UP GRAD : Advanced Certification in Data Science

EDA : Assignment

The general approach followed during working on assignment:

- First working on previous application:
    1. checking the data for duplicates, then checking for missing values.
    2. Fixing the missing values, and some repeated values.
    3. Checking for unbalanced data creating buckets for the same.
    4. Finally do some plotting and analysis and group the data according to previous SK ID and finally create a new Data frame to be merged with current application Data.
- Current Application Data:
    1. Step “1” and “2” above were repeated for the current data.
    2. Merging the previous data with current data on current SK ID and checking for missing values, in the new merged Data frame.
    3. Checking Imbalance and outliers in the new Data frame and fixing the same.
    4. Finally doing univariate, bi variate and regression analysis on the dataset.

Data Understanding:

Previous Application Dataset:

Shape of previous dataset: (1670214, 37)

- While observing the head it was noticed that some columns are having similar values: namely "AMT_APPLICATION" and "AMT_GOODS_PRICE",

```
In [9]: df_0.AMT_APPLICATION.equals(df_0.AMT_GOODS_PRICE)

Out[9]: False
```

- On further investigation it was revealed that for the rows for which the above columns were not equal were nan in all of the columns, except for 232 rows which were rejected cases, so the column with nan’s was dropped. Reason columns with similar values.
- In cases where the client had made more than one application per contract, except for the last application which was flagged as Y, all the applications except the last/flagged one were rejected so the column “FLAG_LAST_APPL_PER_CONTRACT” was dropped as well as the N flagged rows. Reason for duplicate application for same contract on last one is considered by the bank.
- While checking SK ID current: it was noted that many of the current applicants has applied for multiple loans previously.

```
In [25]: df_0.SK_ID_CURR.value_counts()

Out[25]: 187868    77
         265681    73
         173680    72
         206783    67
         156367    65
                   ..
         241434     1
         419859     1
         249128     1
         240397     1
         191629     1
         Name: SK_ID_CURR, Length: 338857, dtype: int64
```

Handling NAN :

- Drop columns with nan greater than 50% and also categorical columns with sum of "XAP" and "XNA"(Information Not Available) grater than 50%.
- While checking the remaining NAN's it was noted that "DAYS_FIRST_DRAWING"
  And the other columns with DAYS as prefix were having same number and exactly same rows as nan's.
  it was further noted that the rows with "NAN's" were mostly rejected or cancelled, except for "39632 rows, so the status of rows were changed to cancelled and the values of columns with "DAYS" prefix for the same rows were changed from "NAN" to "365243", as all the dates were nan it appears that the loan was approved but not withdrawn.

```
In [42]: df_0[df_0.DAYS_FIRST_DRAWING.isnull()].NAME_CONTRACT_STATUS.value_counts()

Out[42]: Canceled        316317
         Refused         282205
         Approved         39632
         Unused offer     26436
         Name: NAME_CONTRACT_STATUS, dtype: int64
```

- After making the above adjustment it was noted that the columns with DAYS prefix as null is exactly the same as sum of all the cases not approved, so the above assumption is makes sense, the "NAN's" were retained as it will not affect the analysis for the approved cases.

```
In [45]: df_0.isna().sum().sort_values(ascending=False).head(15)

Out[45]: NFLAG_INSURED_ON_APPROVAL    664590
         DAYS_TERMINATION             664590
         DAYS_LAST_DUE                664590
         DAYS_LAST_DUE_1ST_VERSION    664590
         DAYS_FIRST_DUE               664590
         DAYS_FIRST_DRAWING           664590

In [46]: df_0[df_0.NAME_CONTRACT_STATUS!="Approved"].shape[0]

Out[46]: 664590
```

- "AMT_ANNUITY" and "CNT_PAYMENT" column was null for rejected or cancelled cases except for "4", so the four rows were dropped and "NAN" was retained.
- For the remaining columns as the "NAN's" were less than 500, the subsequent rows were dropped.
- A lot of columns were having multiple rows with 365243 as value:
  Handling 365243.
  - While checking the column "DAYS_FIRST_DRAWING" it was noted that for the cases where "DAYS_FIRST_DRAWING" !="365243" the "Named Portfolio" was equal to cards, and "CNT_PAYMENT" =0, except for "5" cases.

```
In [56]: # it appears that CNT_PAYMENT =0 and NAME_PORTFOLIO = cards for the DAYS_FIRST_DRAWING columns!=365243
         df_0[(df_0.DAYS_FIRST_DRAWING!=365243)&(~df_0.DAYS_FIRST_DRAWING.isnull())].CNT_PAYMENT.value_counts()

Out[56]: 0.0     62700
         24.0        2
         12.0        2
         18.0        1
         Name: CNT_PAYMENT, dtype: int64

In [57]: df_0[(df_0.DAYS_FIRST_DRAWING!=365243)&(~df_0.DAYS_FIRST_DRAWING.isnull())].NAME_PORTFOLIO.value_counts()

Out[57]: Cards    62700
         Cash         5
         Name: NAME_PORTFOLIO, dtype: int64
```

  - On further investigation it was revealed that for rows in "DAYS_FIRST_DRAWING" =365243, all of them were card users.
  - Out of 93759 card users for
    "DAYS_FIRST_DRAWING" =365243, for "31059 Customers" all card users
    "DAYS_FIRST_DRAWING"!=365243, for "62700 Customers" all card users except 5

```
In [61]: df_0[(df_0.DAYS_FIRST_DRAWING==365243)&(df_0.NAME_PORTFOLIO=="Cards")].DAYS_LAST_DUE_1ST_VERSION.value_counts()

Out[61]: 365243.0    31059
         Name: DAYS_LAST_DUE_1ST_VERSION, dtype: int64
```

```
In [58]: # NAMED_PORTFOLIO values
         df_0[df_0.NAME_CONTRACT_STATUS=="Approved"].NAME_PORTFOLIO.value_counts()

Out[58]: POS       614382
         Cash      288786
         Cards      93759
         Cars         218
         Name: NAME_PORTFOLIO, dtype: int64
```

- Finally it was concluded that the rows for which DAYS_FIRST_DRAWING=365243 and NAME_PORTFOLIO=="Cards" are the customers whose credit card was approved but they did not used it. So the rows were converted to cancelled. As the user did not used the loan.
- After the above change only card users who have used the Credit Card at least once are having DAYS_FIRST_DRAWING !=365243.
- On further investigation on 365243 it was revealed that :
  - DAYS_FIRST_DUE is the due date of first instalment in days relative to current application, if the first due date is after current application day its value =365243.
  - DAYS_LAST_DUE_1ST_VERSION =365243 for card users, in other cases it is the number of days w.r.t. current application when last due of the previous application is supposed to be paid.
  - DAYS_LAST_DUE is days relative to current application when the customer paid the last payment, else it is =365243.
  - DAYS_TERMINATION is days relative to current application when the contract was terminated, else=365243. Generally, one week or so after last payment.
- Finally the Days data was converted into months

```
In [82]: # convert data in days to months
         for i in ["DAYS_DECISION","DAYS_FIRST_DRAWING","DAYS_FIRST_DUE","DAYS_LAST_DUE_1ST_VERSION","DAYS_LAST_DUE","DAYS_TERMINATION"]:
             df_0["Months"+i[4::]]=df_0.apply(lambda x: round(x[i]/30,1),axis=1)
```

- A column "late_payment_last_in_month" was created to know the number of months the client defaulted in last payment if MONTHS_LAST_DUE that is the last payment was after MONTHS_LAST_DUE_CURRENT_VERSION. As there are only 947 such customers it will not affect the overall analysis.

```
In [89]: df_0[df_0["late_payment_last_in_month"]>0].shape[0]

Out[89]: 947
```

- A column "Flag previous running" column was created to highlight customers having previous loan running at the beginning of current application. It was noted that there are more than one lakh cases with previous loan still running, some of them multiple loans on same current SK ID.

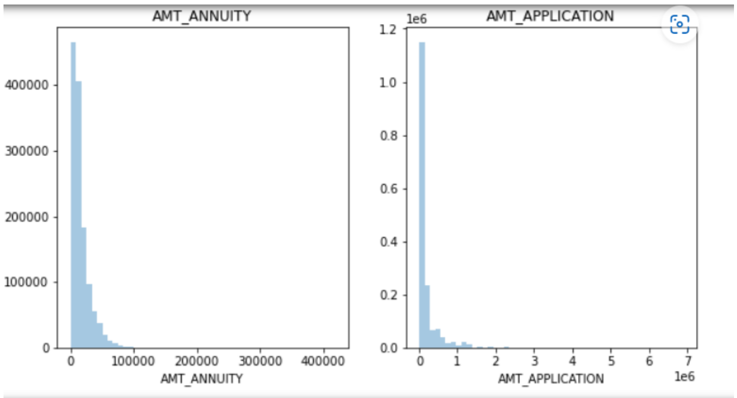
```
In [88]: df_0["Flag previous running"].value_counts()

Out[88]: 0    1522059
         1     139330
         Name: Flag previous running, dtype: int64
```

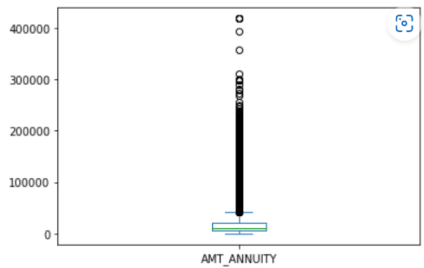
- A column "Annuity_previous_agg" was also created to know the previous annuity amount if the previous loan is running.

Analysis: a small analysis was conducted on the previous dataset to identify imbalances and outliers.

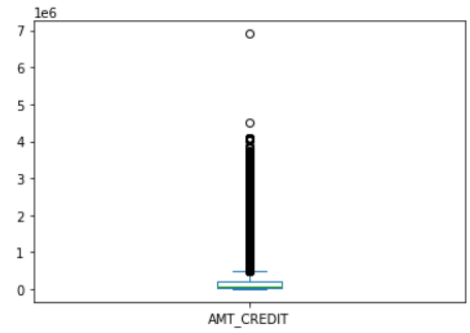
- It was found that the columns "AMT_ANNUITY","AMT_CREDIT","CNT_PAYMENT" the data was highly imbalanced and similar in shape.



Out[95]: <AxesSubplot:>



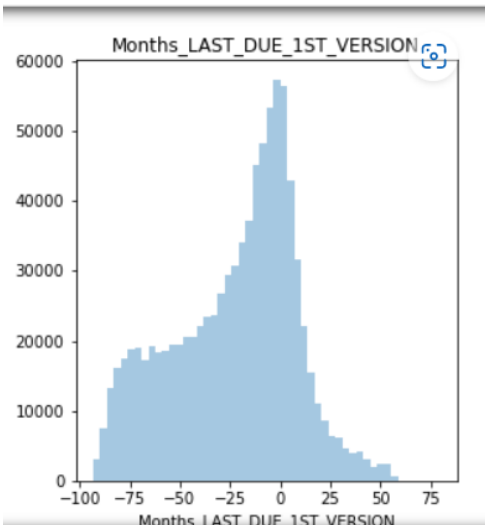
Out[96]: <AxesSubplot:>



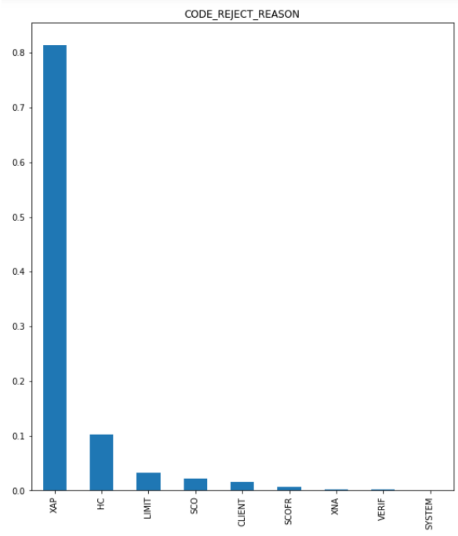
- Annuity was imputed to 90th Quantile,
- While the data for the remaining two columns was divided into categorical buckets.

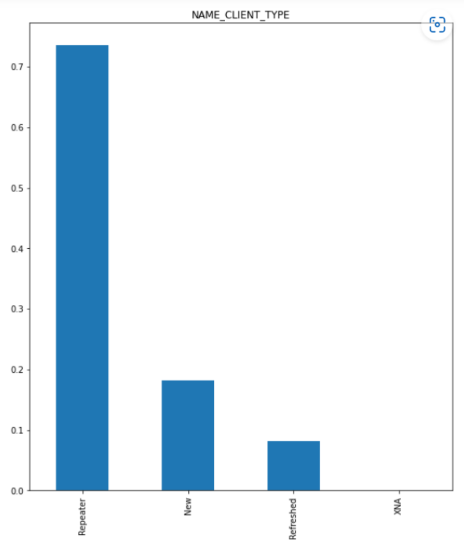
```
In [102]: df_0["Credit_buckets"]=pd.cut(df_0.AMT_CREDIT,[0,10000,50000,500000,2000000],labels=["<10K","10K-50K","50K-500k","500K+"])
```

- There are lot of clients for whom the last due date of previous loan is after the current application was applied, but sum of them have already settled the previous loan as MONTHS_LAST_DUE is not 365243.

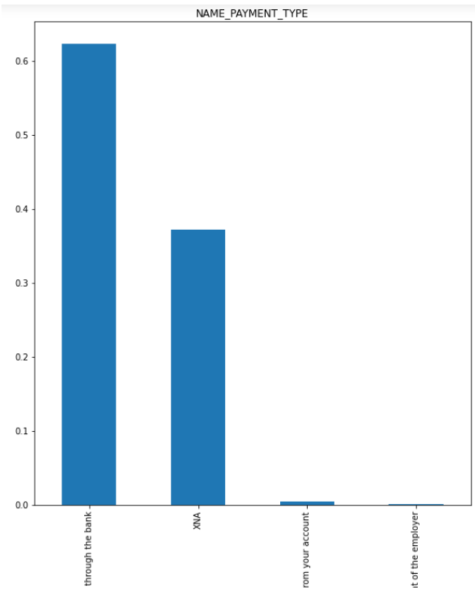


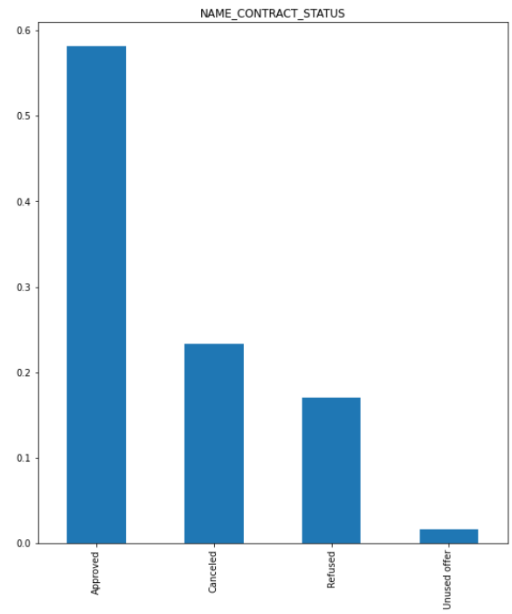
- Categorical Analysis:



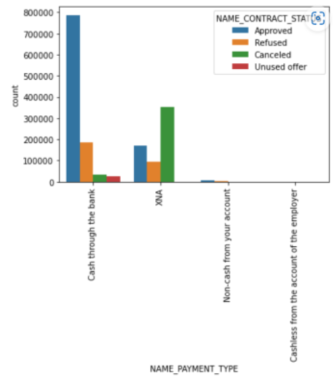


- HC is the top rejection reason in the previous process.
- 70% of the clients are repeater in the previous process.
- “NAME_CONTRACT_TYPE” and “NAME_PAYMENT_TYPE” bar charts were having similar shape so a bivariate analysis was conducted.

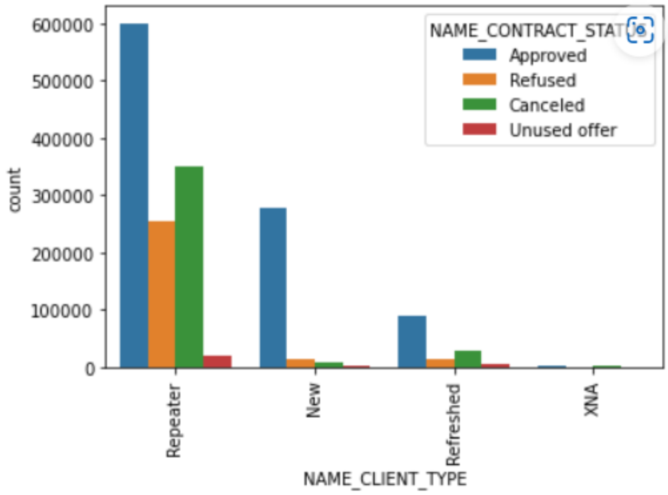




- Most of cash through bank cases were approved.



- Approval rate for new customers is higher than repeaters.



<u>Aggregating the data:</u>

- The data was aggregated on CURR_SK_ID using pivot tables and variable that could pe used to analyse the current data frame were aggregated.
- Finally the DF was saved in a new csv file. “agg_func” =sum for all the variables caried forward.
- “Annuity_Prev_agg” is sum of previous annuities if still running.

```
In [110]: df_1=df_0.pivot_table(index="SK_ID_CURR",values=["Annuity_previous_agg","Flag_Approved","Flag_Canceled","Flag_Refused"],
```

Application Data.

- Data understanding

  - Shape of the dataset: (307511, 122)
  - Target

```
In [4]: df.TARGET.value_counts()
Out[4]: 0    282686
        1     24825
        Name: TARGET, dtype: int64
```

Data cleaning and Manipulation:

- Remove columns with nan values grater than 50%. The columns were reduced from 120 to 81.

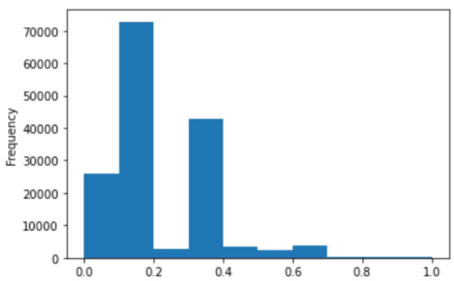
```
In [13]: for i in df.columns:
             if df[i].isnull().sum()>=.5*x:
                 df.drop(i,axis=1,inplace=True)

In [14]: df.shape
Out[14]: (307511, 81)
```

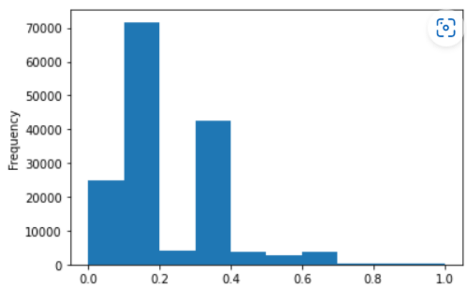
- The columns “FLOORSMAX_AVG”, “FOORSMAX_MODE”, “FLOORSMAX_MEDI” are having same no and exactly same rows as nan’s. Same is true for columns with “YEARS_” prefix in the table below. The distribution plot for the above tables was also same and highly imbalanced so a column "Prop_Discription" was created to flag if the columns are “NAN’s” as approx. 50% of the data is missing and the remaining data is also imbalanced. All the six columns were dropped afterwards.

```
In [15]: df.isna().sum().sort_values(ascending=False).head(20)
Out[15]: FLOORSMAX_AVG                   153020
         FLOORSMAX_MODE                  153020
         FLOORSMAX_MEDI                  153020
         YEARS_BEGINEXPLUATATION_AVG     150007
         YEARS_BEGINEXPLUATATION_MODE    150007
         YEARS_BEGINEXPLUATATION_MEDI    150007
```

```
In [19]: df.FLOORSMAX_MODE.plot.hist()
Out[19]: &lt;AxesSubplot:ylabel='Frequency'&gt;
```




```
In [18]: df.FLOORSMAX_MEDI.plot.hist()
Out[18]: &lt;AxesSubplot:ylabel='Frequency'&gt;
```



- For the column “OCCUPATION_TYPE” a new category “Undisclosed” was created and NAN were replaced by “Undisclosed”.
- The column “EXT_SOURCE_3” and “EXT_SOURCE_2” have same mean, and S.D. and similar quantiles values, so the nan in both columns were replaced by mean value and finally a new

column "External_Source_avg" was created and the average of the two column was taken row wise, and the above two columns were dropped afterwards.

```
In [29]: df.EXT_SOURCE_3.describe()

Out[29]: count    246546.000000
         mean          0.510853
         std           0.194844
         min           0.000527
         25%           0.370650
         50%           0.535276
         75%           0.669057
         max           0.896010
         Name: EXT_SOURCE_3, dtype: float64
```

```
In [30]: round(df.EXT_SOURCE_2.describe(),6)

Out[30]: count    306851.000000
         mean          0.514393
         std           0.191060
         min           0.000000
         25%           0.392457
         50%           0.565961
         75%           0.663617
         max           0.855000
         Name: EXT_SOURCE_2, dtype: float64
```

- The columns with “AMT_REQ_CREDIT_BUREAU_” as prefix are having same no of NAN’s, and exactly the same rows as null as further revealed.

```
In [46]: df[df.AMT_REQ_CREDIT_BUREAU_QRT.isnull()].isna().sum().sort_values(ascending=False).head(20)

Out[46]: AMT_REQ_CREDIT_BUREAU_YEAR    41519
         AMT_REQ_CREDIT_BUREAU_QRT     41519
         AMT_REQ_CREDIT_BUREAU_MON     41519
         AMT_REQ_CREDIT_BUREAU_WEEK    41519
         AMT_REQ_CREDIT_BUREAU_DAY     41519
         AMT_REQ_CREDIT_BUREAU_HOUR    41519
         OBS_30_CNT_SOCIAL_CIRCLE        170
         OBS_60_CNT_SOCIAL_CIRCLE        170
         DEF_30_CNT_SOCIAL_CIRCLE        170
         DEF_60_CNT_SOCIAL_CIRCLE        170
```

- Creating additional categorical column "Credit_bureau_Missing" for the above columns if credit bureau data is missing, to flag the missing rows, and retaining the nan’s.
- For “NAME_TYPE_SUITE” column nan’s were replaced by “Unaccompanied” as it appears logical.
- “OBS_30_CNT_SOCIAL_CIRCLE, DEF_30_CNT_SOCIAL_CIRCLE” the nan’s were replaced with “0”.

```
In [53]: df.OBS_30_CNT_SOCIAL_CIRCLE.value_counts().head(10)

Out[53]: 0.0    163910
         1.0     48783
         2.0     29808
         3.0     20322
         4.0     14143
         5.0      9553
         6.0      6453
         7.0      4390
         8.0      2967
         9.0      2003
         Name: OBS_30_CNT_SOCIAL_CIRCLE, dtype: int64
```

```
In [54]: df.DEF_30_CNT_SOCIAL_CIRCLE.value_counts()

Out[54]: 0.0     271324
         1.0      28328
         2.0       5323
         3.0       1192
         4.0        253
         5.0         56
         6.0         11
         7.0          1
         34.0         1
         8.0          1
         Name: DEF_30_CNT_SOCIAL_CIRCLE, dtype: int64
```

Fill NAN with zero

- Replace the nan's in "AMT_GOODS_PRICE" column by values in "AMT_CREDIT" as both columns have same values.
- Drop all rows if the nan's are 12 or less.

Handling Imbalances:

- There is high imbalance in FLAG_DOCUMENTS columns if it is 0 for greater than 80% it means the document is not important drop such cols.

```
In [66]: lst_1=list(df.columns)
         x =lst_1.index("FLAG_DOCUMENT_2")
         y =lst_1.index("FLAG_DOCUMENT_21")
         print(x,y)

         45 64
```

```
In [67]: for i in range(x,y+1):
             if len(df[df[lst_1[i]]==1])<=.2*df.shape[0]:
                 df.drop([lst_1[i]],axis=1, inplace=True)
```

- Replace 365243 in days employed with "0".
- Converting Days to years/months where applicable. And rename the columns replacing days with years/months.

```
In [77]:
         for i in range (x,y+1):
             df.loc[:,lst_1[i]]=round(np.abs((df[lst_1[i]]/365)),2)
```

<u>Load the previous dataset aggregated on Current SK ID.</u>

- Merge it with the current dataset on SK_ID_CURR.
- Check for nan's: there are 16453 "nan" values in the columns added, which means there are 16453 customers whose data is not available in the previous dataset.

```
In [86]: df_1.isna().sum().sort_values(ascending=False).head(10)

Out[86]: AMT_REQ_CREDIT_BUREAU_QRT     41516
         AMT_REQ_CREDIT_BUREAU_YEAR    41516
         AMT_REQ_CREDIT_BUREAU_HOUR    41516
         AMT_REQ_CREDIT_BUREAU_DAY     41516
         AMT_REQ_CREDIT_BUREAU_WEEK    41516
         AMT_REQ_CREDIT_BUREAU_MON     41516
         Refused_Ratio                 16600
         Annuity_previous_agg          16453
         Flag_Approved                 16453
         Flag_Canceled                 16453
         dtype: int64
```

- Create a new column to flag the new customers as "new", and old customers as "repeater". There are 16453 new customers.
- Flag if annuity including previous pending is greater than income. There are 223 such customers small fraction which will not affect the analysis.

```
In [88]: df_1["Flag Annuity_Inc"].value_counts()

Out[88]: 0    307273
         1       223
         Name: Flag Annuity_Inc, dtype: int64
```

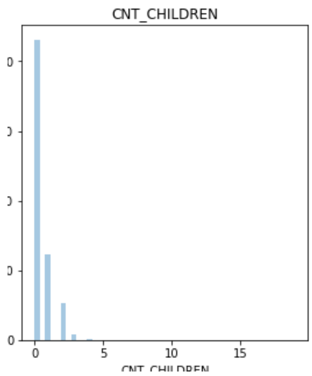
- There are 1540 applicants for whom the loan application has never been approved in previous process, again a small fraction.

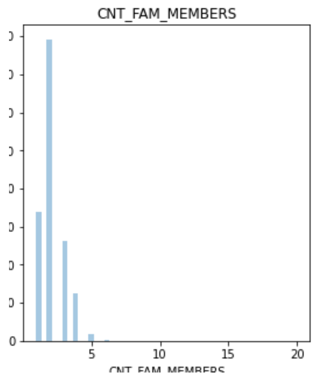
```
In [90]: df_1.Flag_Approved.value_counts().head(10)

Out[90]: 1.0    83008
         2.0    72219
         3.0    51985
         4.0    33374
         5.0    20617
         6.0    12246
         7.0     7005
         8.0     3876
         9.0     2224
         0.0     1540
         Name: Flag_Approved, dtype: int64
```

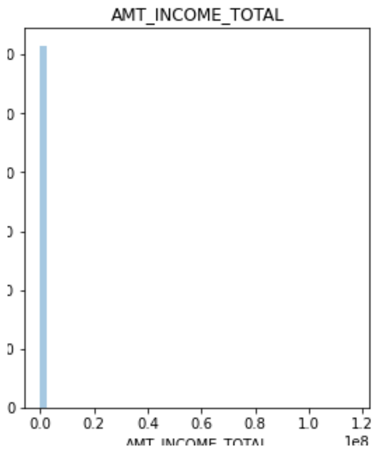
<u>CHECKING FOR IMBALANCE Using histograms and box plots:</u>

- Replace count children by 4 and count family member by 5, due to high imbalance.

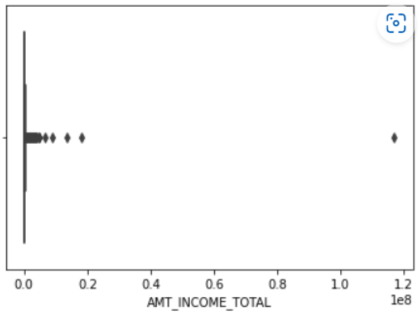




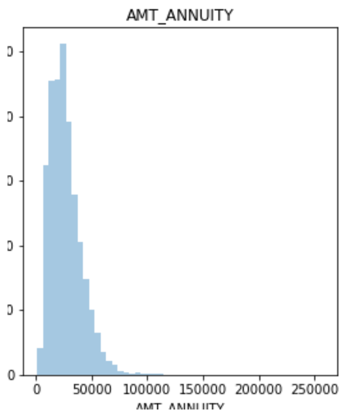
- Converting Highly imbalance data into Categorical Columns using quantiles. Namely:AMT_INCOME_TOTAL)



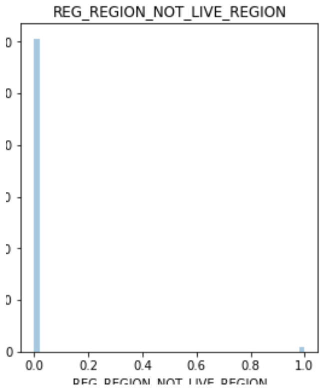

```
In [99]: sns.boxplot(df_1.AMT_INCOME_TOTAL)
         plt.show()
```



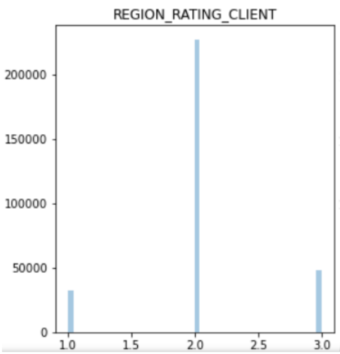
- Converting imbalanced data with similar shape in categorical columns. Namely ("AMT_ANNUITY", "AMT_CREDIT","AMT_GOODS_PRICE","REGION_POPULATION_RELATIVE")

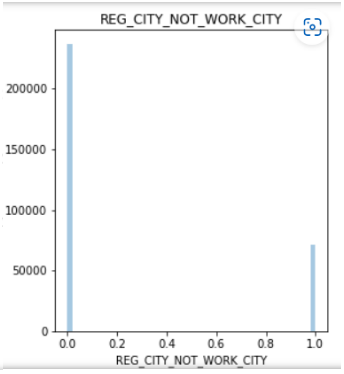


- Drop columns with “REGION_FLAG” prefix if 0 for 90% of rows.



- convert binary or three value numerical columns to categorical columns.

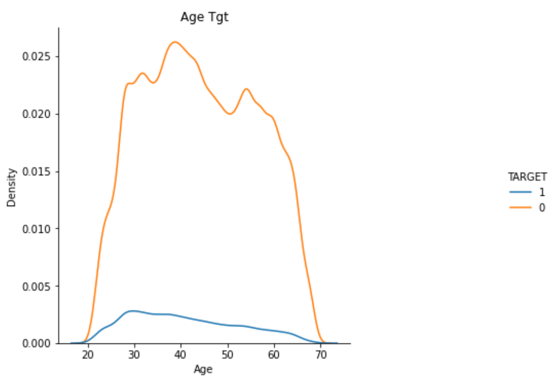




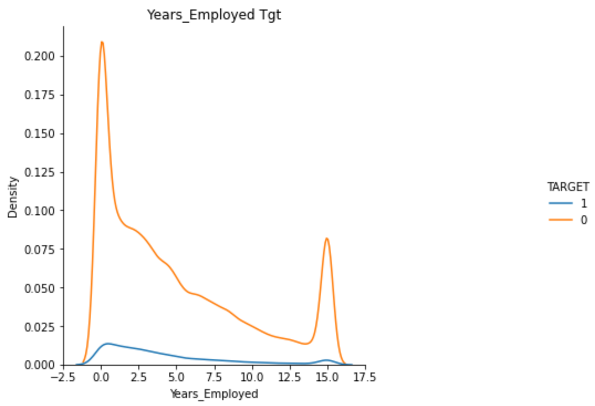
- Flag if phone changed in last three months. "Flag phone changed in 3 month" and drop phone_change_months column.

<u>Data Analysis:</u>

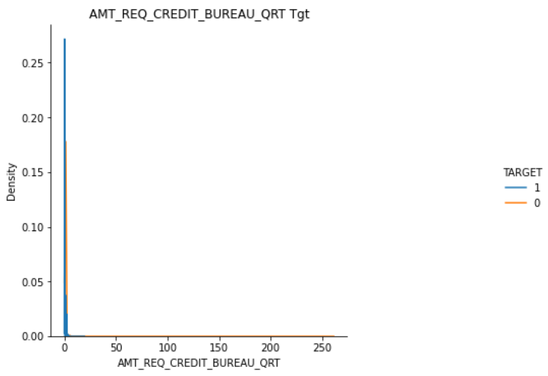
- Segmented Numerical: segmenting the data w.r.t. target and doing analysis.
  - People aged around 30-40, the default rate is higher. Whereas for the non-defaulters the density is almost same.



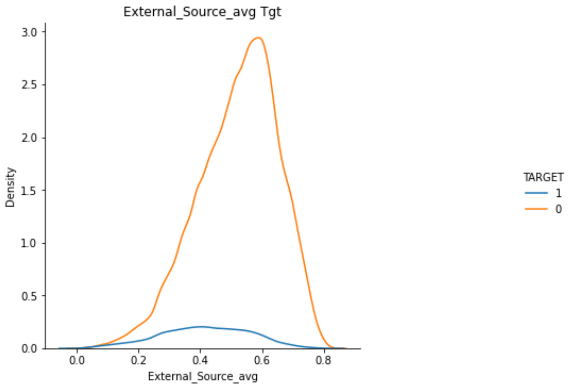
- Years employed: Default rate is highest if the years employed is zero and falls slowly up till 2.5 years, which most probably means unemployed (as nan was replaced by "0") people and people who recently changed their job are defaulting at a higher rate.



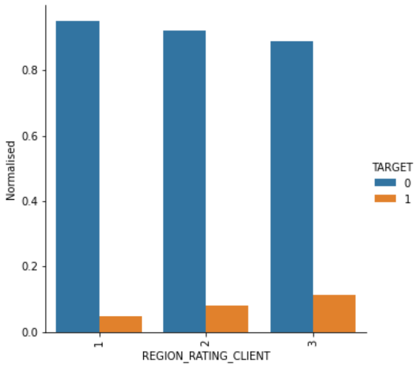
- AMT_REQ_CREDIT_BERAU_QUARTLY: the density function though highly unbalanced, but the trend is noticeably opposite to rest of AMT_REQ plots, i.e. the density of TARGET "0" is much higher for defaulters around 2.8 for AMT_REQ_CREDIT_BERAU_QUARTLY=0, as compared to non-defaulters. Which is opposite to the trend.

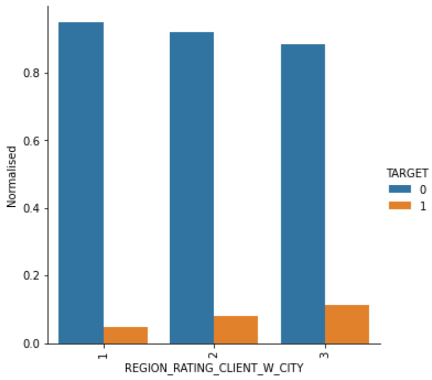


- EXT_SOURCE_AVERAGE: for defaulters the average peaks around 4, for non-defaulters the average is higher and peaks around 6,
  Conclusion: external source 2, and 3 are reliable factors as they rate non defaulters highly.

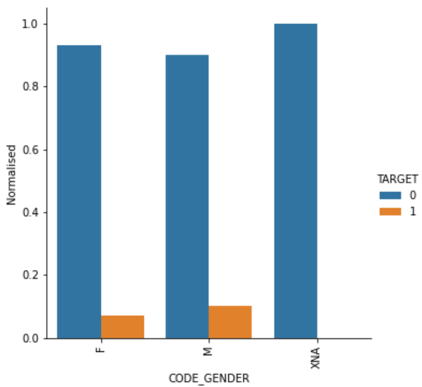


- Categorical Columns:
  - Region Rating Client: Clients living in region rated "3" tends to default at higher rate, and the clients living in "1" rated the default rate is low.
  - Similar trend is shown by Region Rating Client W City: Clients living in third tier city default at higher rate.

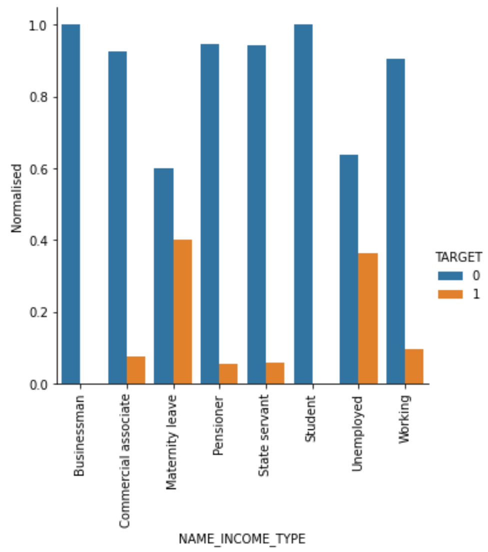




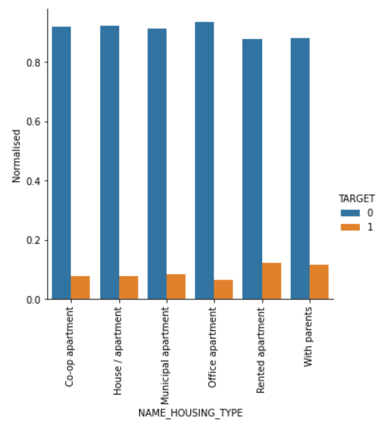
- Gender Code: men tend to default at higher rate compared to females, those who do not mention gender the default rate is zero.



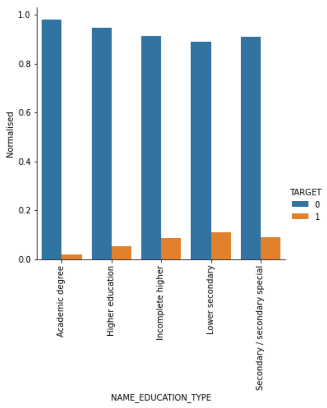
- Name Income type: For businessmen default rate is zero,
  For Students also the default rate is zero
  For clients who mention Income tipe as maternity leave default rate is 40%.
  Unimployed clients also default at around 35-40 percent



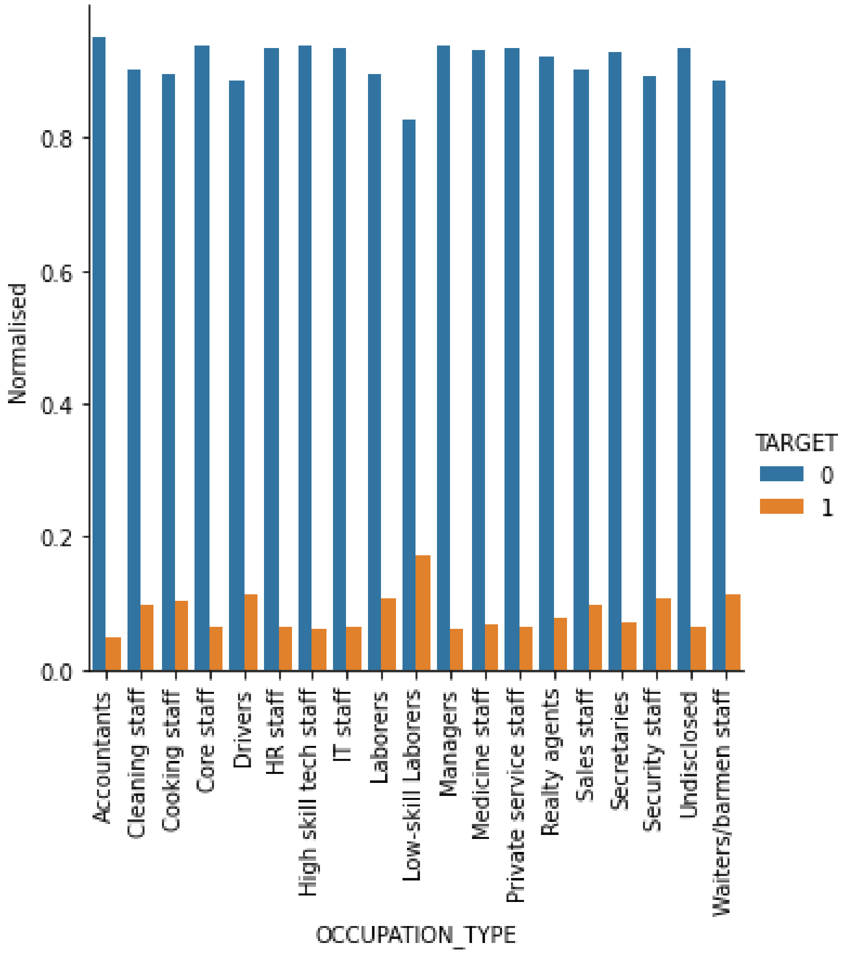
- Name housing type : Clients living with parents and in rented apartments tend to default at higher rate than average.



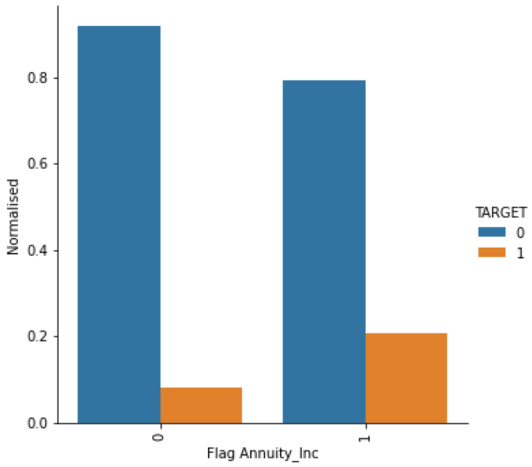
- Name education Type : Clients with Academic Degree the default rate is very low.
  Clients with Higher Education also default at a rate significantly less than average



- 20 percent of Low Skilled Workers have defaulted.
- Accountants has defaulted at around 5% lowest.

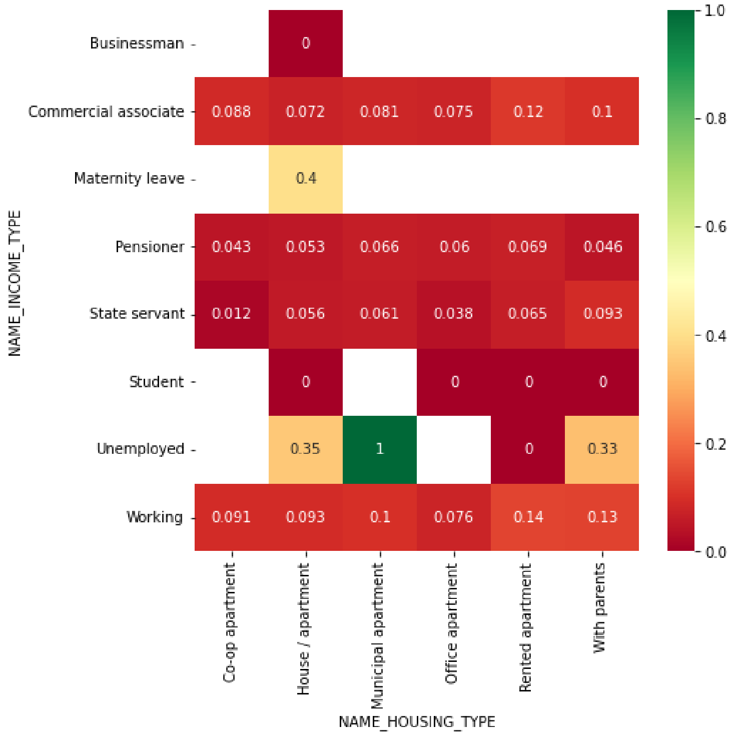


- For clients whose aggregate annuity (including previous annuity if previous loan is running) exceeds their income tend to default at a rate of 25%, although there are only 223 such customers.



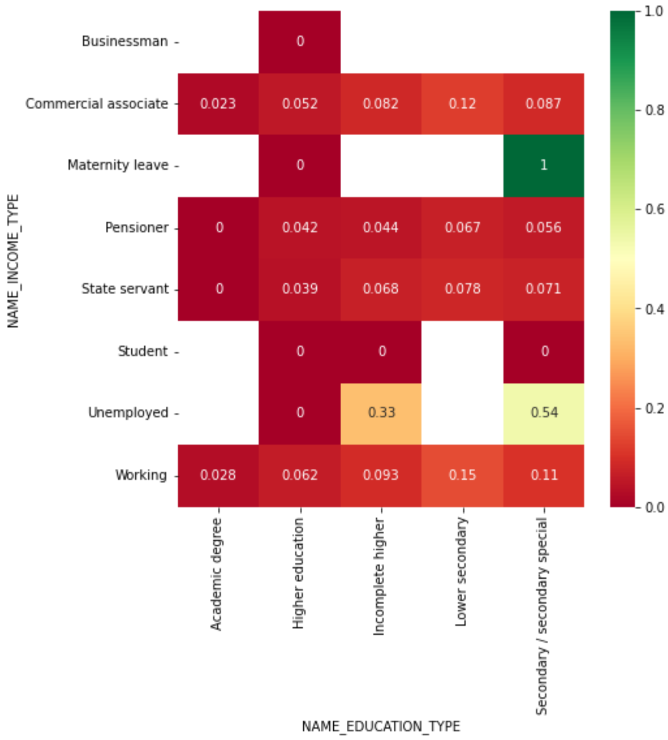
<u>Multivariate Analysis</u>: For the categories showing trends that are different from overall average a multivariate analysis was conducted.

- For unemployed customers living in municipal apartments the default rate is 100%.
- Maternity leave clients with House/Apartment default at a rate of 40%

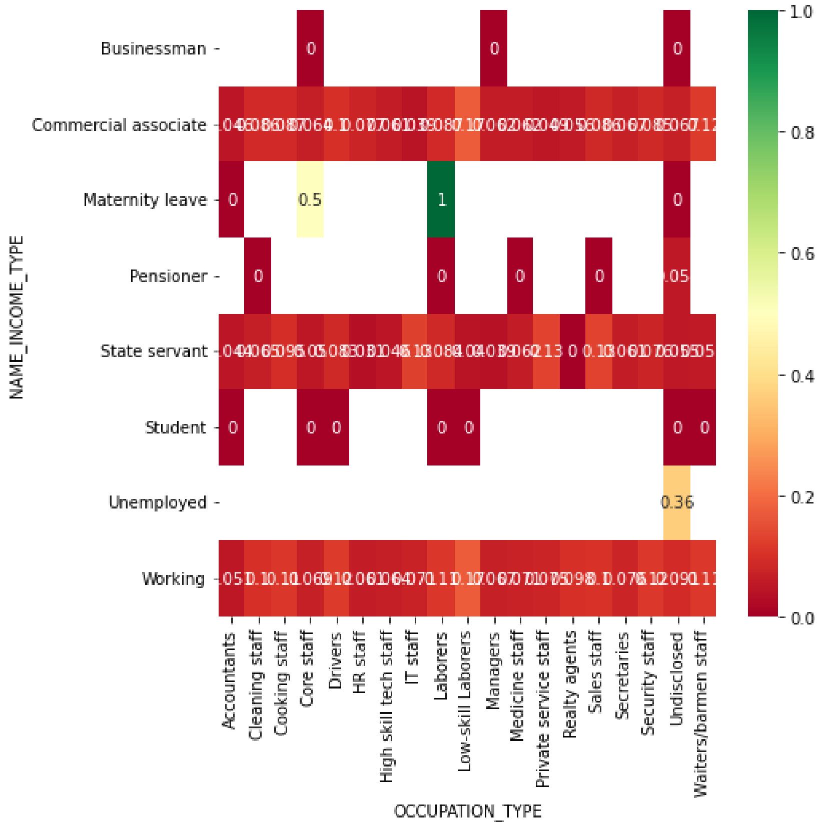


- Clients with maternity leave living in 3 tier city/region default rate is 10%.
- for unemployed default rate is higher than 33%
- clients with maternity leave living in 2nd tier city 33%,

- Clients with maternity leave and Secondary education default at a rate of 100%



- All of the Labourers with maternity leave have defaulted
- Core staff with maternity leave defaulted at 50% rate.
- All Unemployed applicants has occupation undisclosed and default rate is 36%



Top Correlation

Positive correlation: High correlation between similar quantities like (OBS_30_CNT_SOCIAL_CIRCLE and OBS_60_CNT_SOCIAL_CIRCLE) is expected and is same for flag =0 and 1.

```
In [134]: print(top_corr(df_1[df_1.TARGET==0],20))

OBS_30_CNT_SOCIAL_CIRCLE     OBS_60_CNT_SOCIAL_CIRCLE     0.998510
CNT_CHILDREN                 CNT_FAM_MEMBERS              0.873931
DEF_30_CNT_SOCIAL_CIRCLE     DEF_60_CNT_SOCIAL_CIRCLE     0.859370
Flag_Refused                 Refused_Ratio                0.745370
AMT_REQ_CREDIT_BUREAU_YEAR   Flag_Canceled                0.554039
                             Flag_Approved                0.429358
Flag_Canceled                Flag_Refused                 0.346328
DEF_30_CNT_SOCIAL_CIRCLE     OBS_60_CNT_SOCIAL_CIRCLE     0.331723
OBS_30_CNT_SOCIAL_CIRCLE     DEF_30_CNT_SOCIAL_CIRCLE     0.329596
Age                          Years_Registration           0.316848
AMT_REQ_CREDIT_BUREAU_YEAR   Flag_Refused                 0.308456
Annuity_previous_agg         Flag_Approved                0.306665
Flag_Approved                Flag_Canceled                0.284309
                             Flag_Refused                 0.279066
Age                          Years_ID_Publish             0.272049
AMT_REQ_CREDIT_BUREAU_YEAR   Annuity_previous_agg         0.263338
OBS_60_CNT_SOCIAL_CIRCLE     DEF_60_CNT_SOCIAL_CIRCLE     0.255337
OBS_30_CNT_SOCIAL_CIRCLE     DEF_60_CNT_SOCIAL_CIRCLE     0.253369
AMT_REQ_CREDIT_BUREAU_HOUR   AMT_REQ_CREDIT_BUREAU_DAY    0.229065
AMT_REQ_CREDIT_BUREAU_DAY    AMT_REQ_CREDIT_BUREAU_WEEK   0.220120
dtype: float64
```

```
In [135]: print(top_corr(df_1[df_1.TARGET==1],20))

OBS_30_CNT_SOCIAL_CIRCLE     OBS_60_CNT_SOCIAL_CIRCLE     0.998270
CNT_CHILDREN                 CNT_FAM_MEMBERS              0.878688
DEF_30_CNT_SOCIAL_CIRCLE     DEF_60_CNT_SOCIAL_CIRCLE     0.869016
Flag_Refused                 Refused_Ratio                0.720303
AMT_REQ_CREDIT_BUREAU_YEAR   Flag_Canceled                0.519337
                             Flag_Approved                0.405737
Flag_Canceled                Flag_Refused                 0.352438
Annuity_previous_agg         Flag_Approved                0.352323
DEF_30_CNT_SOCIAL_CIRCLE     OBS_60_CNT_SOCIAL_CIRCLE     0.337389
OBS_30_CNT_SOCIAL_CIRCLE     DEF_30_CNT_SOCIAL_CIRCLE     0.334035
AMT_REQ_CREDIT_BUREAU_YEAR   Flag_Refused                 0.317062
Flag_Approved                Flag_Canceled                0.296135
                             Flag_Refused                 0.282273
Age                          Years_Registration           0.273642
OBS_60_CNT_SOCIAL_CIRCLE     DEF_60_CNT_SOCIAL_CIRCLE     0.264357
OBS_30_CNT_SOCIAL_CIRCLE     DEF_60_CNT_SOCIAL_CIRCLE     0.261209
Age                          Years_ID_Publish             0.253248
AMT_REQ_CREDIT_BUREAU_HOUR   AMT_REQ_CREDIT_BUREAU_DAY    0.246741
AMT_REQ_CREDIT_BUREAU_YEAR   Annuity_previous_agg         0.236412
AMT_REQ_CREDIT_BUREAU_QRT    Flag_Canceled                0.225143
dtype: float64
```

- Top meaningful +ve correlated Quantities
  1. Amount req credit bureau year and Flag _Cancelled coefficient of correlation is .55 for non defaulters and .52 for defaulters.
     Which means grater the no of enquiries to credit berau higher the chance of default.
  2. For Non Defaulters the correlation between Age and Year ID Published is .27,
     Whereas for defaulters the correlation between Age and Years ID Published is .25
  3. For defaulters the correlation between Age and Years registration is .27.

Negative correlation:

In [137]:
```
print(top_corr(df_1[df_1.TARGET==0],20))
```

```
CNT_CHILDREN               Age                          -0.339841
Age                        CNT_FAM_MEMBERS              -0.288369
CNT_CHILDREN               Years_Registration           -0.184747
Years_Registration         CNT_FAM_MEMBERS              -0.173397
External_Source_avg        Refused_Ratio                -0.128870
                           Flag_Refused                 -0.125556
Age                        HOUR_APPR_PROCESS_START      -0.095905
                           Years_Employed               -0.085447
AMT_REQ_CREDIT_BUREAU_YEAR External_Source_avg          -0.060099
CNT_CHILDREN               Annuity_previous_agg         -0.059634
Years_Employed             Years_ID_Publish             -0.052642
CNT_CHILDREN               Flag_Canceled                -0.045041
                           AMT_REQ_CREDIT_BUREAU_YEAR   -0.042972
External_Source_avg        Flag_Canceled                -0.042076
DEF_60_CNT_SOCIAL_CIRCLE   External_Source_avg          -0.039438
DEF_30_CNT_SOCIAL_CIRCLE   External_Source_avg          -0.038606
Years_Registration         Refused_Ratio                -0.035647
CNT_CHILDREN               External_Source_avg          -0.035424
Years_ID_Publish           HOUR_APPR_PROCESS_START      -0.034145
Age                        Refused_Ratio                -0.031091
dtype: float64
```

In [138]:
```
print(top_corr(df_1[df_1.TARGET==1],20))
```

```
CNT_CHILDREN               Age                          -0.262780
Age                        CNT_FAM_MEMBERS              -0.206495
CNT_CHILDREN               Years_Registration           -0.148025
Years_Registration         CNT_FAM_MEMBERS              -0.144078
External_Source_avg        Refused_Ratio                -0.136089
                           Flag_Refused                 -0.127321
Age                        HOUR_APPR_PROCESS_START      -0.062173
CNT_CHILDREN               Annuity_previous_agg         -0.052935
Years_Registration         Refused_Ratio                -0.044897
External_Source_avg        Flag_Canceled                -0.042705
CNT_CHILDREN               Flag_Canceled                -0.041888
                           AMT_REQ_CREDIT_BUREAU_YEAR   -0.034684
Annuity_previous_agg       Refused_Ratio                -0.034120
AMT_REQ_CREDIT_BUREAU_QRT  External_Source_avg          -0.033782
HOUR_APPR_PROCESS_START    Annuity_previous_agg         -0.033704
AMT_REQ_CREDIT_BUREAU_YEAR External_Source_avg          -0.033169
HOUR_APPR_PROCESS_START    AMT_REQ_CREDIT_BUREAU_YEAR   -0.032692
CNT_FAM_MEMBERS            HOUR_APPR_PROCESS_START      -0.029558
Years_ID_Publish           Refused_Ratio                -0.028399
CNT_CHILDREN               HOUR_APPR_PROCESS_START      -0.025680
dtype: float64
```

- Meaningful -ve correlated quantities.
    1. No of children and age is having a correlation of -.33 for non defaulters and -.26 for defaulters.
    2. Similar trend is shown by age and family members.