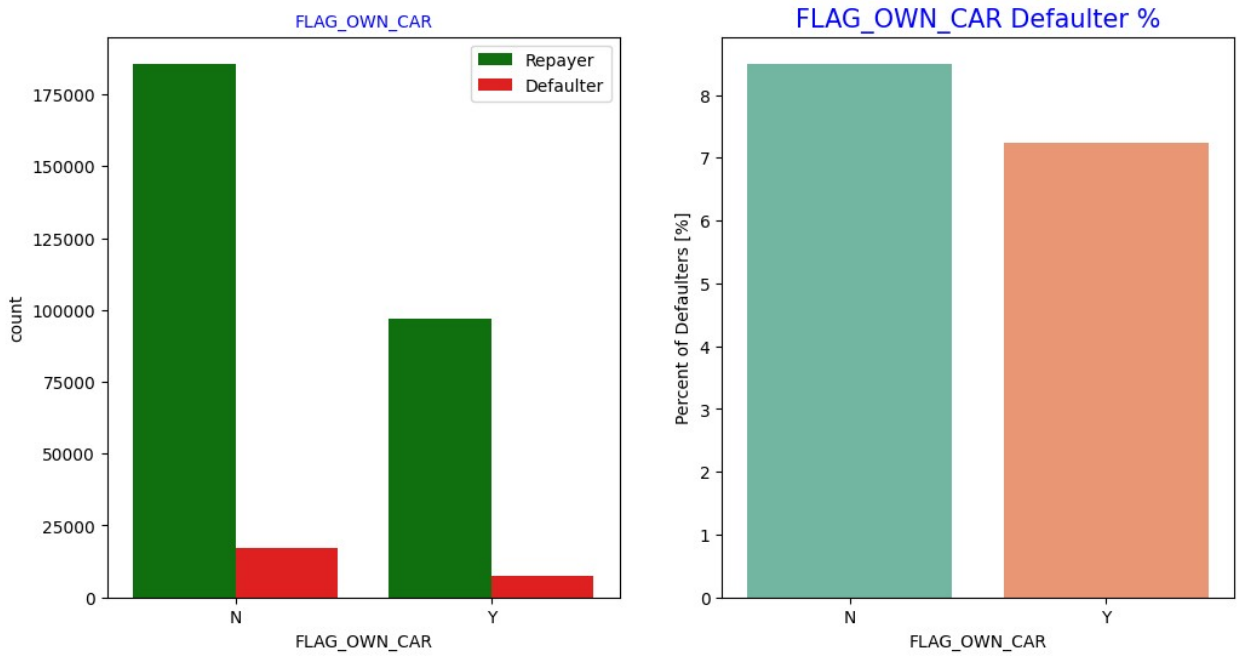
```
# Checking the contract type based on loan repayment status
univariate_categorical('NAME_CONTRACT_TYPE',True)
```



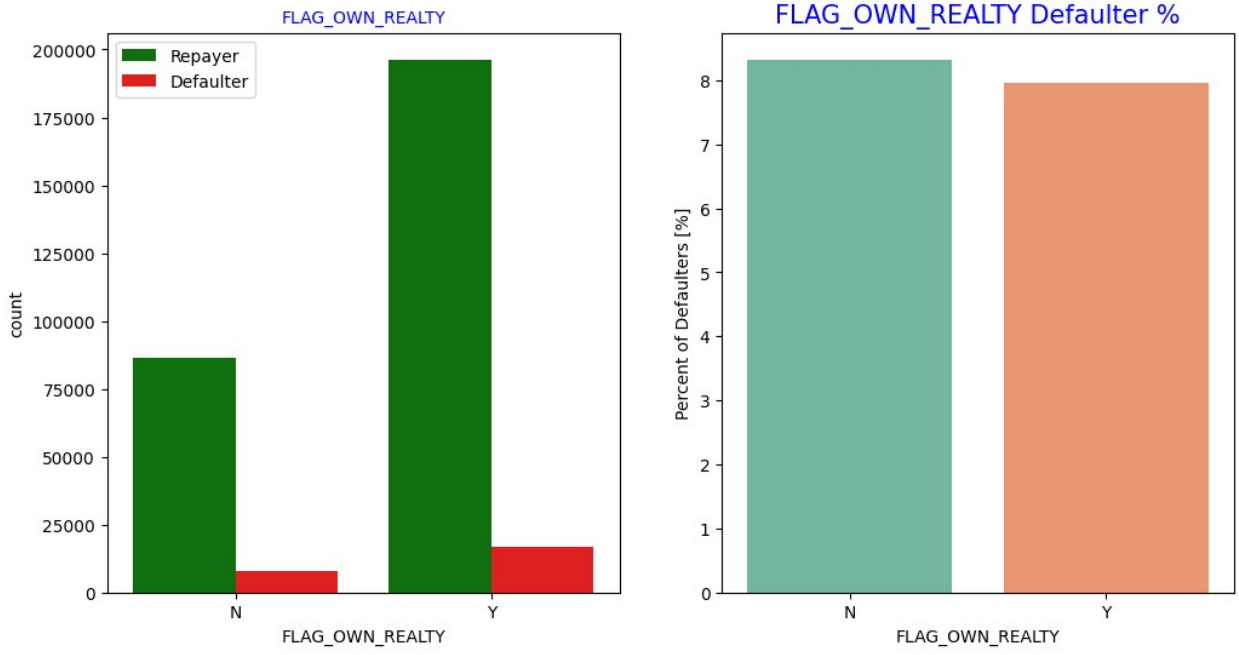

```
# Checking the type of Gender on loan repayment status
univariate_categorical('CODE_GENDER')
```



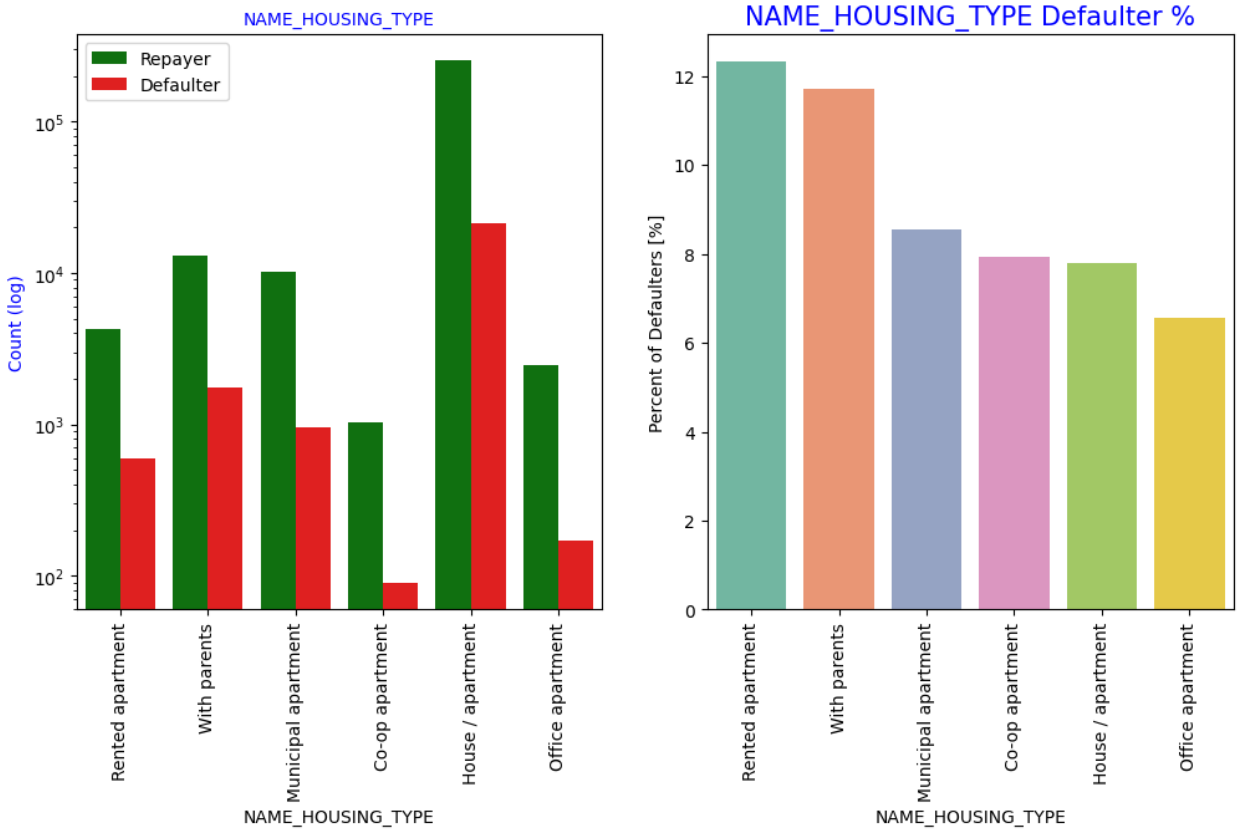
```
# Checking if owning a car is related to loan repayment status
univariate_categorical('FLAG_OWN_CAR')
```



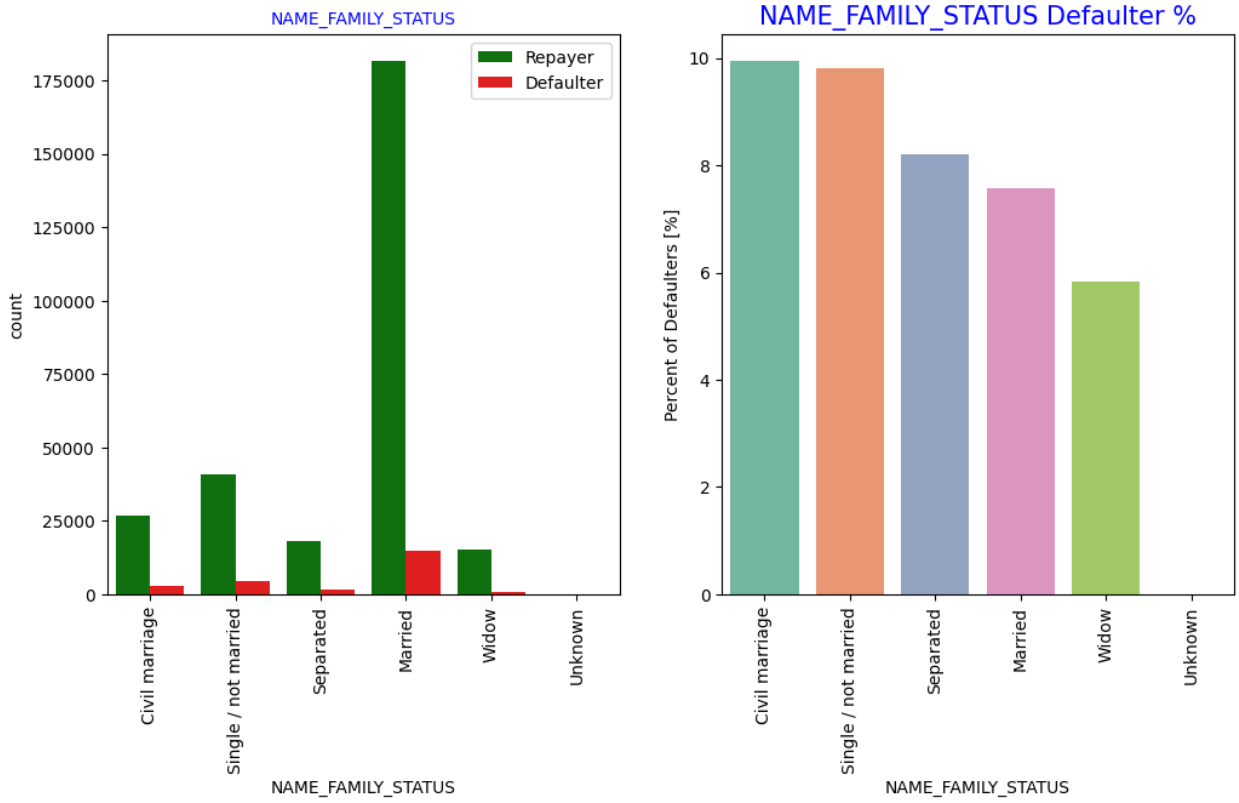

```
# Checking if owning a realty is related to loan repayment status
univariate_categorical('FLAG_OWN_REALTY')
```



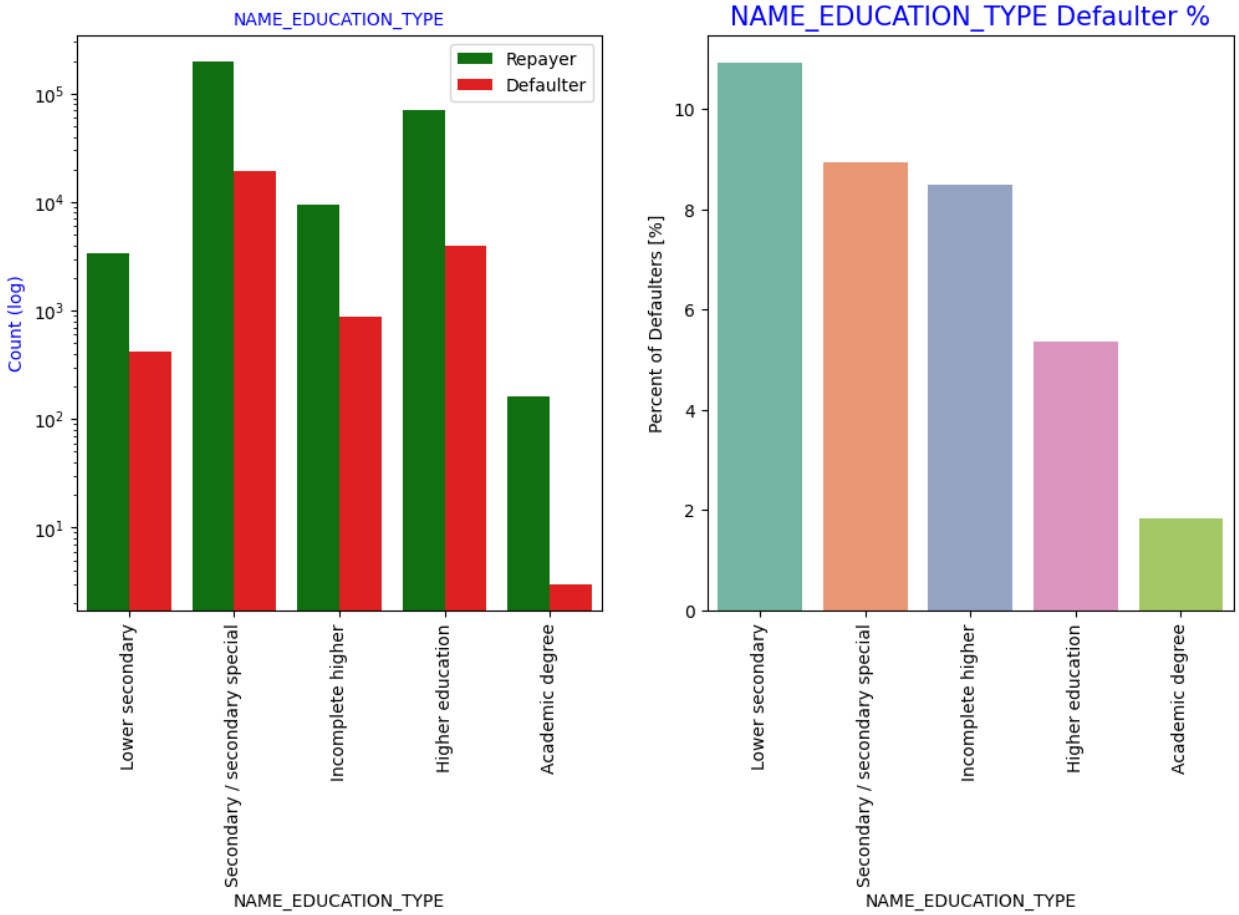

```
# Analyzing Housing Type based on loan repayment status
univariate_categorical("NAME_HOUSING_TYPE",True,True,True)
```



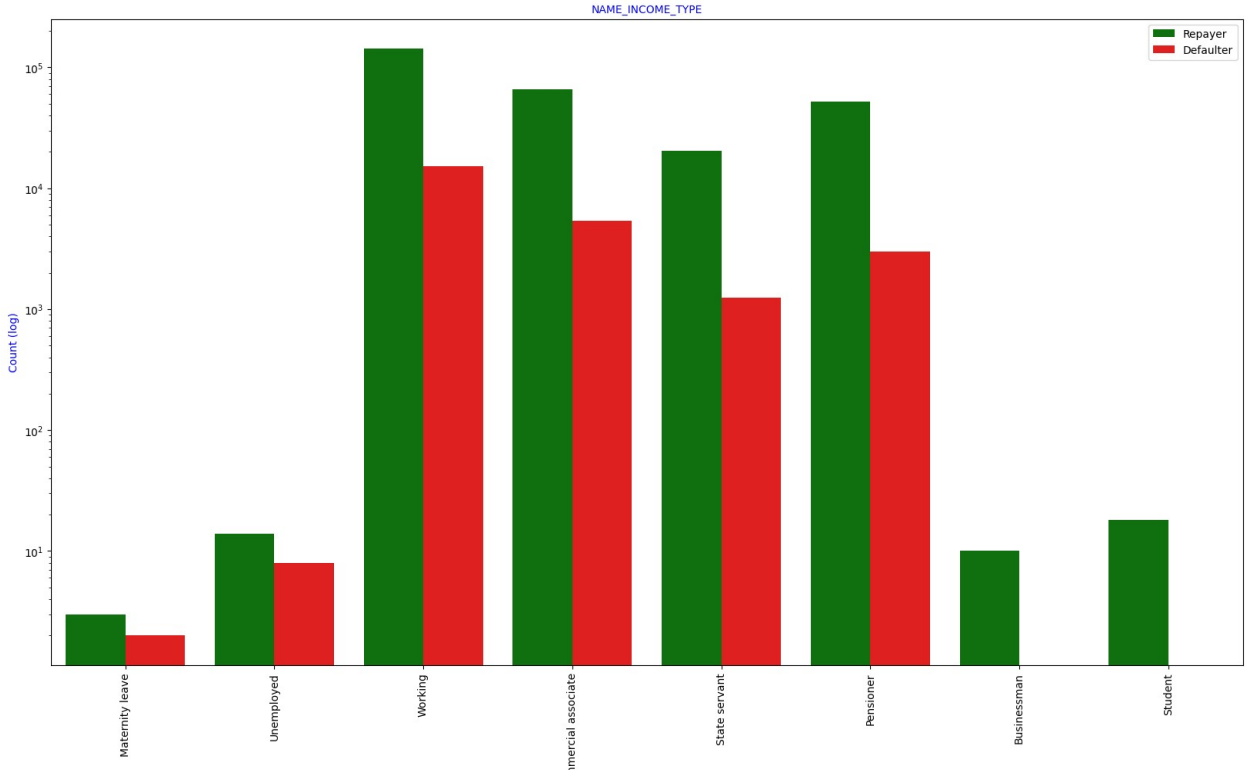

```
# Analyzing Family status based on loan repayment status
univariate_categorical("NAME_FAMILY_STATUS",False,True,True)
```

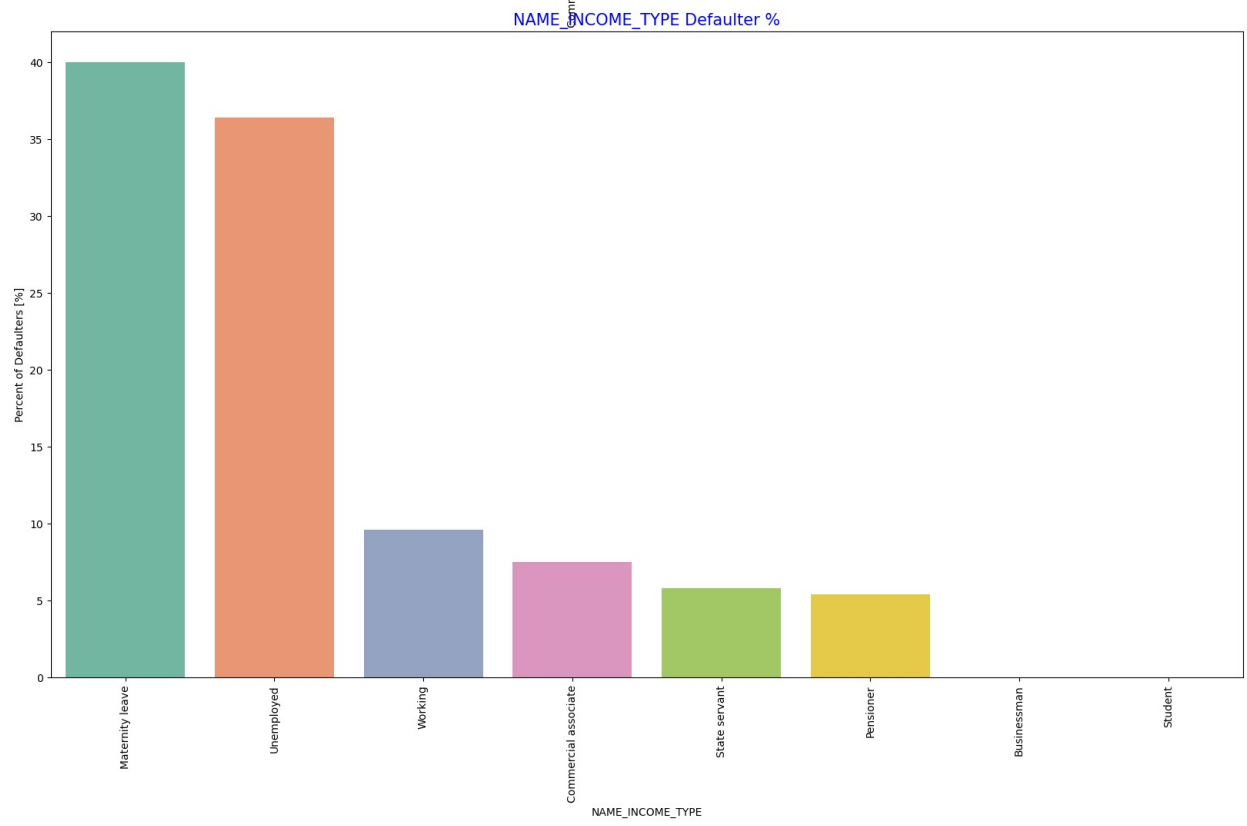



```
# Analyzing Education Type based on loan repayment status
univariate_categorical("NAME_EDUCATION_TYPE",True,True,True)
```

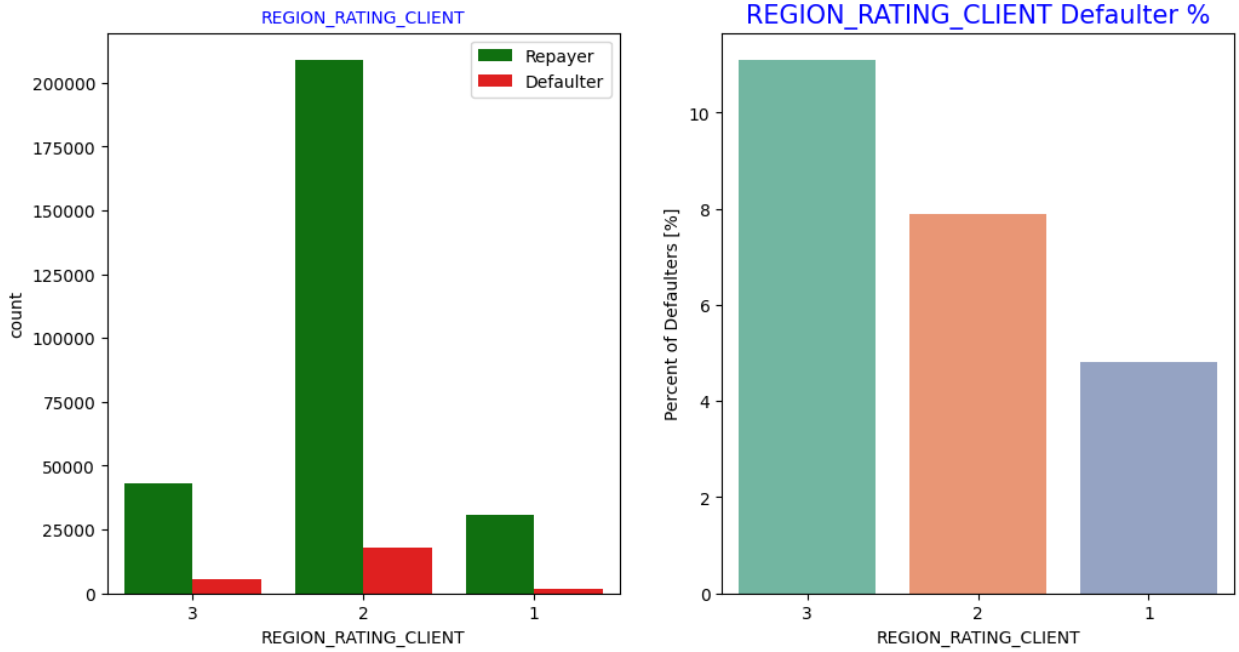



```
# Analyzing Income Type based on loan repayment status
univariate_categorical("NAME_INCOME_TYPE",True,True,False)
```

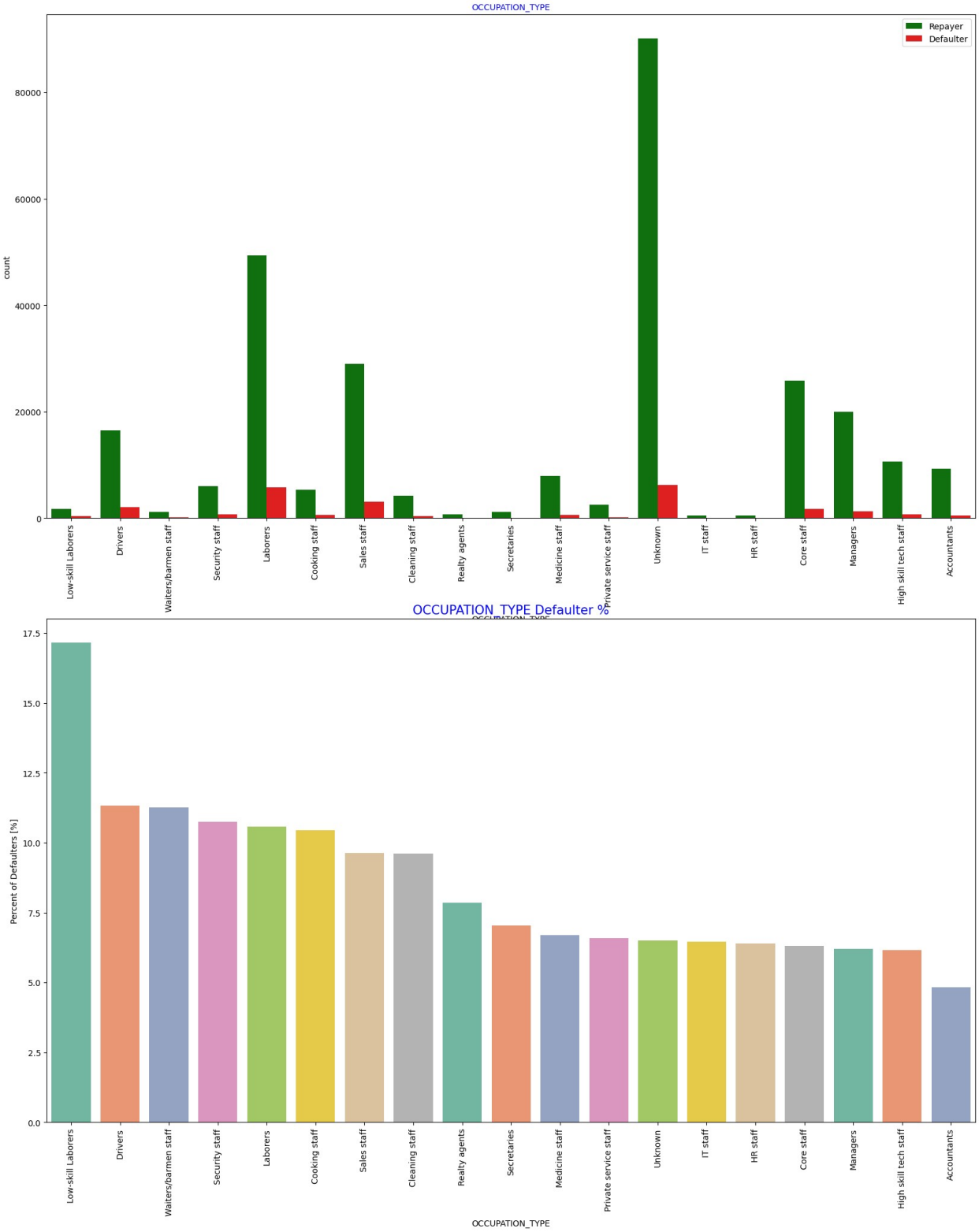

NAME_INCOME_TYPE
Repayer
Defaulter
Count (log)
10^5
10^4
10^3
10^2
10^1
Maternity leave
Unemployed
Working
Commercial associate
State servant
Pensioner
Businessman
Student


NAME_INCOME_TYPE Defaulter %
Percent of Defaulters [%]
40
35
30
25
20
15
10
5
0
Maternity leave
Unemployed
Working
Commercial associate
State servant
Pensioner
Businessman
Student
NAME_INCOME_TYPE

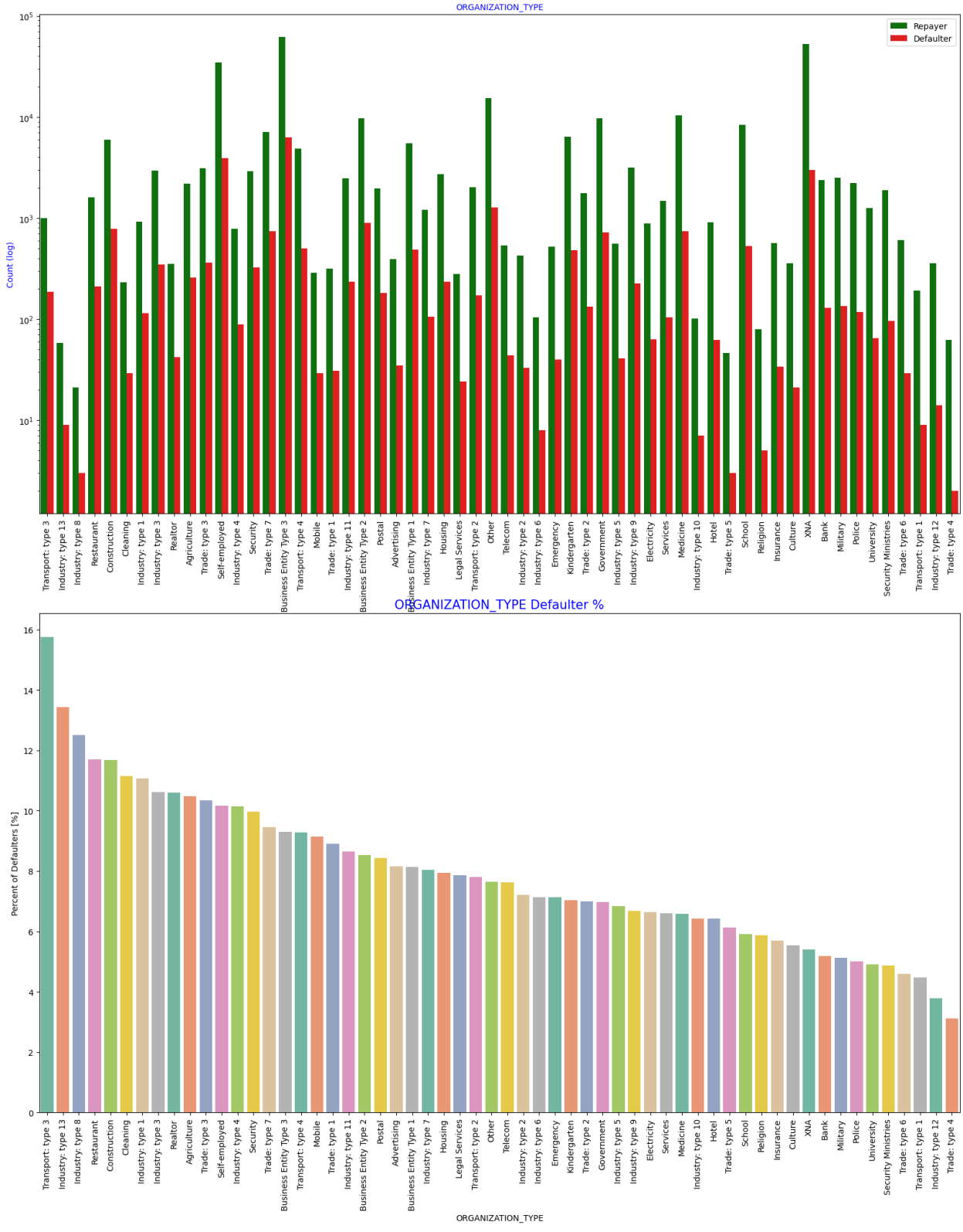
```
# Analyzing Region rating where applicant lives based on loan repayment status
univariate_categorical("REGION_RATING_CLIENT",False,False,True)
```



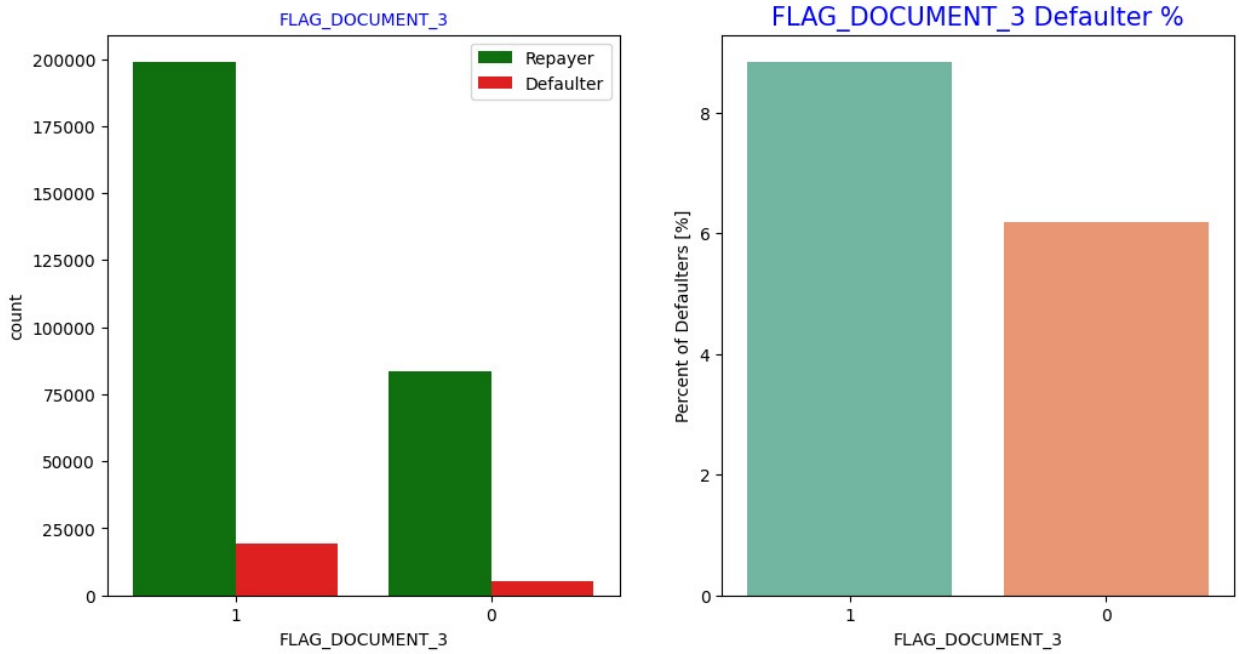

```
# Analyzing Occupation Type where applicant lives based on loan repayment status
univariate_categorical("OCCUPATION_TYPE",False,True,False)
```



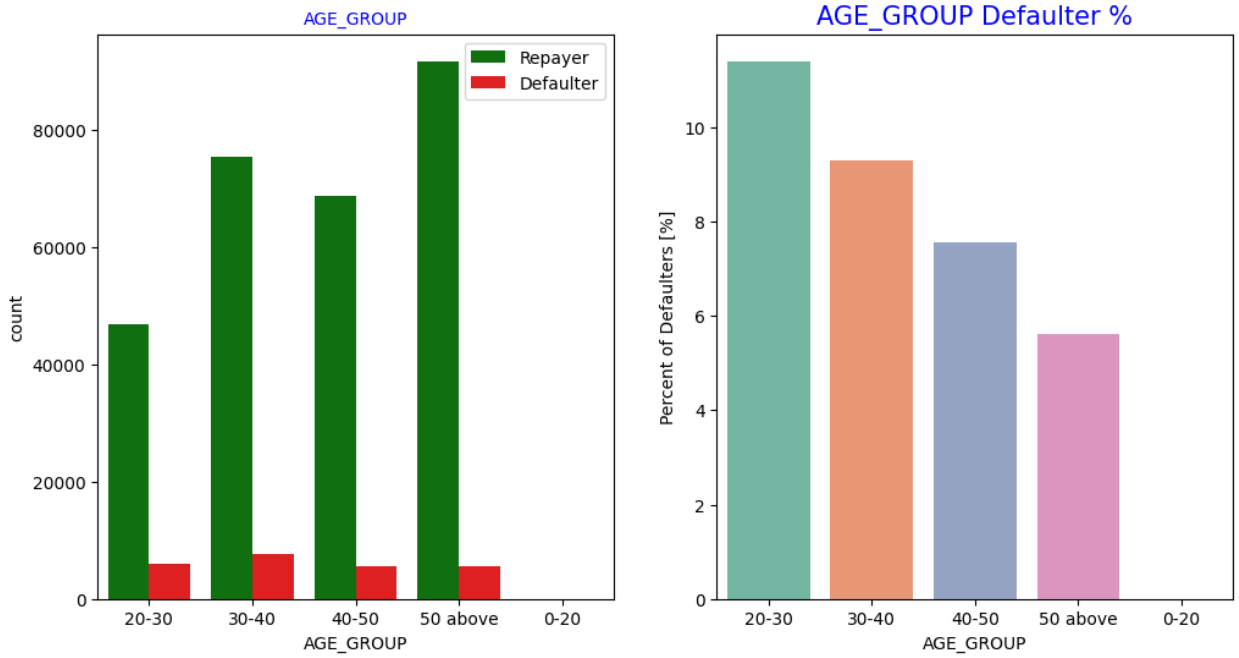

```
# Checking Loan repayment status based on Organization type
univariate_categorical("ORGANIZATION_TYPE",True,True,False)
```


ORGANIZATION_TYPE
Repayer
Defaulter
Count (log)
Transport: type 3
Industry: type 13
Industry: type 8
Restaurant
Construction
Cleaning
Industry: type 1
Industry: type 3
Realtor
Agriculture
Trade: type 3
Self-employed
Industry: type 4
Security
Trade: type 7
Business Entity Type 3
Transport: type 4
Mobile
Trade: type 1
Industry: type 11
Business Entity Type 2
Postal
Advertising
Business Entity Type 1
Industry: type 7
Housing
Legal Services
Transport: type 2
Other
Telecom
Industry: type 2
Industry: type 6
Emergency
Kindergarten
Trade: type 2
Government
Industry: type 5
Industry: type 9
Electricity
Services
Medicine
Industry: type 10
Hotel
Trade: type 5
School
Religion
Insurance
Culture
XNA
Bank
Military
Police
University
Security Ministries
Trade: type 6
Transport: type 1
Industry: type 12
Trade: type 4
ORGANIZATION_TYPE Defaulter %
Percent of Defaulters [%]
ORGANIZATION_TYPE

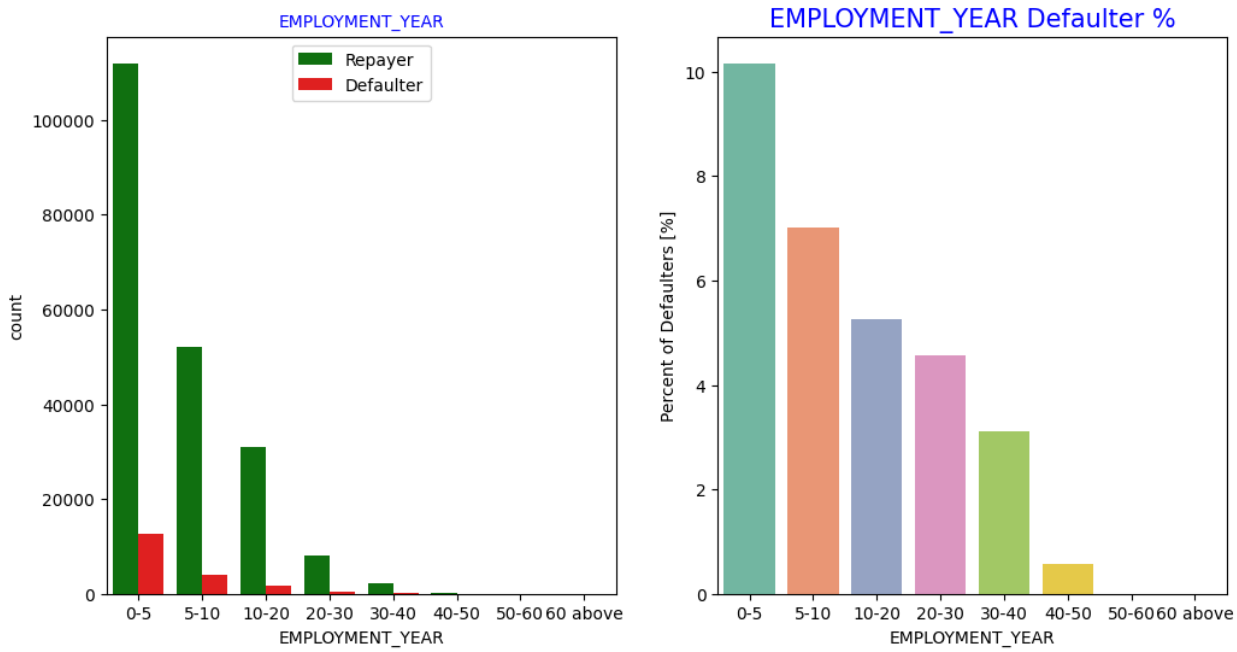
```
# Analyzing Flag_Doc_3 submission status based on loan repayment status
univariate_categorical("FLAG_DOCUMENT_3",False,False,True)
```



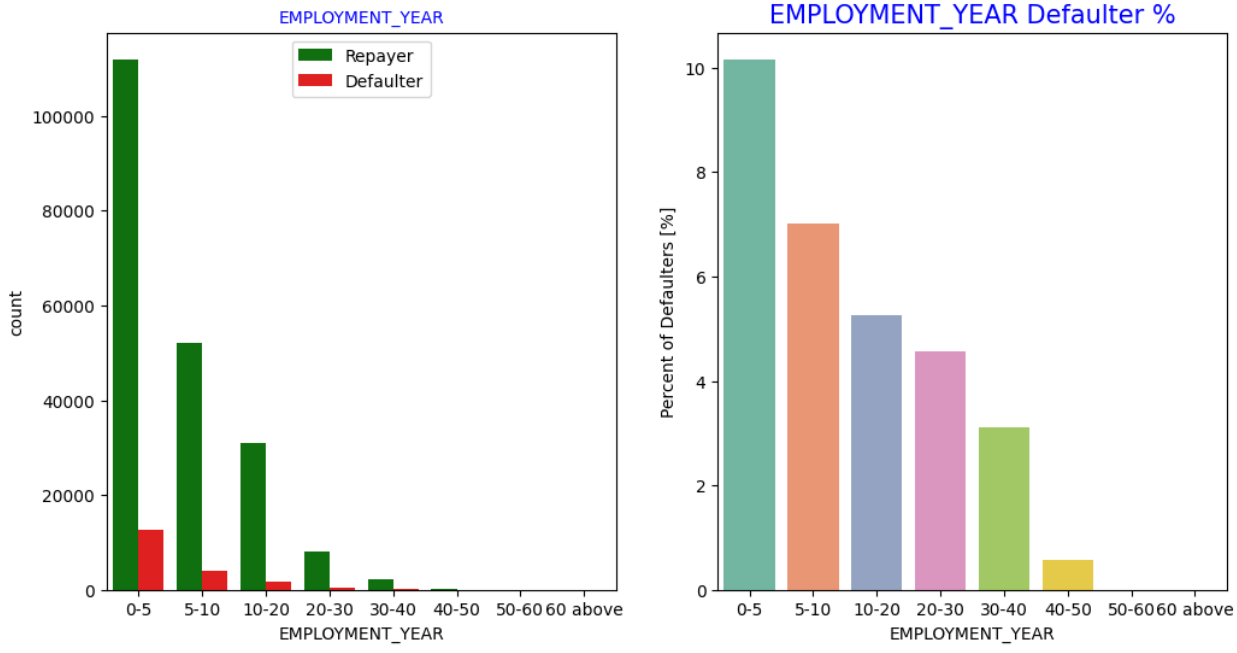

```
# Analyzing Age Group based on loan repayment status
univariate_categorical("AGE_GROUP",False,False,True)
```



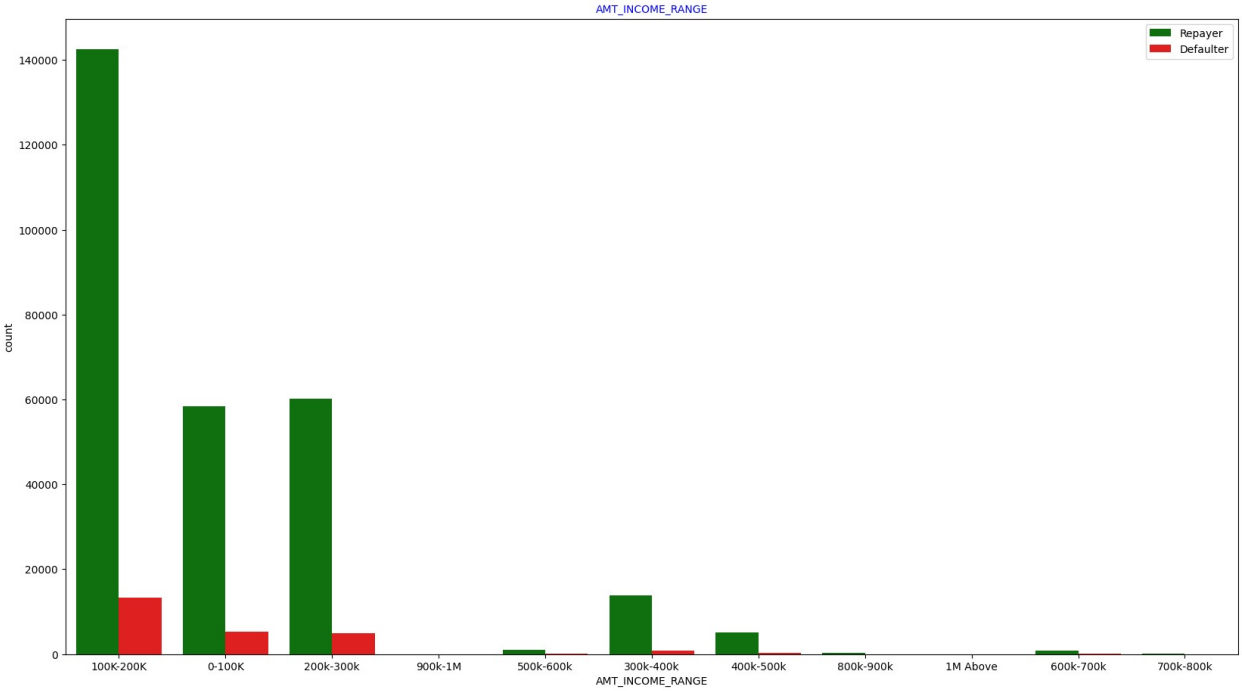
```
# Analyzing Employment_Year based on loan repayment status
univariate_categorical("EMPLOYMENT_YEAR",False,False,True)
```

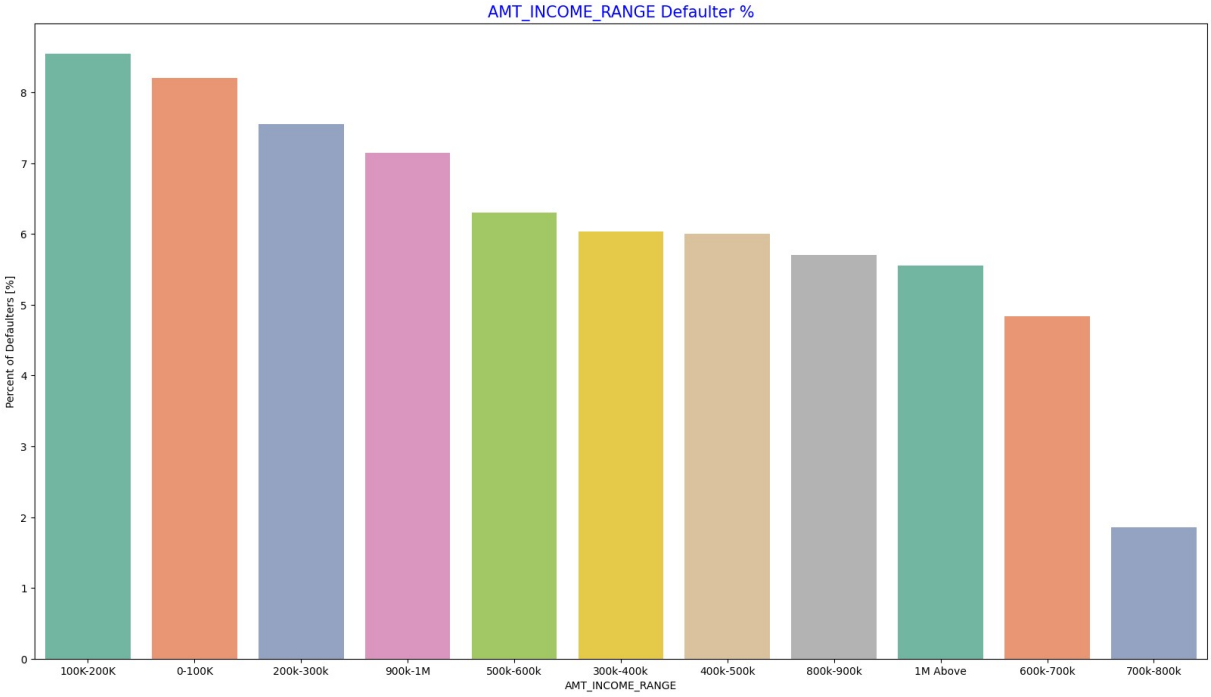



```
# Analyzing Employment_Year based on loan repayment status
univariate_categorical("EMPLOYMENT_YEAR",False,False,True)
```

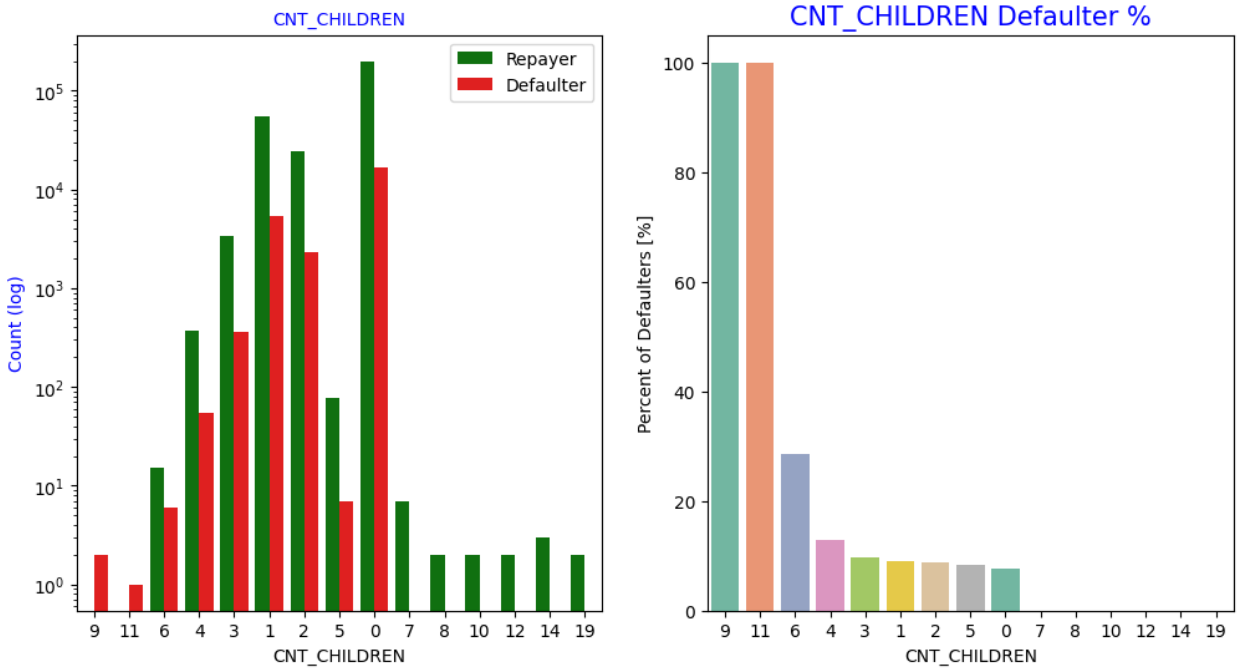



```
# Analyzing Amount_Income Range based on loan repayment status
univariate_categorical("AMT_INCOME_RANGE",False,False,False)
```

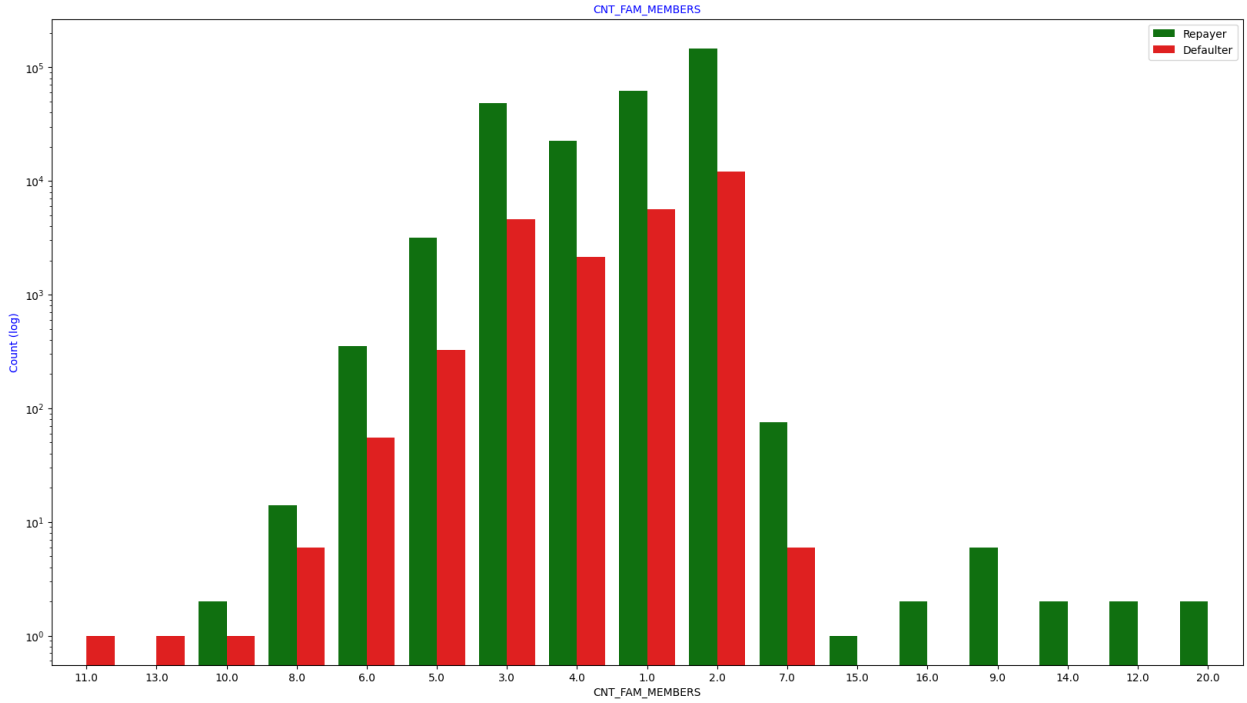


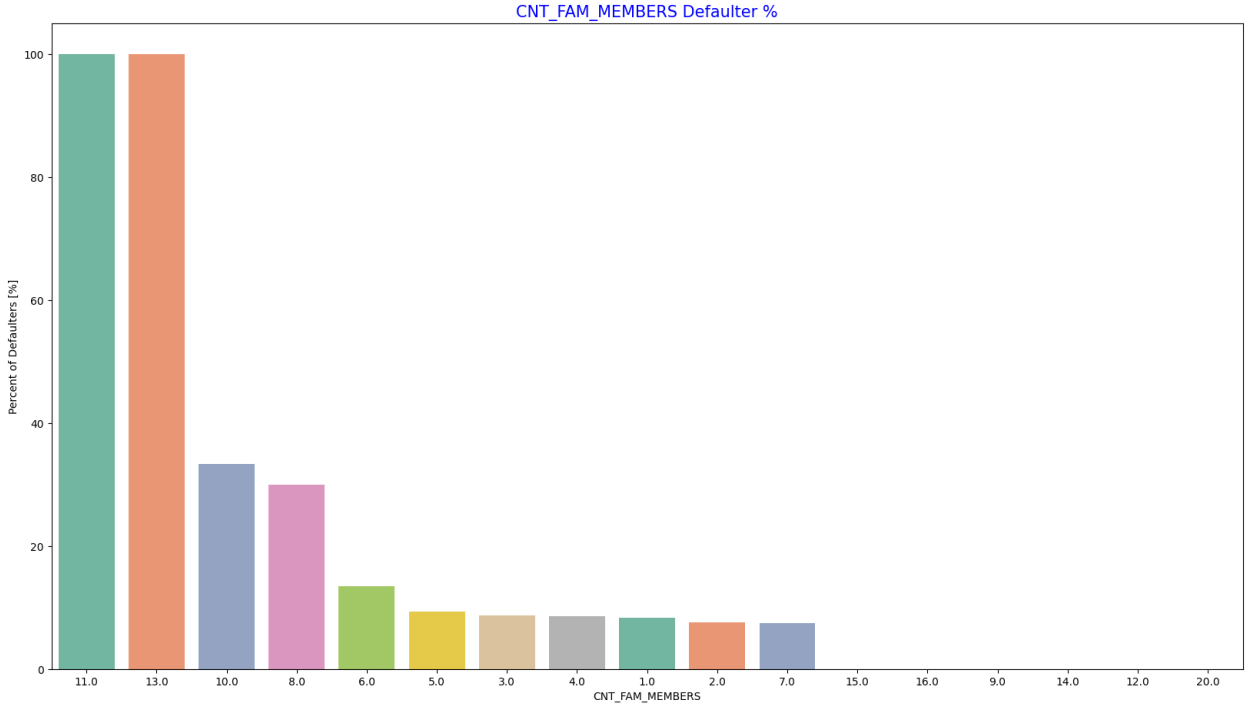



```
# Analyzing Number of children based on loan repayment status
univariate_categorical("CNT_CHILDREN",True)
```




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



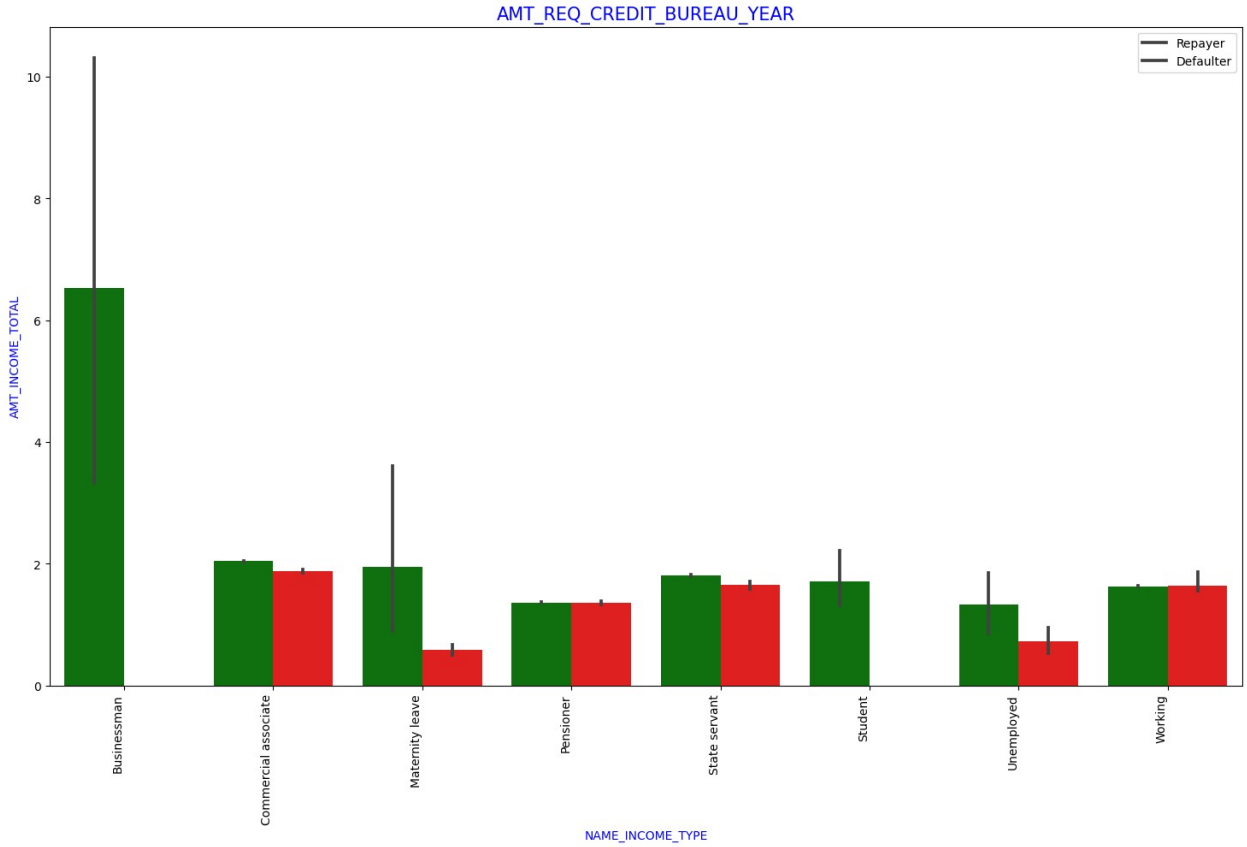



```
applicationDF.groupby('NAME_INCOME_TYPE')
['AMT_INCOME_TOTAL'].describe()
```

```
                        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
```

```
# Income type vs Income Amount Range
bivariate_bar("NAME_INCOME_TYPE","AMT_INCOME_TOTAL",applicationDF,"TAR
GET",(18,10))
```



Numeric Variables Analysis

```
applicationDF.columns
```

```
Index(['SK_ID_CURR', 'TARGET', 'NAME_CONTRACT_TYPE', 'CODE_GENDER',
'FLAG_OWN_CAR', 'FLAG_OWN_REALTY', 'CNT_CHILDREN', 'AMT_INCOME_TOTAL',
'AMT_CREDIT', 'AMT_ANNUITY', 'AMT_GOODS_PRICE', 'NAME_TYPE_SUITE',
'NAME_INCOME_TYPE', 'NAME_EDUCATION_TYPE', 'NAME_FAMILY_STATUS',
'NAME_HOUSING_TYPE', 'REGION_POPULATION_RELATIVE', 'DAYS_BIRTH',
'DAYS_EMPLOYED', 'DAYS_REGISTRATION', 'DAYS_ID_PUBLISH',
'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')
```

```
# Bifurcating the applicationDF dataframe based on Target value 0 and
1 for correlation and other analysis
cols_for_correlation = ['NAME_CONTRACT_TYPE', 'CODE_GENDER',
'FLAG_OWN_CAR', 'FLAG_OWN_REALTY',
                        'CNT_CHILDREN', 'AMT_INCOME_TOTAL',
'AMT_CREDIT', 'AMT_ANNUITY', 'AMT_GOODS_PRICE',
                        'NAME_TYPE_SUITE', 'NAME_INCOME_TYPE',
'NAME_EDUCATION_TYPE', 'NAME_FAMILY_STATUS',
                        'NAME_HOUSING_TYPE',
'REGION_POPULATION_RELATIVE', 'DAYS_BIRTH', 'DAYS_EMPLOYED',
                        'DAYS_REGISTRATION', 'DAYS_ID_PUBLISH',
'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_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']
```

```
Repayer_df = applicationDF.loc[applicationDF['TARGET']==0,
cols_for_correlation] # Repayers
Defaulter_df = applicationDF.loc[applicationDF['TARGET']==1,
cols_for_correlation] # Defaulters
```

Correlation between numeric variable

```
# Assuming Repayer_df is already defined

# Selecting only numeric columns
numeric_cols = Repayer_df.select_dtypes(include=[np.number]).columns
numeric_repayer_df = Repayer_df[numeric_cols]

# Getting the correlation matrix
corr_repayer = numeric_repayer_df.corr()

# Mask to get the upper triangle of the correlation matrix
mask = np.triu(np.ones_like(corr_repayer, dtype=bool))

# Applying the mask to the correlation matrix
corr_repayer = corr_repayer.where(mask)

# Unstacking the correlation matrix
corr_df_repayer = corr_repayer.unstack().reset_index()
corr_df_repayer.columns = ['VAR1', 'VAR2', 'Correlation']

# Dropping NaN values
corr_df_repayer.dropna(subset=["Correlation"], inplace=True)

# Taking the absolute value of the correlations
corr_df_repayer["Correlation"] = corr_df_repayer["Correlation"].abs()

# Sorting the dataframe by correlation values
corr_df_repayer.sort_values(by='Correlation', ascending=False,
inplace=True)

# Displaying the top 10 correlations
top_10_correlations = corr_df_repayer.head(10)
print(top_10_correlations)
```

```
                          VAR1                          VAR2
Correlation
0                  CNT_CHILDREN                 CNT_CHILDREN
1.0
120  REGION_POPULATION_RELATIVE  REGION_POPULATION_RELATIVE
1.0
168               DAYS_EMPLOYED               DAYS_EMPLOYED
1.0
```

```
192           DAYS_REGISTRATION           DAYS_REGISTRATION
1.0
216             DAYS_ID_PUBLISH             DAYS_ID_PUBLISH
1.0
240             CNT_FAM_MEMBERS             CNT_FAM_MEMBERS
1.0
264     HOUR_APPR_PROCESS_START     HOUR_APPR_PROCESS_START
1.0
288  REG_REGION_NOT_LIVE_REGION  REG_REGION_NOT_LIVE_REGION
1.0
312    OBS_60_CNT_SOCIAL_CIRCLE    OBS_60_CNT_SOCIAL_CIRCLE
1.0
336    DEF_60_CNT_SOCIAL_CIRCLE    DEF_60_CNT_SOCIAL_CIRCLE
1.0
```

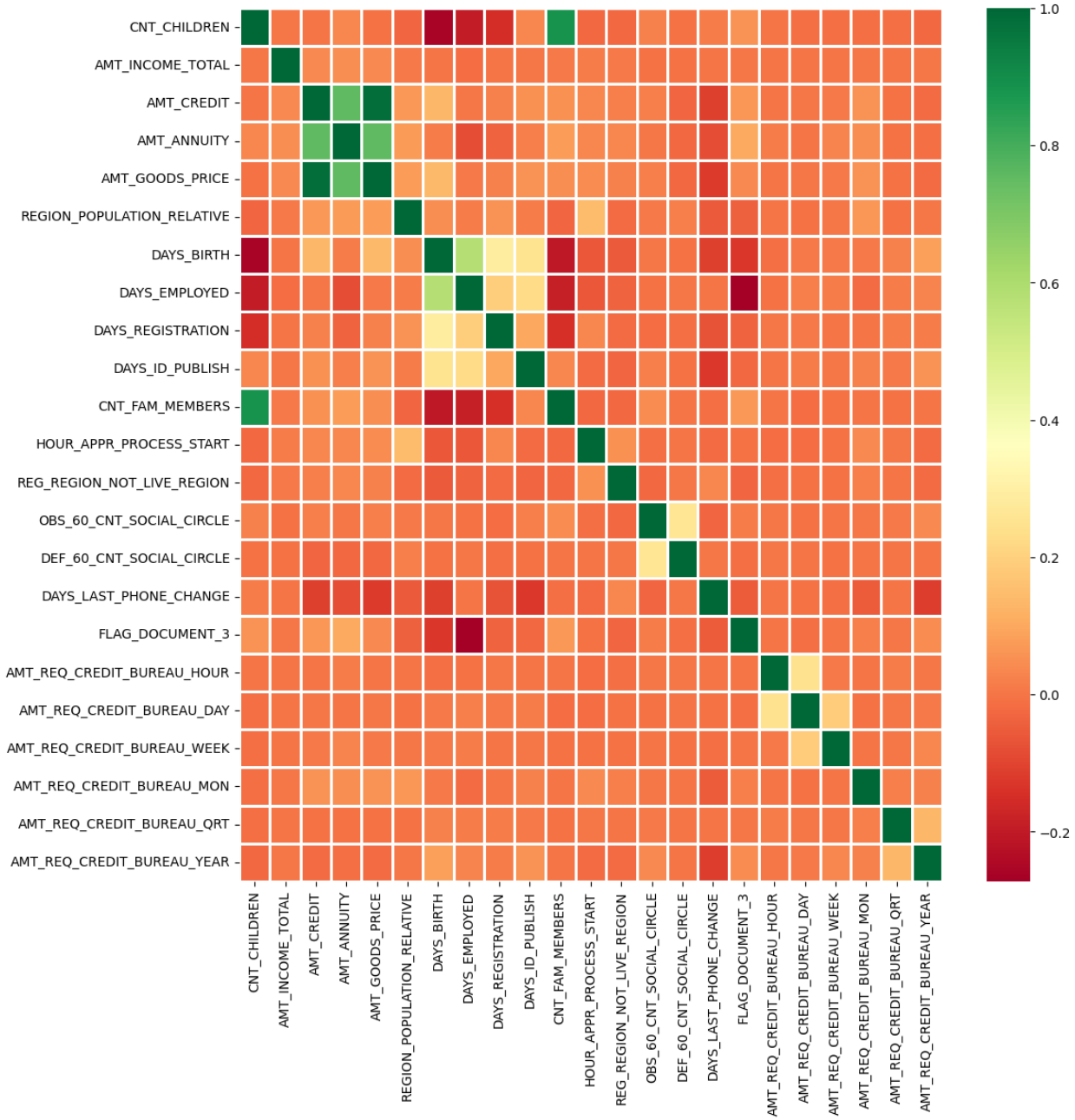
```
# Assuming Repayer_df is already defined

# Selecting only numeric columns
numeric_cols = Repayer_df.select_dtypes(include=[np.number]).columns
numeric_repayer_df = Repayer_df[numeric_cols]

# Creating the correlation matrix
corr_matrix = numeric_repayer_df.corr()

# Plotting the heatmap
fig = plt.figure(figsize=(12,12))
ax = sns.heatmap(corr_matrix, cmap="RdYlGn", annot=False,
linewidths=1)
plt.title('Heatmap of Correlation Matrix for Repayers Data')
plt.show()
```


```
# Assuming Defaulter_df is already defined

# Selecting only numeric columns
numeric_cols = Defaulter_df.select_dtypes(include=[np.number]).columns
numeric_defaulter_df = Defaulter_df[numeric_cols]

# Creating the correlation matrix
corr_Defaulter = numeric_defaulter_df.corr()

# Getting the upper triangle of the correlation matrix
```

```
corr_Defaulter =
corr_Defaulter.where(np.triu(np.ones(corr_Defaulter.shape),
k=1).astype(bool))

# Unstacking and resetting the index
corr_df_Defaulter = corr_Defaulter.unstack().reset_index()
corr_df_Defaulter.columns = ['VAR1', 'VAR2', 'Correlation']

# Dropping NaN values
corr_df_Defaulter.dropna(subset=["Correlation"], inplace=True)

# Taking absolute value of correlations
corr_df_Defaulter["Correlation"] =
corr_df_Defaulter["Correlation"].abs()

# Sorting by correlation in descending order
corr_df_Defaulter.sort_values(by='Correlation', ascending=False,
inplace=True)

# Getting the top 10 correlations
top_10_correlations = corr_df_Defaulter.head(10)

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

	VAR1	VAR2	Correlation
94	AMT_GOODS_PRICE	AMT_CREDIT	0.983103
230	CNT_FAM_MEMBERS	CNT_CHILDREN	0.885484
95	AMT_GOODS_PRICE	AMT_ANNUITY	0.752699
71	AMT_ANNUITY	AMT_CREDIT	0.752195
167	DAYS_EMPLOYED	DAYS_BIRTH	0.582185
190	DAYS_REGISTRATION	DAYS_BIRTH	0.289114
375	FLAG_DOCUMENT_3	DAYS_EMPLOYED	0.272169
335	DEF_60_CNT_SOCIAL_CIRCLE	OBS_60_CNT_SOCIAL_CIRCLE	0.264159
138	DAYS_BIRTH	CNT_CHILDREN	0.259109
213	DAYS_ID_PUBLISH	DAYS_BIRTH	0.252863

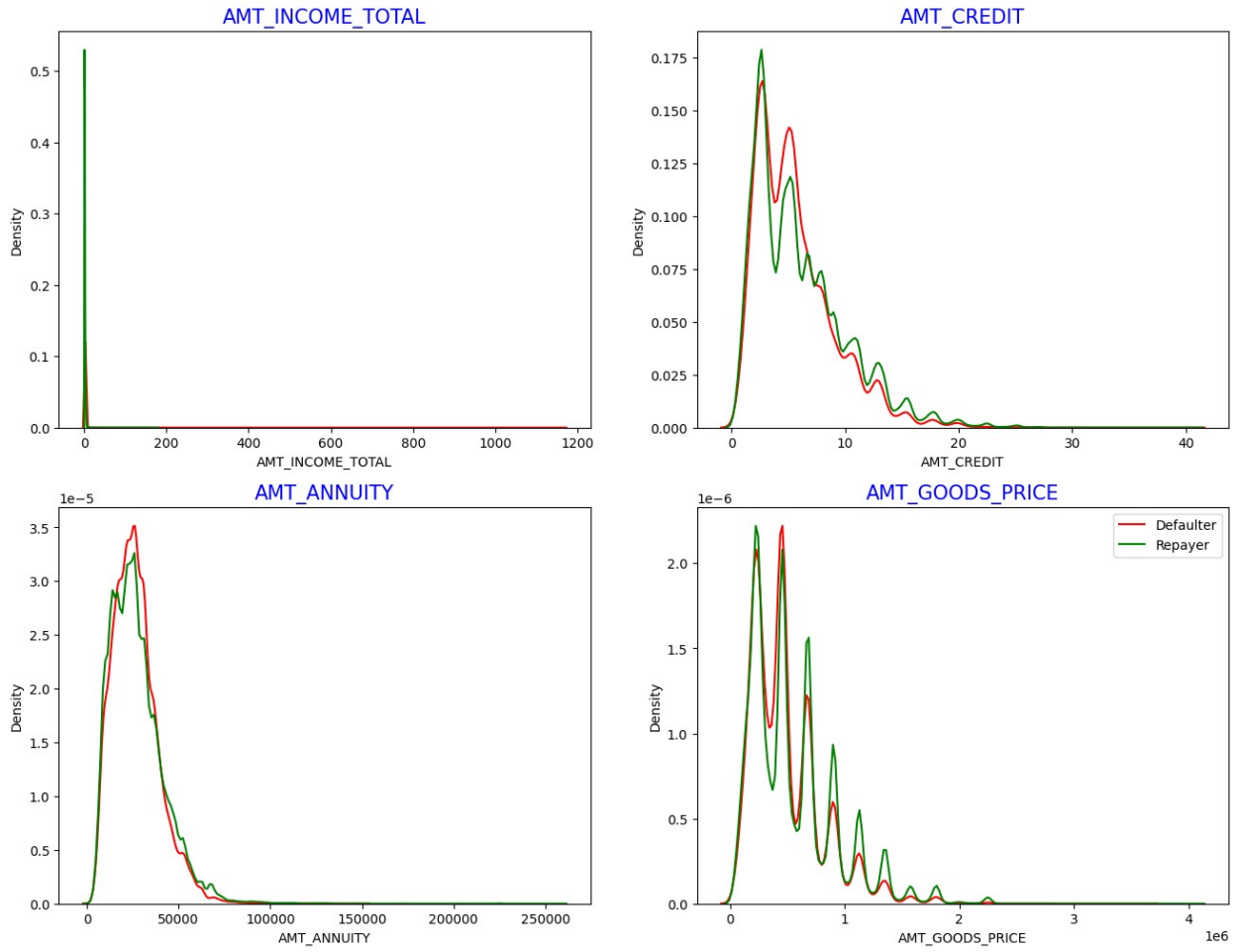
```
# Assuming Defaulter_df is already defined

# Selecting only numeric columns
numeric_cols = Defaulter_df.select_dtypes(include=[np.number]).columns
numeric_defaulter_df = Defaulter_df[numeric_cols]

# Creating the correlation matrix
corr_Defaulter = numeric_defaulter_df.corr()

# Plotting the heatmap
fig = plt.figure(figsize=(12, 12))
ax = sns.heatmap(corr_Defaulter, cmap="RdYlGn", annot=False,
linewidths=1)
plt.show()
```



Numerical Univariate Analysis

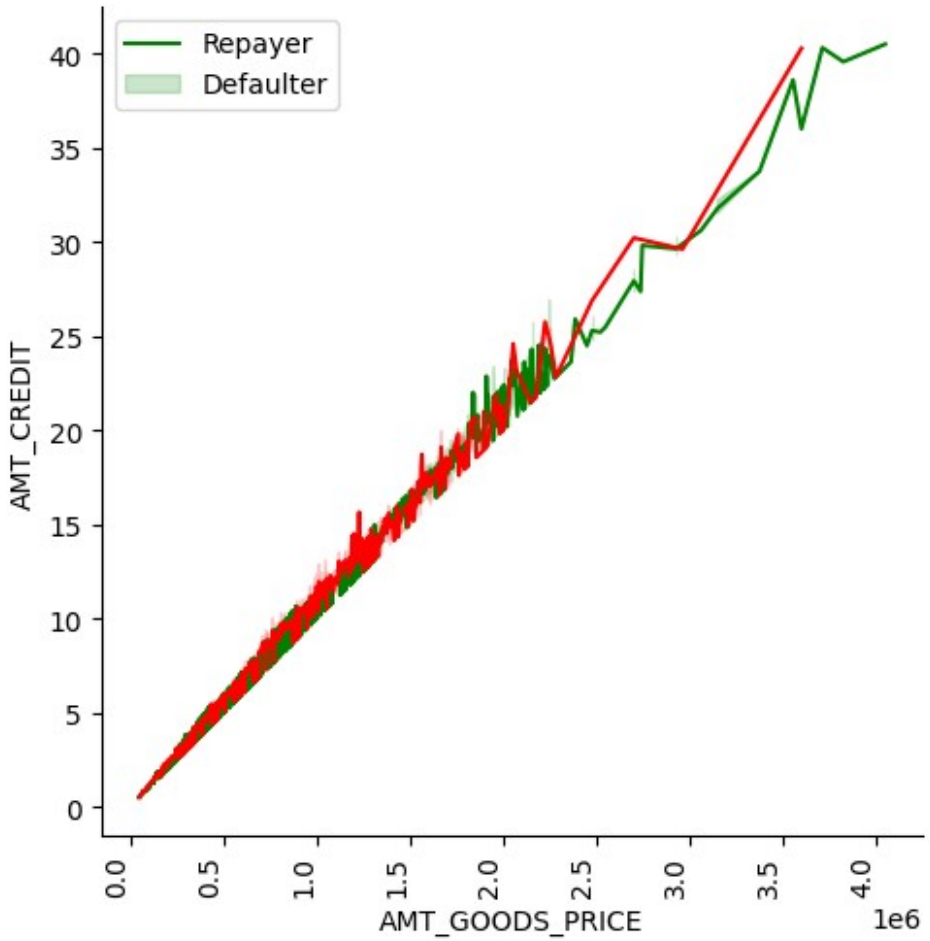
```
# Plotting the numerical columns related to amount as distribution
plot to see density
amount = applicationDF[[
'AMT_INCOME_TOTAL','AMT_CREDIT','AMT_ANNUITY', 'AMT_GOODS_PRICE']]

fig = plt.figure(figsize=(16,12))

for i in enumerate(amount):
    plt.subplot(2,2,i[0]+1)
```

```
    sns.distplot(Defaulter_df[i[1]], hist=False, color='r',label
="Defaulter")
    sns.distplot(Repayer_df[i[1]], hist=False, color='g', label
="Repayer")
    plt.title(i[1], fontdict={'fontsize' : 15, 'fontweight' : 5,
'color' : 'Blue'})

plt.legend()

plt.show()
```

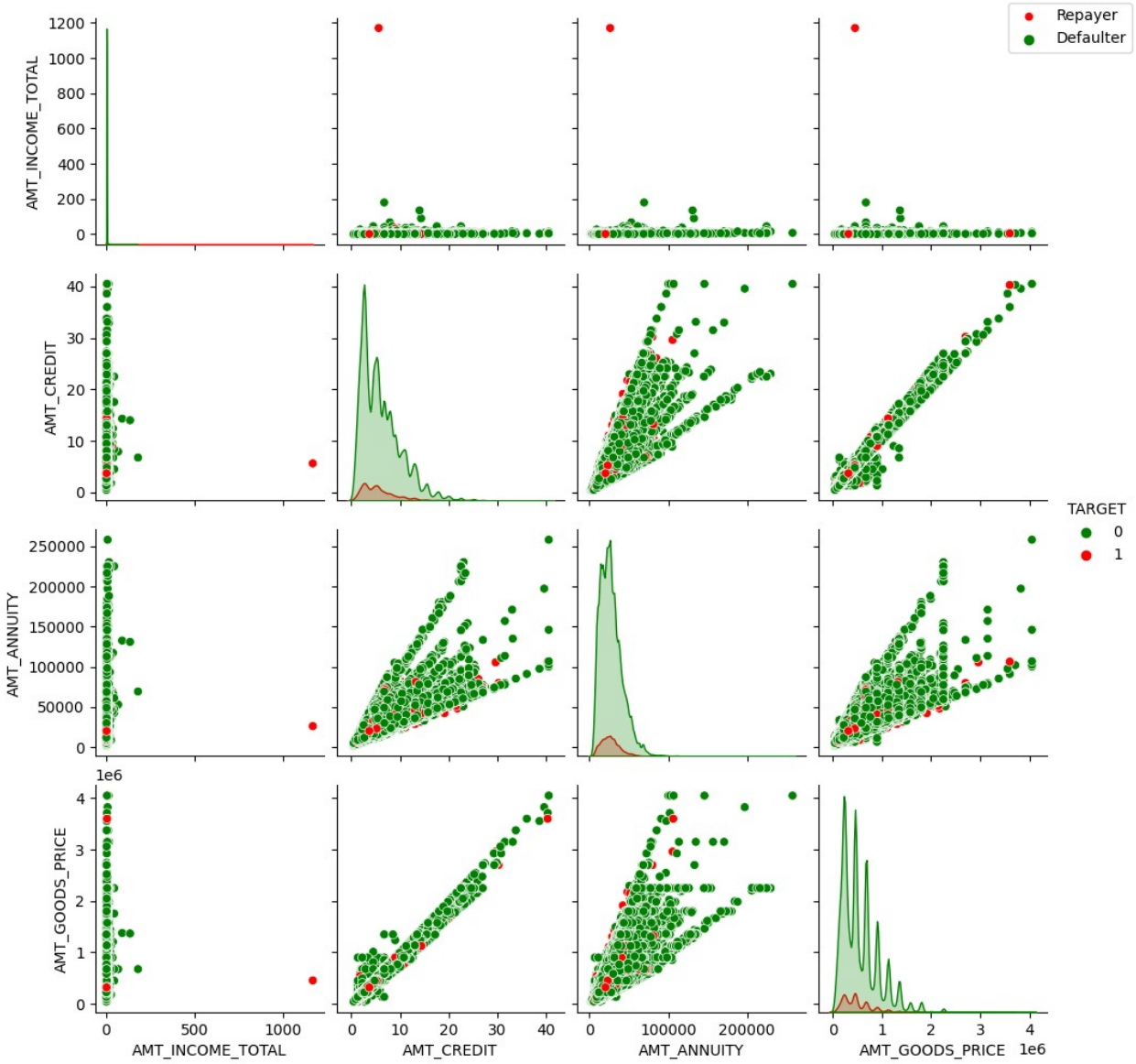



```
# Checking the relationship between Goods price and credit and
comparing with loan repayment staus
bivariate_rel('AMT_GOODS_PRICE','AMT_CREDIT',applicationDF,"TARGET",
"line", ['g','r'], False,(15,6))
```

```
<Figure size 1500x600 with 0 Axes>
```




```
# Plotting pairplot between amount variable to draw reference against
loan repayment status
amount = applicationDF[[ 'AMT_INCOME_TOTAL','AMT_CREDIT',
                         'AMT_ANNUITY', 'AMT_GOODS_PRICE','TARGET']]
amount = amount[(amount["AMT_GOODS_PRICE"].notnull()) &
(amount["AMT_ANNUITY"].notnull())]
ax= sns.pairplot(amount,hue="TARGET",palette=["g","r"])
ax.fig.legend(labels=['Repayer','Defaulter'])
plt.show()
```



Merged Dataframes Analysis

```
#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()
```

```
   SK_ID_CURR  TARGET NAME_CONTRACT_TYPE_x CODE_GENDER FLAG_OWN_CAR
FLAG_OWN_REALTY  CNT_CHILDREN  AMT_INCOME_TOTAL  AMT_CREDIT_x
AMT_ANNUITY_x  AMT_GOODS_PRICE_x NAME_TYPE_SUITE NAME_INCOME_TYPE
NAME_EDUCATION_TYPE    NAME_FAMILY_STATUS  NAME_HOUSING_TYPE
REGION_POPULATION_RELATIVE  DAYS_BIRTH  DAYS_EMPLOYED
DAYS_REGISTRATION  DAYS_ID_PUBLISH 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  SK_ID_PREV
NAME_CONTRACT_TYPE_y  AMT_ANNUITY_y  AMT_APPLICATION  AMT_CREDIT_y
AMT_GOODS_PRICE_y NAME_CASH_LOAN_PURPOSE NAME_CONTRACT_STATUS
DAYS_DECISION      NAME_PAYMENT_TYPE CODE_REJECT_REASON
NAME_CLIENT_TYPE NAME_GOODS_CATEGORY NAME_PORTFOLIO NAME_PRODUCT_TYPE
CHANNEL_TYPE  SELLERPLACE_AREA NAME_SELLER_INDUSTRY  CNT_PAYMENT
NAME_YIELD_GROUP          PRODUCT_COMBINATION DAYS_DECISION_GROUP
0      100002       1           Cash loans           M            N
Y              0             2.025      4.065975        24700.5
351000.0   Unaccompanied          Working  Secondary / secondary
special  Single / not married  House / apartment
0.018801        9461            637             3648.0
2120        Laborers              1.0                    2
2                  WEDNESDAY                        10
0                         0                           0
0                      0                        0  Business Entity Type
3                       2.0                       2.0
2.0                       2.0                   -1134.0                  1
0.0                        0.0                          0.0
0.0                        0.0                          1.0
200k-300k        400k-500k   25     20-30                1
0-5     1038818       Consumer loans       9251.775         179055.0
179055.0           179055.0                   XAP
Approved            606                    XNA                  XAP
New            Vehicles            POS               XNA
Stone               500      Auto technology         24.0
low_normal      POS other with interest             400-800
1      100003       0           Cash loans           F            N
N             0             2.700     12.935025        35698.5
1129500.0          Family    State servant                Higher
education               Married  House / apartment
0.003541       16765           1188             1186.0
291      Core staff              2.0                    1
1                     MONDAY                       11
0                         0                           0
0                       0                        0
School                       1.0                          0.0
1.0                      0.0                   -828.0                   1
0.0                        0.0                          0.0
0.0                        0.0                          0.0
```

```
200k-300k         1M Above   45     40-50                3
0-5     1810518           Cash loans      98356.995         900000.0
1035882.0           900000.0                     XNA
Approved            746                    XNA                XAP
Repeater                 XNA           Cash            x-sell  Credit
and cash offices               -1                  XNA         12.0
low_normal             Cash X-Sell: low             400-800
2      100003       0           Cash loans           F             N
N             0             2.700     12.935025        35698.5
1129500.0          Family    State servant               Higher
education              Married  House / apartment
0.003541       16765           1188             1186.0
291      Core staff              2.0                    1
1                      MONDAY                        11
0                           0                           0
0                       0                       0
School                        1.0                        0.0
1.0                        0.0                   -828.0                  1
0.0                         0.0                         0.0
0.0                         0.0                         0.0
200k-300k         1M Above   45     40-50                3
0-5     2636178       Consumer loans      64567.665         337500.0
348637.5           337500.0                     XAP
Approved            828  Cash through the bank               XAP
Refreshed           Furniture            POS               XNA
Stone               1400            Furniture          6.0
middle   POS industry with interest            800-1200
3      100004       0      Revolving loans           M             Y
Y             0             0.675      1.350000         6750.0
135000.0   Unaccompanied          Working  Secondary / secondary
special  Single / not married  House / apartment
0.010032       19046            225             4260.0
2531        Laborers              1.0                    2
2                      MONDAY                         9
0                           0                           0
0                       0                       0
Government                       0.0                         0.0
0.0                        0.0                   -815.0                  0
0.0                         0.0                         0.0
0.0                         0.0                         0.0
0-100K        100K-200K   52  50 above               0              NaN
1564014       Consumer loans       5357.250          24282.0
20106.0            24282.0                    XAP            Approved
815  Cash through the bank               XAP              New
Mobile            POS               XNA         Regional / Local
30         Connectivity          4.0           middle  POS mobile
without interest            800-1200
4      100006       0           Cash loans           F             N
Y             0             1.350      3.126825        29686.5
```

```
297000.0   Unaccompanied          Working  Secondary / secondary
special        Civil marriage  House / apartment
0.008019       19005           3039             9833.0
2437        Laborers              2.0                    2
2                  WEDNESDAY                        17
0                          0                           0
0                      0                         0  Business Entity Type
3                       2.0                        0.0
2.0                       0.0                   -617.0                 1
0.0                        0.0                          0.0
0.0                        0.0                          1.0
100K-200K        300k-400k   52  50 above                8
5-10     2078043           Cash loans      24246.000         675000.0
675000.0           675000.0                     XNA
Approved            181  Cash through the bank                XAP
Repeater                 XNA           Cash           x-sell  Credit
and cash offices                -1                  XNA         48.0
low_normal             Cash X-Sell: low               0-400
```

```
#Checking the details of the merged dataframe
loan_process_df.shape
```

```
(887347, 74)
```

```
# Checking the element count of the dataframe
loan_process_df.size
```

```
65663678
```

```
# checking the columns and column types of the dataframe
loan_process_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 887347 entries, 0 to 887346
Data columns (total 74 columns):
 #   Column                       Non-Null Count   Dtype
---  ------                       --------------   -----
 0   SK_ID_CURR                   887347 non-null  int64
 1   TARGET                       887347 non-null  int64
 2   NAME_CONTRACT_TYPE_x         887347 non-null  category
 3   CODE_GENDER                  887347 non-null  category
 4   FLAG_OWN_CAR                 887347 non-null  category
 5   FLAG_OWN_REALTY              887347 non-null  category
 6   CNT_CHILDREN                 887347 non-null  int64
 7   AMT_INCOME_TOTAL             887347 non-null  float64
 8   AMT_CREDIT_x                 887347 non-null  float64
 9   AMT_ANNUITY_x                887287 non-null  float64
 10  AMT_GOODS_PRICE_x            886608 non-null  float64
 11  NAME_TYPE_SUITE              887347 non-null  category
 12  NAME_INCOME_TYPE             887347 non-null  category
 13  NAME_EDUCATION_TYPE          887347 non-null  category
```

```
14  NAME_FAMILY_STATUS            887347 non-null  category
15  NAME_HOUSING_TYPE             887347 non-null  category
16  REGION_POPULATION_RELATIVE    887347 non-null  float64
17  DAYS_BIRTH                    887347 non-null  int64
18  DAYS_EMPLOYED                 887347 non-null  int64
19  DAYS_REGISTRATION             887347 non-null  float64
20  DAYS_ID_PUBLISH               887347 non-null  int64
21  OCCUPATION_TYPE               887347 non-null  category
22  CNT_FAM_MEMBERS               887347 non-null  float64
23  REGION_RATING_CLIENT          887347 non-null  category
24  REGION_RATING_CLIENT_W_CITY   887347 non-null  category
25  WEEKDAY_APPR_PROCESS_START    887347 non-null  category
26  HOUR_APPR_PROCESS_START       887347 non-null  int64
27  REG_REGION_NOT_LIVE_REGION    887347 non-null  int64
28  REG_REGION_NOT_WORK_REGION    887347 non-null  category
29  LIVE_REGION_NOT_WORK_REGION   887347 non-null  category
30  REG_CITY_NOT_LIVE_CITY        887347 non-null  category
31  REG_CITY_NOT_WORK_CITY        887347 non-null  category
32  LIVE_CITY_NOT_WORK_CITY       887347 non-null  category
33  ORGANIZATION_TYPE             887347 non-null  category
34  OBS_30_CNT_SOCIAL_CIRCLE      885364 non-null  float64
35  DEF_30_CNT_SOCIAL_CIRCLE      885364 non-null  float64
36  OBS_60_CNT_SOCIAL_CIRCLE      885364 non-null  float64
37  DEF_60_CNT_SOCIAL_CIRCLE      885364 non-null  float64
38  DAYS_LAST_PHONE_CHANGE        887347 non-null  float64
39  FLAG_DOCUMENT_3               887347 non-null  int64
40  AMT_REQ_CREDIT_BUREAU_HOUR    887347 non-null  float64
41  AMT_REQ_CREDIT_BUREAU_DAY     887347 non-null  float64
42  AMT_REQ_CREDIT_BUREAU_WEEK    887347 non-null  float64
43  AMT_REQ_CREDIT_BUREAU_MON     887347 non-null  float64
44  AMT_REQ_CREDIT_BUREAU_QRT     887347 non-null  float64
45  AMT_REQ_CREDIT_BUREAU_YEAR    887347 non-null  float64
46  AMT_INCOME_RANGE              886908 non-null  category
47  AMT_CREDIT_RANGE              887347 non-null  category
48  AGE                           887347 non-null  int64
49  AGE_GROUP                     887347 non-null  category
50  YEARS_EMPLOYED                887347 non-null  int64
51  EMPLOYMENT_YEAR               647974 non-null  category
52  SK_ID_PREV                    887347 non-null  int64
53  NAME_CONTRACT_TYPE_y          887347 non-null  category
54  AMT_ANNUITY_y                 887347 non-null  float64
55  AMT_APPLICATION               887347 non-null  float64
56  AMT_CREDIT_y                  887347 non-null  float64
57  AMT_GOODS_PRICE_y             887347 non-null  float64
58  NAME_CASH_LOAN_PURPOSE        887347 non-null  category
59  NAME_CONTRACT_STATUS          887347 non-null  category
60  DAYS_DECISION                 887347 non-null  int64
61  NAME_PAYMENT_TYPE             887347 non-null  category
62  CODE_REJECT_REASON            887347 non-null  category
```

```
 63  NAME_CLIENT_TYPE             887347 non-null  category
 64  NAME_GOODS_CATEGORY          887347 non-null  category
 65  NAME_PORTFOLIO               887347 non-null  category
 66  NAME_PRODUCT_TYPE            887347 non-null  category
 67  CHANNEL_TYPE                 887347 non-null  category
 68  SELLERPLACE_AREA             887347 non-null  int64
 69  NAME_SELLER_INDUSTRY         887347 non-null  category
 70  CNT_PAYMENT                  887347 non-null  float64
 71  NAME_YIELD_GROUP             887347 non-null  category
 72  PRODUCT_COMBINATION          887143 non-null  category
 73  DAYS_DECISION_GROUP          887347 non-null  category
dtypes: category(37), float64(23), int64(14)
memory usage: 281.8 MB

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

          SK_ID_CURR         TARGET   CNT_CHILDREN  AMT_INCOME_TOTAL
AMT_CREDIT_x  AMT_ANNUITY_x  AMT_GOODS_PRICE_x
REGION_POPULATION_RELATIVE     DAYS_BIRTH  DAYS_EMPLOYED
DAYS_REGISTRATION  DAYS_ID_PUBLISH  CNT_FAM_MEMBERS
HOUR_APPR_PROCESS_START  REG_REGION_NOT_LIVE_REGION
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            AGE  YEARS_EMPLOYED
SK_ID_PREV  AMT_ANNUITY_y  AMT_APPLICATION  AMT_CREDIT_y
AMT_GOODS_PRICE_y  DAYS_DECISION  SELLERPLACE_AREA    CNT_PAYMENT
count  887347.000000  887347.000000  887347.000000     887347.000000
887347.000000  887287.000000       8.866080e+05
887347.000000  887347.000000  887347.000000      887347.000000
887347.000000    887347.000000            887347.000000
887347.000000             885364.000000             885364.000000
885364.000000             885364.000000           887347.000000
887347.000000               887347.000000             887347.000000
887347.000000              887347.000000             887347.000000
887347.000000  887347.000000   887347.000000  8.873470e+05
887347.000000     8.873470e+05  8.873470e+05       8.873470e+05
887347.000000      8.873470e+05  887347.000000
mean   278573.355245       0.086533       0.404863          1.735124
5.875341   27014.933342       5.276874e+05                   0.020758
16316.688747   72698.338446        5003.100439      3034.758355
2.150632                11.982459                    0.012007
1.542822                  0.153992                   1.524985
0.107933            -1083.148511         0.737876
0.005373                   0.005909                    0.034321
0.265025                   0.318782                    2.688320
44.202104      198.645220  1.922461e+06   14795.547439
```

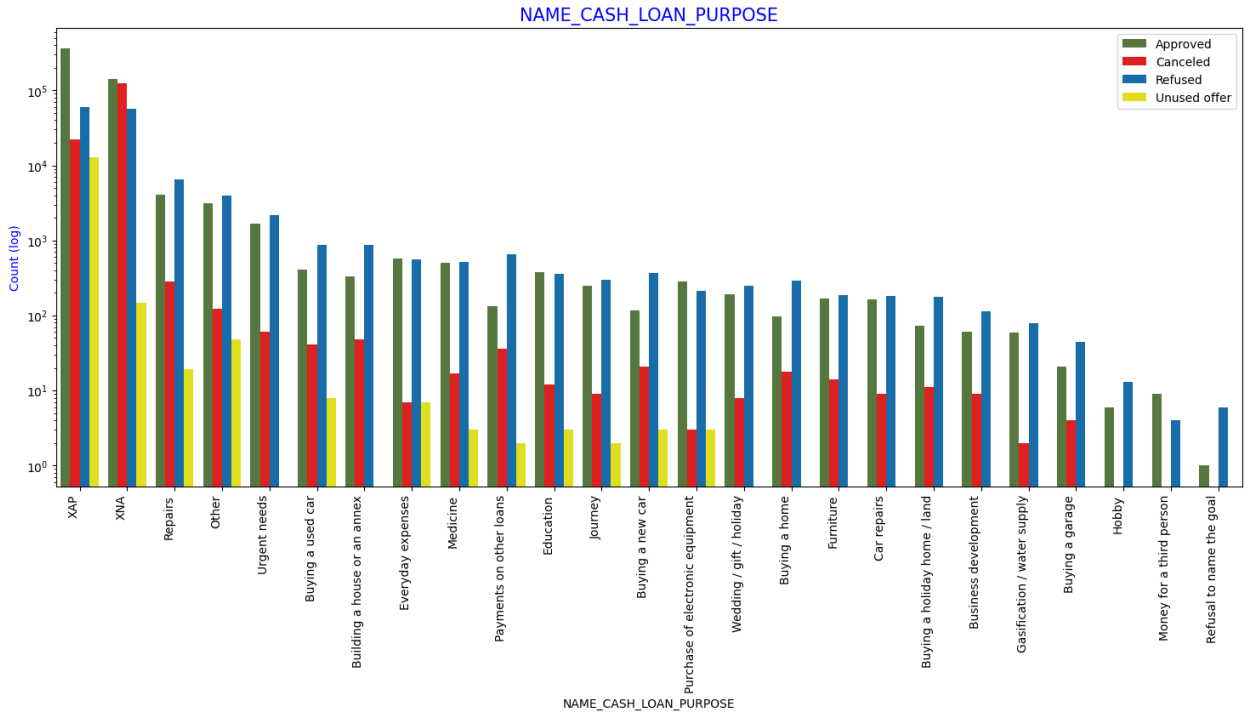
```
1.743646e+05  1.953514e+05       1.845025e+05     881.570688
3.200226e+02      12.536602
std    102862.966629       0.281150       0.716925          2.388415
3.851740   13958.001832       3.533979e+05                    0.013363
4348.574147  143365.009447        3550.314322      1507.730283
0.900076                3.234360                    0.108915
2.509418                 0.464619                   2.487924
0.378546              799.371187         0.439790
0.075963                   0.098344                    0.201901
0.921289                   0.968889                    2.157655
11.912967      392.713628  5.330660e+05   13131.959867
2.919124e+05  3.179320e+05       2.864381e+05     783.545077
8.643867e+03      14.439936
min    100002.000000       0.000000       0.000000          0.256500
0.450000    1615.500000       4.050000e+04                    0.000290
7489.000000       0.000000           0.000000         0.000000
1.000000                0.000000                    0.000000
0.000000                 0.000000                   0.000000
0.000000            -4292.000000         0.000000
0.000000                   0.000000                    0.000000
0.000000                   0.000000                    0.000000
20.000000        0.000000  1.000001e+06       0.000000
0.000000e+00  0.000000e+00       0.000000e+00       2.000000     -
1.000000e+00       0.000000
25%    189427.000000       0.000000       0.000000          1.125000
2.700000   16807.500000       2.385000e+05                    0.010032
12726.000000    1041.000000        2002.000000      1781.000000
2.000000               10.000000                    0.000000
0.000000                 0.000000                   0.000000
0.000000            -1681.000000         0.000000
0.000000                   0.000000                    0.000000
0.000000                   0.000000                    1.000000
34.000000        2.000000  1.460106e+06    7376.085000
2.000250e+04  2.505600e+04       4.500000e+04     272.000000     -
1.000000e+00       0.000000
50%    279118.000000       0.000000       0.000000          1.575000
5.084955   24907.500000       4.500000e+05                    0.018850
16036.000000    2399.000000        4507.000000      3331.000000
2.000000               12.000000                    0.000000
0.000000                 0.000000                   0.000000
0.000000            -1008.000000         1.000000
0.000000                   0.000000                    0.000000
0.000000                   0.000000                    2.000000
43.000000        6.000000  1.923040e+06   11250.000000
7.071750e+04  8.048250e+04       7.071750e+04     584.000000
5.000000e+00      10.000000
75%    367786.000000       0.000000       1.000000          2.070000
8.086500   34537.500000       6.795000e+05                    0.028663
19980.000000    6319.000000        7512.000000      4319.000000
```

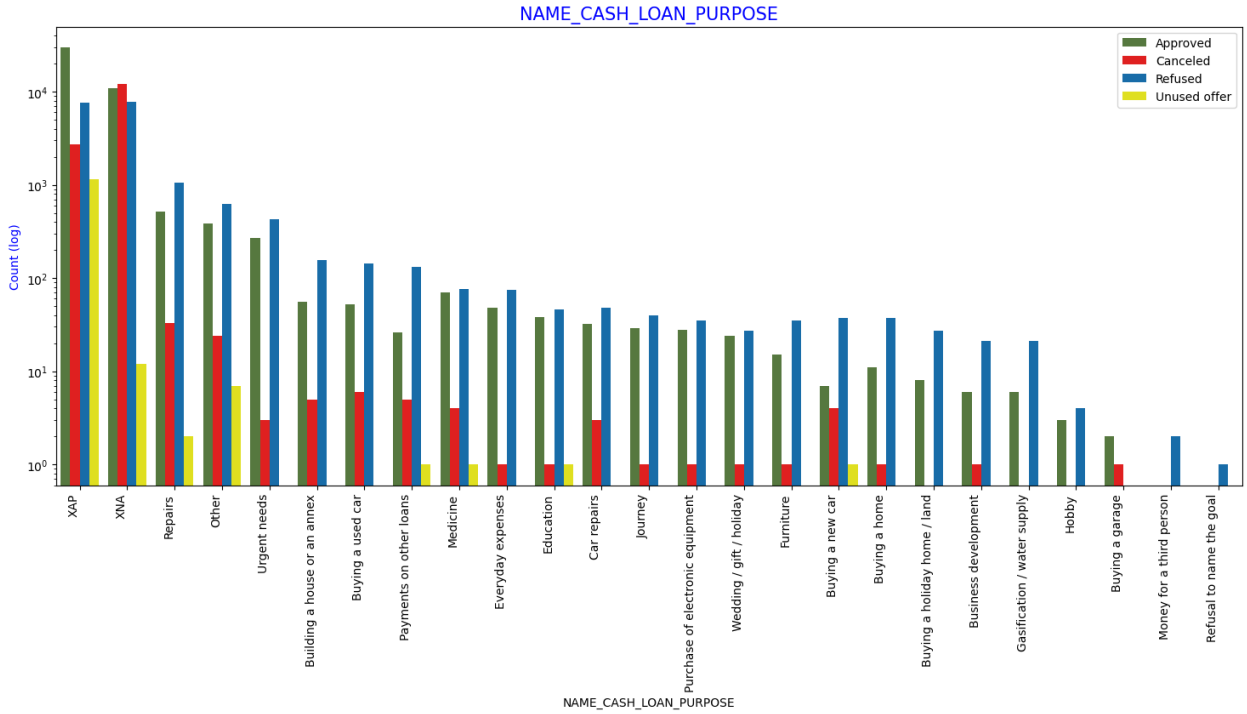
```
3.000000                14.000000                    0.000000
2.000000                  0.000000                   2.000000
0.000000             -395.000000         1.000000
0.000000                  0.000000                    0.000000
0.000000                  0.000000                    4.000000
54.000000       17.000000  2.384212e+06   16676.910000
1.800000e+05  2.126250e+05       1.800000e+05    1315.000000
8.900000e+01      16.000000
max    456255.000000       1.000000      19.000000       1170.000000
39.562740  220297.500000       3.825000e+06
0.072508   25201.000000  365243.000000       23416.000000
7197.000000        20.000000                23.000000
1.000000               348.000000                34.000000
344.000000                24.000000                0.000000
1.000000                  4.000000                   9.000000
8.000000                 27.000000                  261.000000
25.000000      69.000000     1000.000000  2.845381e+06  418058.145000
4.050000e+06  4.104351e+06       4.050000e+06    2922.000000
4.000000e+06      84.000000
```

```
# 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
```

Plotting Contract Status vs purpose of the loan

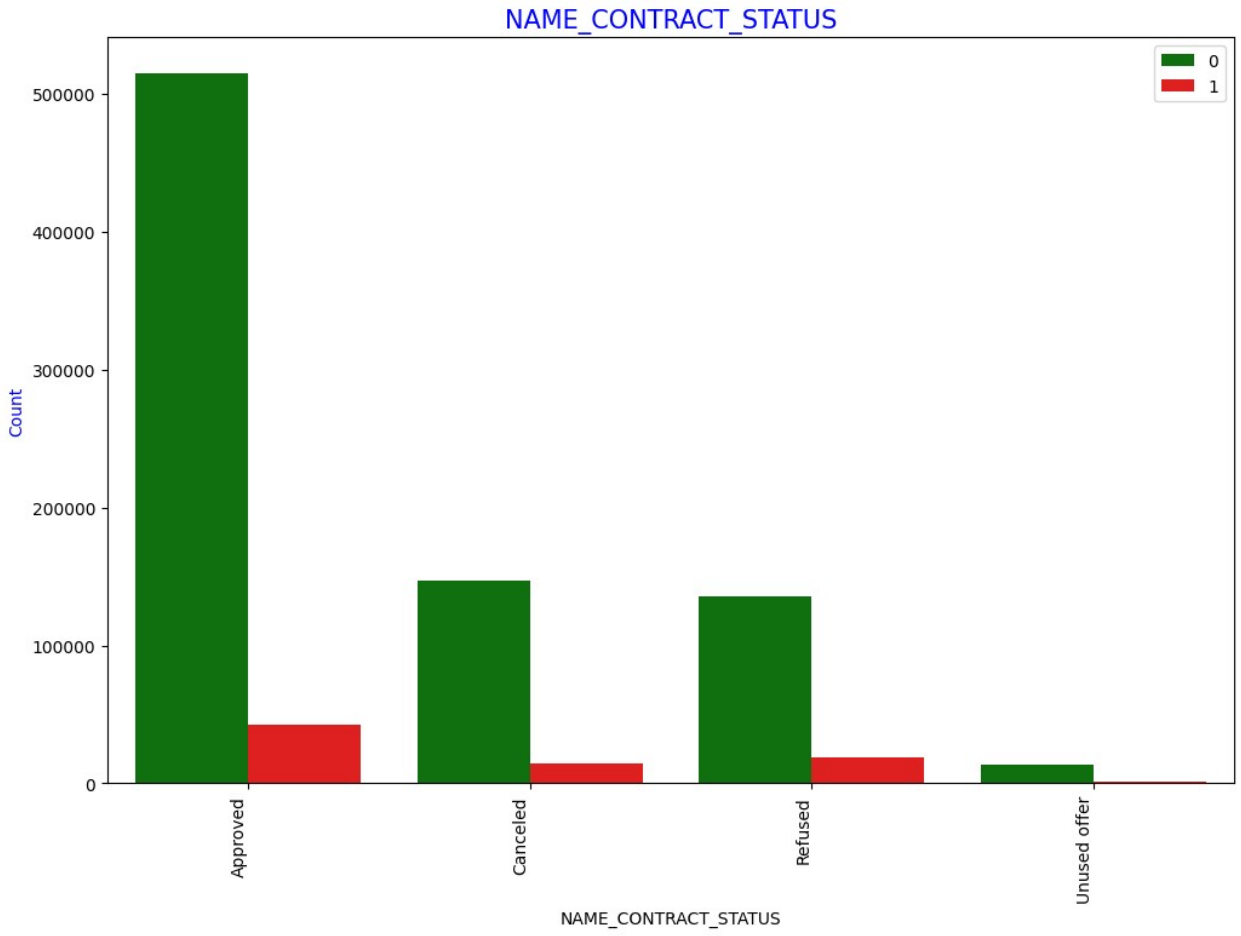
```
univariate_merged("NAME_CASH_LOAN_PURPOSE",L0,"NAME_CONTRACT_STATUS",
["#548235","#FF0000","#0070C0","#FFFF00"],True,(18,7))

univariate_merged("NAME_CASH_LOAN_PURPOSE",L1,"NAME_CONTRACT_STATUS",
["#548235","#FF0000","#0070C0","#FFFF00"],True,(18,7))
```





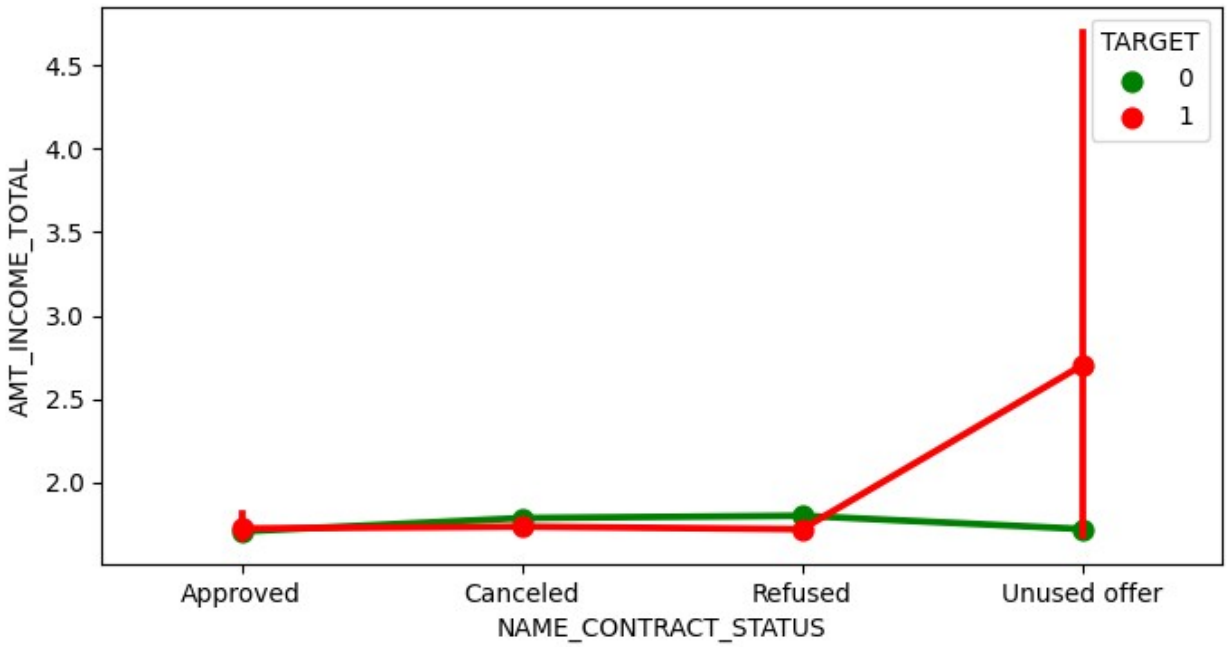

```
# Checking the Contract Status based on loan repayment status and
whether there is any business loss or financial loss
univariate_merged("NAME_CONTRACT_STATUS",loan_process_df,"TARGET",
['g','r'],False,(12,8))
g = loan_process_df.groupby("NAME_CONTRACT_STATUS")["TARGET"]
df1 =
```

```
pd.concat([g.value_counts(),round(g.value_counts(normalize=True).mul(1
00),2)],axis=1, keys=('Counts','Percentage'))
df1['Percentage'] = df1['Percentage'].astype(str) +"%" # adding
percentage symbol in the results for understanding
print (df1)
```

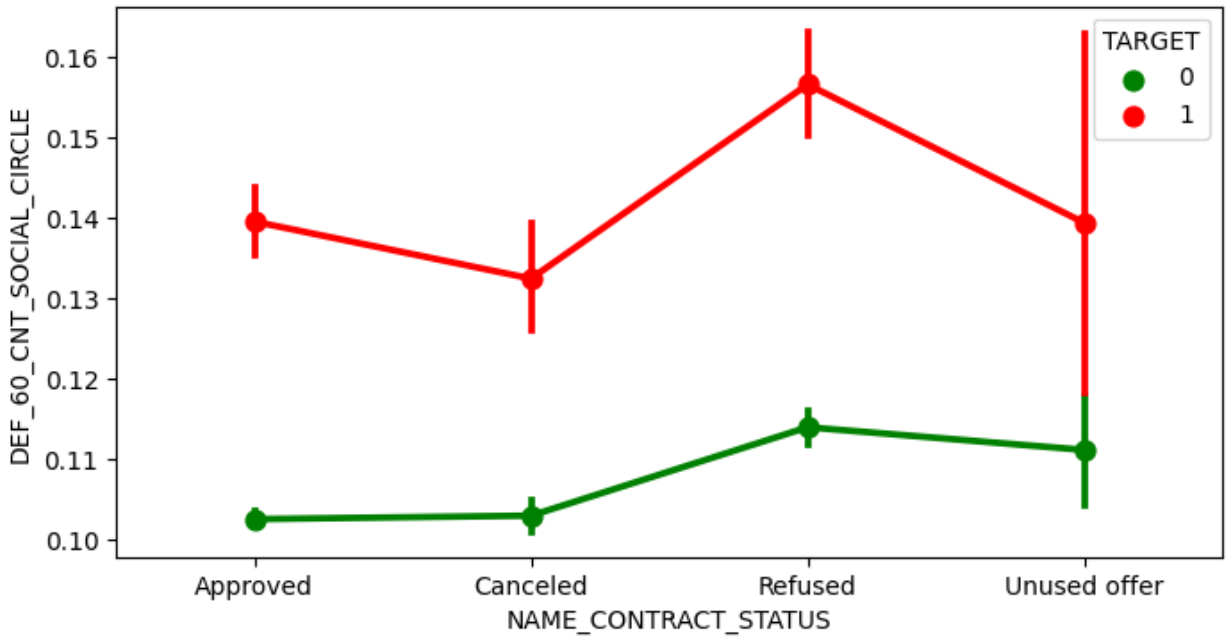


		Counts	Percentage
NAME_CONTRACT_STATUS	TARGET		
Approved	0	515220	92.41%
	1	42313	7.59%
Canceled	0	146729	90.82%
	1	14823	9.18%
Refused	0	135396	88.0%
	1	18463	12.0%
Unused offer	0	13217	91.77%
	1	1186	8.23%

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




```
# 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')
```



Conclusions

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

#NAME_FAMILY_STATUS : People who have civil marriage or who are single
default a lot.
#NAME_EDUCATION_TYPE: People with Lower Secondary & Secondary
education
#NAME_INCOME_TYPE: Clients who are either at Maternity leave OR
Unemployed default a lot.
#REGION_RATING_CLIENT: People who live in Rating 3 has highest
defaults.
#OCCUPATION_TYPE: Avoid Low-skill Laborers, Drivers and Waiters/barmen
staff, Security staff, Laborers and Cooking staff as the default rate
is huge.
#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%). 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.
#DAYS_BIRTH: Avoid young people who are in age group of 20-40 as they
have higher probability of defaulting
#DAYS_EMPLOYED: People who have less than 5 years of employment have
high default rate.
#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.
#AMT_GOODS_PRICE: When the credit amount goes beyond 3M, there is an
increase in defaulters.
```