BANK LOAN CASE STUDY

BY SUJAL VERMA

PROJECT DESCRIPTION

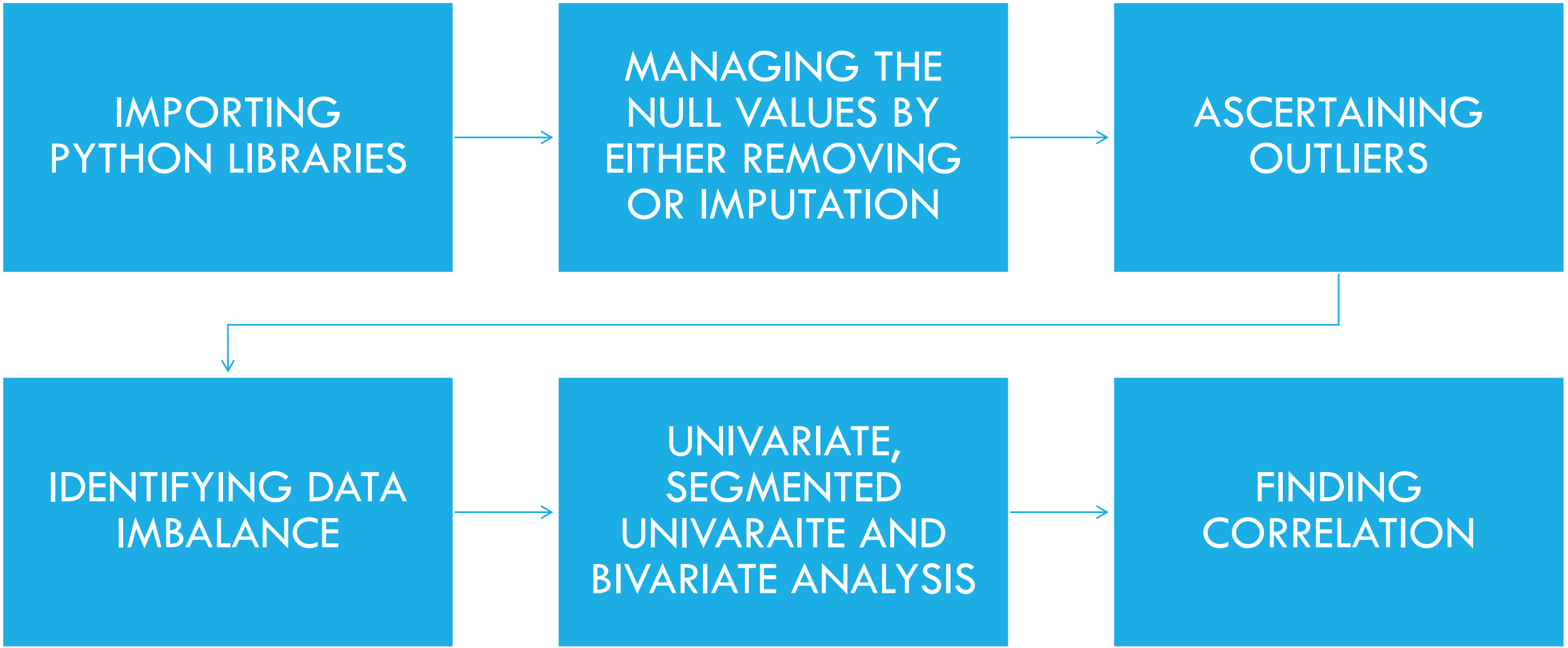
The given project asks us to apply Risk Analytics by analysing the data given for bank loan.

The project expects us to perform Exploratory Data Analysis to understand how consumer attribute which influence the tendency to default.

By EDA we will identify patterns of loan default and the driving factors which leads to these defaults. For this which has strong indicators of default are found.

We are expected to manage missing data, identify outliers, report any data imbalance, explain the result of univariate,segmented univariate, bivariate analysis and determine the top 10 correlation for the client with payment difficulties and all other cases.

APPROACH


IMPORTING PYTHON LIBRARIES
MANAGING THE NULL VALUES BY EITHER REMOVING OR IMPUTATION
ASCERTAINING OUTLIERS
IDENTIFYING DATA IMBALANCE
UNIVARIATE, SEGMENTED UNIVARAITE AND BIVARIATE ANALYSIS
FINDING CORRELATION

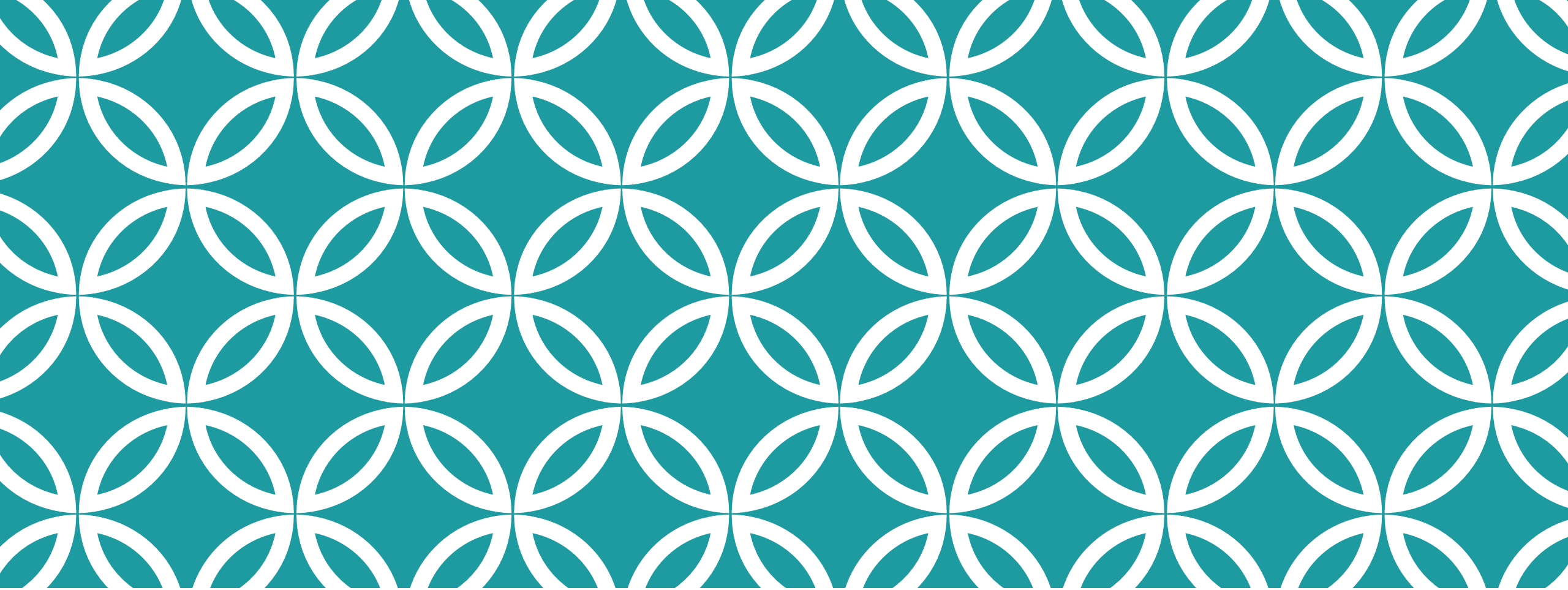
TECH STACH USED

JUPYTER Labs was used because they can help in easy python compilations and visualisation.

The file can be viewed in the link below:-

Click here to see the IPYNB file

Due to size constraint please don't hesitate to view the visualisations in the IPYNB file.



INSIGHTS

TASK 1-DATA CLEANING AND MANIPULATION

- FINDING AND HANDLING MISSING DATA
- REMOVING UNWANTED COLUMNS
- DROPPING COLUMNS WITH HIGH MISSING VALUE PERCENTAGE
- DATA IMPUTATION IN IMPORTANT COLUMNS WHICH CAN NOT BE DROPPED AND DATA WITH LOW COUNT OF NULL VALUES
- STANDARDISING VALUES

FINDING AND HANDLING MISSING DATA DATASET -1 ->APPLICATION DATA

Number of Rows =307511

Number of Columns=122

```
#checking rows and columns
print ("application_data:",apd.shape)
```

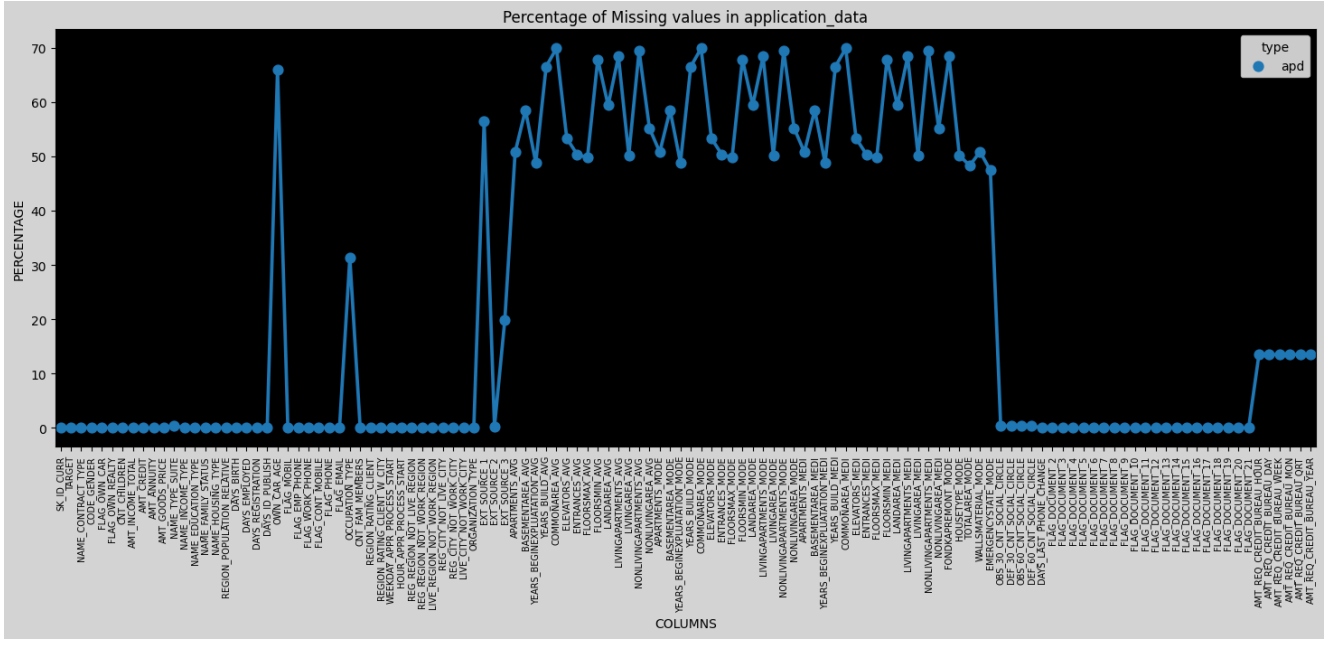
```
application_data: (307511, 122)
```

There are some data set related to days which is negative would have to manage that

The given line plot shows the columns and number of null values(extracted Jupyter File attached in Tech

We will drop those columns with missing greater than 40%

As per Industrial Standard, max Threshold limit can be between 40% to 50 % depending acquired in specific sector



There are 49 col with null values greater than 40%
After dropping 49 col new shape is 307511 rows and 73 columns

```
apd.shape
```

```
(307511, 73)
```

Remaining columns with null value

```
nullval(apd)[nullval(apd)>0]
```

```
OCCUPATION_TYPE               31.35
EXT_SOURCE_3                  19.83
AMT_REQ_CREDIT_BUREAU_YEAR    13.50
AMT_REQ_CREDIT_BUREAU_QRT     13.50
AMT_REQ_CREDIT_BUREAU_MON     13.50
AMT_REQ_CREDIT_BUREAU_WEEK    13.50
AMT_REQ_CREDIT_BUREAU_DAY     13.50
AMT_REQ_CREDIT_BUREAU_HOUR    13.50
NAME_TYPE_SUITE                0.42
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
EXT_SOURCE_2                   0.21
AMT_GOODS_PRICE                0.09
dtype: float64
```

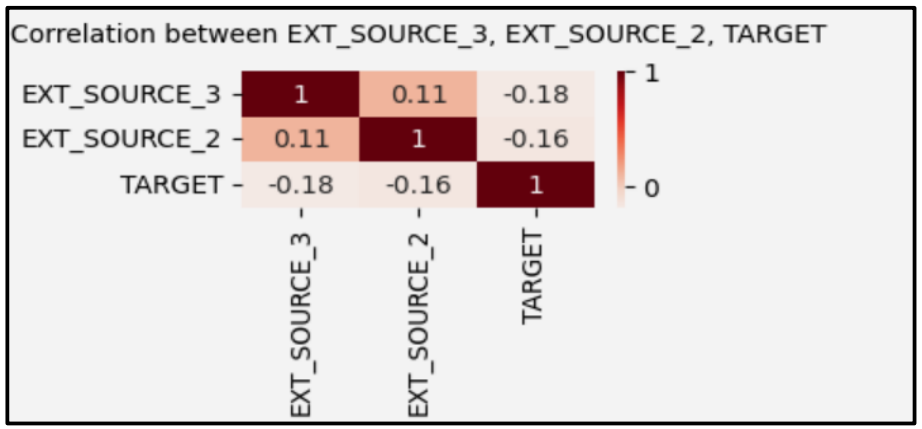
As in the dictionary we see a normalised data set of ext source 1, because ext source 2 and ext source 3 have no linear correlation with target the the column ext source 2 and ext source 3 are dropped
After dropping we have 71 columns

```
(307511, 71)
```

```
#creating a variable null_col_40 for storing null columns having missing values more than 40%
null_col_40 = nullval(apd)[nullval(apd)>40]
print("following columns have null value more than 40%")
print(null_col_40.index)
print("No. of columns with more than 40% missing value:",len(null_col_40.index))
```

```
following columns have null value more than 40%
Index(['COMMONAREA_MEDI', 'COMMONAREA_AVG', 'COMMONAREA_MODE',
       'NONLIVINGAPARTMENTS_MODE', 'NONLIVINGAPARTMENTS_AVG',
       'NONLIVINGAPARTMENTS_MEDI', 'FONDKAPREMONT_MODE',
       'LIVINGAPARTMENTS_MODE', 'LIVINGAPARTMENTS_AVG',
       'LIVINGAPARTMENTS_MEDI', 'FLOORSMIN_AVG', 'FLOORSMIN_MODE',
       'FLOORSMIN_MEDI', 'YEARS_BUILD_MEDI', 'YEARS_BUILD_MODE',
       'YEARS_BUILD_AVG', 'OWN_CAR_AGE', 'LANDAREA_MEDI', 'LANDAREA_MODE',
       'LANDAREA_AVG', 'BASEMENTAREA_MEDI', 'BASEMENTAREA_AVG',
       'BASEMENTAREA_MODE', 'EXT_SOURCE_1', 'NONLIVINGAREA_MODE',
       'NONLIVINGAREA_AVG', 'NONLIVINGAREA_MEDI', 'ELEVATORS_MEDI',
       'ELEVATORS_AVG', 'ELEVATORS_MODE', 'WALLSMATERIAL_MODE',
       'APARTMENTS_MEDI', 'APARTMENTS_AVG', 'APARTMENTS_MODE',
       'ENTRANCES_MEDI', 'ENTRANCES_AVG', 'ENTRANCES_MODE', 'LIVINGAREA_AVG',
       'LIVINGAREA_MODE', 'LIVINGAREA_MEDI', 'HOUSETYPE_MODE',
       'FLOORSMAX_MODE', 'FLOORSMAX_MEDI', 'FLOORSMAX_AVG',
       'YEARS_BEGINEXPLUATATION_MODE', 'YEARS_BEGINEXPLUATATION_MEDI',
       'YEARS_BEGINEXPLUATATION_AVG', 'TOTALAREA_MODE', 'EMERGENCYSTATE_MODE'],
      dtype='object')
No. of columns with more than 40% missing value: 49
```

```
#removing extra columns
#putting irrelevent columns in 1 variable
irrev = ["EXT_SOURCE_3","EXT_SOURCE_2"]
#making a heatmap
plt.figure(figsize= [3,1])
sns.heatmap(apd[irrev+["TARGET"]].corr(), cmap="Reds",annot=True)
plt.title("Correlation between EXT_SOURCE_3, EXT_SOURCE_2, TARGET", fontdict={"fontsize":10}, pad=12)
plt.show()
```



Checking column with FLAGS and determining their relationship with target column

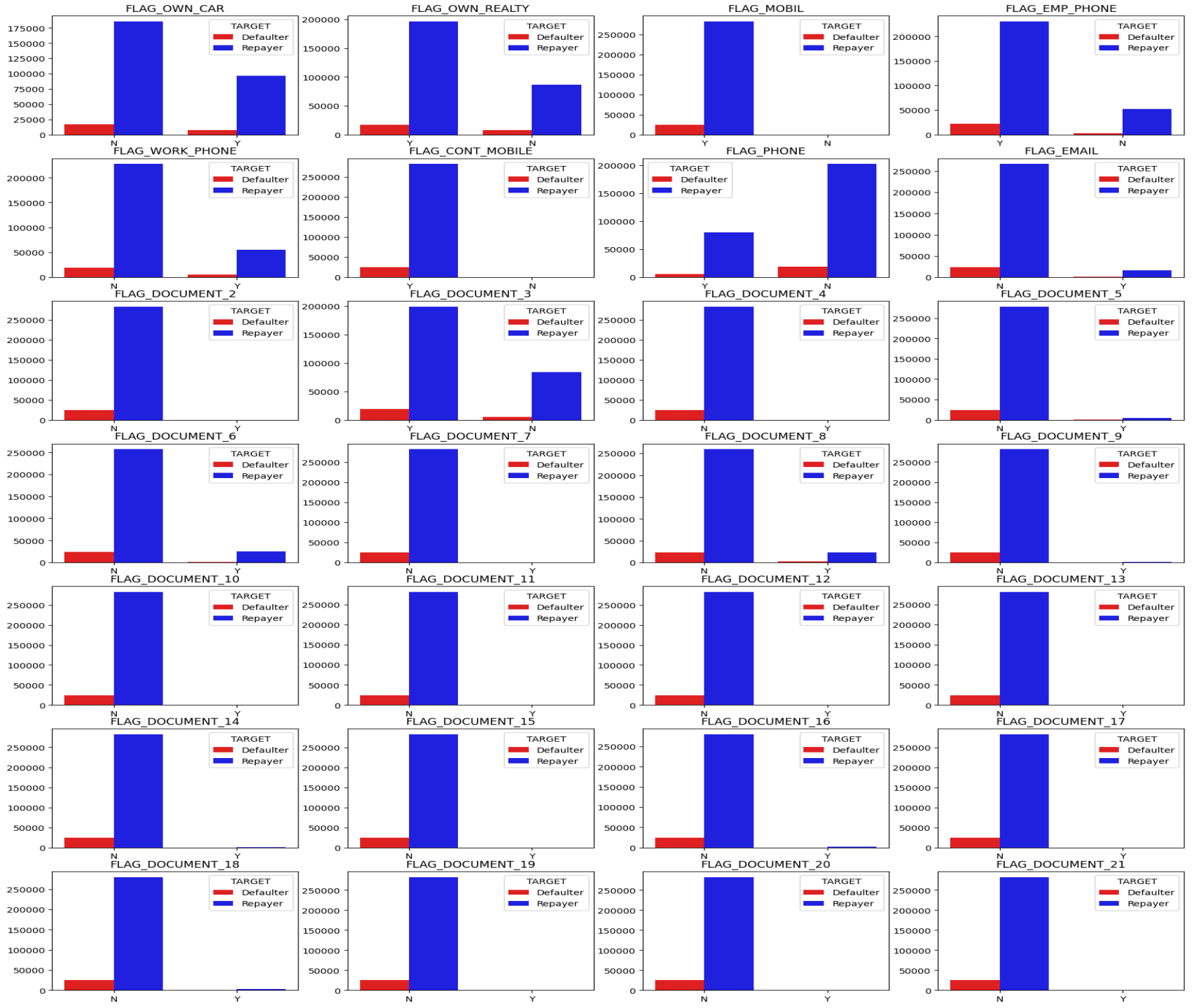
Then checking correlation with target and removing unwanted columns

Columns (FLAG_OWN_REALTY, FLAG_MOBIL ,FLAG_EMP_PHONE, FLAG_CONT_MOBILE, FLAG_DOCUMENT_3) have more **yeses** than **nos** and from these keeping FLAG_DOCUMENT_3,FLAG_OWN_REALTY, FLAG_MOBIL more sense thus we can include these columns and remove all other FLAG columns for further analysis.

Hence we drop these columns and after removing all unnecessary columns now we have 46 relevant columns

```
apd.shape
```

```
(307511, 46)
```



DATA IMPUTATION-DATASET1 APPLICATION DATA

- In the column occupation type there are 31.35% null values hence we will change those to 'Unknown', this unknown column has the highest percentage.
- Similarly in the column Name Type Suite, the missing values will be replaced with "Unaccompanied" which is the mode of the data.

For rest of the columns with null value we will replace them with their median values.

```
apd["OCCUPATION_TYPE"]=apd["OCCUPATION_TYPE"].fillna("Unknown")
```

```
apd["OCCUPATION_TYPE"].isnull().sum() #zero null value left
```

```
0
```

```
apd["NAME_TYPE_SUITE"].value_counts()
```

```
Unaccompanied      248526
Family              40149
Spouse, partner     11370
Children             3267
Other_B              1770
Other_A               866
Group of people       271
Name: NAME_TYPE_SUITE, dtype: int64
```

```
apd["NAME_TYPE_SUITE"]=apd["NAME_TYPE_SUITE"].fillna("Unaccompanied")
```

```
apd[amt_credit].median()
```

```
AMT_REQ_CREDIT_BUREAU_YEAR    1.0
AMT_REQ_CREDIT_BUREAU_QRT     0.0
AMT_REQ_CREDIT_BUREAU_MON     0.0
AMT_REQ_CREDIT_BUREAU_WEEK    0.0
AMT_REQ_CREDIT_BUREAU_DAY     0.0
AMT_REQ_CREDIT_BUREAU_HOUR    0.0
dtype: float64
```

```
apd.fillna(apd[amt_credit].median(),inplace=True)
```

```
apd[a].median()
```

```
DEF_60_CNT_SOCIAL_CIRCLE    0.0
OBS_60_CNT_SOCIAL_CIRCLE    0.0
DEF_30_CNT_SOCIAL_CIRCLE    0.0
OBS_30_CNT_SOCIAL_CIRCLE    0.0
dtype: float64
```

```
nullval(apd).head()
```

```
AMT_GOODS_PRICE             0.09
AMT_ANNUITY                 0.00
CNT_FAM_MEMBERS             0.00
DAYS_LAST_PHONE_CHANGE      0.00
SK_ID_CURR                  0.00
dtype: float64
```

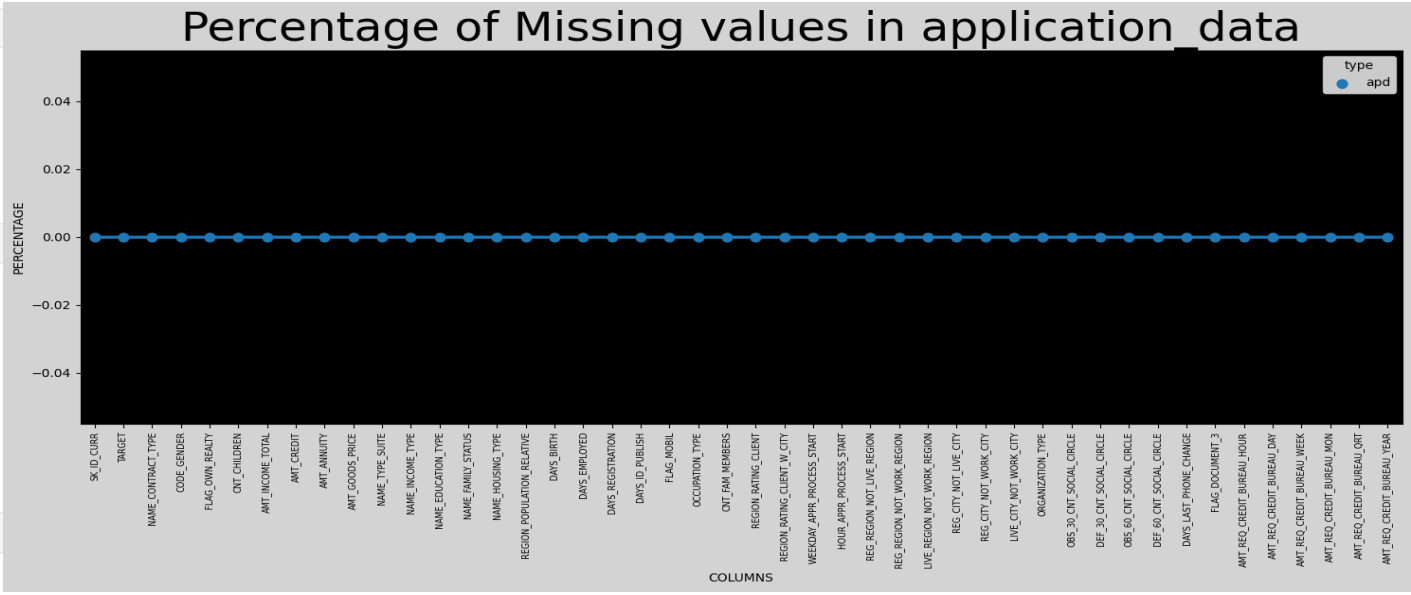
```
apd["AMT_GOODS_PRICE"].describe()
```

```
count    3.072330e+05
mean     5.383962e+05
std      3.694465e+05
min      4.050000e+04
25%      2.385000e+05
50%      4.500000e+05
75%      6.795000e+05
max      4.050000e+06
Name: AMT_GOODS_PRICE, dtype: float64
```

```
apd["AMT_GOODS_PRICE"].isnull().sum()
```

```
278
```

```
apd.fillna(apd["AMT_GOODS_PRICE"].median(),inplace=True)
```



WE CAN SEE NO COLUMN WITH NULL VALUES

STANDARDIZING VALUES – DATASET-1 APPLICATION DATA

STANDARDIZING VALUES – DATASET-1 APPLICATION DATA

```
# Creating bins for Credit amount in term of Lakhs
apd['AMT_CREDIT']=apd['AMT_CREDIT']/100000

bins = [0,1,2,3,4,5,6,7,8,9,10,41] #40.5 is the highest amount
slots = ['0-1L','1L-2L', '2L-3L','3L-4L','4L-5L','5L-6L','6L-7L','7L-8L','8L-9L','9L-10L','10L Above']

apd['AMT_CREDIT_RANGE']=pd.cut(apd['AMT_CREDIT'],bins=bins,labels=slots)
```

```
apd['AMT_CREDIT_RANGE'].value_counts(normalize=True)*100
```

```
2L-3L        17.824728
10L Above    16.254703
5L-6L        11.131960
4L-5L        10.418489
1L-2L         9.801275
3L-4L         8.564897
6L-7L         7.820533
8L-9L         7.086576
7L-8L         6.241403
9L-10L        2.902986
0-1L          1.952450
Name: AMT_CREDIT_RANGE, dtype: float64
```

```
#STANDARDIZING VALUES
# Binning Numerical Columns to create a categorical column
# Creating bins for income amount in term of Lakhs
apd['AMT_INCOME_TOTAL']=apd['AMT_INCOME_TOTAL']/100000
```

```
bins = [0,1,2,3,4,5,6,7,8,9,10,1170]#1170 is the largest value
slot = ['0-1L','1L-2L', '2L-3L','3L-4L','4L-5L','5L-6L','6L-7L','7L-8L','8L-9L','9L-10L','10L Above']

apd['AMT_INCOME_RANGE']=pd.cut(apd['AMT_INCOME_TOTAL'],bins,labels=slot)
```

```
apd["AMT_INCOME_RANGE"].value_counts(normalize=True)*100
```

```
1L-2L        50.696723
2L-3L        21.194689
0-1L         20.714056
3L-4L         4.772512
4L-5L         1.743352
5L-6L         0.356085
6L-7L         0.282592
8L-9L         0.096907
10L Above     0.081298
7L-8L         0.052681
9L-10L        0.009105
Name: AMT_INCOME_RANGE, dtype: float64
```

```
# Creating bins for Price of Goods in term of Lakhs
apd['AMT_GOODS_PRICE']=apd['AMT_GOODS_PRICE']/100000

bins = [0,1,2,3,4,5,6,7,8,9,10,41] #40.5 is the highest value
slots = ['0-1L','1L-2L', '2L-3L','3L-4L','4L-5L','5L-6L','6L-7L','7L-8L','8L-9L','9L-10L','10L Above']

apd['AMT_GOODS_PRICE_RANGE']=pd.cut(apd['AMT_GOODS_PRICE'],bins=bins,labels=slots)
```

```
apd['AMT_GOODS_PRICE_RANGE'].value_counts(normalize=True)*100
```

```
2L-3L        20.409351
4L-5L        18.617545
6L-7L        13.015469
10L Above    11.095213
1L-2L        10.717015
8L-9L         6.986417
3L-4L         6.900241
5L-6L         4.265538
0-1L          2.832094
7L-8L         2.637304
9L-10L        2.523812
Name: AMT_GOODS_PRICE_RANGE, dtype: float64
```

STANDARDIZING VALUES –DATASET-1 APPLICATION DATA

columns DAYS_BIRTH, DAYS_EMPLOYED, DAYS_REGISTRATION, DAYS_ID_PUBLISH, DAYS_LAST_PHONE_CHANGE which counts days have negative values. thus will correct those values

	DAYS_BIRTH	DAYS_EMPLOYED	DAYS_REGISTRATION	DAYS_ID_PUBLISH	DAYS_LAST_PHONE_CHANGE
count	307511.000000	307511.000000	307511.000000	307511.000000	307511.000000
mean	-16036.995067	63815.045904	-4986.120328	-2994.202373	-961.392295
std	4363.988632	141275.766519	3522.886321	1509.450419	1159.717257
min	-25229.000000	-17912.000000	-24672.000000	-7197.000000	-4292.000000
25%	-19682.000000	-2760.000000	-7479.500000	-4299.000000	-1570.000000
50%	-15750.000000	-1213.000000	-4504.000000	-3254.000000	-757.000000
75%	-12413.000000	-289.000000	-2010.000000	-1720.000000	-274.000000
max	-7489.000000	365243.000000	0.000000	0.000000	450000.000000

```
apd[days_col]=abs(apd[days_col])
apd[days_col].describe()
```

	DAYS_BIRTH	DAYS_EMPLOYED	DAYS_REGISTRATION	DAYS_ID_PUBLISH	DAYS_LAST_PHONE_CHANGE
count	307511.000000	307511.000000	307511.000000	307511.000000	307511.000000
mean	16036.995067	67724.742149	4986.120328	2994.202373	964.319019
std	4363.988632	139443.751806	3522.886321	1509.450419	1157.284784
min	7489.000000	0.000000	0.000000	0.000000	0.000000
25%	12413.000000	933.000000	2010.000000	1720.000000	274.000000
50%	15750.000000	2219.000000	4504.000000	3254.000000	757.000000
75%	19682.000000	5707.000000	7479.500000	4299.000000	1570.000000
max	25229.000000	365243.000000	24672.000000	7197.000000	450000.000000

STANDARDIZING VALUES –DATASET-1 APPLICATION DATA

convert DAYS_BIRTH to AGE in years , DAYS_EMPLOYED to YEARS EMPLOYED

```
#days brith to age
apd["AGE"] = abs(apd["DAYS_BIRTH"]/365)
bins = [0,20,25,30,35,40,45,50,55,60,100]
slots = ["0-20","20-25","25-30","30-35","35-40","40-45","45-50","50-55","55-60","60 Above"]

apd["AGE_GROUP"] = pd.cut(apd["AGE"], bins=bins, labels=slots)
```

```
apd["AGE_GROUP"].value_counts(normalize=True)*100
```

```
35-40       13.940314
40-45       13.464884
30-35       12.825557
60 Above    11.569993
45-50       11.425608
50-55       11.362846
55-60       10.770346
25-30       10.686447
20-25        3.954005
0-20         0.000000
Name: AGE_GROUP, dtype: float64
```

```
#creating column "EMPLOYEMENT_YEARS" from "DAYS_EMPLOYED"

apd["YEARS_EMPLOYED"] = apd["DAYS_EMPLOYED"]/365
bins = [0,5,10,15,20,25,30,1001] #1000.5 was largest value
slots = ["0-5","5-10","10-15","15-20","20-25","25-30","30 Above"]

apd["EMPLOYEMENT_YEARS"] = pd.cut(apd["YEARS_EMPLOYED"], bins=bins, labels=slots)
```

```
apd["EMPLOYEMENT_YEARS"].value_counts(normalize=True)*100
```

```
0-5         44.326833
5-10        21.095968
30 Above    18.984485
10-15        8.958762
15-20        3.528027
20-25        2.030184
25-30        1.075741
Name: EMPLOYEMENT_YEARS, dtype: float64
```

FINDING AND HANDLING MISSING DATA DATASET -2 ->PREVIOUS APPLICATION

Number of Rows =1670214

Number of Columns=37

```
print("Previous Application:", pra.shape)
```

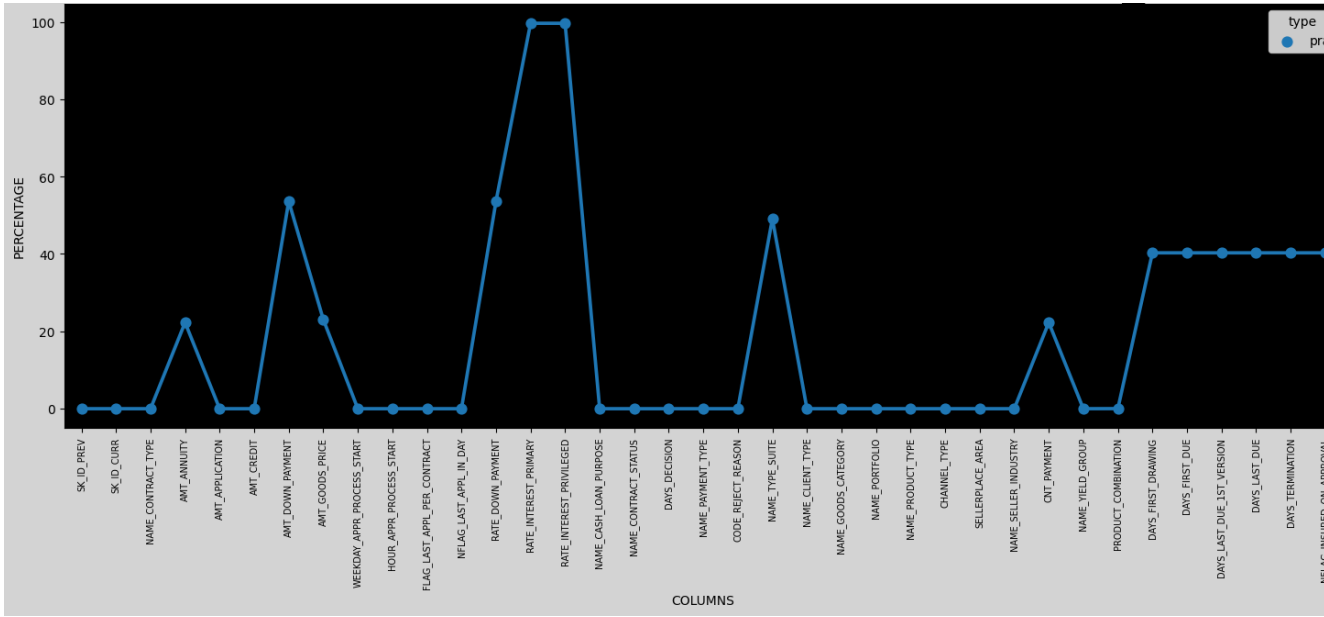
```
Previous Application: (1670214, 37)
```

There are some data set related to days which is negative would have to manage that

The given line plot shows the columns and number of null values(extracted Jupyter File attached in Tech

We will drop those columns with missing greater than 50%

As per Industrial Standard, max Threshold limit can be between 40% to 50 % depending acquired in specific sector



DROPPING UNWANTED COLUMNS

Removed 4 columns with null values greater than 50% , now there are 33 columns left
Dropped 4 more columns which were not necessary for the analysis, 29 columns Left

```
# Listing down columns which are not needed

Unnecessary_col = ['WEEKDAY_APPR_PROCESS_START','HOUR_APPR_PROCESS_START','FLAG_LAST_APPL_PER_CONTRACT',
                   'NFLAG_LAST_APPL_IN_DAY']
#pra.drop(Unnecessary_col,axis =1, inplace = True)

pra.shape
```

```
(1670214, 29)
```

```
pnullcol50=nullval(pra)[nullval(pra)>50]
```

```
pnullcol50
```

```
RATE_INTEREST_PRIVILEGED    99.64
RATE_INTEREST_PRIMARY       99.64
AMT_DOWN_PAYMENT            53.64
RATE_DOWN_PAYMENT           53.64
dtype: float64
```

```
pra.drop(columns=pnullcol50.index,inplace=True)
```

```
pra.shape
```

```
(1670214, 33)
```

DATA IMPUTATION

In column Name_type_suit, unknown values filled

```
# IMputing values "Unknown" as this a categorical column
pra["NAME_TYPE_SUITE"] = pra["NAME_TYPE_SUITE"].fillna("Unknown")

pra["NAME_TYPE_SUITE"].value_counts(normalize=True)*100
```

```
Unknown             49.119754
Unaccompanied       30.473341
Family              12.768603
Spouse, partner      4.015593
Children             1.889937
Other_B              1.055194
Other_A              0.543463
Group of people      0.134115
Name: NAME_TYPE_SUITE, dtype: float64
```

In column N_FLAG_INSURED_ON_APPROVAL null values were filled with mode i.e. 0

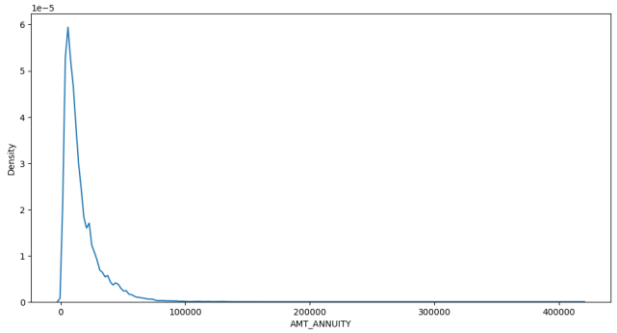
```
pra['NFLAG_INSURED_ON_APPROVAL'].value_counts()
```

```
0.0    665527
1.0    331622
Name: NFLAG_INSURED_ON_APPROVAL, dtype: int64
```

```
pra['NFLAG_INSURED_ON_APPROVAL']=pra['NFLAG_INSURED_ON_APPROVAL'].fillna(0)
```

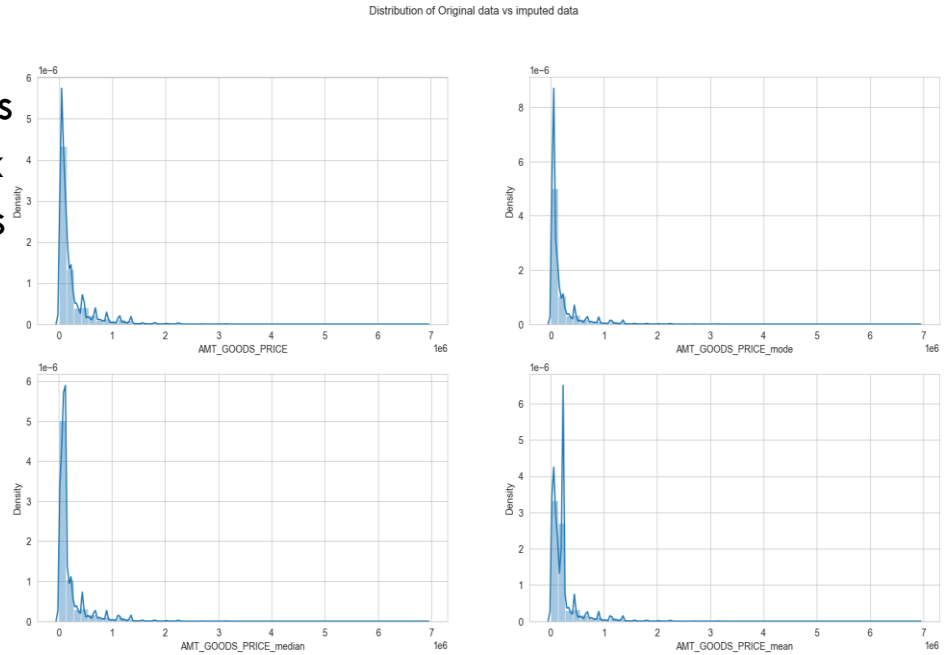
For AMT_ANNUITY as the kde plot is skewed with 1peak, null values were replaced by median

```
#imputing missing values with median as there is only 1 peak implying outliers

pra['AMT_ANNUITY'].fillna(pra['AMT_ANNUITY'].median(),inplace = True)
```



For AMT_GOOD_PRICE since kdeplot has Multiple peaks, the plot with replacing null values with mode median and mean was made to check which plot resembles the original one the most as it was the mode one null values were replaced with mode

```
# Imputing null values with mode

pra['AMT_GOODS_PRICE'].fillna(pra['AMT_GOODS_PRICE'].mode()[0], inplace=True)
```



DATA IMPUTATION
DATASET -2 ->PREVIOUS APPLICATION

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

```
#taking out values count for NAME_CONTRACT_STATUS categories where CNT_PAYMENT have null values.
pra.loc[pra['CNT_PAYMENT'].isnull(),'NAME_CONTRACT_STATUS'].value_counts()
```

```
Canceled        305805
Refused          40897
Unused offer     25524
Approved             4
Name: NAME_CONTRACT_STATUS, dtype: int64
```

```
#imputing null values as 0
pra['CNT_PAYMENT'].fillna(0,inplace = True)
```

In column PRODUCT_COMBINATION the null values "unknown"

```
pra['PRODUCT_COMBINATION']=pra['PRODUCT_COMBINATION'].fillna("Unkown")
pra['PRODUCT_COMBINATION'].value_counts(normalize=True)*100
```

```
Cash                              17.122956
POS household with interest       15.783726
POS mobile with interest          13.212079
Cash X-Sell: middle                8.614645
Cash X-Sell: low                   7.798282
Card Street                        6.740573
POS industry with interest         5.917385
POS household without interest     4.963915
Card X-Sell                        4.824651
Cash Street: high                  3.570740
Cash X-Sell: high                  3.550503
Cash Street: middle                2.075063
Cash Street: low                   2.025728
POS mobile without interest        1.441851
POS other with interest            1.429697
POS industry without interest      0.754514
POS others without interest        0.152974
Unkown                             0.020716
Name: PRODUCT_COMBINATION, dtype: float64
```

STANDARDIZING VALUES – DATASET-2 PREVIOUS APPLICATION

There are negative values in days column which needs to be rectified

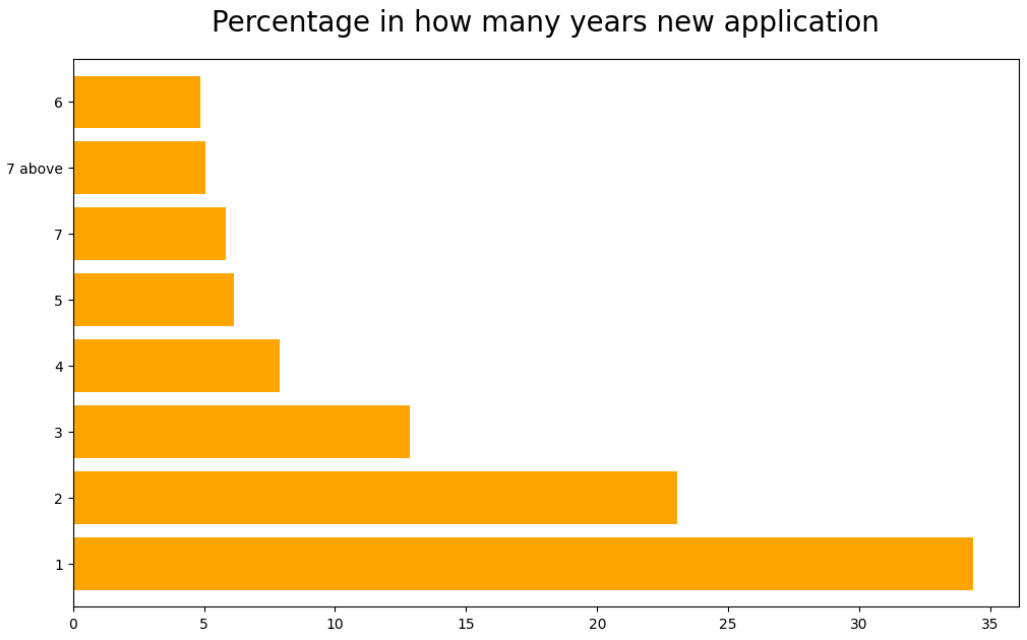
	DAYS_DECISION	DAYS_FIRST_DRAWING	DAYS_FIRST_DUE	DAYS_LAST_DUE_1ST_VERSION	DAYS_LAST_DUE	DAYS_TERMI
count	1.670214e+06	997149.000000	997149.000000	997149.000000	997149.000000	99714
mean	-8.806797e+02	342209.855039	13826.269337	33767.774054	76582.403064	8199
std	7.790997e+02	88916.115833	72444.869708	106857.034789	149647.415123	15330
min	-2.922000e+03	-2922.000000	-2892.000000	-2801.000000	-2889.000000	-287
25%	-1.300000e+03	365243.000000	-1628.000000	-1242.000000	-1314.000000	-127
50%	-5.810000e+02	365243.000000	-831.000000	-361.000000	-537.000000	-49
75%	-2.800000e+02	365243.000000	-411.000000	129.000000	-74.000000	-4
max	-1.000000e+00	365243.000000	365243.000000	365243.000000	365243.000000	36524

```
# Converting Negative days to positive days

pra[p_days_col] = abs(pra[p_days_col])

pra[p_days_col].describe() # analysing after conversion
```

	DAYS_DECISION	DAYS_FIRST_DRAWING	DAYS_FIRST_DUE	DAYS_LAST_DUE_1ST_VERSION	DAYS_LAST_DUE
count	1.670214e+06	997149.000000	997149.000000	997149.000000	997149.000000
mean	8.806797e+02	342340.056543	15949.224065	35163.363265	78152.730207
std	7.790997e+02	88413.495220	72007.270877	106405.950190	148833.342466
min	1.000000e+00	2.000000	2.000000	0.000000	2.000000
25%	2.800000e+02	365243.000000	475.000000	257.000000	455.000000
50%	5.810000e+02	365243.000000	921.000000	741.000000	1155.000000
75%	1.300000e+03	365243.000000	1825.000000	1735.000000	2418.000000
max	2.922000e+03	365243.000000	365243.000000	365243.000000	365243.000000

STANDARDIZING VALUES – DATASET-2 PREVIOUS APPLICATION

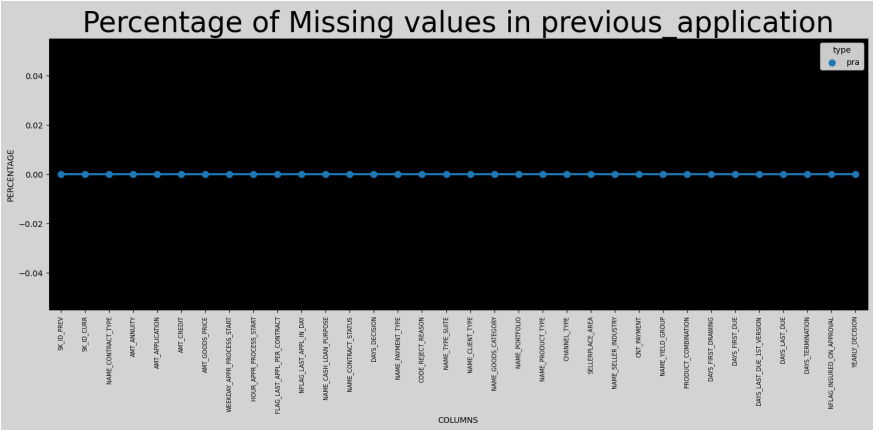
Standardizing values for days_decision column by converting them into years by binning, for better understanding of data after replacing the null values with mode.




```
#days group calculation

bins = [0,1*365,2*365,3*365,4*365,5*365,6*365,7*365,10*365]
slots = ["1","2","3","4","5","6","7","7 above"]
pra['YEARLY_DECISION'] = pd.cut(pra['DAYS_DECISION'],bins,labels=slots)
```

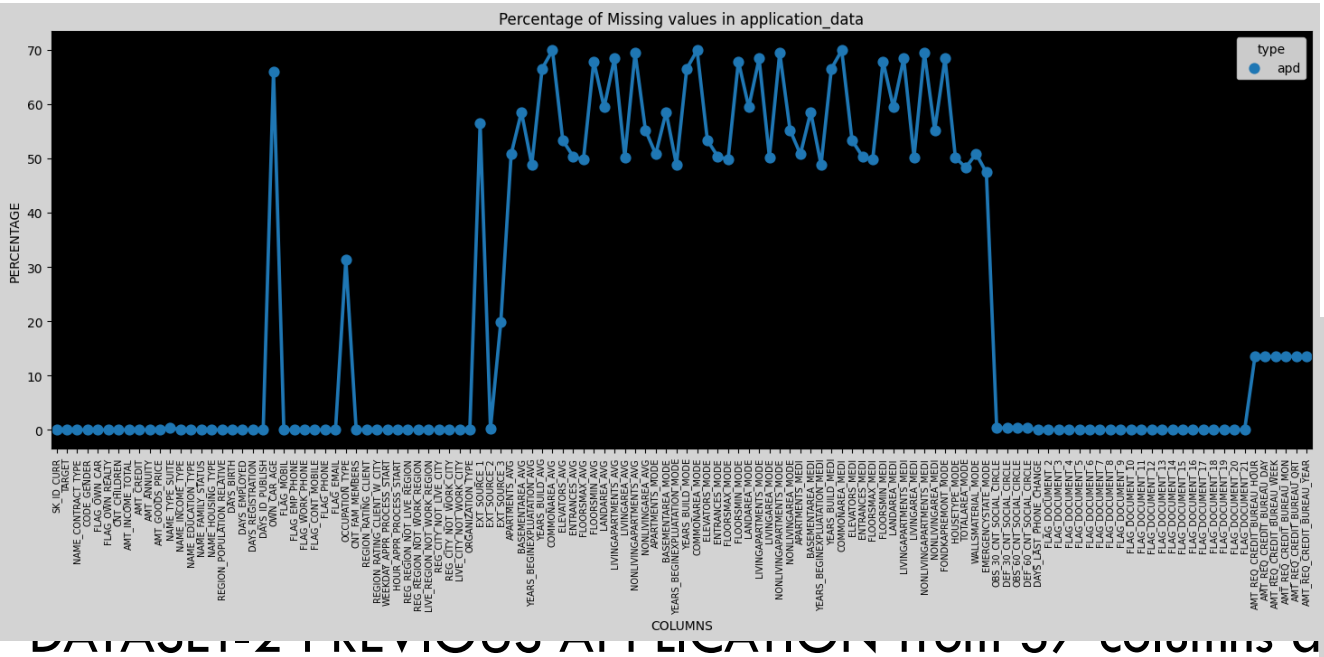
About 35% applicant apply again after decision on previous loan

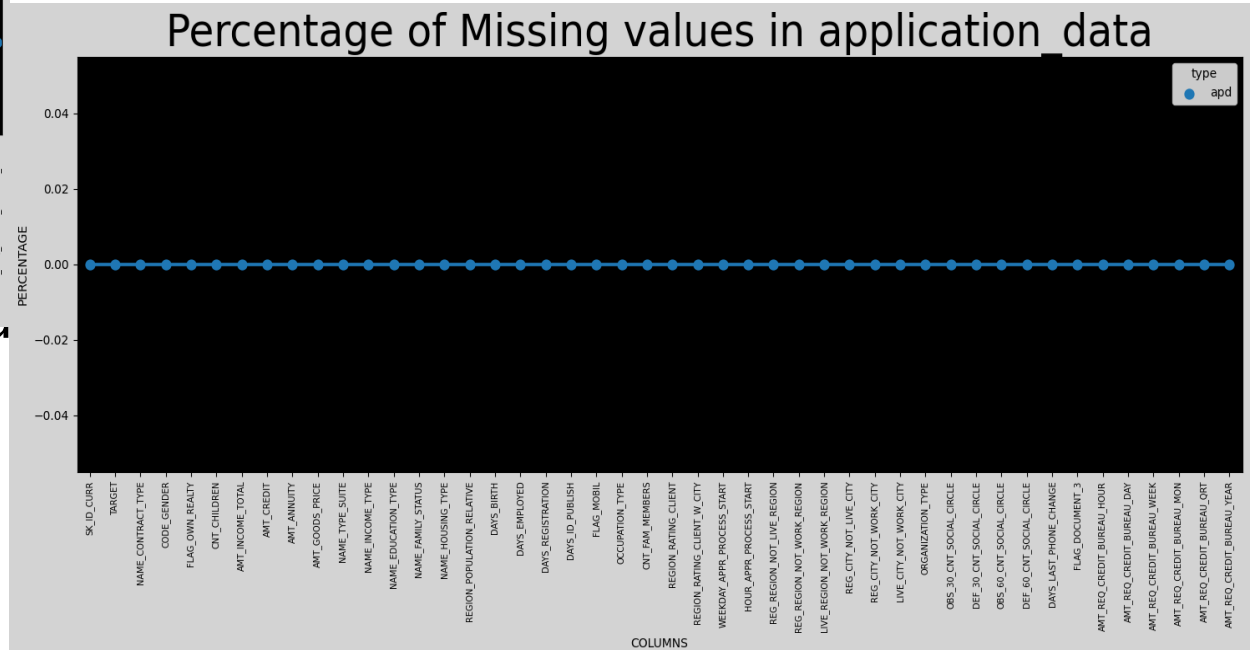
All Null Values have been removed



DATA CLEANING CONCLUSION

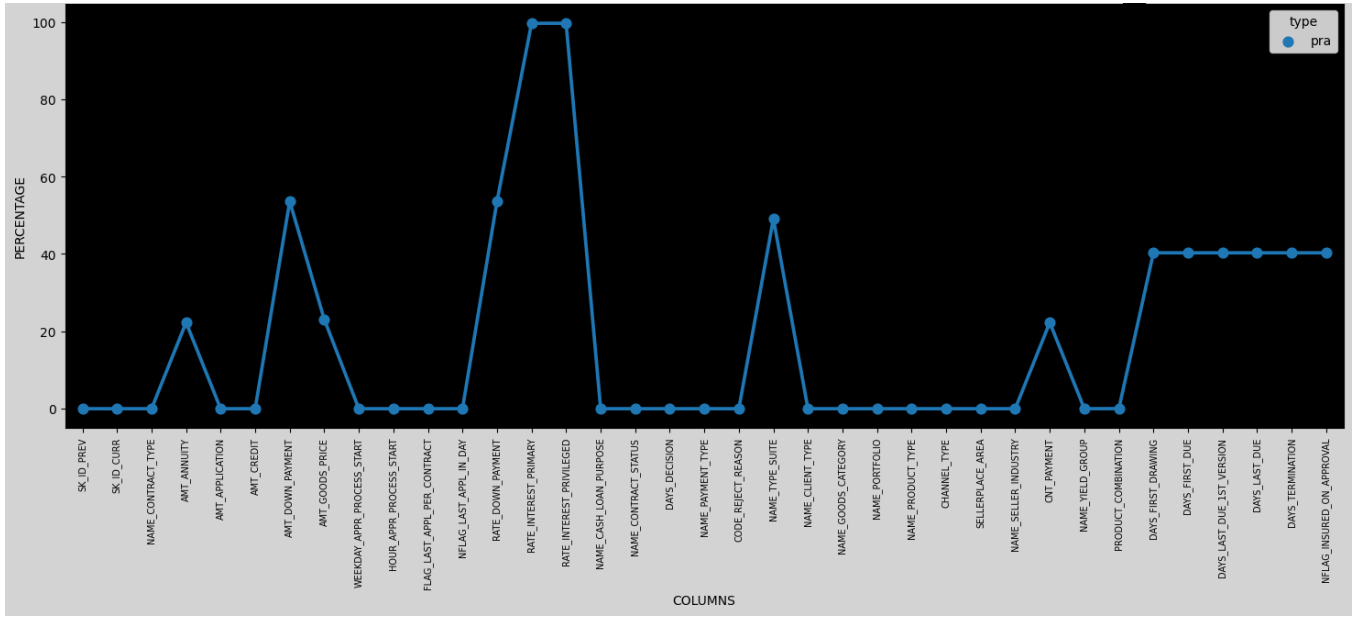
DATASET-1 APPLICATION DATA-> from 122 columns down to 53 columns

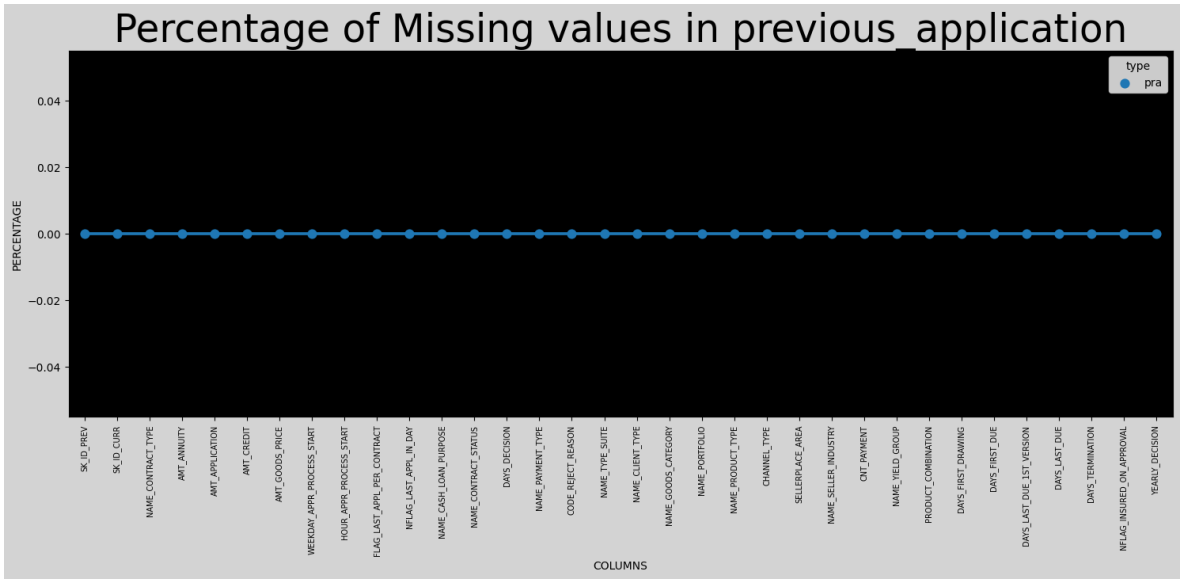


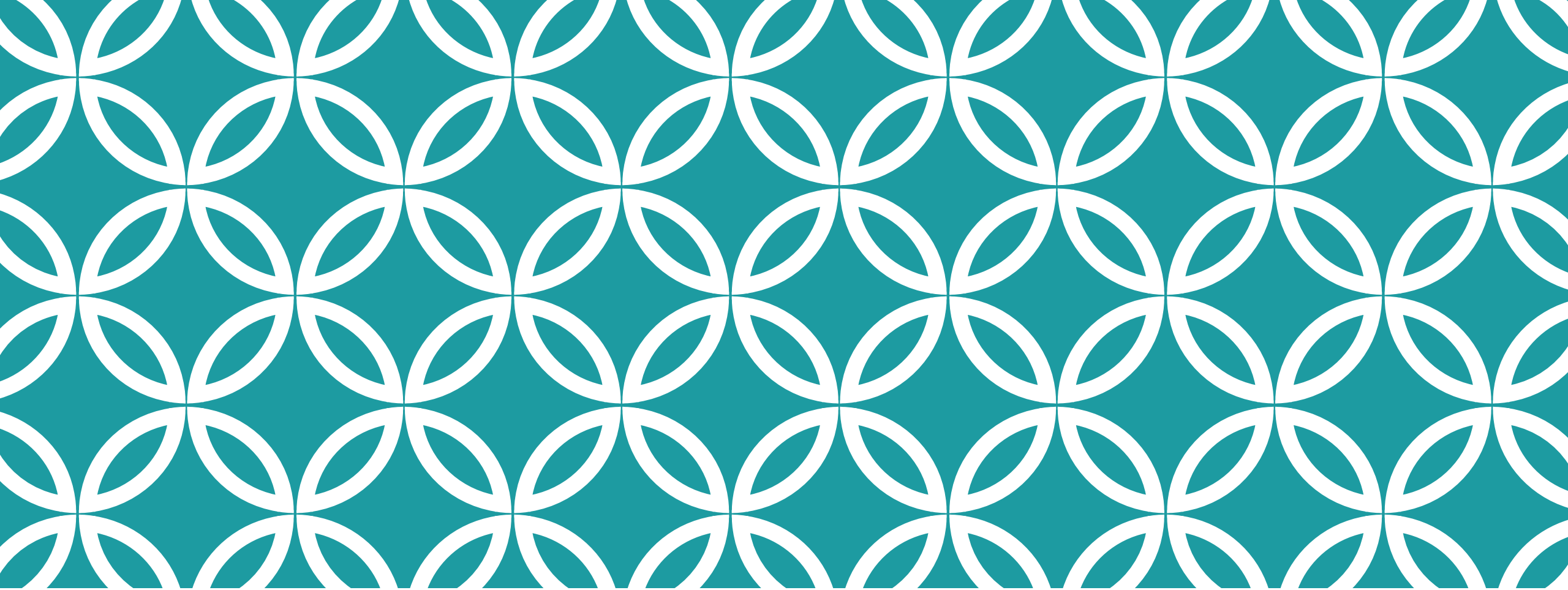


DATA CLEANING CONCLUSION

DATASET-2 PREVIOUS APPLICATION from 37 columns down to 34 columns







TASK-2-IDENTIFYING OUTLIER

OUTLIER IDENTIFICATION
DATA SET-1 APPLICATION DATA

From describe we could find all the columns those who have high difference between max and 75 percentile and the ones which makes no sense having max value to be so high are captured below:

```
outlier_col = ["CNT_CHILDREN","AMT_INCOME_TOTAL", "AMT_CREDIT", "AMT_ANNUITY", "AMT_GOODS_PRICE",
               "DAYS_BIRTH", "DAYS_EMPLOYED", "DAYS_REGISTRATION", "CNT_FAM_MEMBERS"]
```

The box plot of these columns will reflect if they have outliers or not.

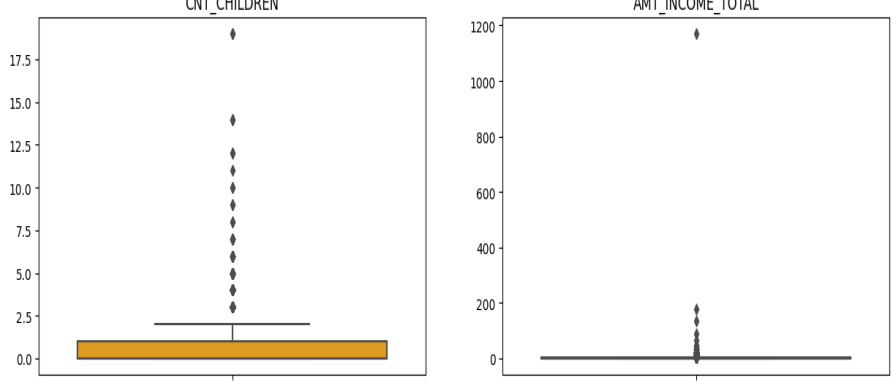
•AMT_ANNUITY, AMT_CREDIT, AMT_GOODS_PRICE,CNT_CHILDREN have some number of outliers.

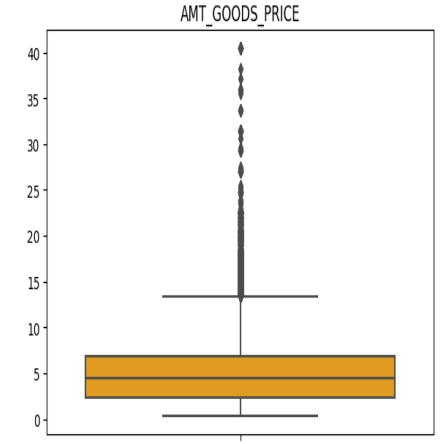
•AMT_INCOME_TOTAL has huge number of outliers which indicate that few of the loan applicants have high income when compared to the others.

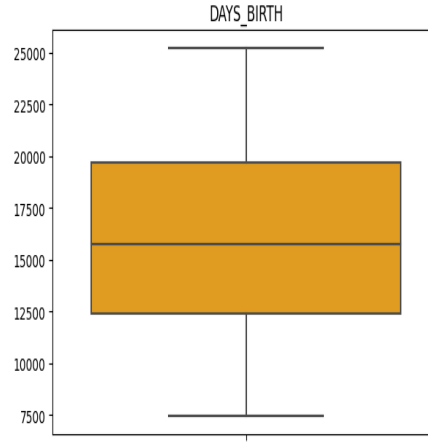
•DAYS_BIRTH has no outliers which means the data available is reliable.

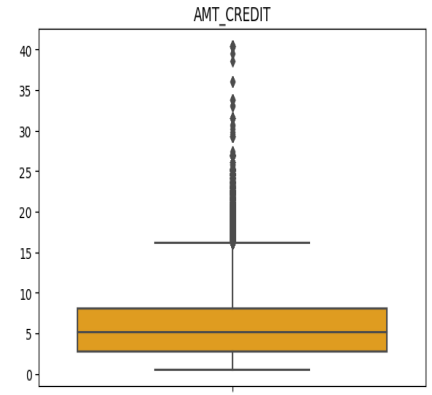
•DAYS_EMPLOYED has outlier values around 350000(days) which is around 958 years which is impossible and hence this has to be incorrect entry.

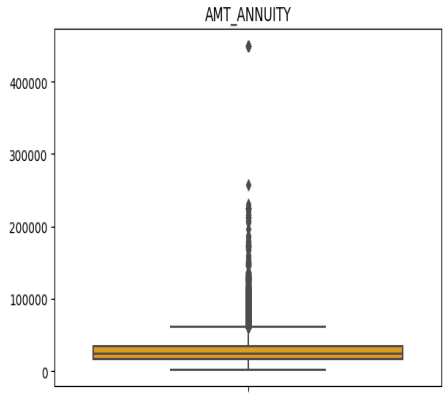
•CNT_FAM_MEMBERS has outlier value more than 45000 which is impossible hence it is an incorrect entry

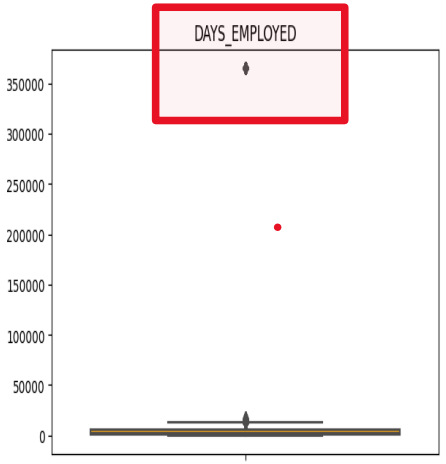


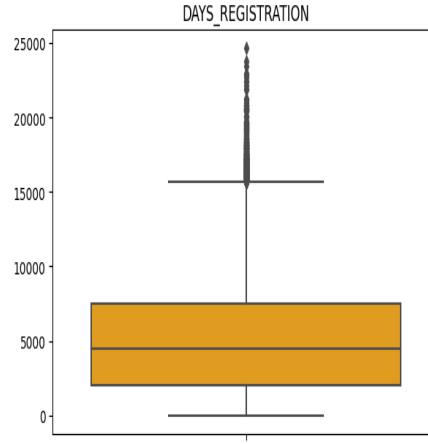


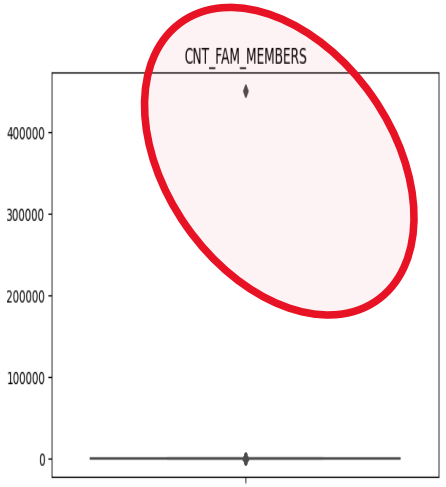












OUTLIER IDENTIFICATION
DATA SET-2 PREVIOUS APPLICATION

From describe we could find all the columns those who have high difference between max and 75 percentile and the ones which makes no sense having max value to be so high are captured below:

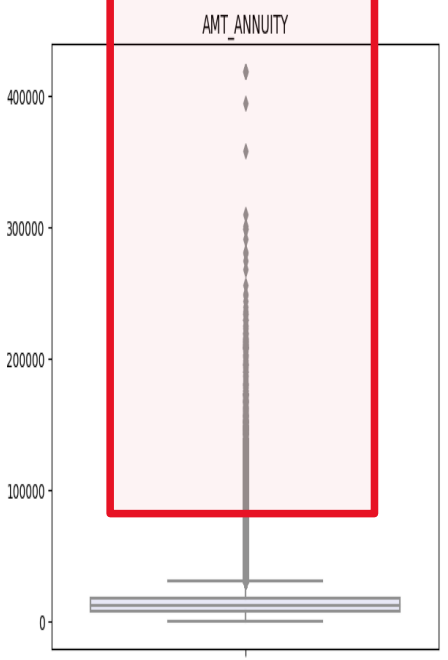
```
p_outlier_col = ['AMT_ANNUITY','AMT_APPLICATION','AMT_CREDIT','AMT_GOODS_PRICE',
                 'SELLERPLACE_AREA','DAYS_DECISION','CNT_PAYMENT']
```

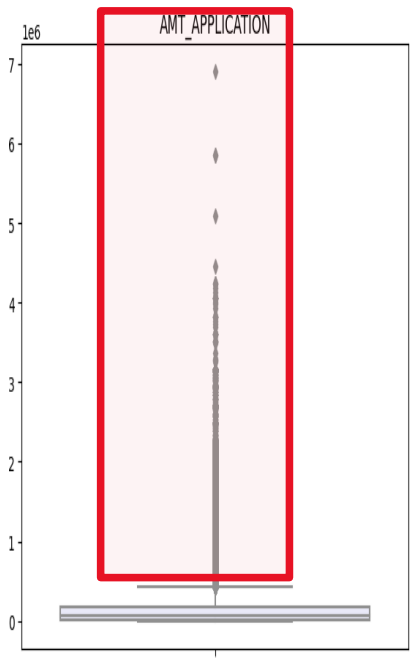
The box plot of these columns will reflect if they have outliers or not.

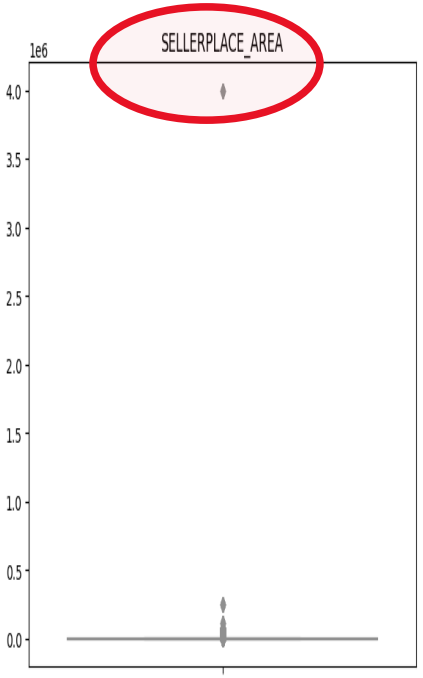
It can be seen that in previous application data
•AMT_ANNUITY, AMT_APPLICATION, AMT_CREDIT, AMT_GOODS_PRICE, SELLERPLACE_AREA have huge number of outliers.

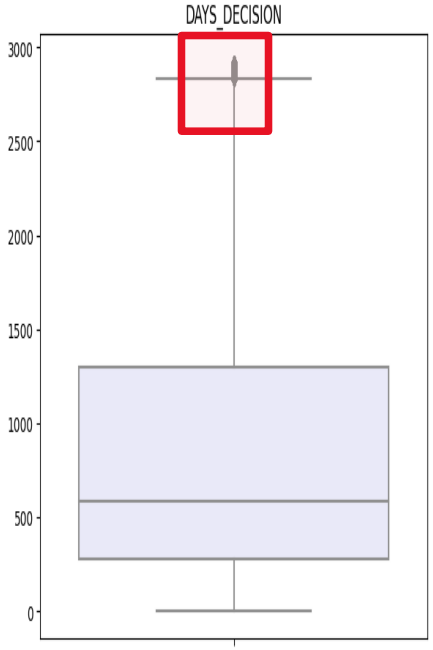
•CNT_PAYMENT has few outlier values.

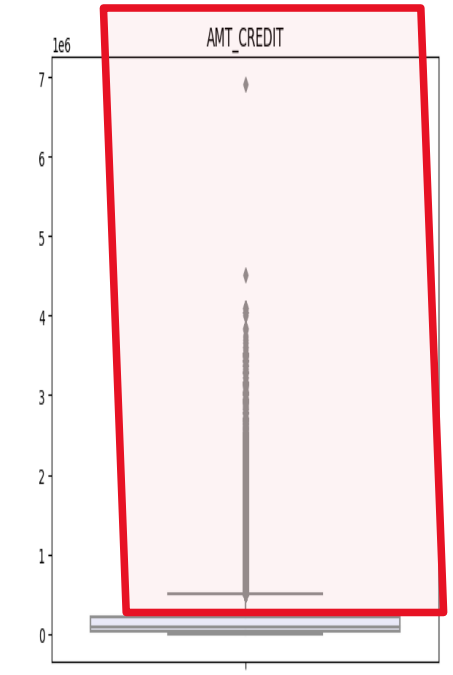
•DAYS_DECISION has little number of outliers indicating that these previous applications decisions were taken recently.

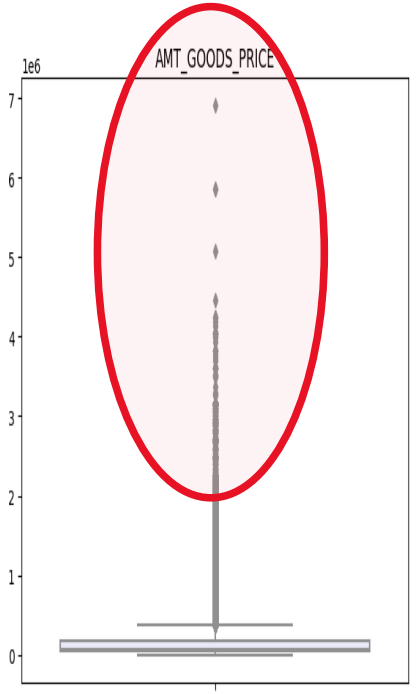


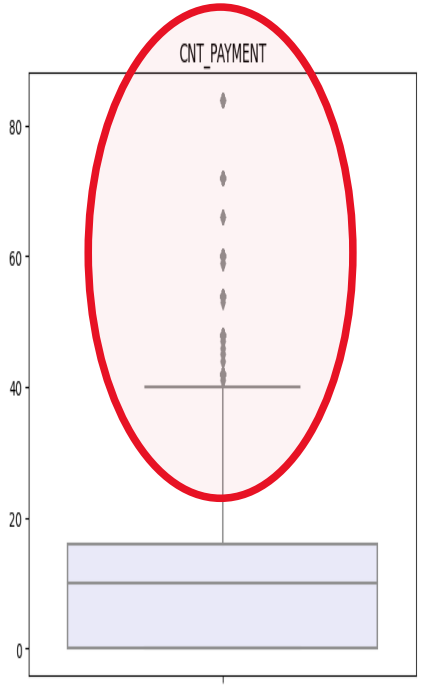


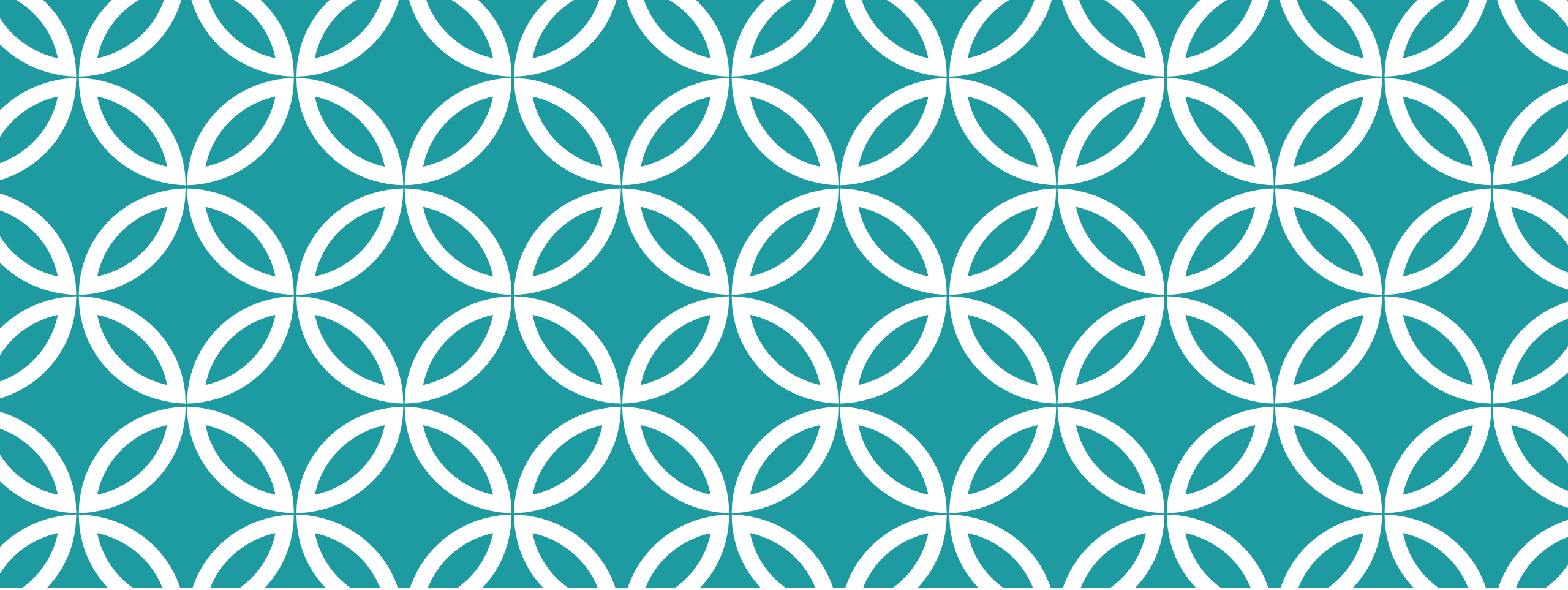












TASK-3-REPORTING DATA IMBALANCE

IMBALANCE BETWEEN DEFAULTER AND REPAYER COUNT

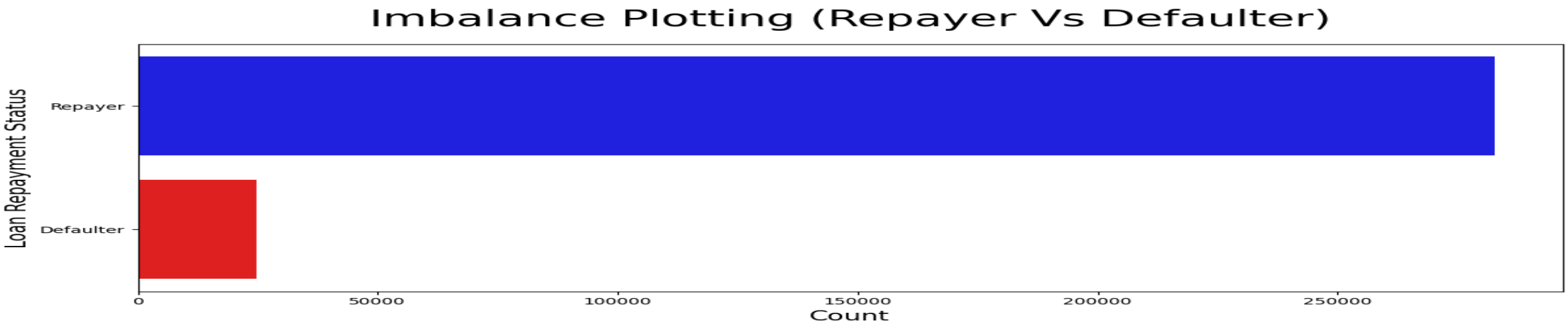
No. of reapyers are 282686

No. of defaulters are 24825

Repayer Percentage is 91.93%

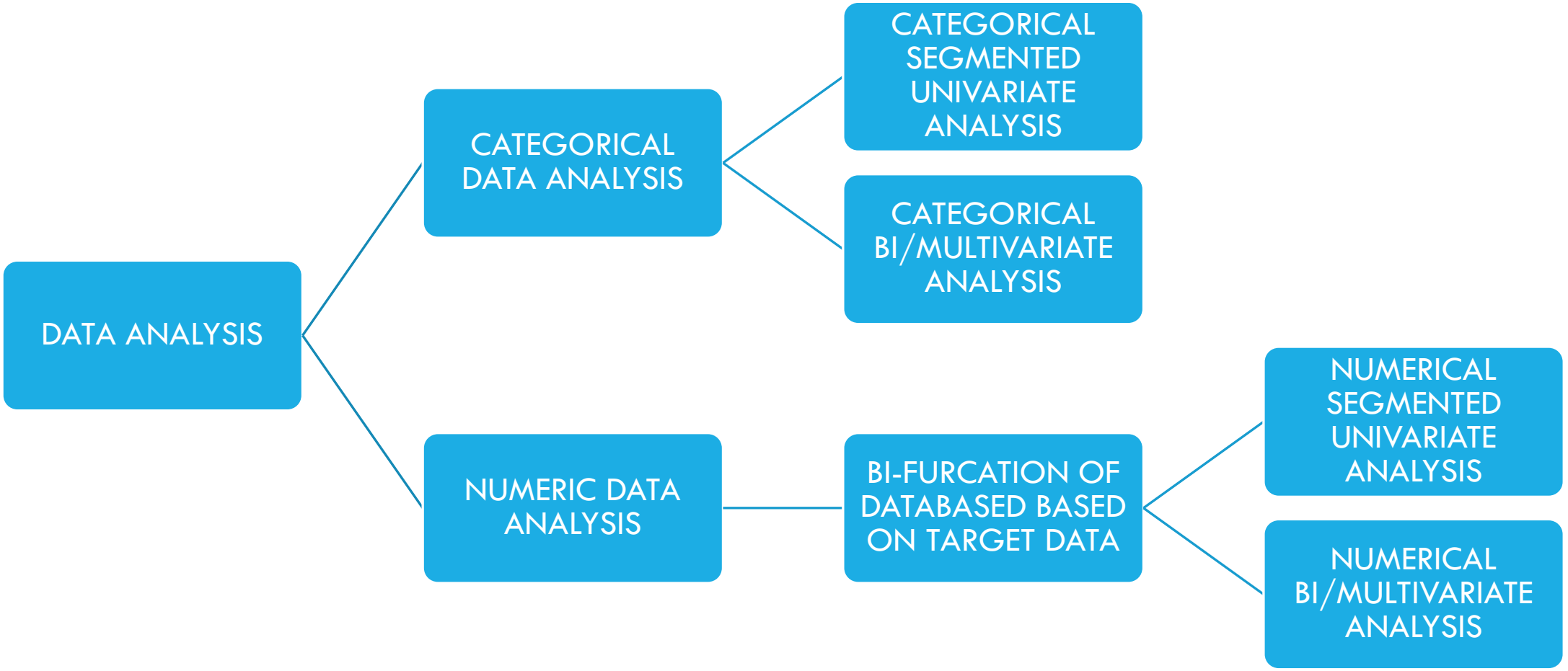
Defaulter Percentage is 8.07%

Imbalance Ratio with respect to Repayer and Defaulter is given: 11.39/1 (approx)



TASK-4 REPORTING RESULTS OF UNIVARIATE,SEGMENTED UNIVARAITE AND BIVARIATE ANALYSIS

METHODOLOGY ADOPTED

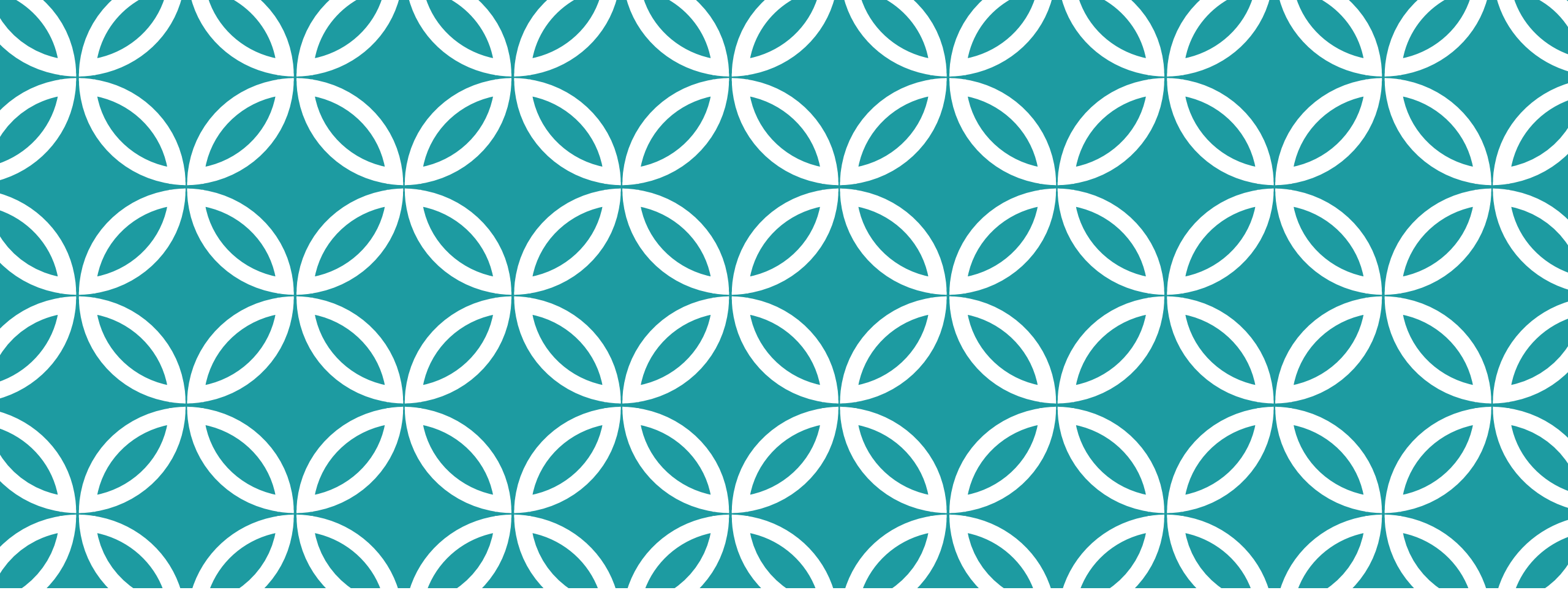


APPRAOCH ADOPTED

To do the analysis, various functions are created to divide the data sets into categorical and numeric datasets

Later new more functions are created to analyse the dataset according to the required analysis (univariate/segmented univariate/ bivariate)

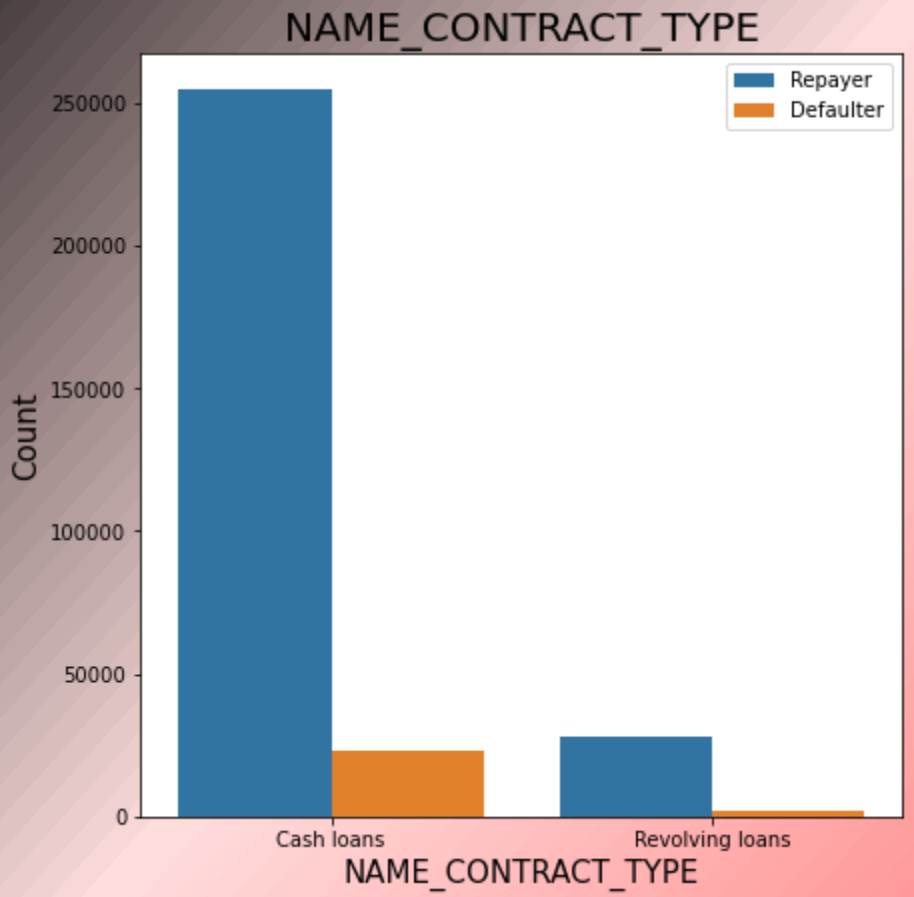
(THE FUNCTIONS CAN BE SEEN IN THE IPYNB ATTATCHED IN TECH STACK USED SLIDE)

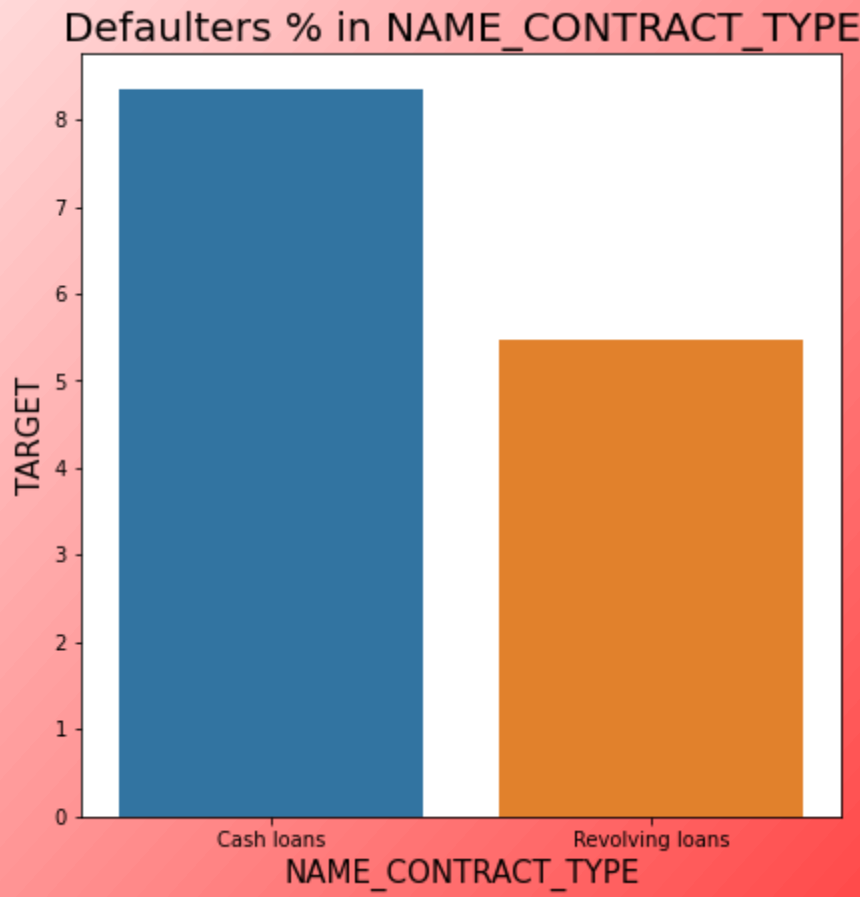


CATEGORICAL VARIABLES ANALYSIS

SEGMENTED UNIVARIATE ANALYSIS

CONTRACT TYPE

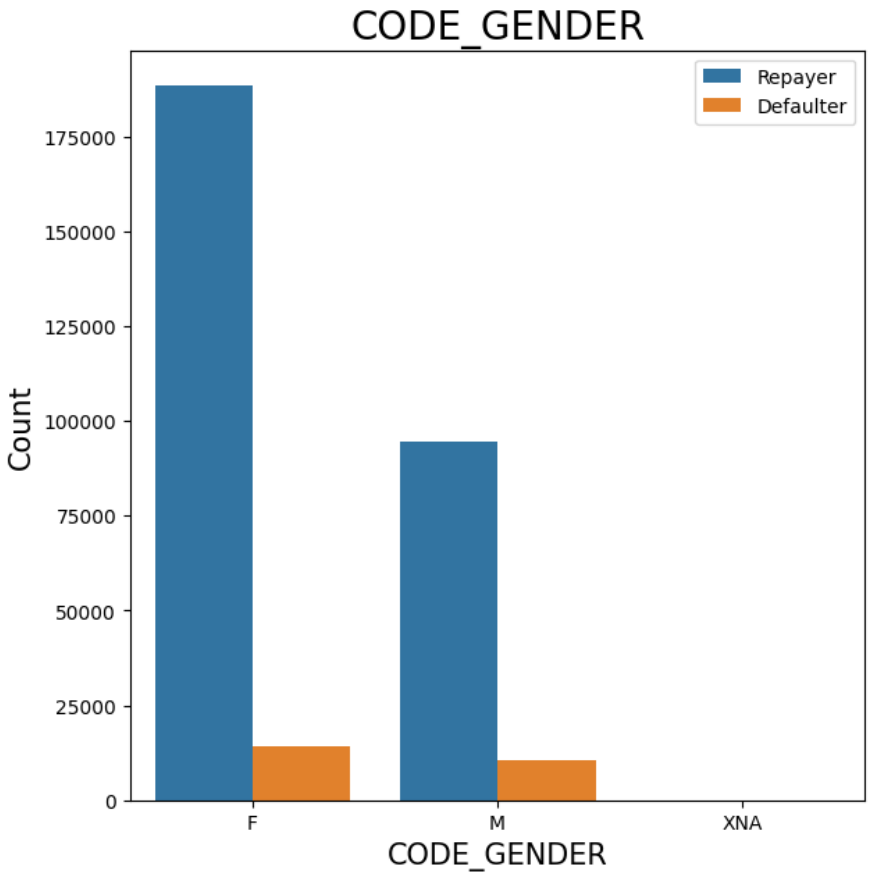


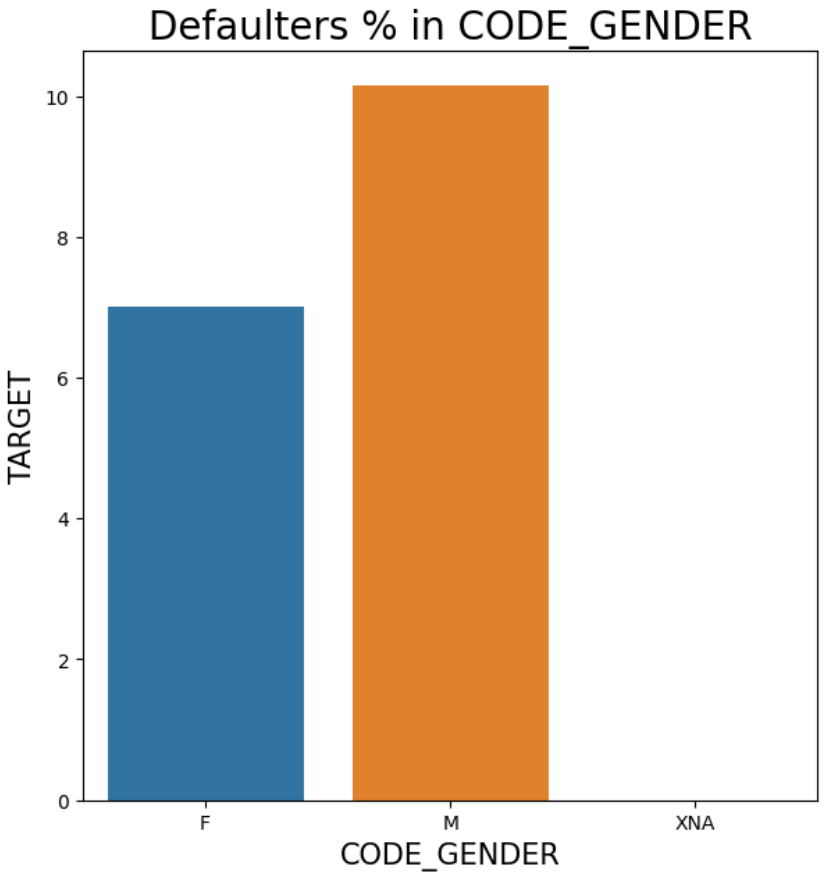


- **Revolving loans are just a small fraction (10%) from the total number of loans.**
- **Around 8-9% Cash loan applicants and 5-6% Revolving loan applicant are in defaulters.**

GENDER WISE LOAN REPAYMENT STATUS

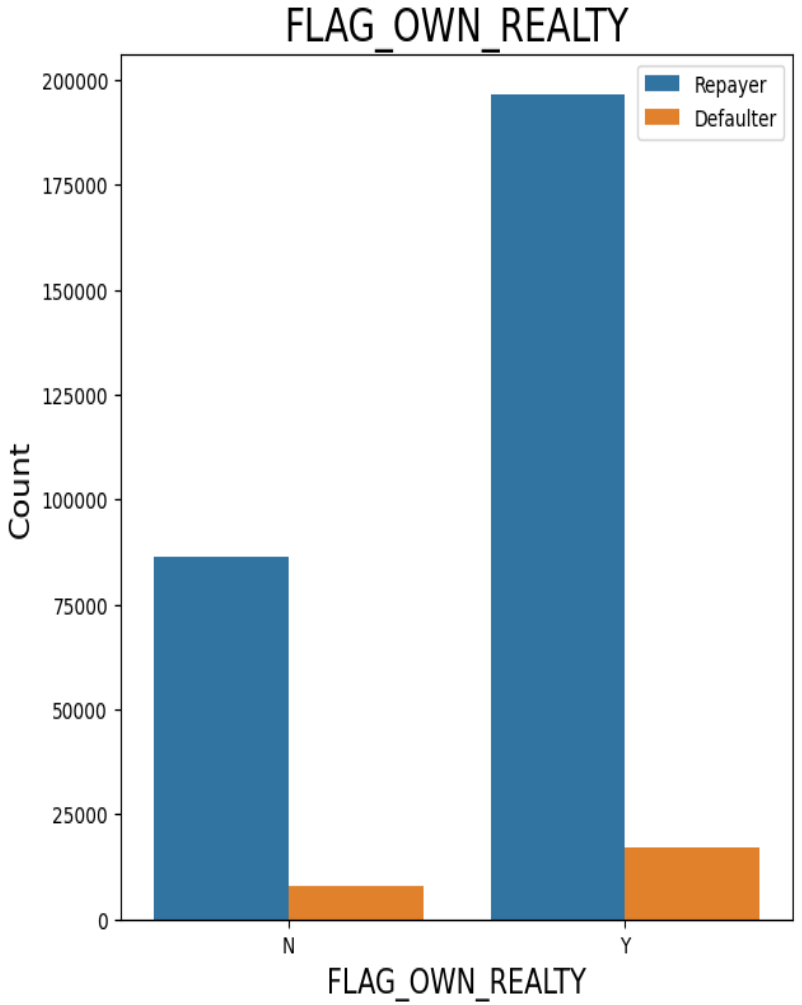
- **The number of female clients is almost double the number of male clients.**
- **Based on the percentage of defaulted credits, males have a higher chance of not returning their loans about 10%, comparing with women about 7%.**

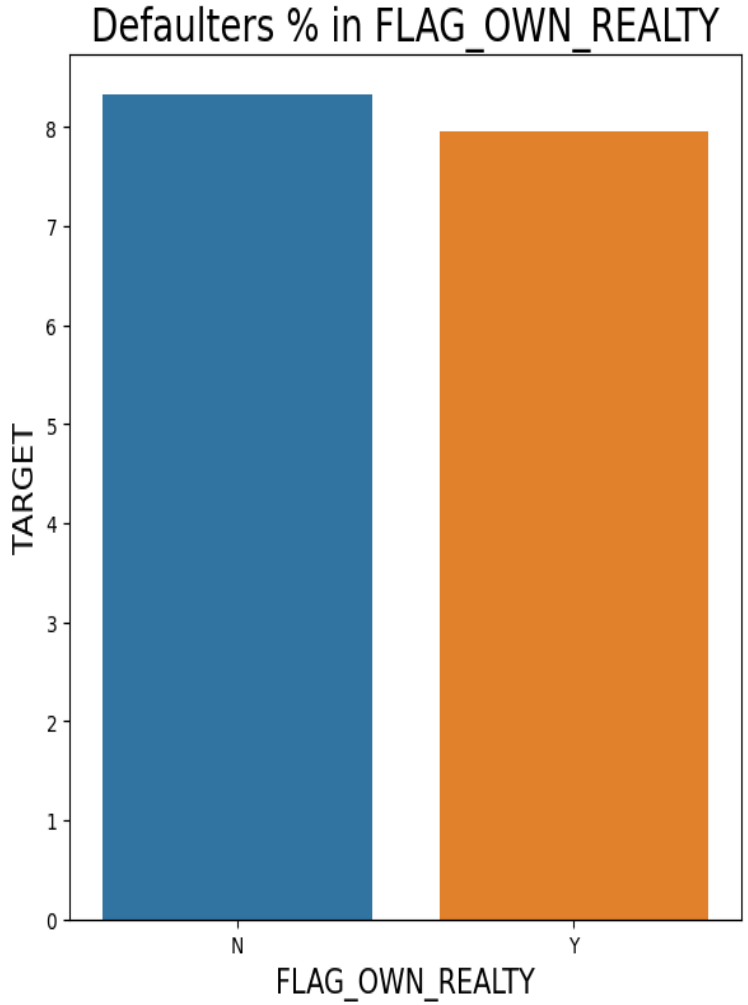




REAL ESTATE OWNERSHIP AND LOAN REPAYMENT STATUS

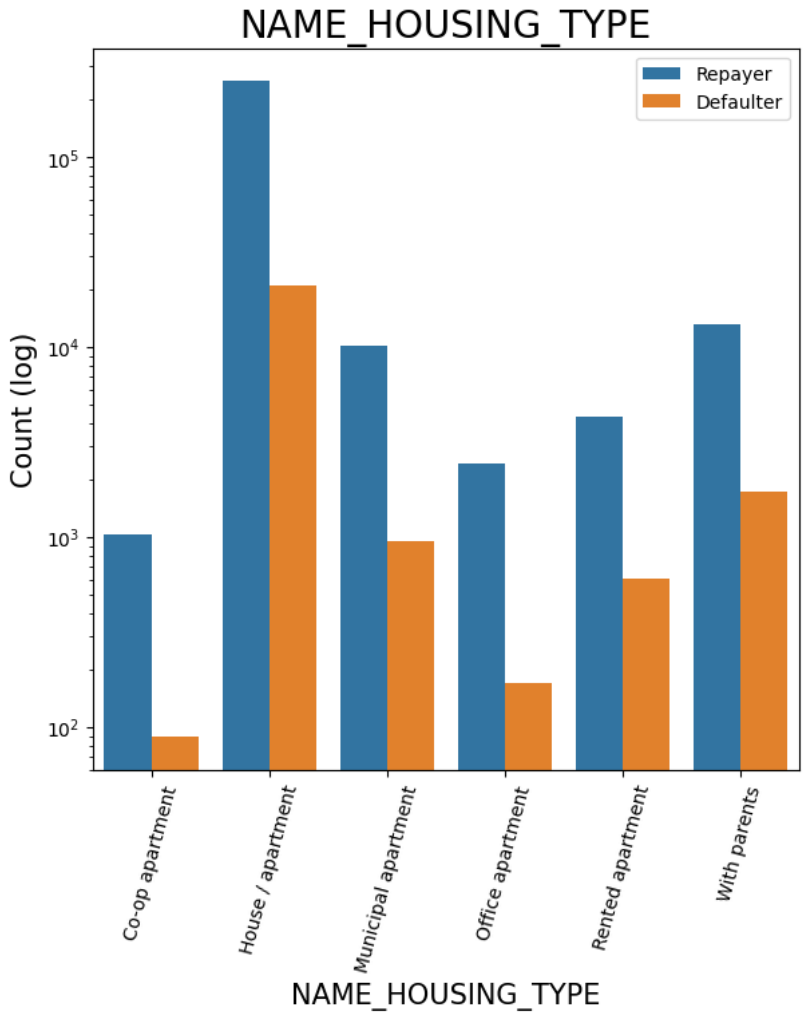
- **The clients who own real estate are more than double of the ones that don't own.**
- **The defaulting rate of both categories are around the same (~8%). Thus we can infer that there is no correlation between owning a reality and defaulting the loan.**

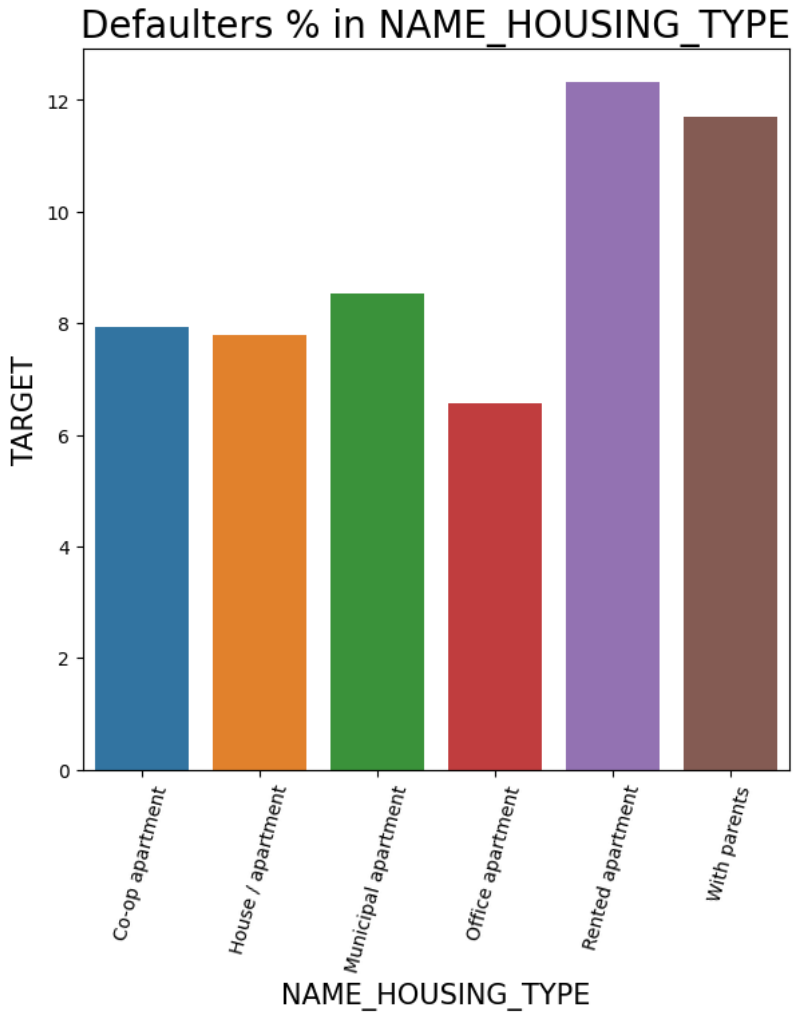




HOUSING TYPE AND LOAN REPAYMENT STATUS

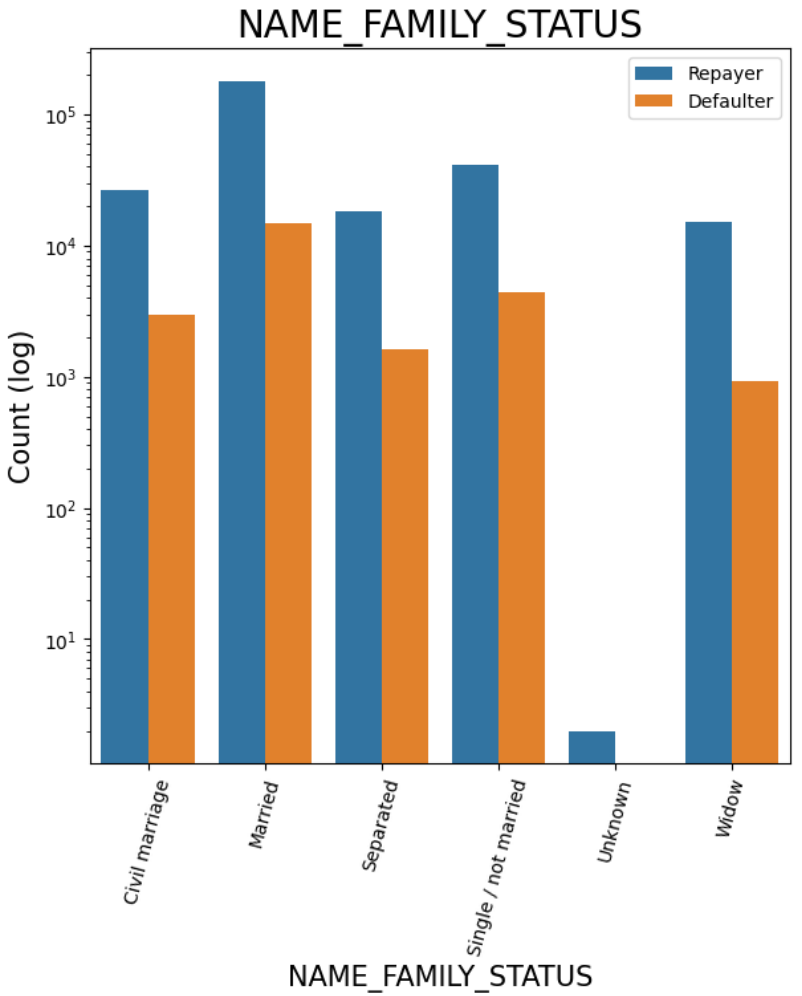
- **Majority of people live in House/apartment.**
- **People living in office apartments have lowest default rate.**
- **People living with parents (~11.5%) and living in rented apartments(>12%) have higher probability of defaulting.**

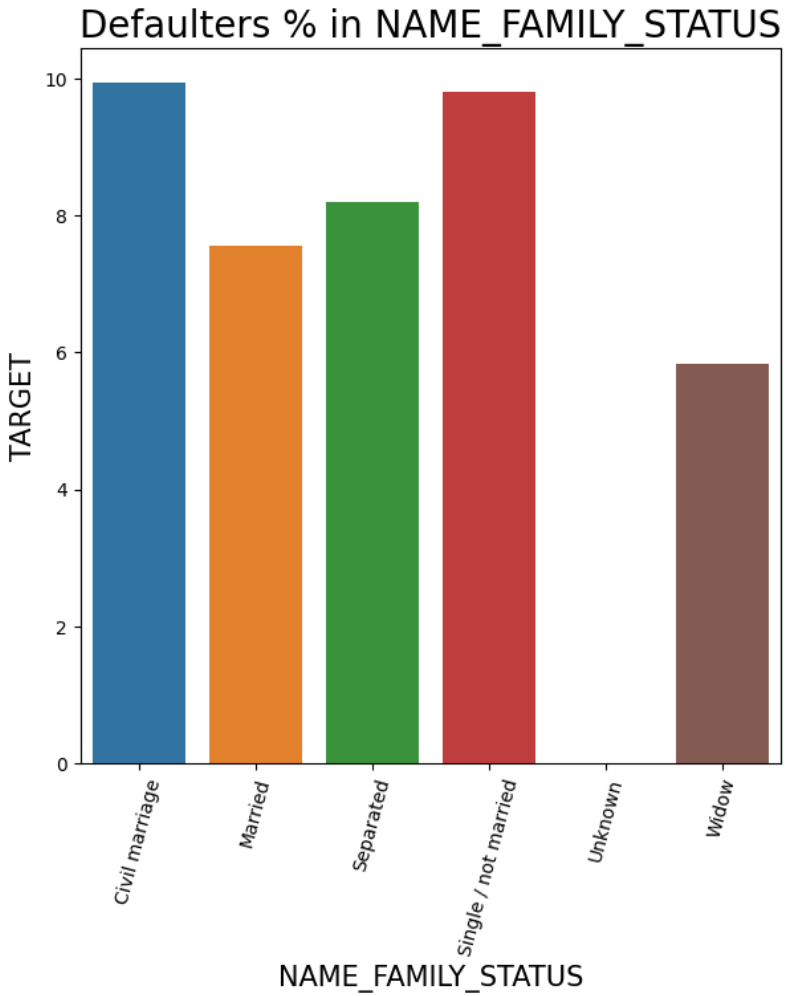




FAMILY STATUS AND LOAN REPAYMENT STATUS

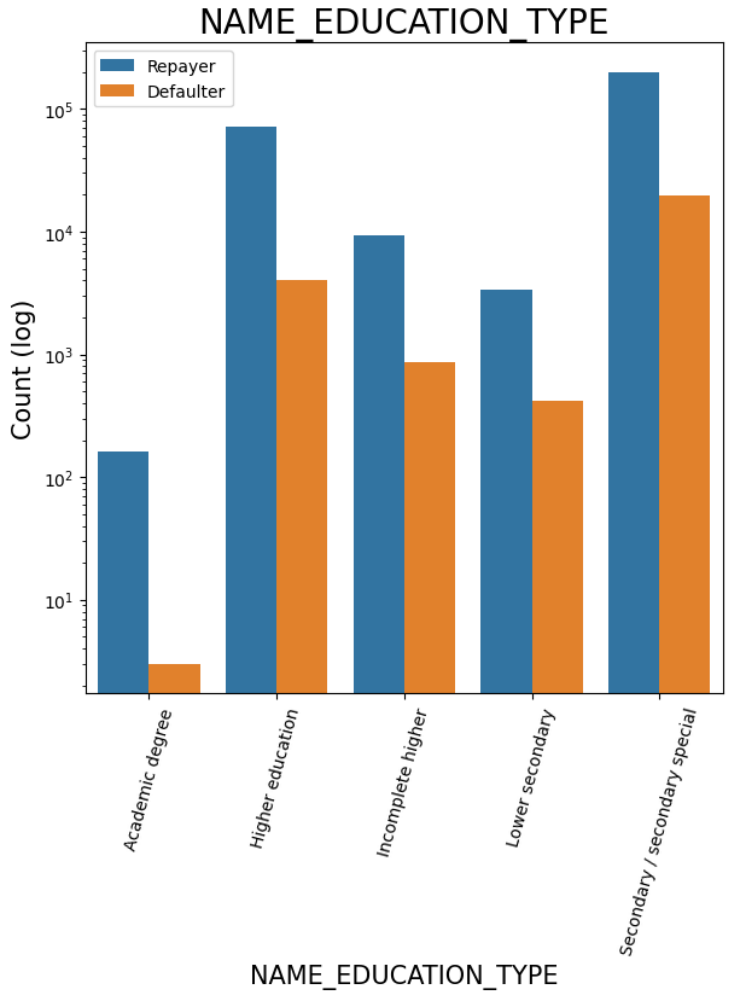
- **Most of the people who have taken loan are married, followed by Single/not married and civil marriage.**
- **In Percentage of defaulters, Civil marriage has the highest percent around (10%) and widow has the lowest around 6% (exception being Unknown).**

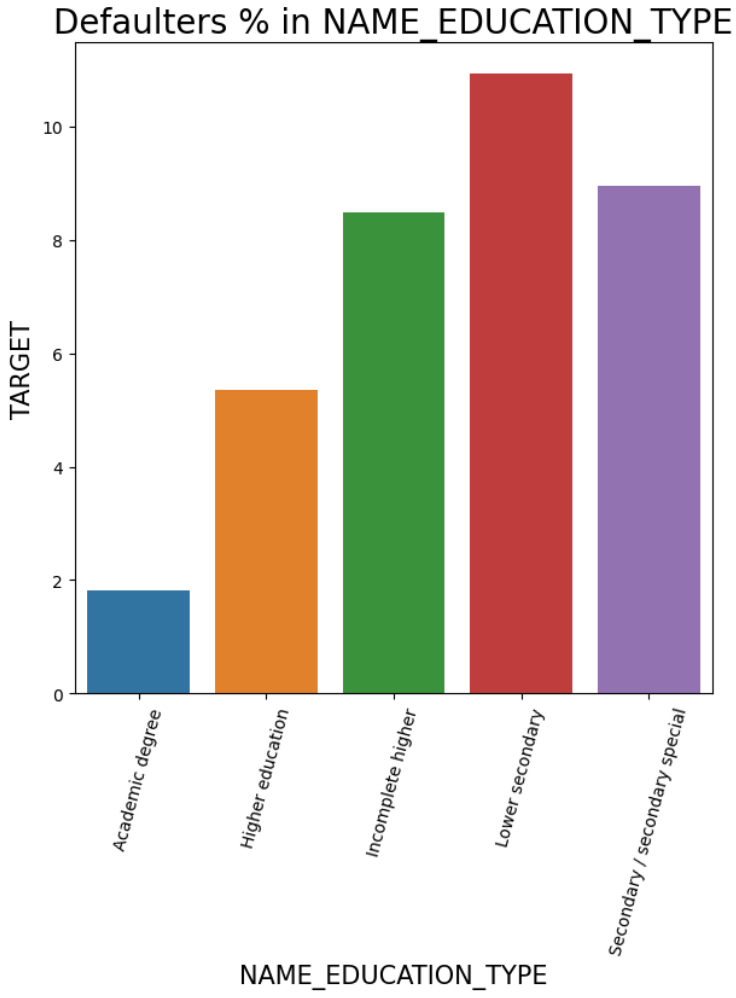




EDUCATION AND LOAN REPAYMENT STATUS

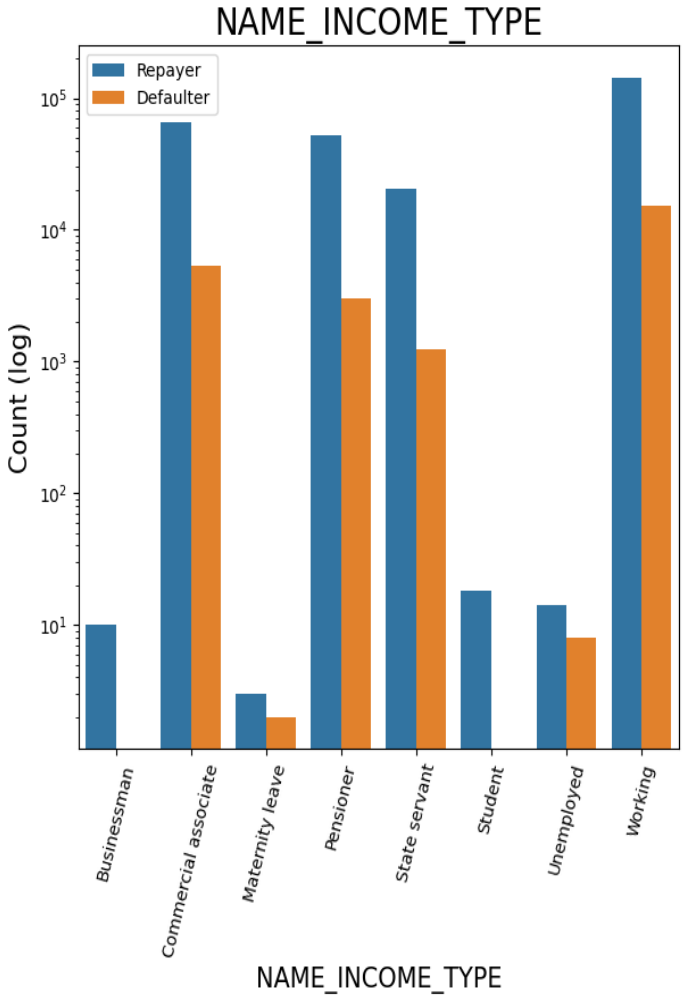
- **Majority of clients have Secondary/secondary special education, followed by clients with Higher education.**
- **Very few clients have an academic degree**
- **Lower secondary category have highest rate of defaulting around 11%.**
- **People with Academic degree are least likely to default.**

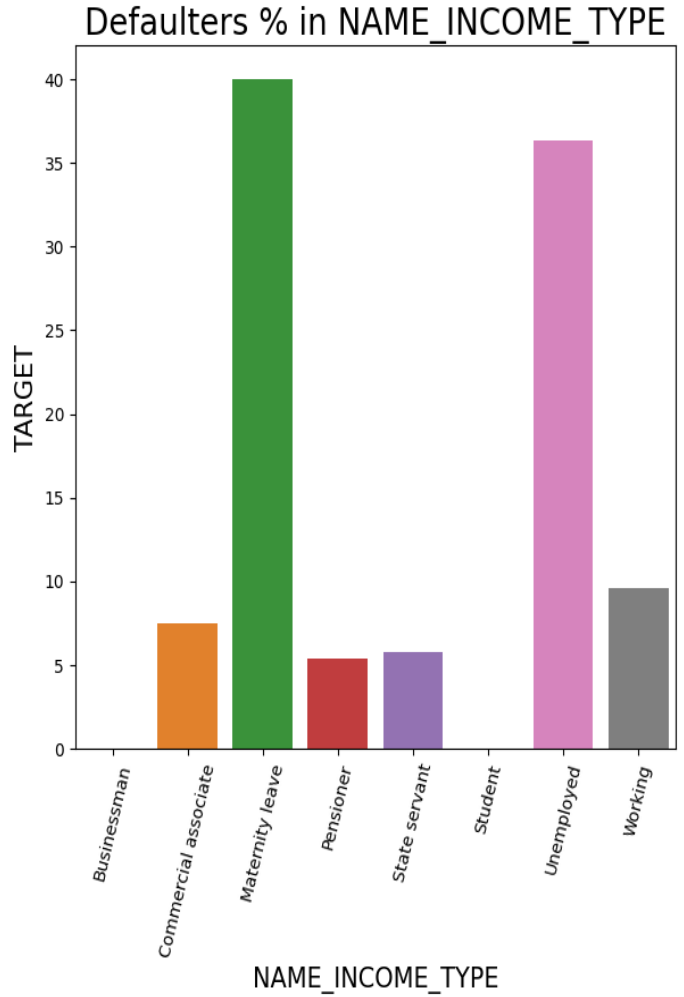




INCOME TYPE AND LOAN REPAYMENT STATUS

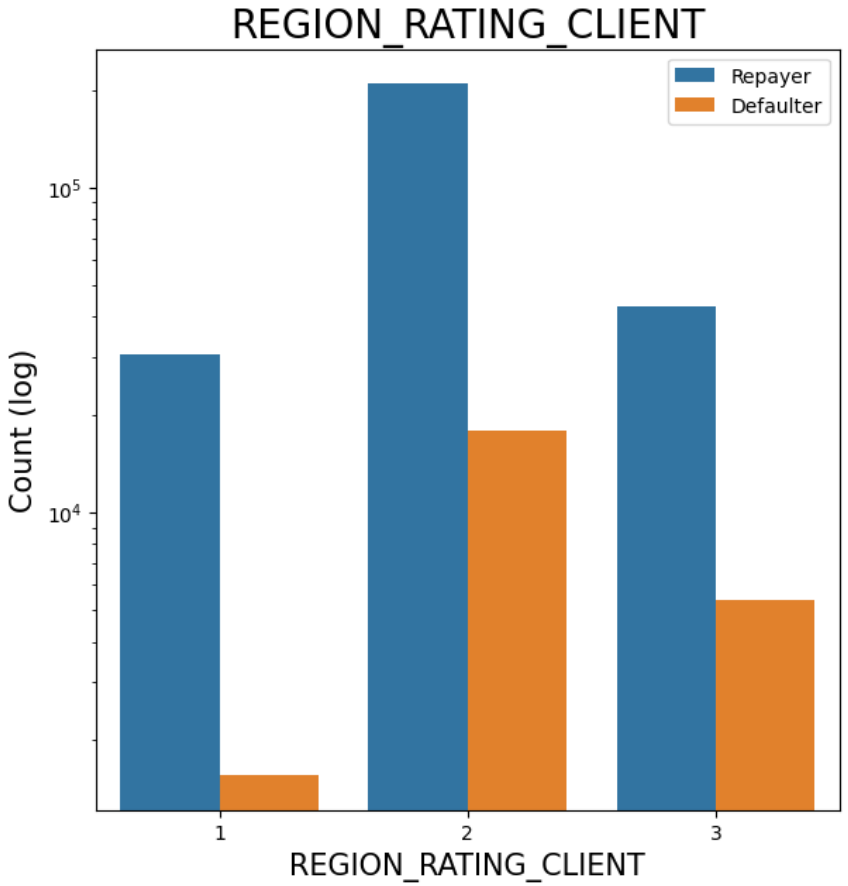
- **Most of applicants for loans income type is Working, followed by Commercial associate, Pensioner and State servant.**
- **The applicants who are on Maternity leave have defaulting percentage of 40% which is the highest, followed by Unemployed (37%). The rest under average around 10% defaulters.**
- **Student and Businessmen though less in numbers, do not have default record. Safest two categories for providing loan.**

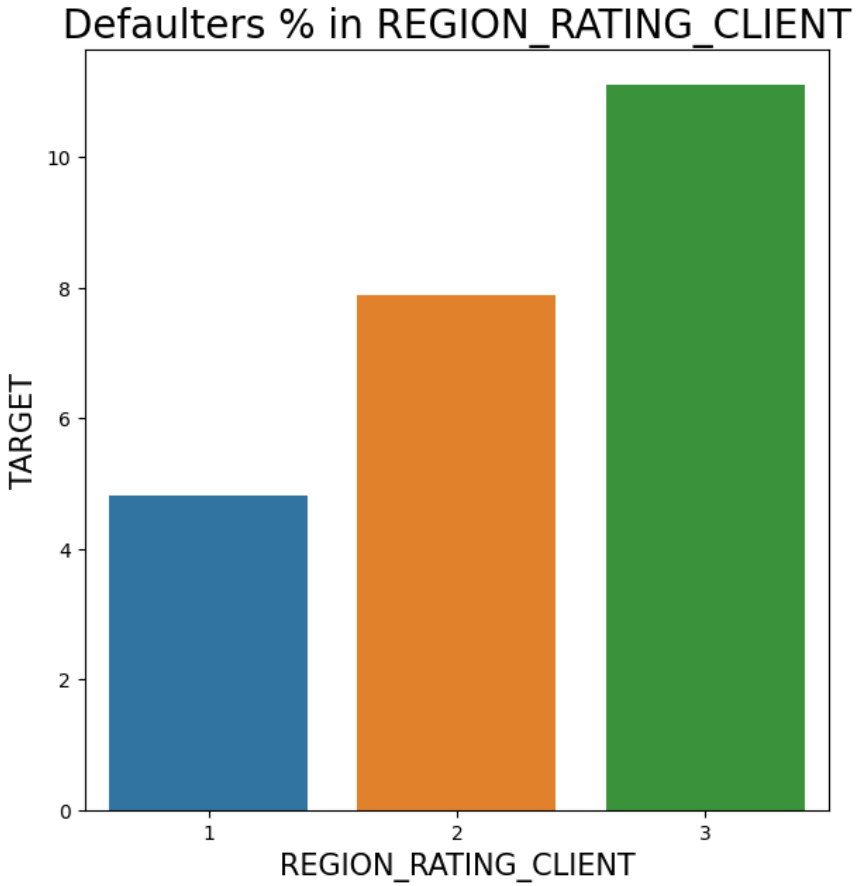




REGION RATING AND LOAN REPAYMENT STATUS

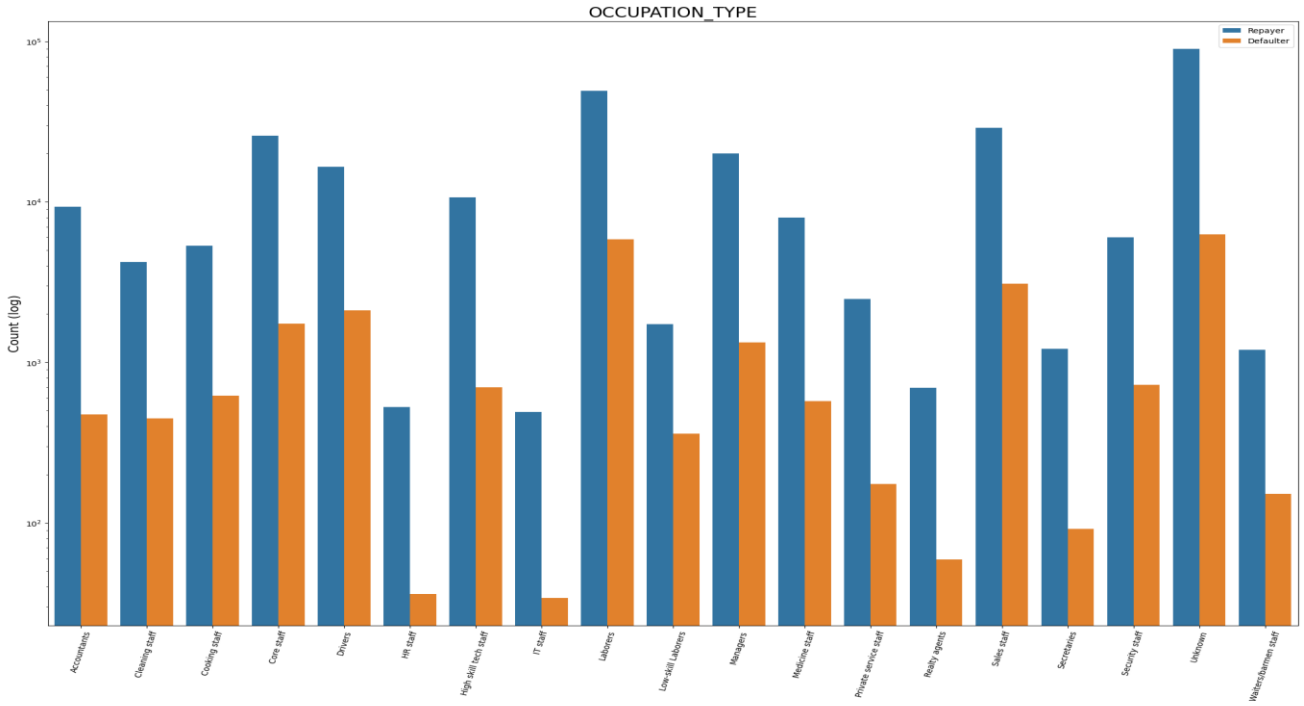
- **Most of the applicants are living in Region with Rating 2 place.**
- **Region Rating 3 has the highest default rate (11%).**
- **Applicant living in Region_Rating 1 has the lowest probability of defaulting, thus safer for approving loans.**

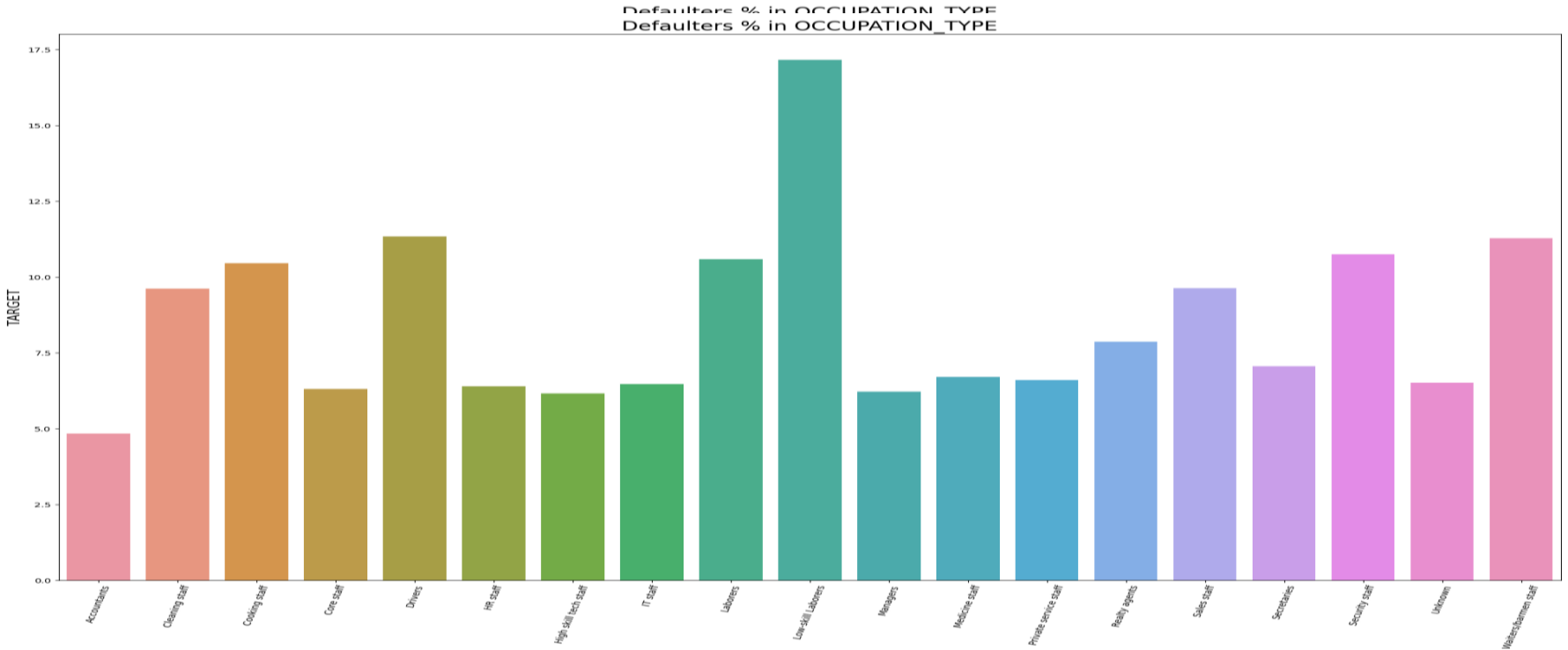




OCCUPATION TYPE AND LOAN REPAYMENY STATUS

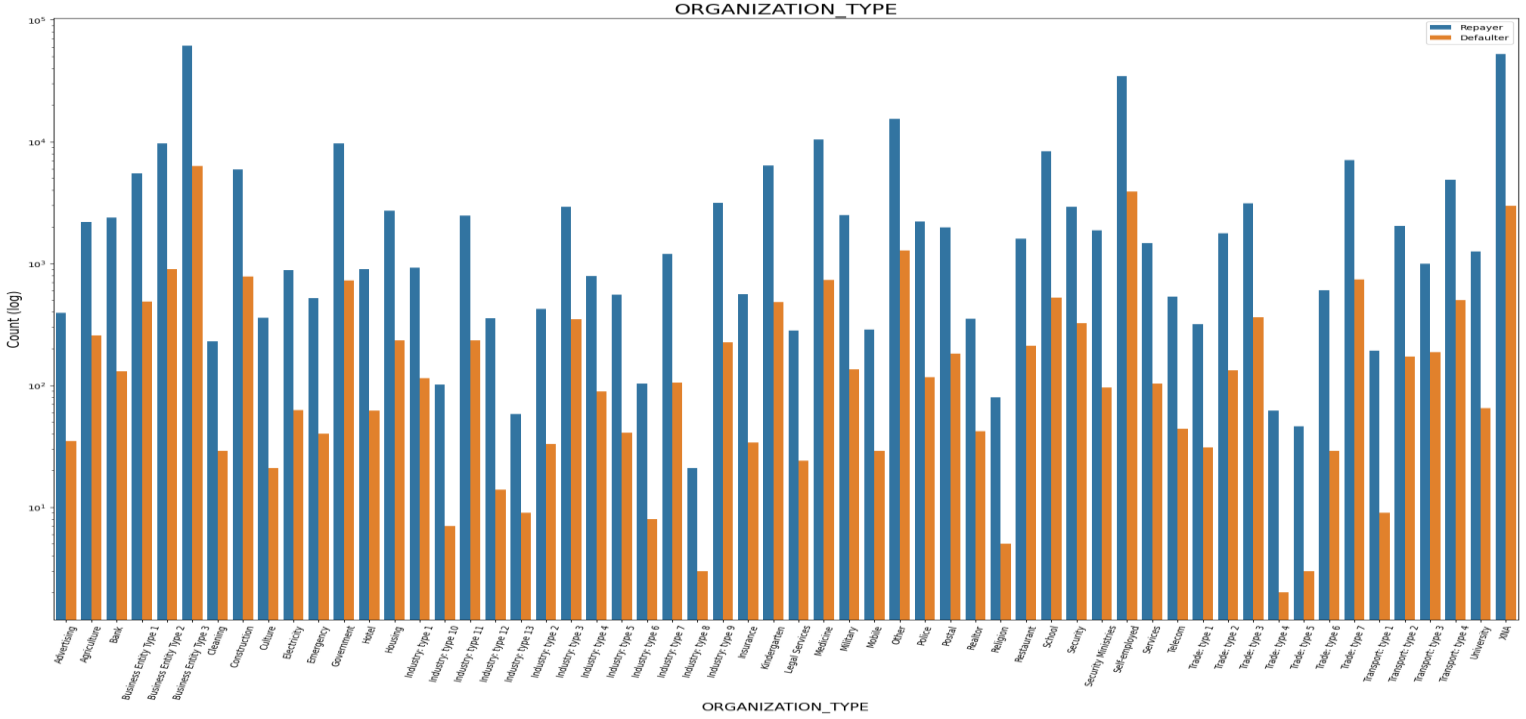
- Most of the loans are taken by Laborers, followed by Sales staff.
- IT staff are less likely to apply for Loan.
- Category with highest percent of defaulters are Low-skill Laborers (above 17%), followed by Drivers and Waiters/barmen staff, Security staff, Laborers and Cooking staff.

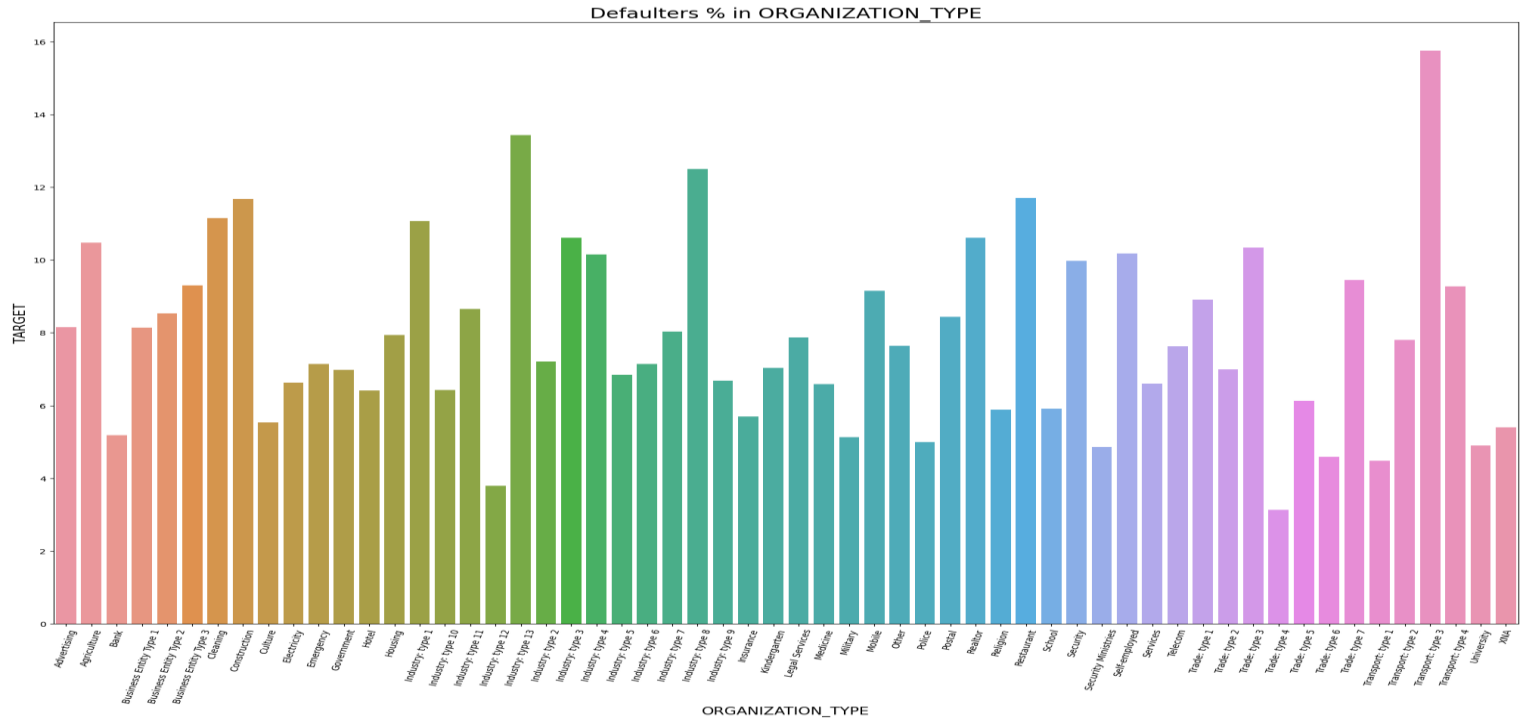




ORGANIZATION TYPE AND LOAN REPAYMENT STATUS

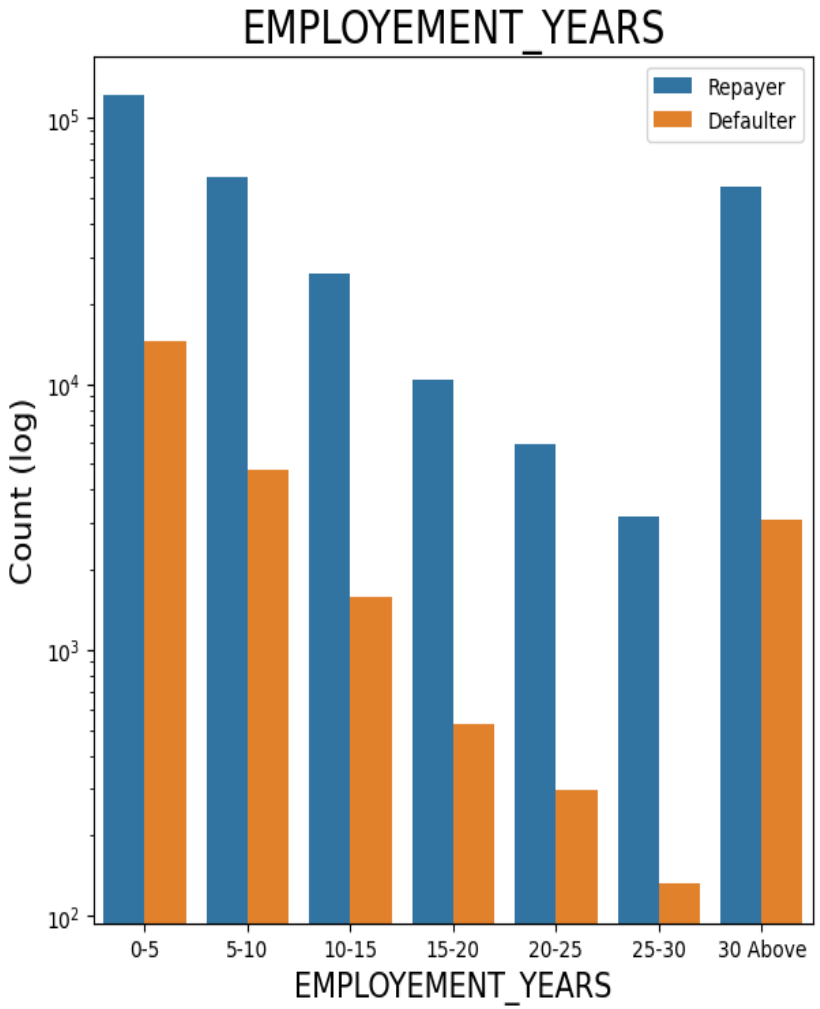
- Organizations with highest percent of defaulters are Transport: type 3 (16%), Industry: type 13 (13.5%), Industry: type 8 (12.5%) and Restaurant (less than 12%).
- Self employed people have relative high defaulting rate, to be on safer side loan disbursement should be avoided or provide loan with higher interest rate to mitigate the risk of defaulting.
- Most of the people application for loan are from Business Entity Type 3.
- For a very high number of applications, Organization type information is unavailable(XNA).
- It can be seen that following category of organization type has lesser defaulters thus safer for providing loans: Trade Type 4 and 5, Industry type 8.

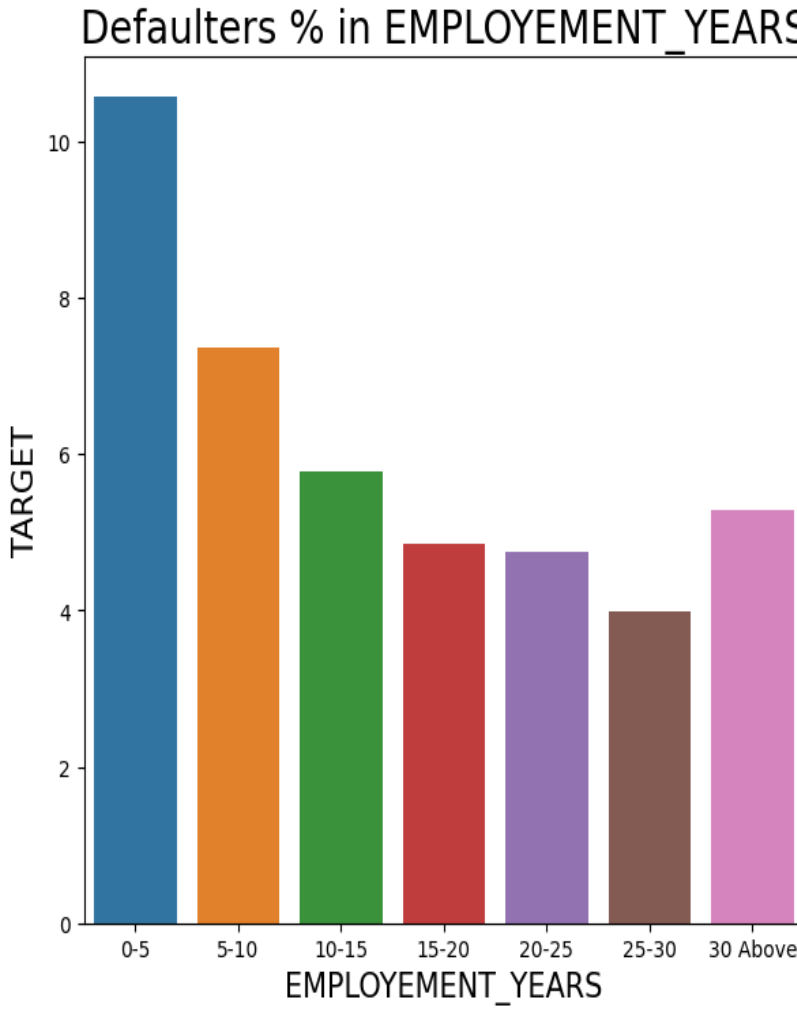




EMPLOYMENT YEARS AND LOAN REPAYMENT STATUS

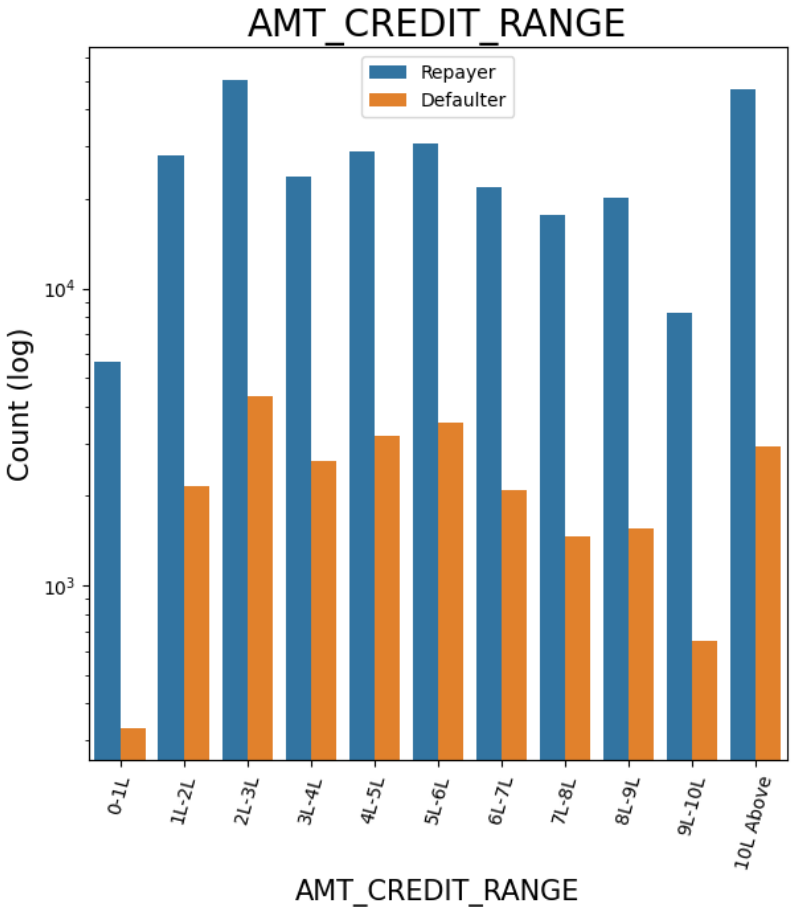
- **Majority of the applicants have working experience between 0-5 years. The defaulting rating of this group is also the highest which is around 10%.**
- **With increase of employment year, defaulting rate is gradually decreasing.**
- **Those with experience of 25-30 years have lowest default percentage.**

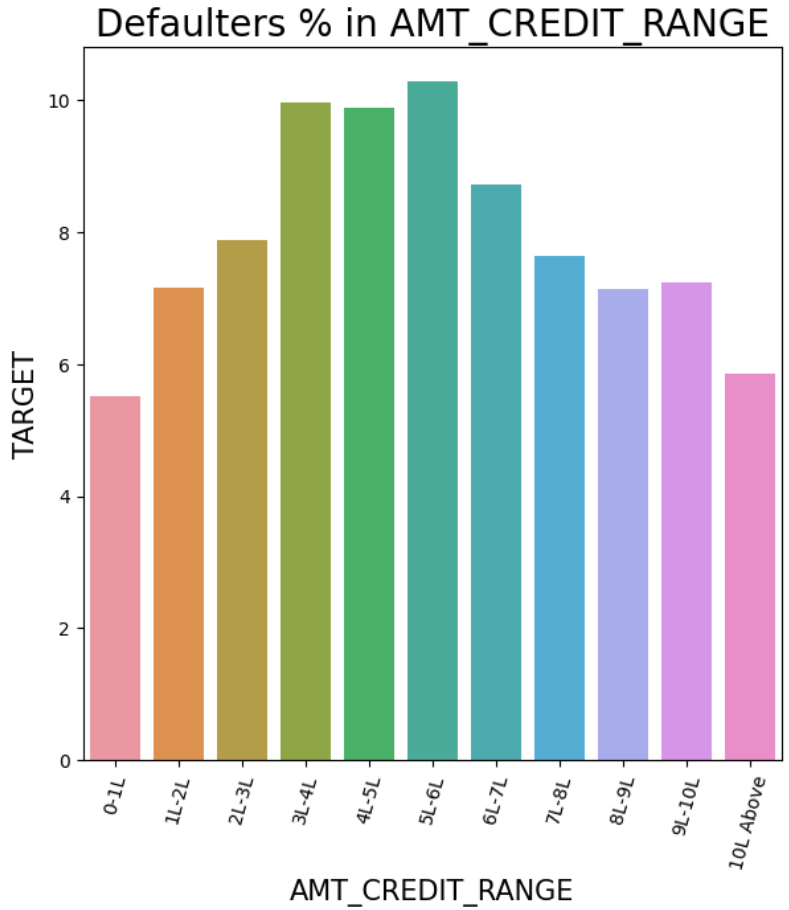




AMOUNT CREDIT AND LOAN REPAYMENT STATUS

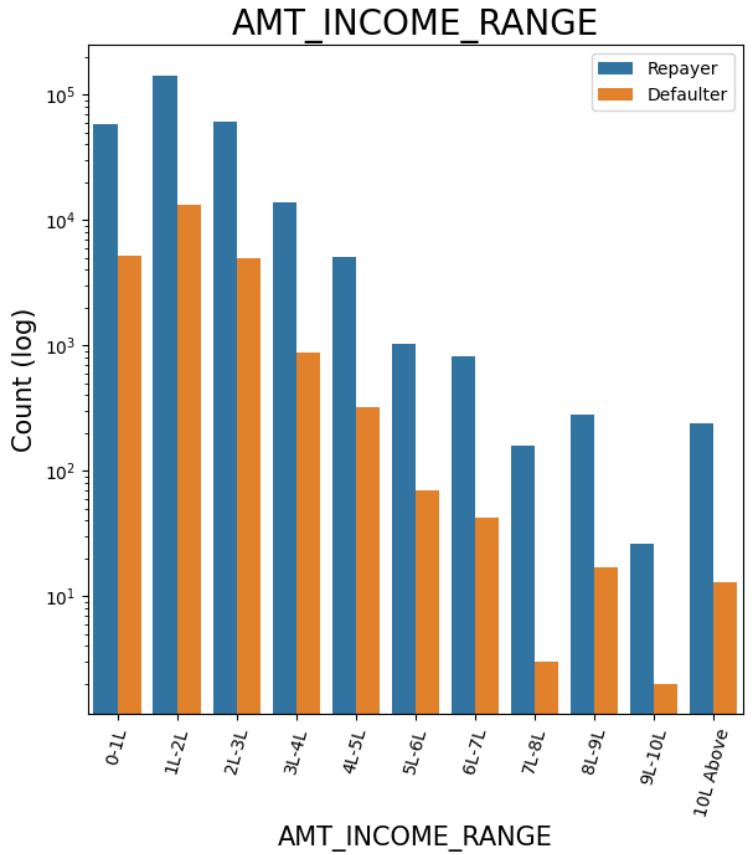
- **there are high number of applicants have loan in range of 2-3 Lakhs followed by 10 Lakh above range.**
- **People who get loan for 3-6 Lakhs have most number of defaulters than other loan range.**

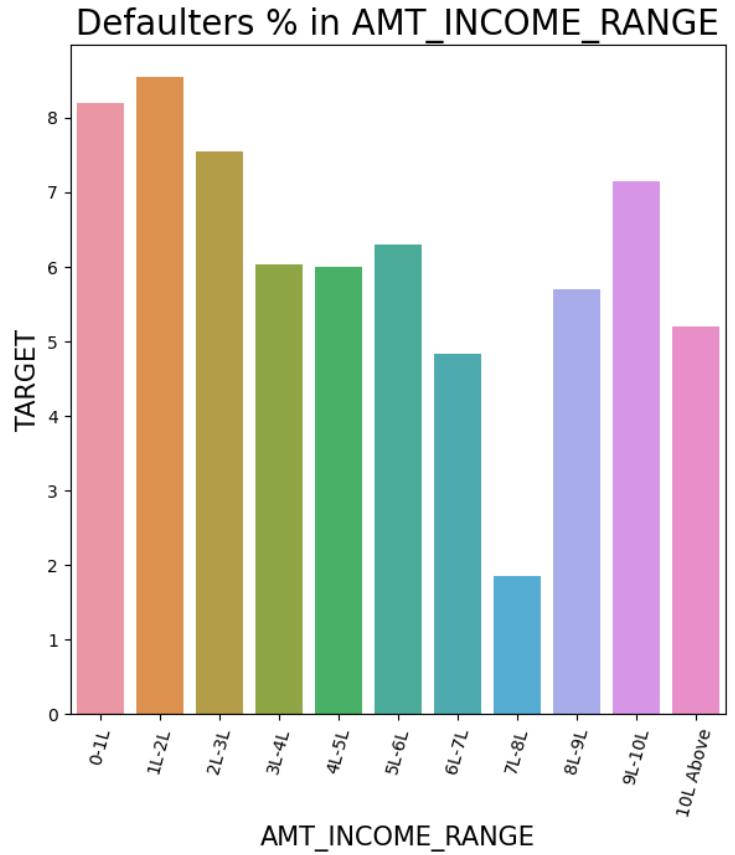




INCOME RANGE AND LOAN REPAYMENT STATUS

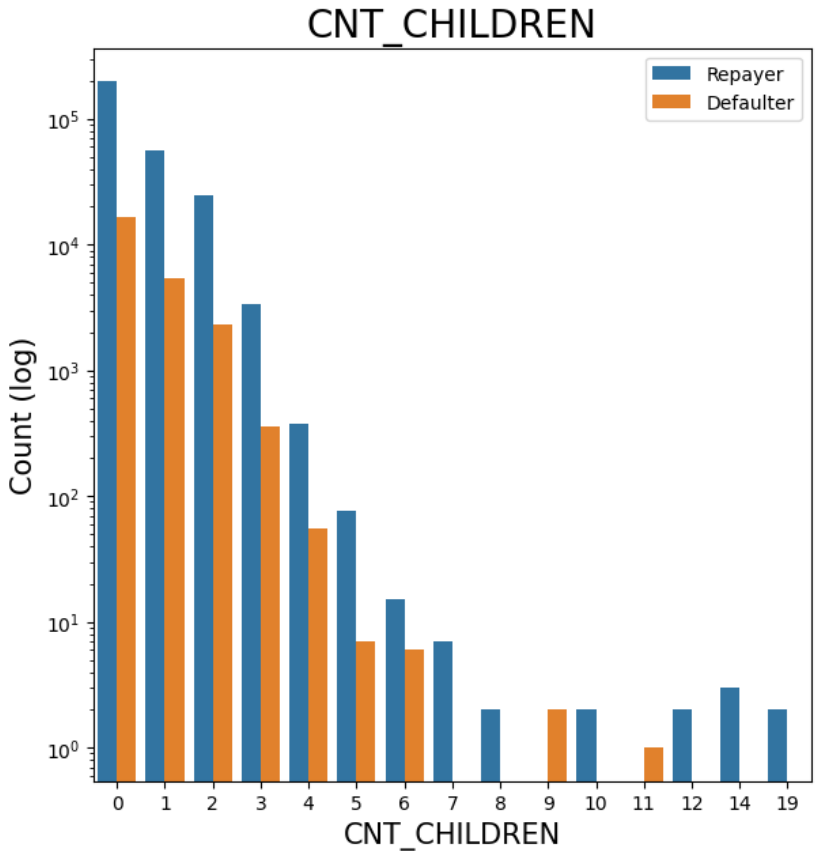
- **Majority of the applications have Income total less than 3 Lakhs.**
- **Application with Income less than 3 Lakhs has high probability of defaulting.**
- **Applicant with Income 7-8 Lakhs are less likely to default.**

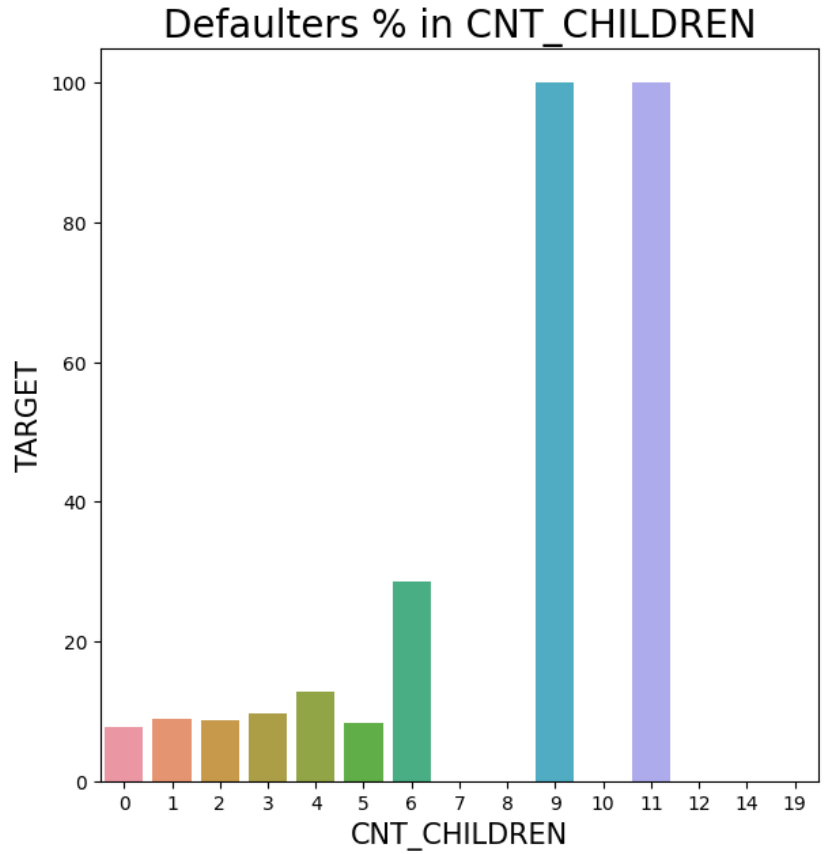




CHILDREN COUNT AND LOAN REPAYMENT STATUS

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

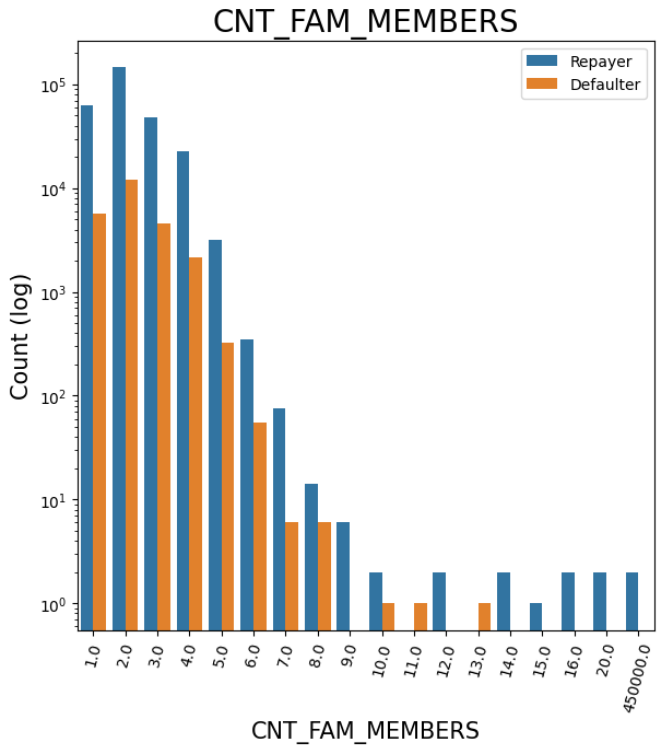


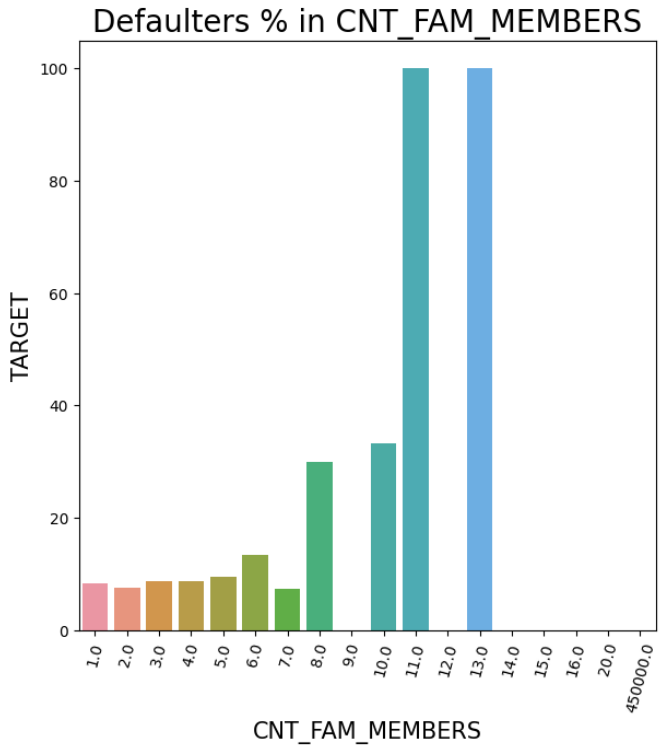


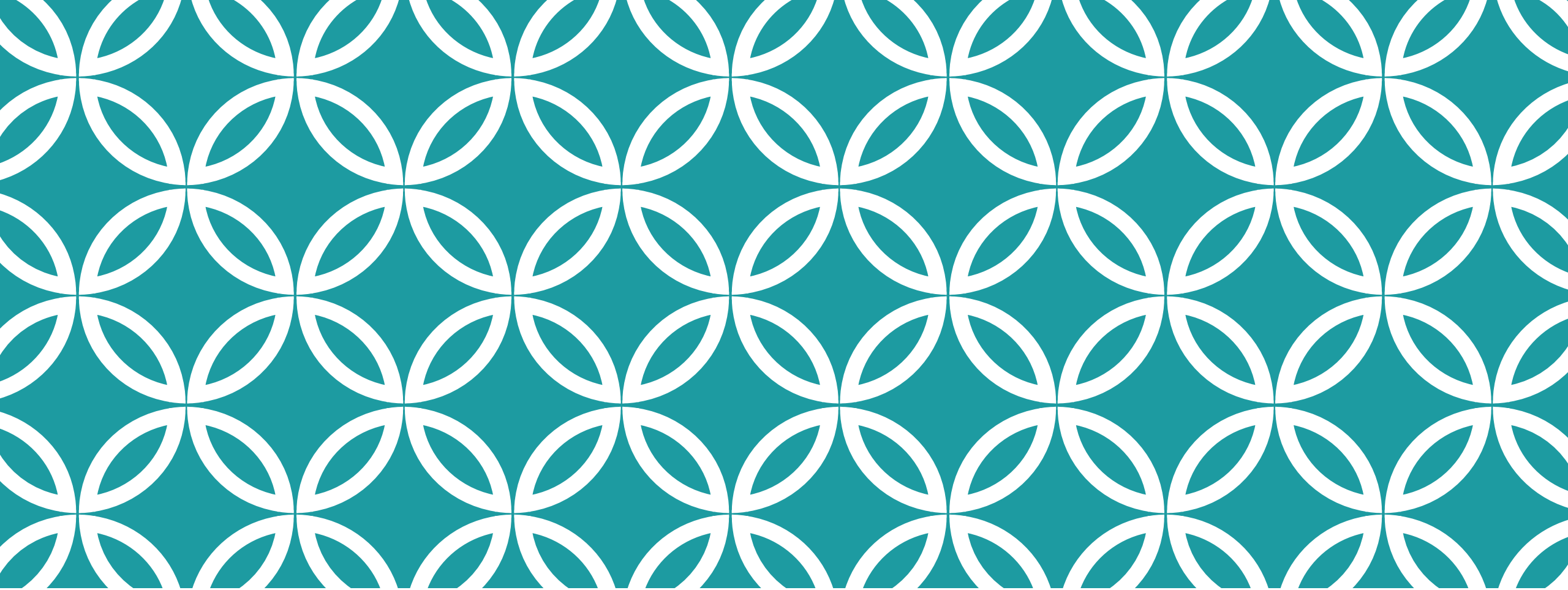
COUNT FAMILY MEMBERS AND LOAN REPAYMENT STATUS

Family member follows the same trend as children where having more family members increases the risk of defaulting.

A Family with 11 & 13 members have 100% default rate.





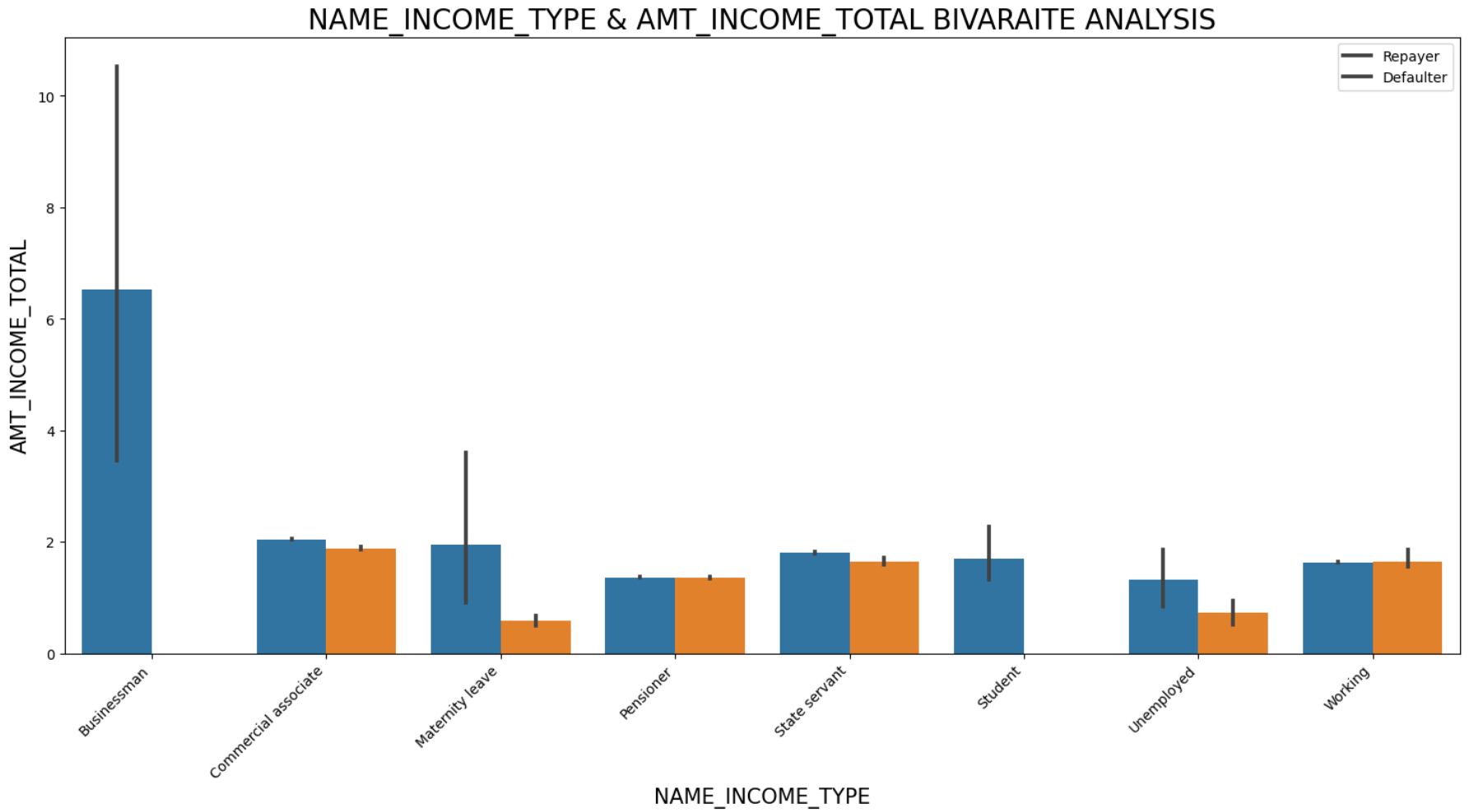


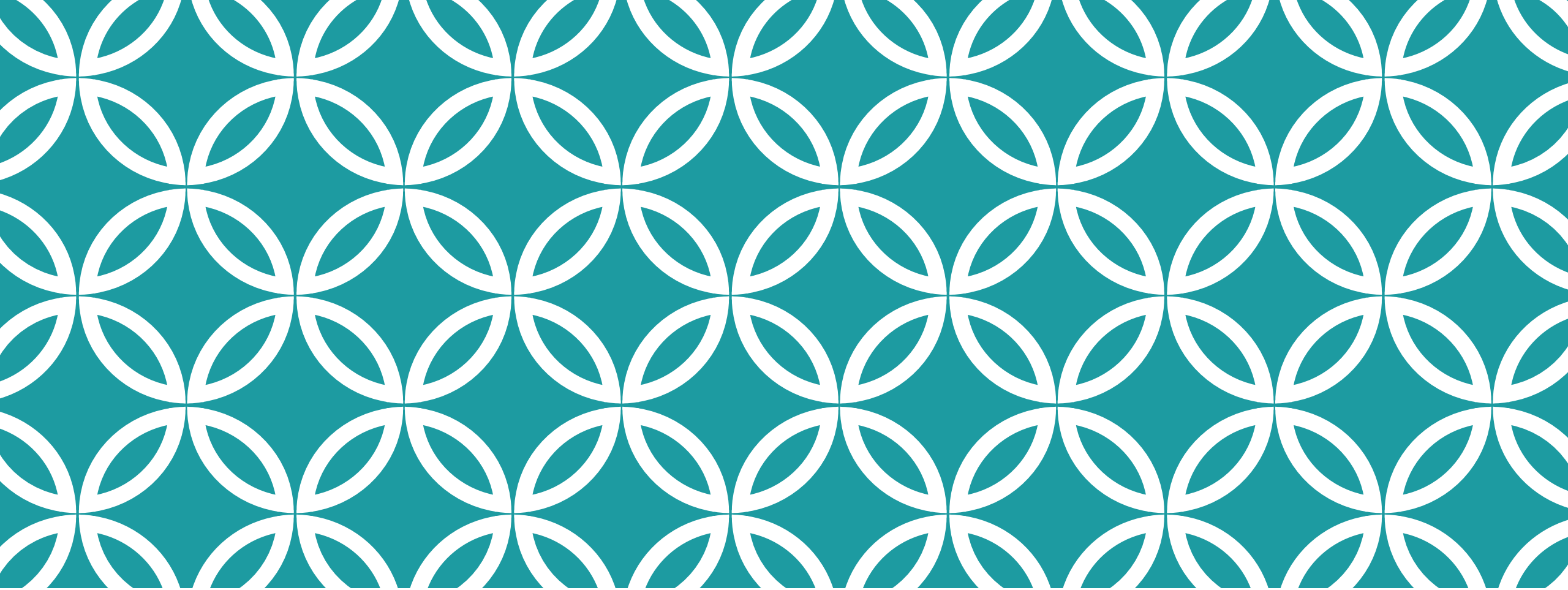
CATEGORICAL VARIABLES ANALYSIS

BIVARIATE/MULTIVARAITE ANALYSIS

INCOME TYPE AND TOTAL INCOME RELATIONSHIP

It can be seen that Businessman income is the highest and the estimated range with default 95% confidence level seem to indicate that the income of a Businessman could be in the range of slightly close to 4 lakhs and slightly above 10 lakhs.



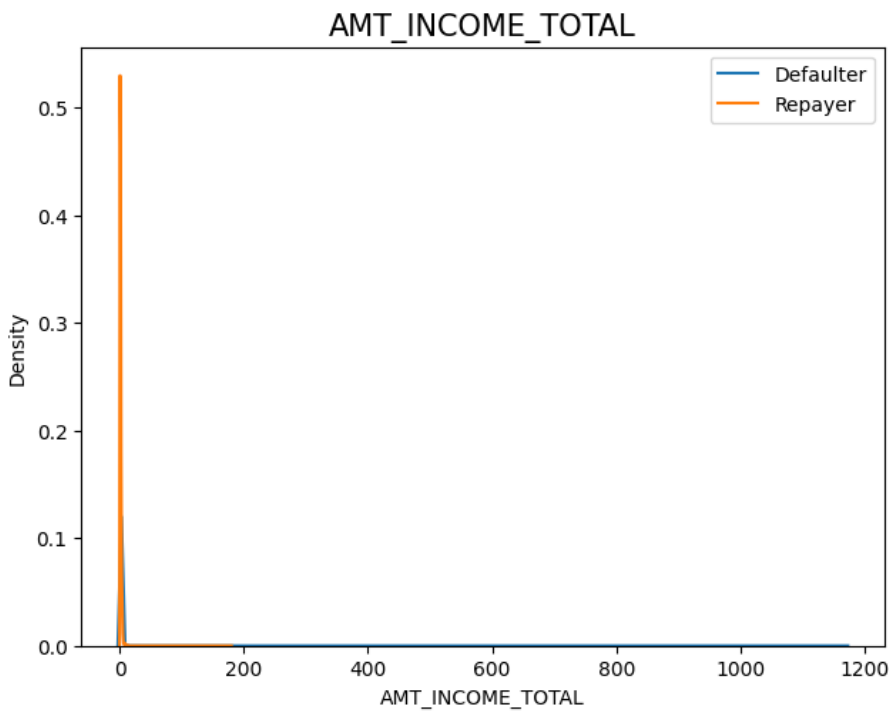


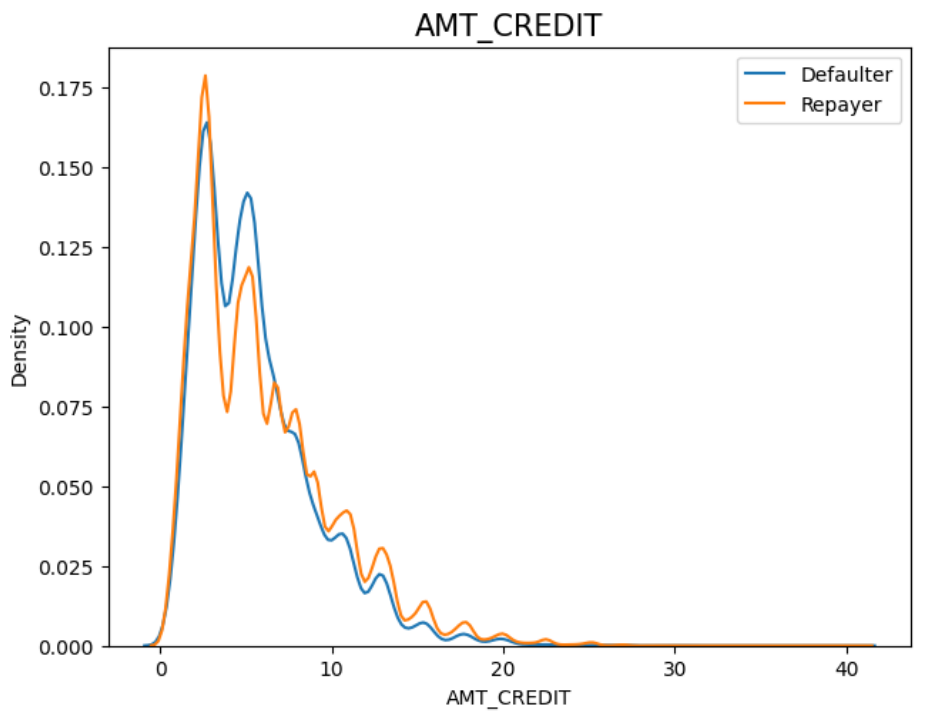
NUMERICAL VARIABLES ANALYSIS

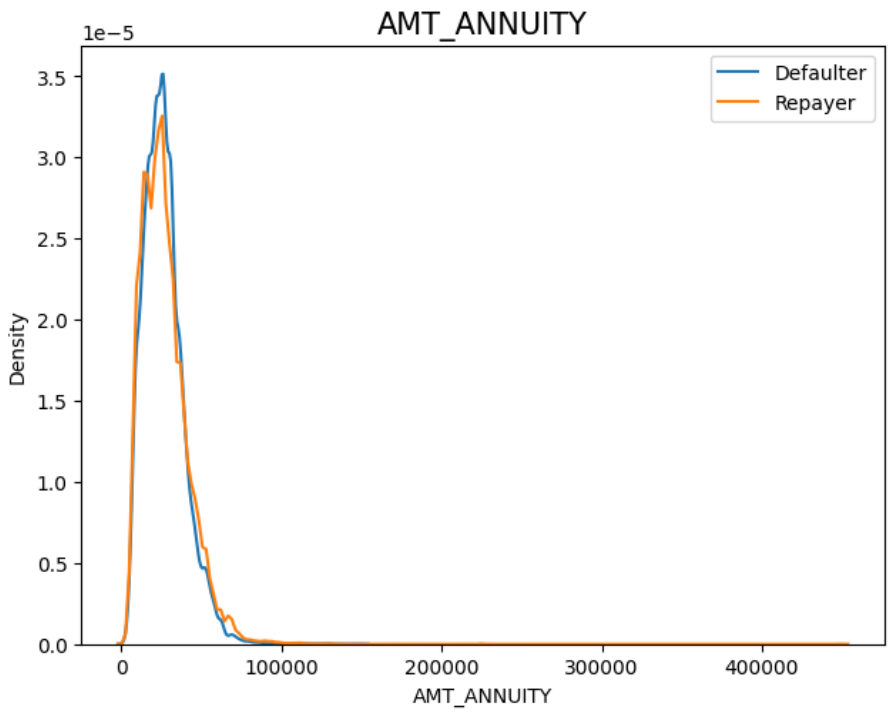
UNIVARIATE ANALYSIS

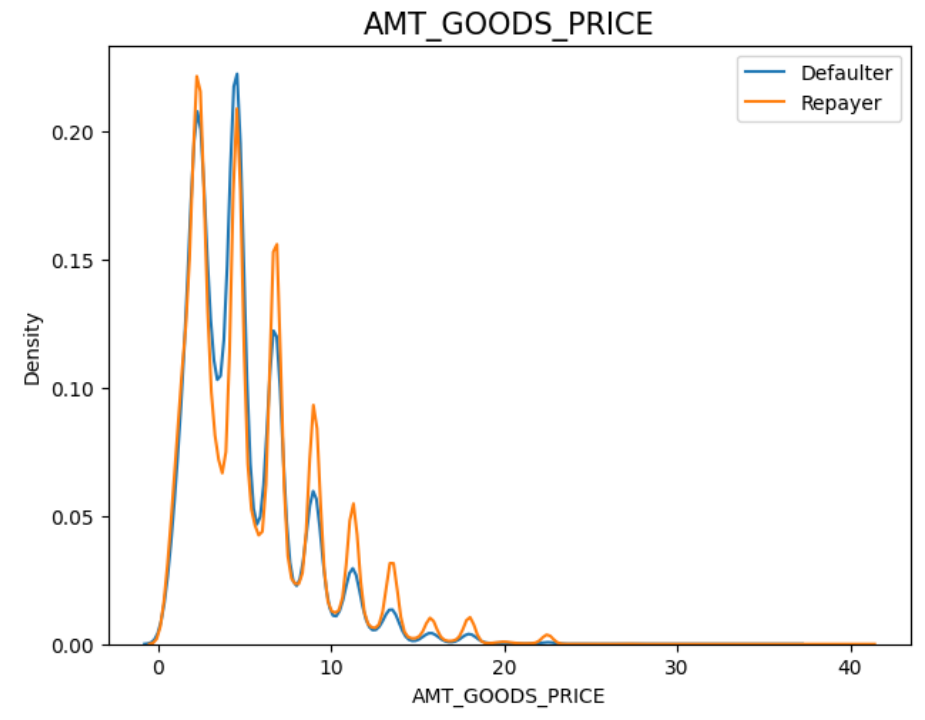
AMOUNT AND LOAN REPAYMENT STATUS

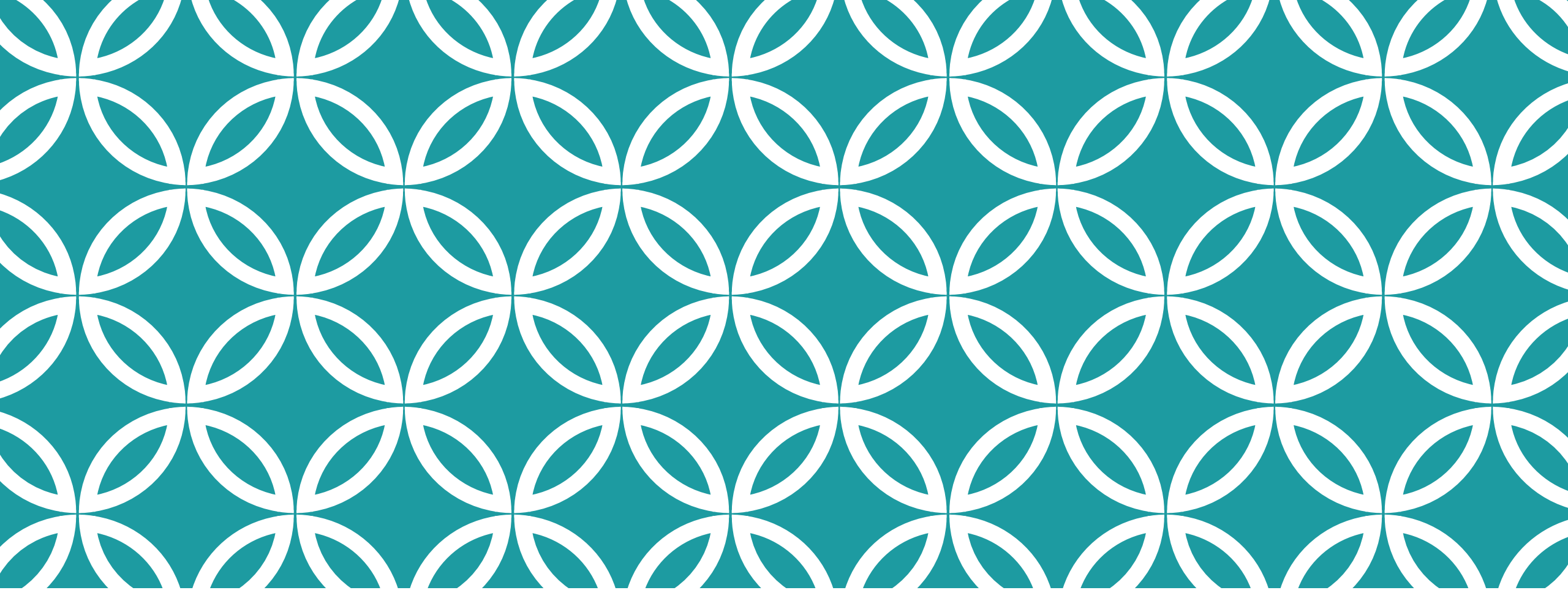
- **Most no of loans are given for goods price below 10 lakhs.**
- **Most people pay annuity below 50K for the credit loan.**
- **Credit amount of the loan is mostly less then 10 lakhs.**
- **The repayers and defaulters distribution overlap in all the plots and hence we cannot use any of these variables in isolation to make a decision.**









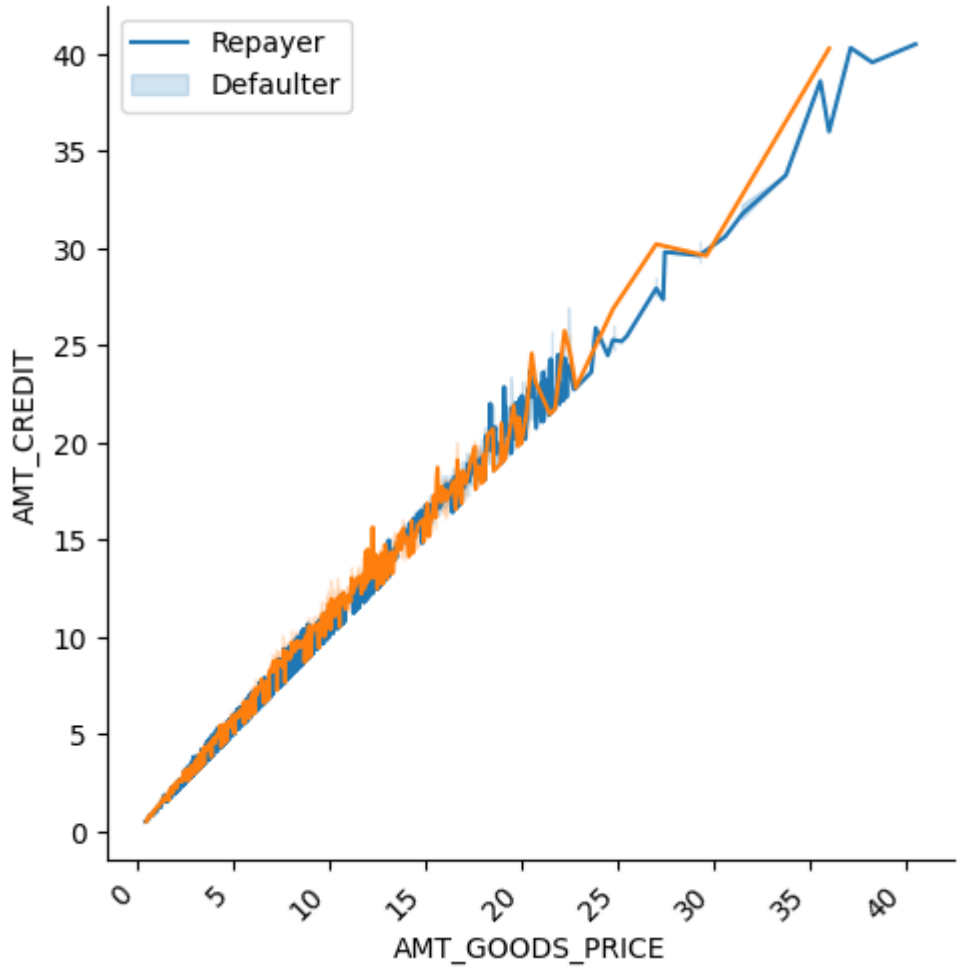


NUMERICAL VARIABLES ANALYSIS

BIVARIATE ANALYSIS

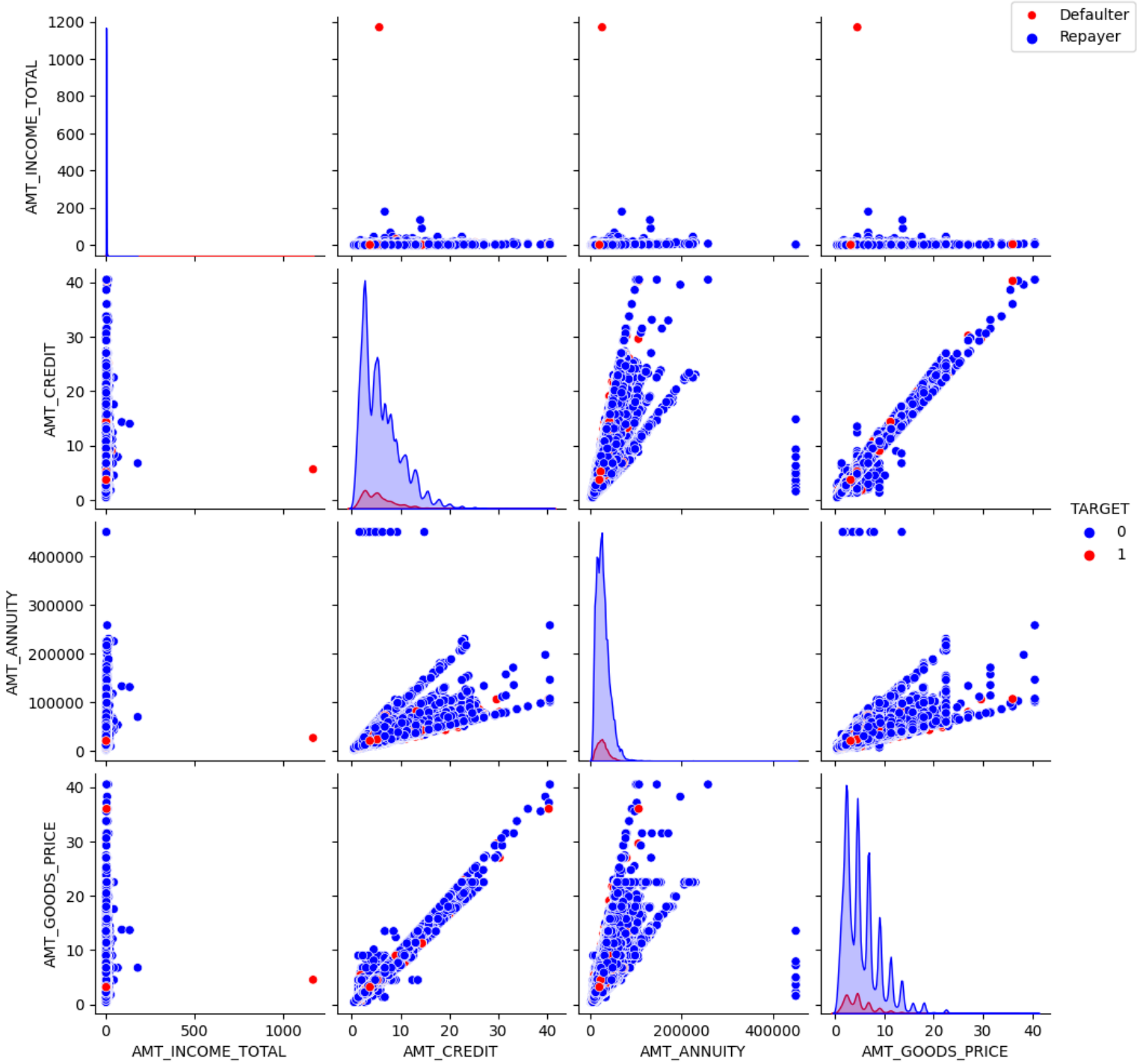
GOOD PRICE AMOUNT AND CREDIT AMOUNT BIVARIATE ANALYSIS

When the credit amount goes beyond 30 Lakhs, there is an increase in defaulters.



LOAN REPAYMENT STATUS BY PLOTTING AGAINST AMOUNT VARIABLES

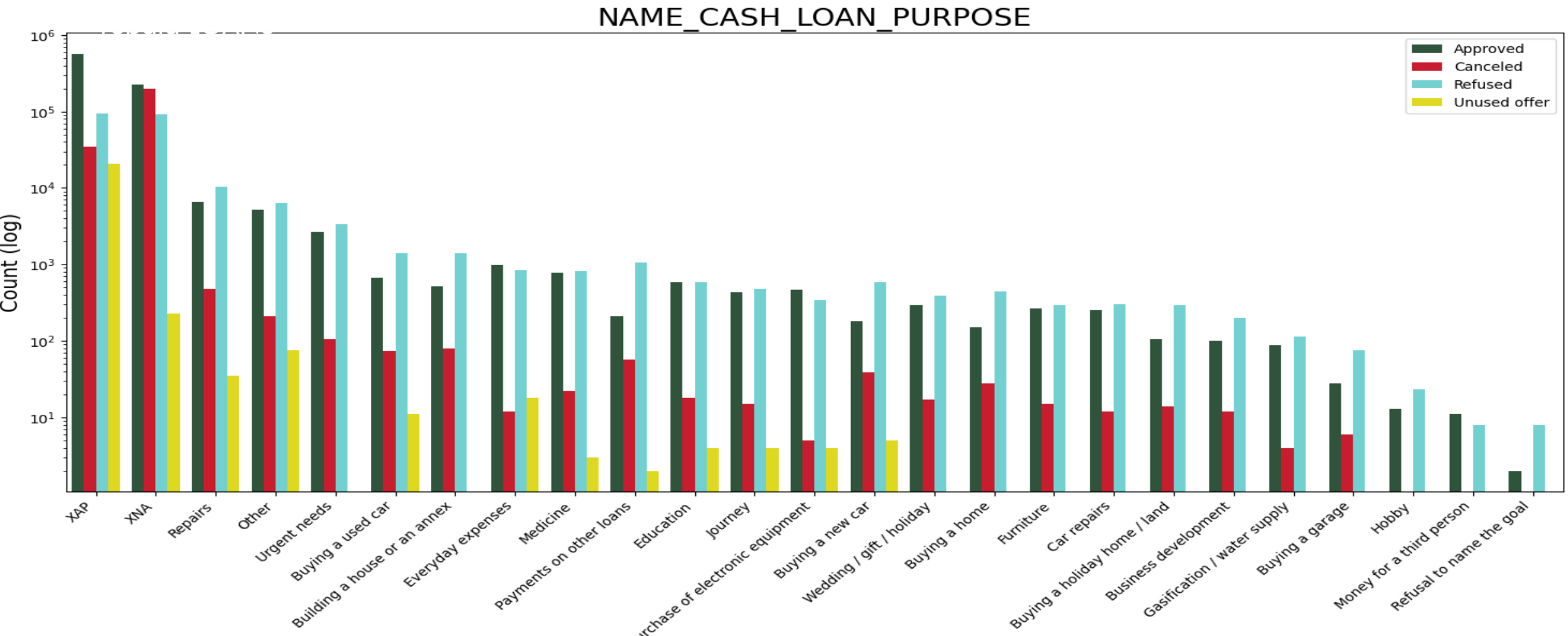
- **When Annuity Amount > 15K and Good Price Amount > 20 Lakhs, there is a lesser chance of defaulters.**
- **Loan Amount(AMT_CREDIT) and Goods price(AMT_GOODS_PRICE) are highly correlated as based on the scatterplot where most of the data are consolidated in form of a line.**
- **There are very less defaulters for AMT_CREDIT >20 Lakhs.**



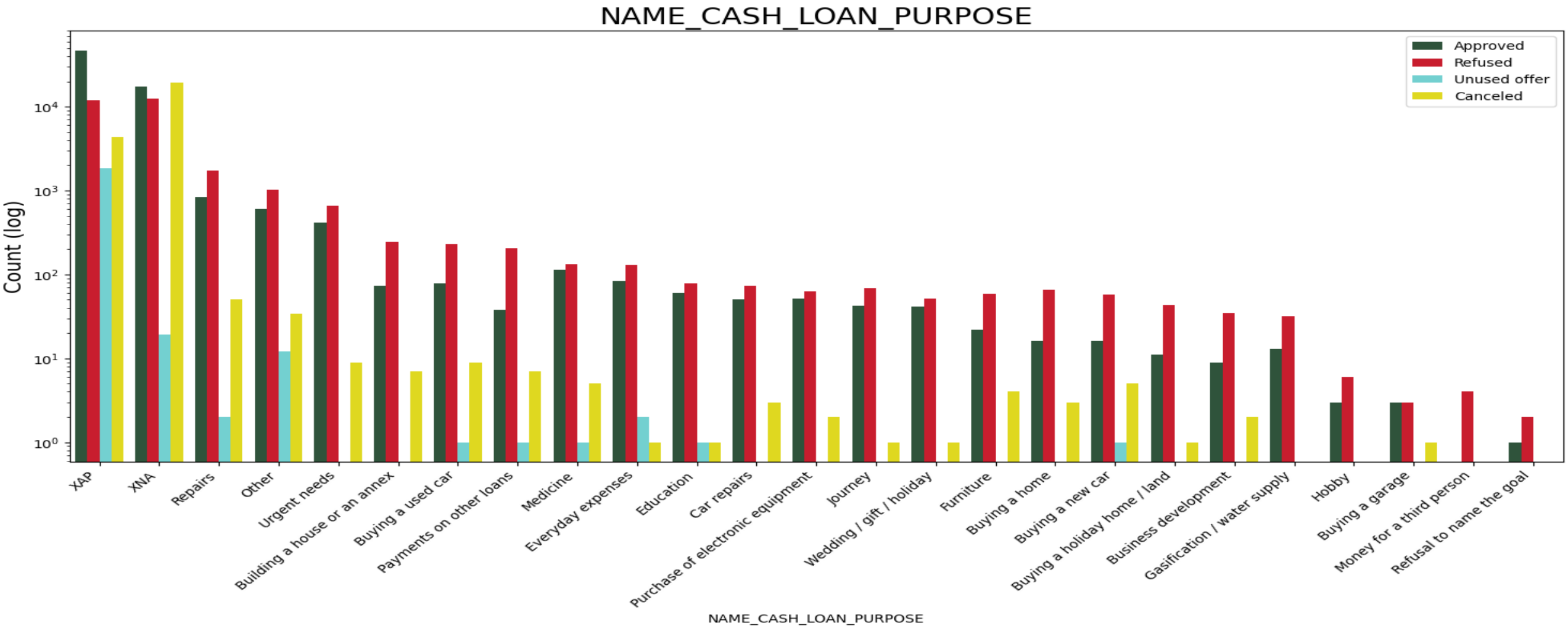
DATA SET-3 VARAIABLE ANALYSIS->MERGEED DATA SETS APPLICATION DATA ANDPREVIOUS APPLICATION

UNIVARAITE AND BIVARIATE ANALYSIS

CONTRACT STATUS AND REPAYMENT OF LOANS



CONTRACT STATUS AND REPAYMENT OF LOANS

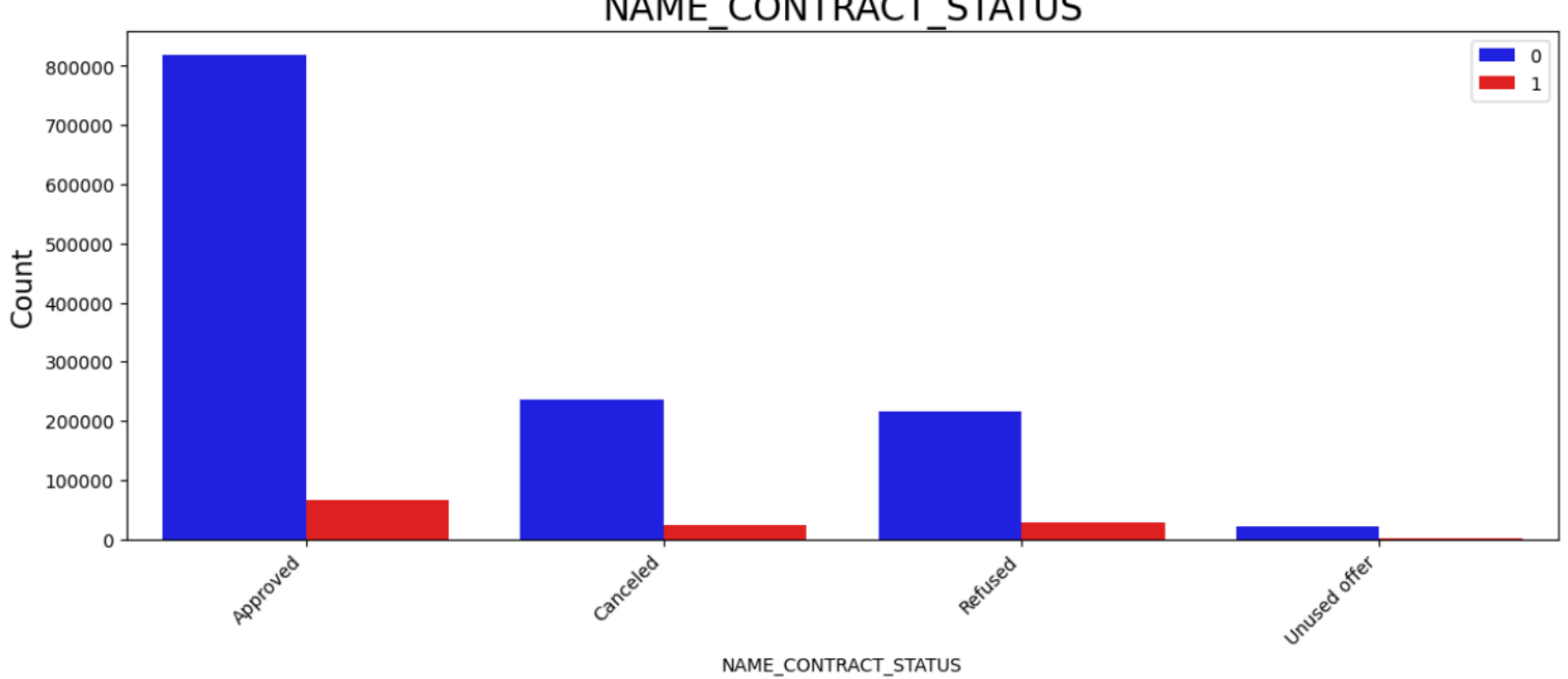


CONTRACT STATUS AND REPAYMENT OF LOANS

FROM THE PERIOUS GRAPHS THE FOLLOWING CAN BE INFERRED

- **Loan purpose has high number of unknown values (XAP, XNA).**
- **Loan taken for the purpose of Repairs looks to have highest default rate.**
- **Huge number application have been rejected by bank or refused by client which are applied for Repair or Other. from this we can infer that repair is considered high risk by bank. Also, either they are rejected or bank offers loan on high interest rate which is not feasible by the clients and they refuse the loan.**

NAME CONTRACT STATUS AND LOAN REPAYMENT STATUS

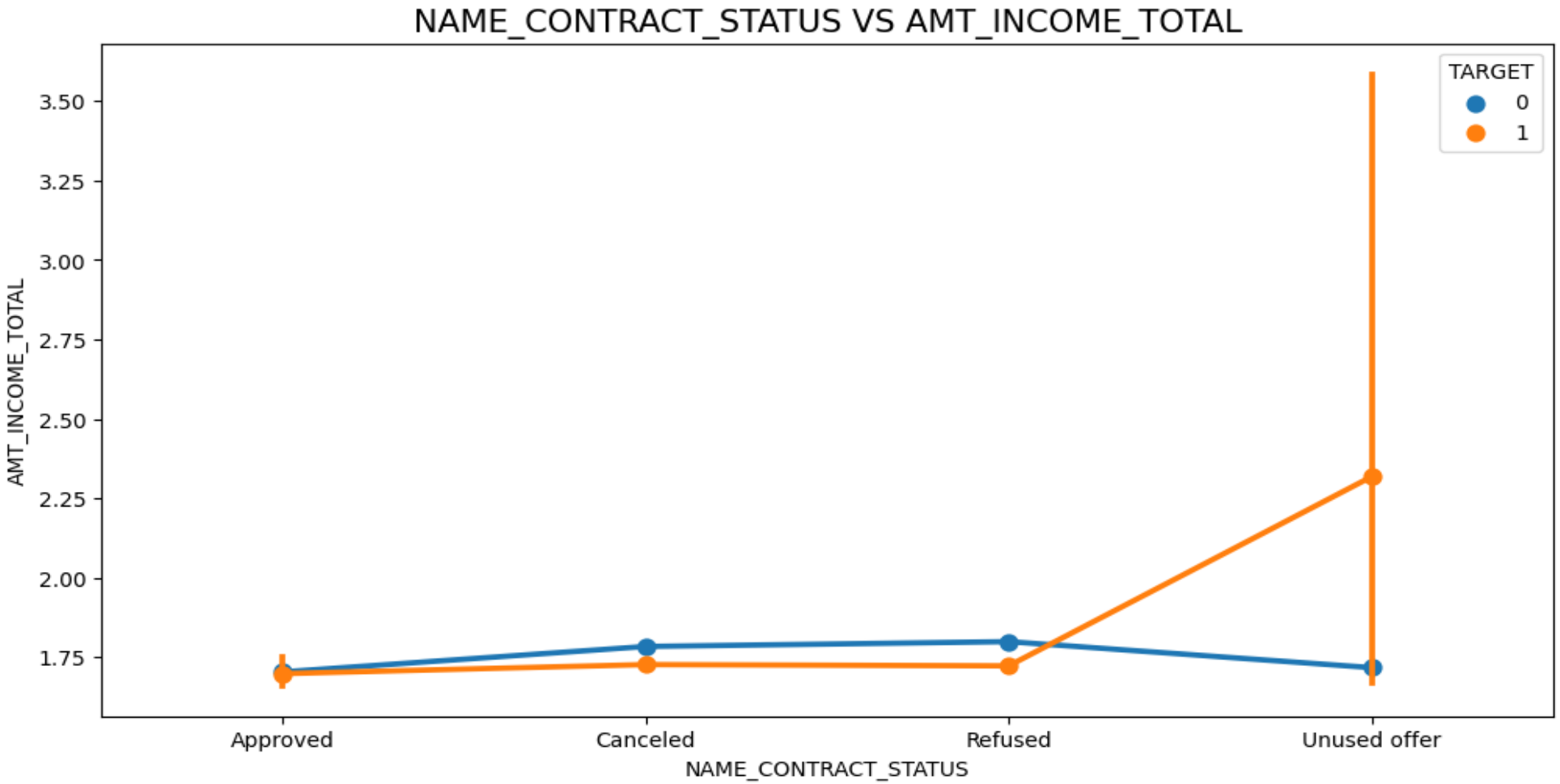


- **90% of the previously cancelled client have actually repaid the loan. Revising the interest rates would increase business opportunity for these clients.**
- **88% of the clients who have been previously refused a loan has paid back the loan in current case.**
- **Refusal reason should be recorded for further analysis as these clients could turn into potential repaying customer.**

		Counts	Percentage
NAME_CONTRACT_STATUS	TARGET		
Approved	0	818856	92.41%
	1	67243	7.59%
Canceled	0	235641	90.83%
	1	23800	9.17%
Refused	0	215952	88.0%
	1	29438	12.0%
Unused offer	0	20892	91.75%
	1	1879	8.25%

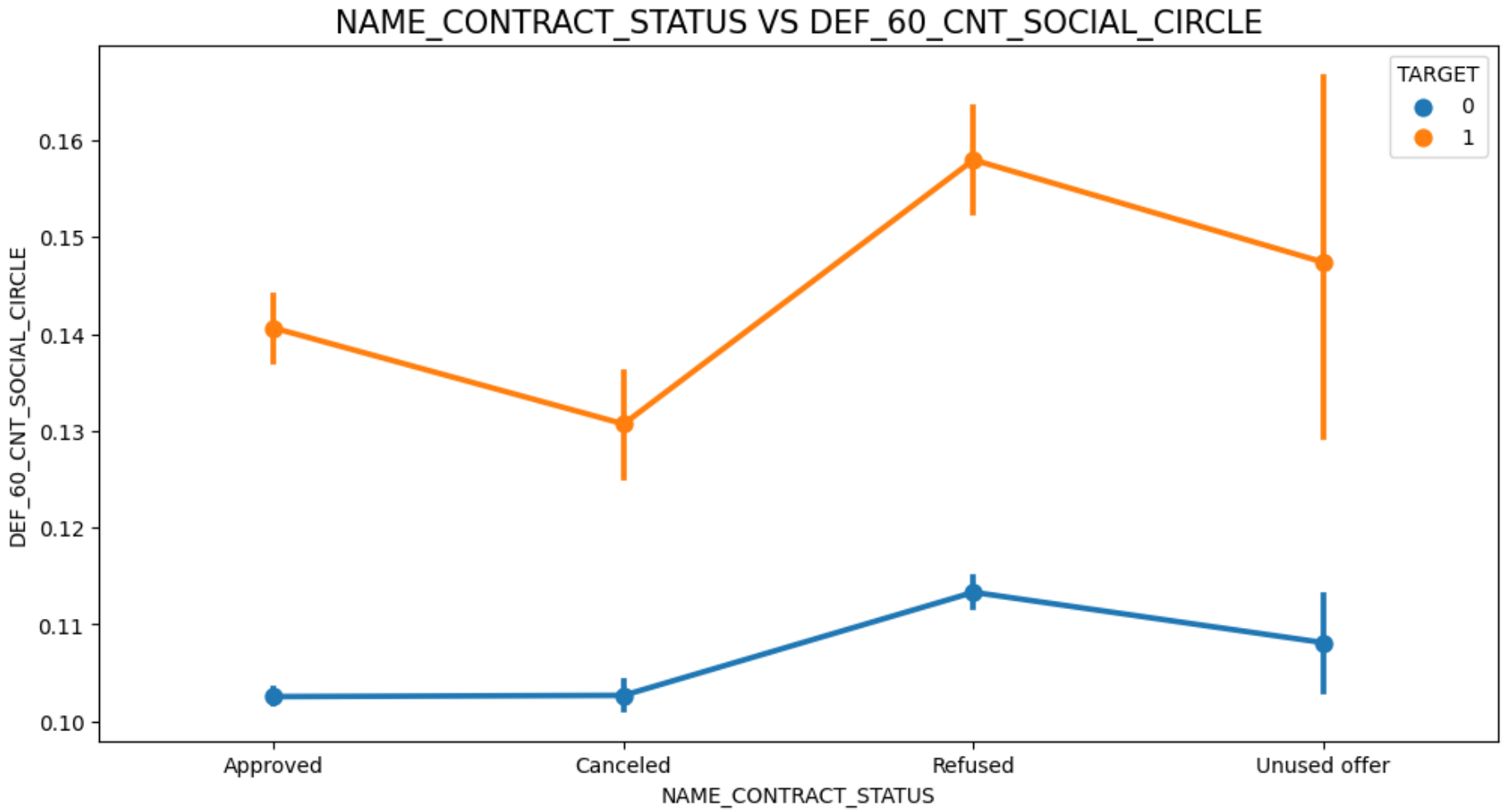
NAME CONTRACT STATUS AND AMOUNT INCOME BIVARIATE ANALYSIS

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



CONTRACT STATUS & SOCIAL CIRCLE BIVARAITE ANALYSIS

Clients who have average of 0.13 or higher their DEF_60_CNT_SOCIAL_CIRCLE score tend to default more and thus analysing client's social circle could help in disbursement of the loan.



TASK-5 CORRELATEION BETWEEN VARIABLES AND REPAYMENT/DEFAULT

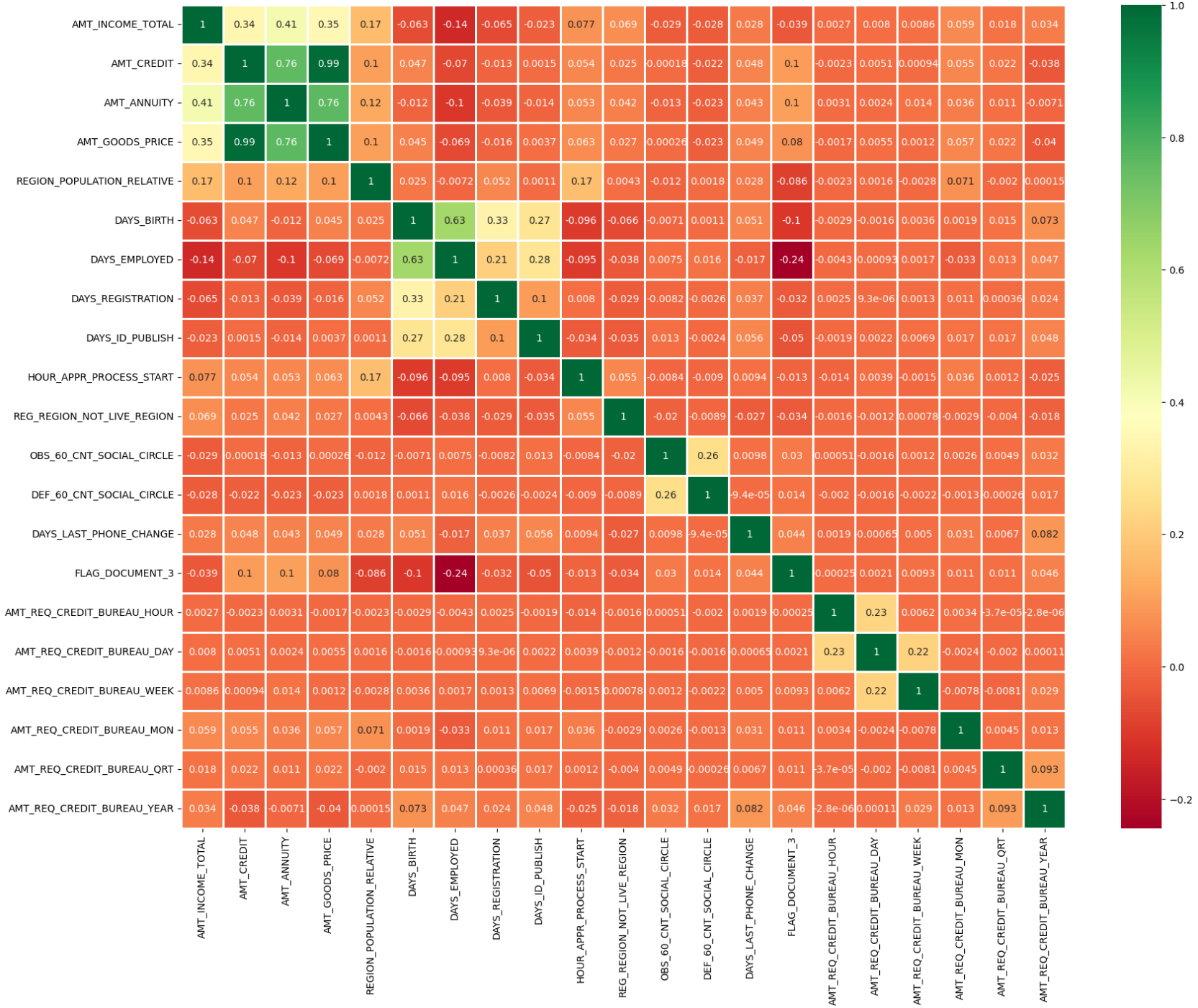
CORRELATION BETWEEN VARIABLES WHEN LOAN REPAID

	VAR1	VAR2	Correlation
64	AMT_GOODS_PRICE	AMT_CREDIT	0.987022
65	AMT_GOODS_PRICE	AMT_ANNUITY	0.762953
43	AMT_ANNUITY	AMT_CREDIT	0.757916
131	DAYS_EMPLOYED	DAYS_BIRTH	0.626114
42	AMT_ANNUITY	AMT_INCOME_TOTAL	0.411929
63	AMT_GOODS_PRICE	AMT_INCOME_TOTAL	0.349426
21	AMT_CREDIT	AMT_INCOME_TOTAL	0.342799
152	DAYS_REGISTRATION	DAYS_BIRTH	0.333151
174	DAYS_ID_PUBLISH	DAYS_EMPLOYED	0.276663
173	DAYS_ID_PUBLISH	DAYS_BIRTH	0.271314

1. Credit amount is highly correlated with:

- **Goods Price Amount**
- **Loan Annuity**
- **Total Income**

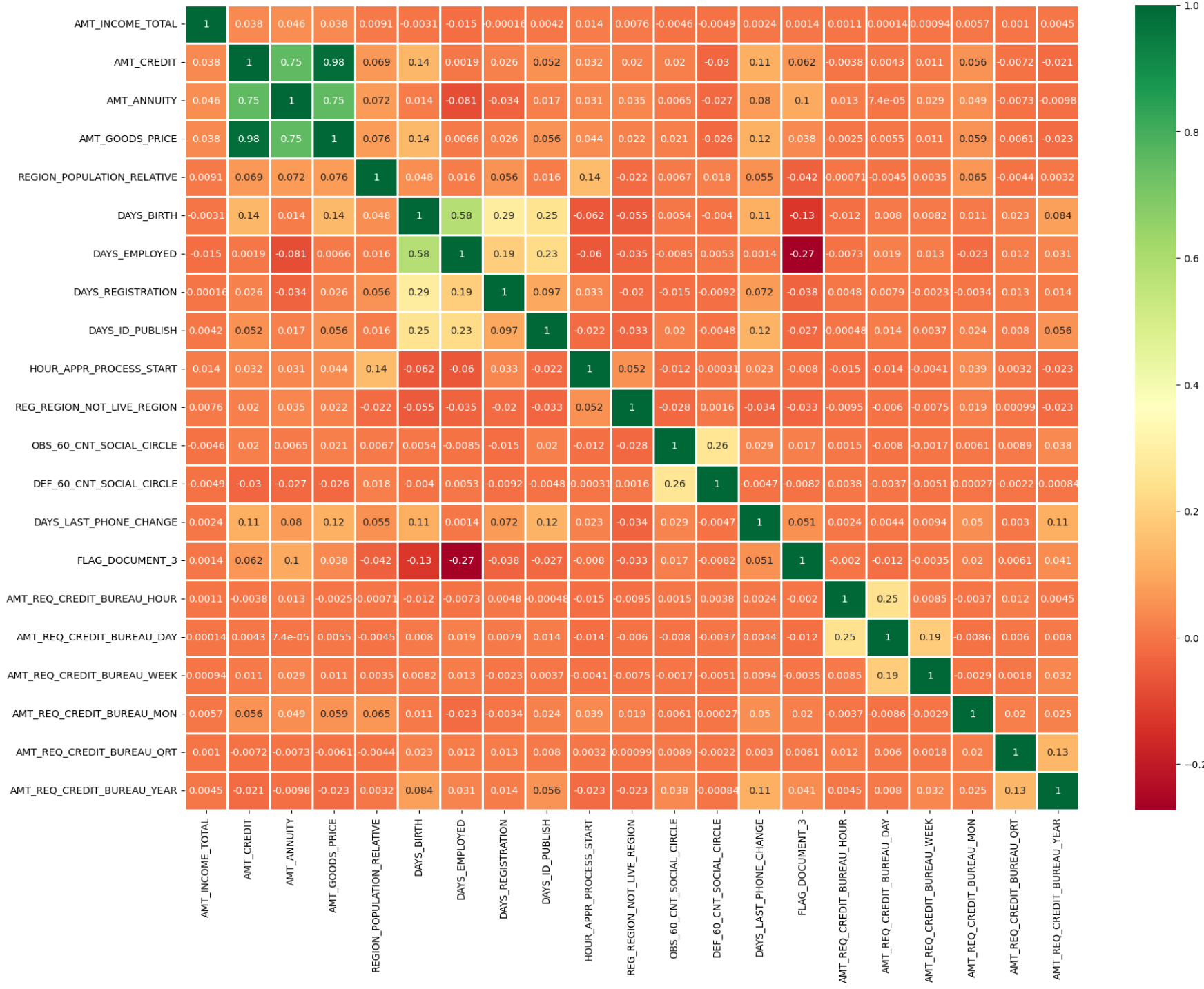
2. We can also see that repayers have high correlation in number of days employed.



CORRELATION BETWEEN VARIABLES WHEN DEFAULT

	VAR1	VAR2	Correlation
64	AMT_GOODS_PRICE	AMT_CREDIT	0.982783
65	AMT_GOODS_PRICE	AMT_ANNUITY	0.752295
43	AMT_ANNUITY	AMT_CREDIT	0.752195
131	DAYS_EMPLOYED	DAYS_BIRTH	0.582185
152	DAYS_REGISTRATION	DAYS_BIRTH	0.289114
300	FLAG_DOCUMENT_3	DAYS_EMPLOYED	0.272169
263	DEF_60_CNT_SOCIAL_CIRCLE	OBS_60_CNT_SOCIAL_CIRCLE	0.264357
173	DAYS_ID_PUBLISH	DAYS_BIRTH	0.252863
351	AMT_REQ_CREDIT_BUREAU_DAY	AMT_REQ_CREDIT_BUREAU_HOUR	0.247511
174	DAYS_ID_PUBLISH	DAYS_EMPLOYED	0.229090

- **Credit amount is highly correlated with good price amount which is same as repaid.**
- **Credit amount is highly correlated with:**
- **Goods Price Amount**
- **Loan Annuity**
- **Total Income**



TASK-6 FINAL SUMMARY CONCLUSION

FINAL SUMMARY AND CONCLUSION

After analysing the datasets, there are few attributes of a client with which the bank would be able to identify if they will repay the loan or not.

DECISIVE FACTOR WHETHER AN APPLICANT WILL BE REPAYER

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

DECISIVE FACTOR WHETHER AN APPLICANT WILL BE DEFAULTER

1. CODE_GENDER: Men are at relatively higher default rate.
2. NAME_FAMILY_STATUS : People who have civil marriage or who are single default a lot.
3. NAME_EDUCATION_TYPE: People with Lower Secondary & Secondary education.
4. NAME_INCOME_TYPE: Clients who are either at Maternity leave OR Unemployed default a lot.
5. REGION_RATING_CLIENT: People who live in Rating 3 has highest defaults.
6. OCCUPATION_TYPE: Avoid Low-skill Laborers, Drivers and Waiters/barmen staff, Security staff, Laborers and Cooking staff as their default rate is huge.
7. ORGANIZATION_TYPE: Organizations with highest percent of loans not repaid are Transport: type 3 (16%), Industry: type 13 (13.5%), Industry: type 8 (12.5%) and Restaurant (less than 12%). 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.
8. DAYS_BIRTH: Avoid young people who are in age group of 20-40 as they have higher probability of defaulting.
9. DAYS_EMPLOYED: People who have less than 5 years of employment have high default rate.
10. 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.
11. AMT_GOODS_PRICE: When the credit amount goes beyond 3lakhs, there is an increase in defaulters.

FACTORS THAT LOAN CAN BE GIVEN ON CONDITION OF HIGH INTEREST RATE TO MITIGATE ANY DEFAULT RISK LEADING TO BUSINESS LOSS:

1. NAME_HOUSING_TYPE: High number of loan applications are from the category of people who live in Rented apartments & living with parents and hence offering the loan would mitigate the loss if any of those default.

2. AMT_CREDIT: People who get loan for 3-6 Lakhs tend to default more than others and hence having higher interest specifically for this credit range would be ideal.

3. AMT_INCOME: Since 90% of the applications have Income total less than 3Lakhs and they have high probability of defaulting, they could be offered loan with higher interest compared to other income category.

4. CNT_CHILDREN & CNT_FAM_MEMBERS: Clients who have 4 to 8 children has a very high default rate and hence higher interest should be imposed on their loans.

5. NAME_CASH_LOAN_PURPOSE: Loan taken for the purpose of Repairs seems to have highest default rate. A very high number applications have been rejected by bank or refused by client in previous applications as well which has purpose as repair or other. This shows that purpose repair is taken as high risk by bank and either they are rejected, or bank offers very high loan interest rate which is not feasible by the clients, thus they refuse the loan. The same approach could be followed in future as well.

SUGGESTIONS

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

RESULT

The project helped in strengthening ability to use python for data analysis.

Due to the detailed nature of the project multiple rows and columns, the project helped strengthen the usage of pandas, NumPy, matplotlib and seaborn libraires.

The project helped in perfecting my data cleaning and data visualisation methodologies.

I have also worked on making a more effective presentation.