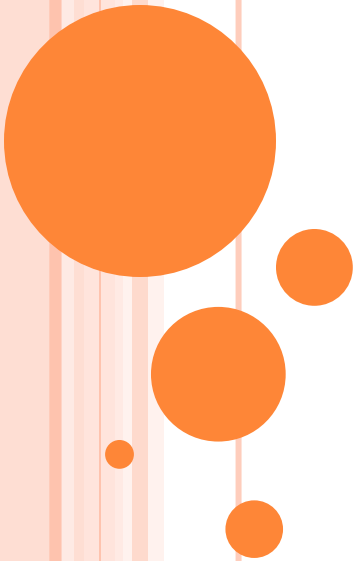
EDA

CASE STUDY



Selecting Data

- As provided, we have worked on application_data.csv and previous_application.csv for our whole case study

- Firstly we worked on application_data.csv ,
- Then previous_application.csv
- Finally we merged the two dataframes on their common column

```
In [3]: data.head()
```

Out[3]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CA
0	100002	1	Cash loans	M	
1	100003	0	Cash loans	F	
2	100004	0	Revolving loans	M	
3	100006	0	Cash loans	F	
4	100007	0	Cash loans	M	

5 rows × 122 columns

```
In [4]: data.describe()
```

Out[4]:

	SK_ID_CURR	TARGET	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CRE
count	307511.000000	307511.000000	307511.000000	3.075110e+05	3.075110e
mean	278180.518577	0.080729	0.417052	1.687979e+05	5.990260e
std	102790.175348	0.272419	0.722121	2.371231e+05	4.024908e
min	100002.000000	0.000000	0.000000	2.565000e+04	4.500000e
25%	189145.500000	0.000000	0.000000	1.125000e+05	2.700000e
50%	278202.000000	0.000000	0.000000	1.471500e+05	5.135310e
75%	367142.500000	0.000000	1.000000	2.025000e+05	8.086500e
max	456255.000000	1.000000	19.000000	1.170000e+08	4.050000e

- Initially we checked what file have and description of file.

- Shape of application_data is (307511,122)

- Now we have checked for null values in dataframe

- We dropped columns which have 50% and more null values

```
In [12]: data = data[data.columns[data.isnull().sum() < mid]]
         data.shape

Out[12]: (307511, 81)
```

- After dropping number of column reduced to 81 from 122

Categorical column VS Numeric column

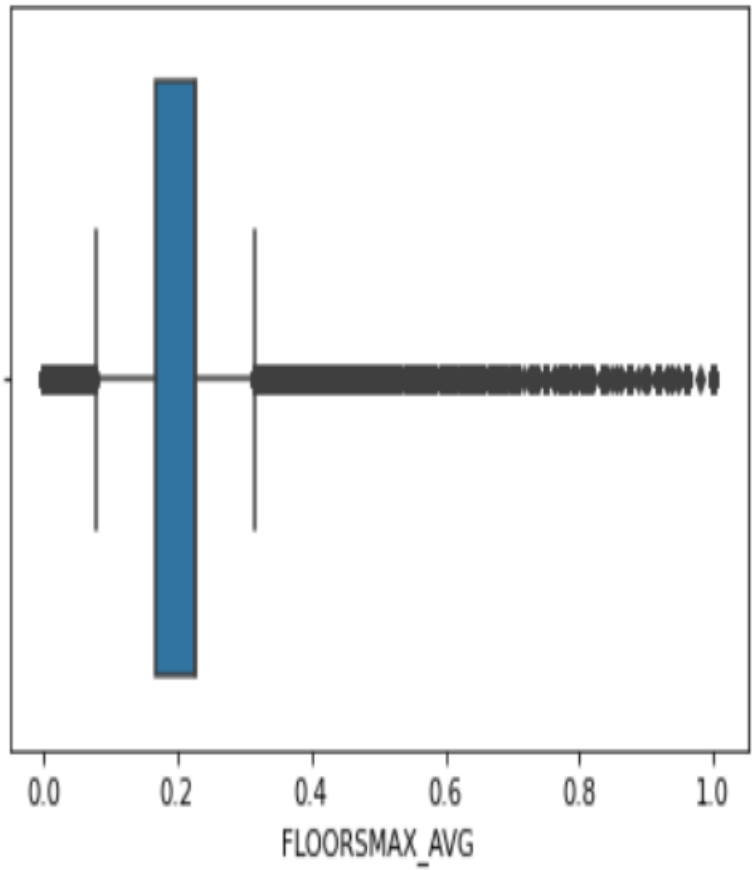
- This will list all the categorical column in dataframe

```
In [15]: # Categorical columns

         list(set(data.columns) - set(data.describe().columns))
```

- This will list all the numeric column in dataframe

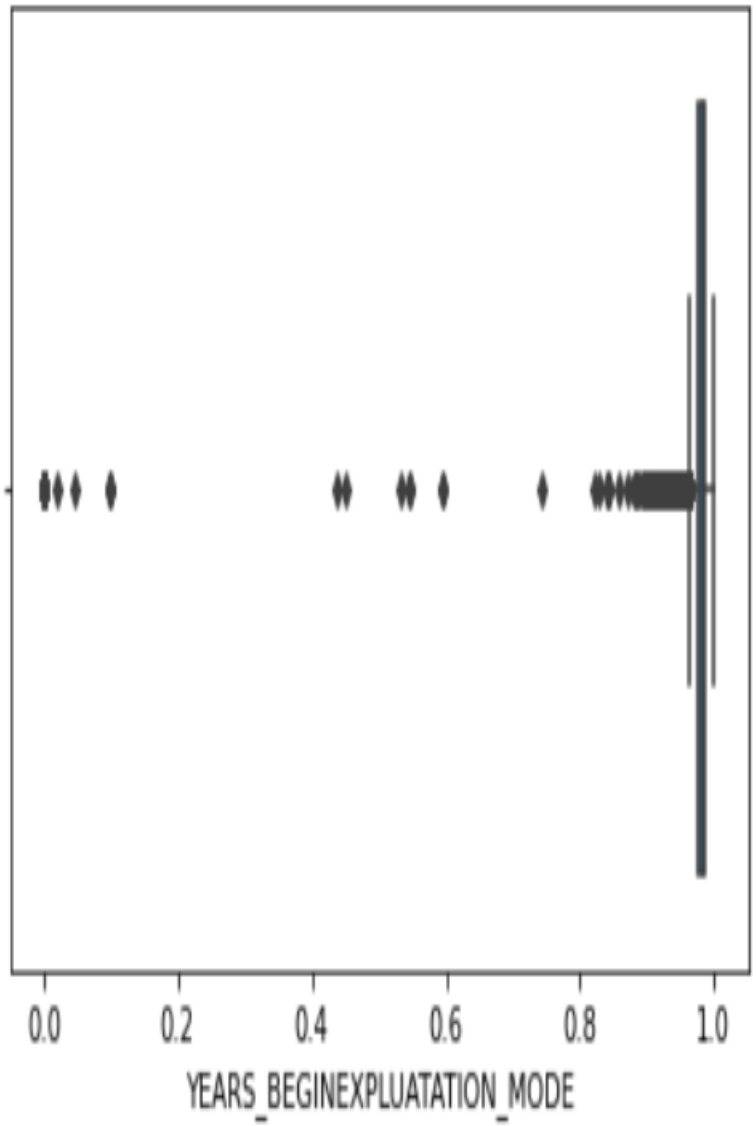
```
In [16]: #Numeric columns

         data.describe().columns
```

Fill the Missing values



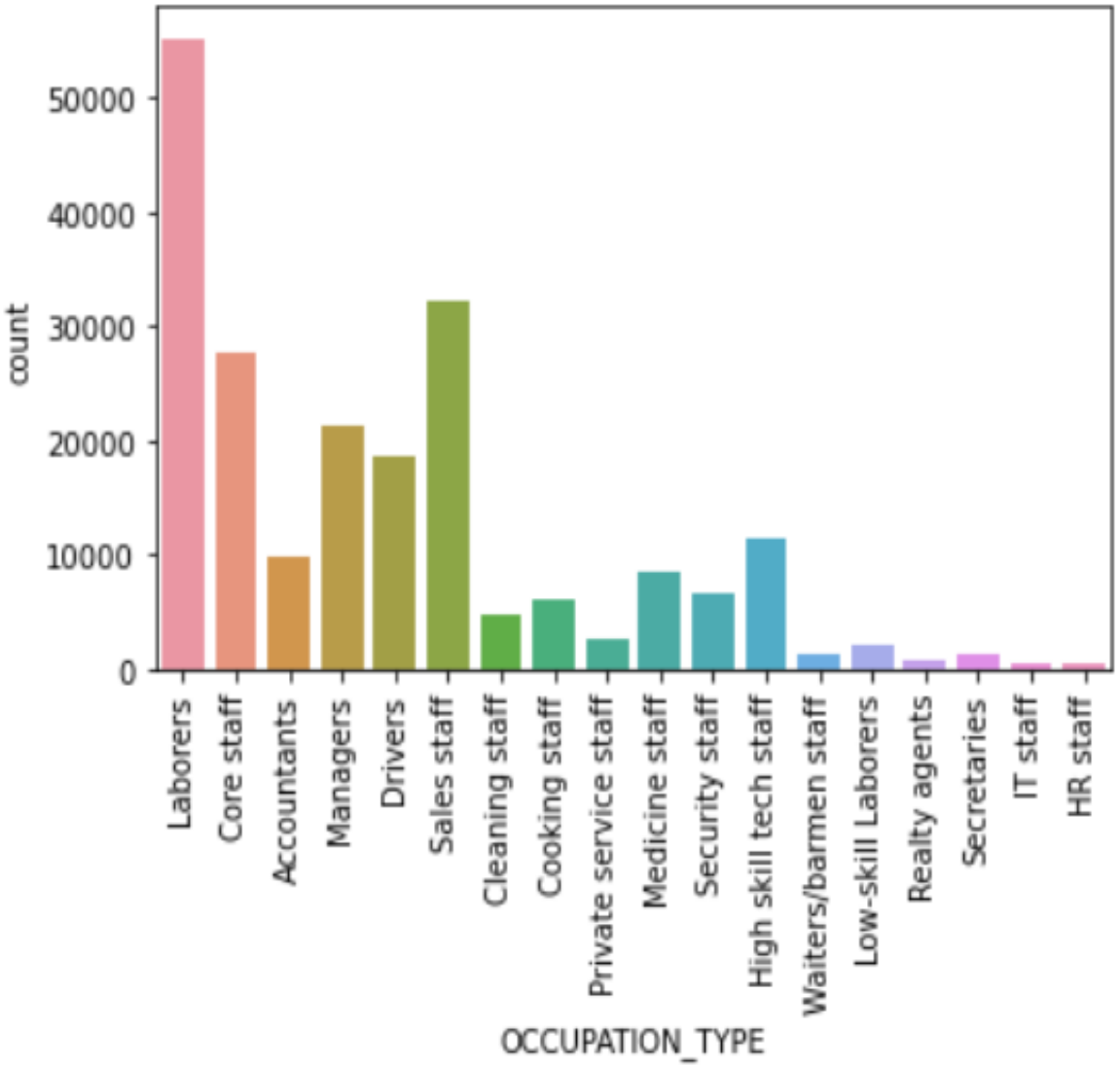
For FLOORMAX_AVG we will fillna with mean of FLOORMAX_AVG

Similarly for FLOORMAX_MEDI and FLOORMAX_MODE we filled missing with mean of respective column



For YEARS_BEGINEXPLUATATION_MODE we will fillna with median of FLOORMAX_AVG Because maximum value belongs to their

Similarly for YEARS_BEGINEXPLUATATION_AVG YEARS_BEGINEXPLUATATION_MEDI with median of their respective column



Customers having Occupation Type as 'Labourers' are the highest sector ging for loans

Handelling column with unique values having null values

- In this scenerio we will fillna() with mode of respective column and iloc[0]

- We have used this in :

  AMT_REQ_CREDIT_BUREAU_YEAR
  AMT_REQ_CREDIT_BUREAU_QRT
  AMT_REQ_CREDIT_BUREAU_MON
  AMT_REQ_CREDIT_BUREAU_WEEK
  AMT_REQ_CREDIT_BUREAU_DAY
  AMT_REQ_CREDIT_BUREAU_HOUR

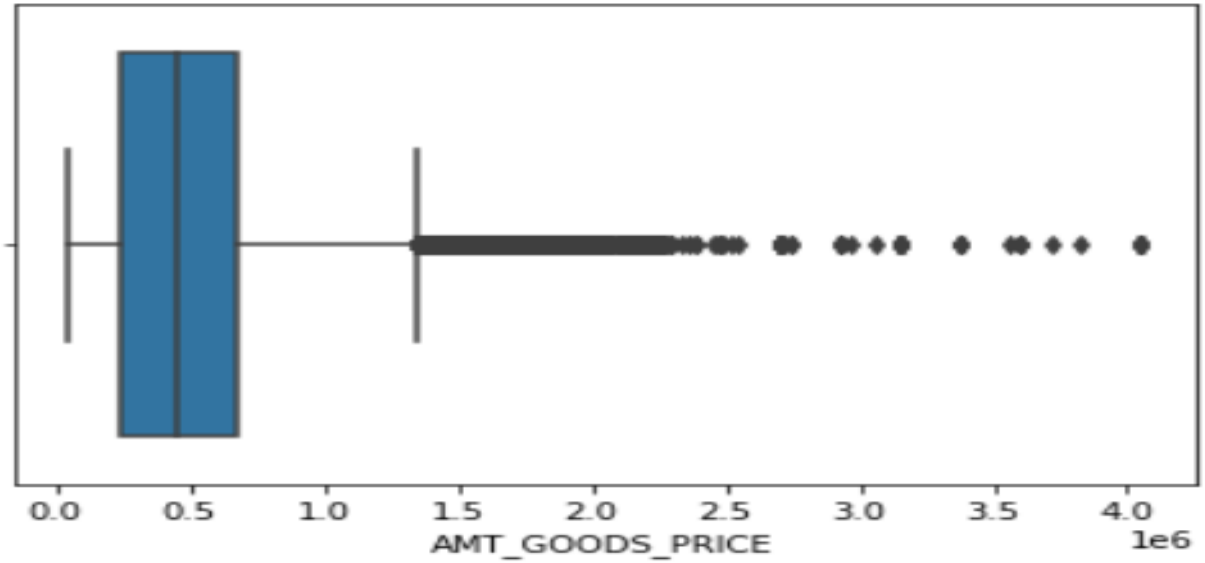
Handelling column with object type values having null values

- For such value we replaced all the missing value with most frequent value

- We done this in columns like:
  - NAME_TYPE_SUITE
  - EMERGENCYSTATE_MODE
  - OCCUPATION_TYPE

  Using fillna(, inplace=True)

Identify outlier using plots



- In this value after 400000 is clearly an oulier
- This is why we used mode to fill null values

- This is how we handeled all missing values using different methods to get clean and good dataframe
- So, that we can carry our analysis in smooth way
- In this way we will have more accurate observations

Observation on column starting from date

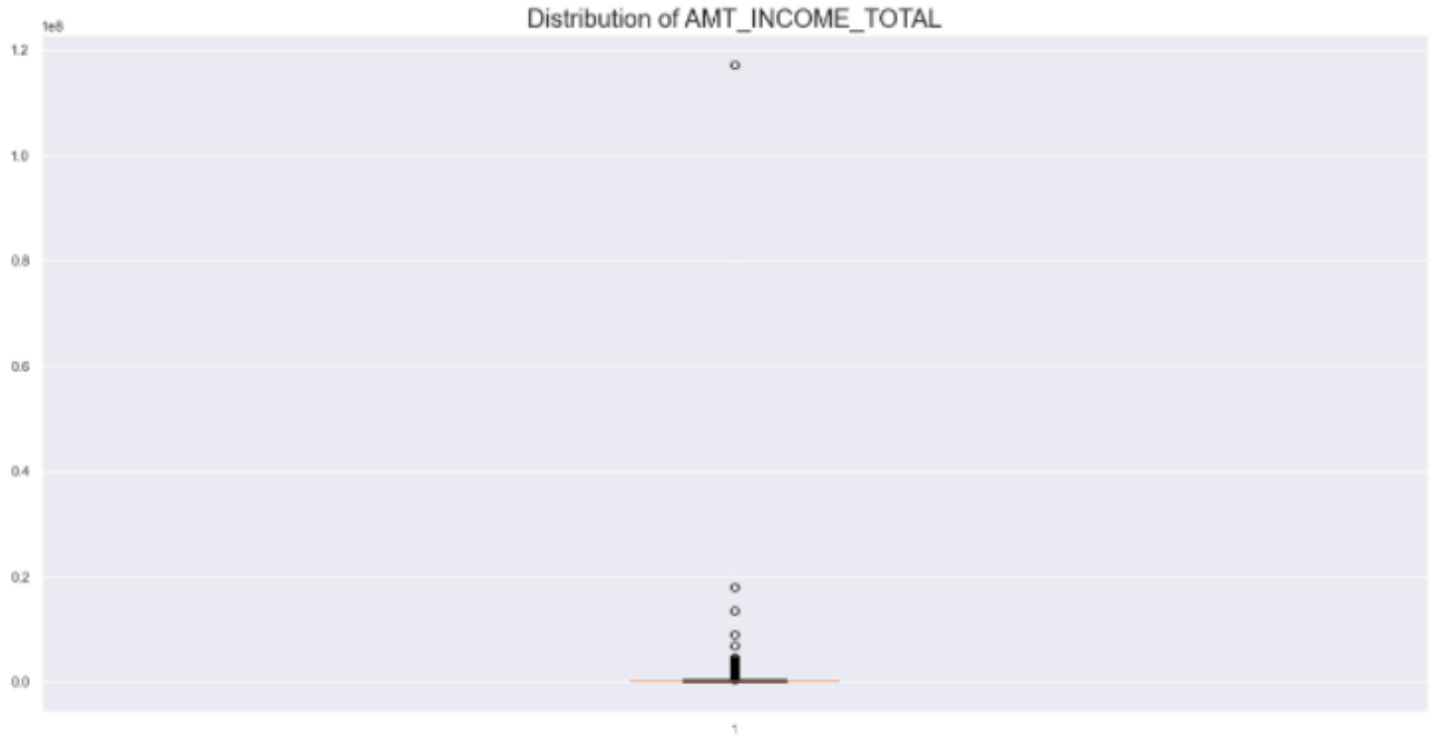
- With help of unique we know that there are some specific value for columns

In [174]:
```
# Checking the values present in columns starting with 'DAYS'
print(data['DAYS_BIRTH'].unique())
print(data['DAYS_EMPLOYED'].unique())
print(data['DAYS_REGISTRATION'].unique())
print(data['DAYS_ID_PUBLISH'].unique())
print(data['DAYS_LAST_PHONE_CHANGE'].unique())
```

```
[ -9461 -16765 -19046 ...  -7951  -7857 -25061]
[  -637  -1188   -225 ... -12971 -11084  -8694]
[ -3648.  -1186.  -4260. ... -16396. -14558. -14798.]
[-2120  -291 -2531 ... -6194 -5854 -6211]
[-1134.  -828.  -815. ... -3988. -3899. -3538.]
```

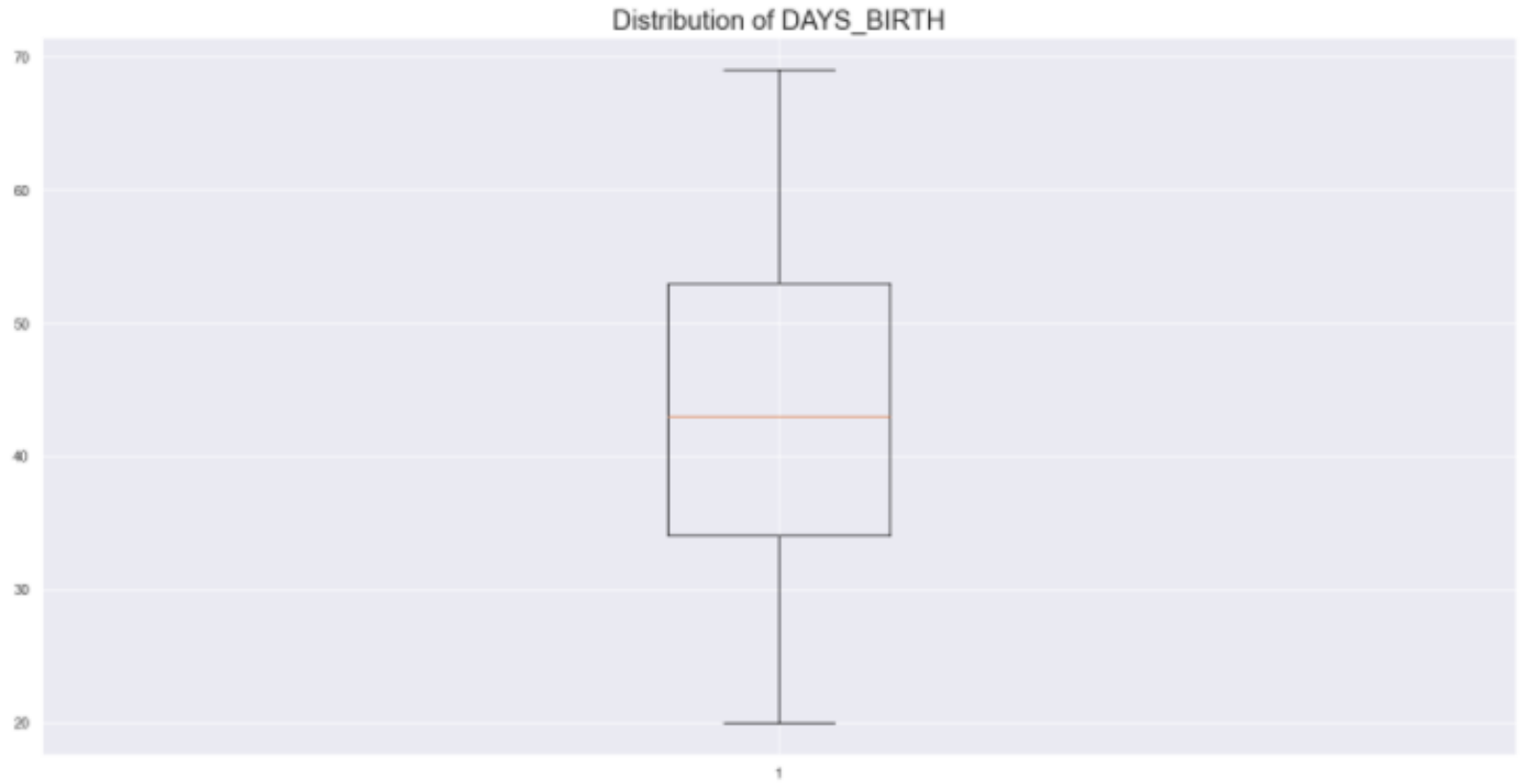
Analysing AMT_INCOME_TOTAL

- From graph we can surely say that 117 million is surely an outlier



Analysis w.r.t to customer age

- No outlier and median in range 40-45

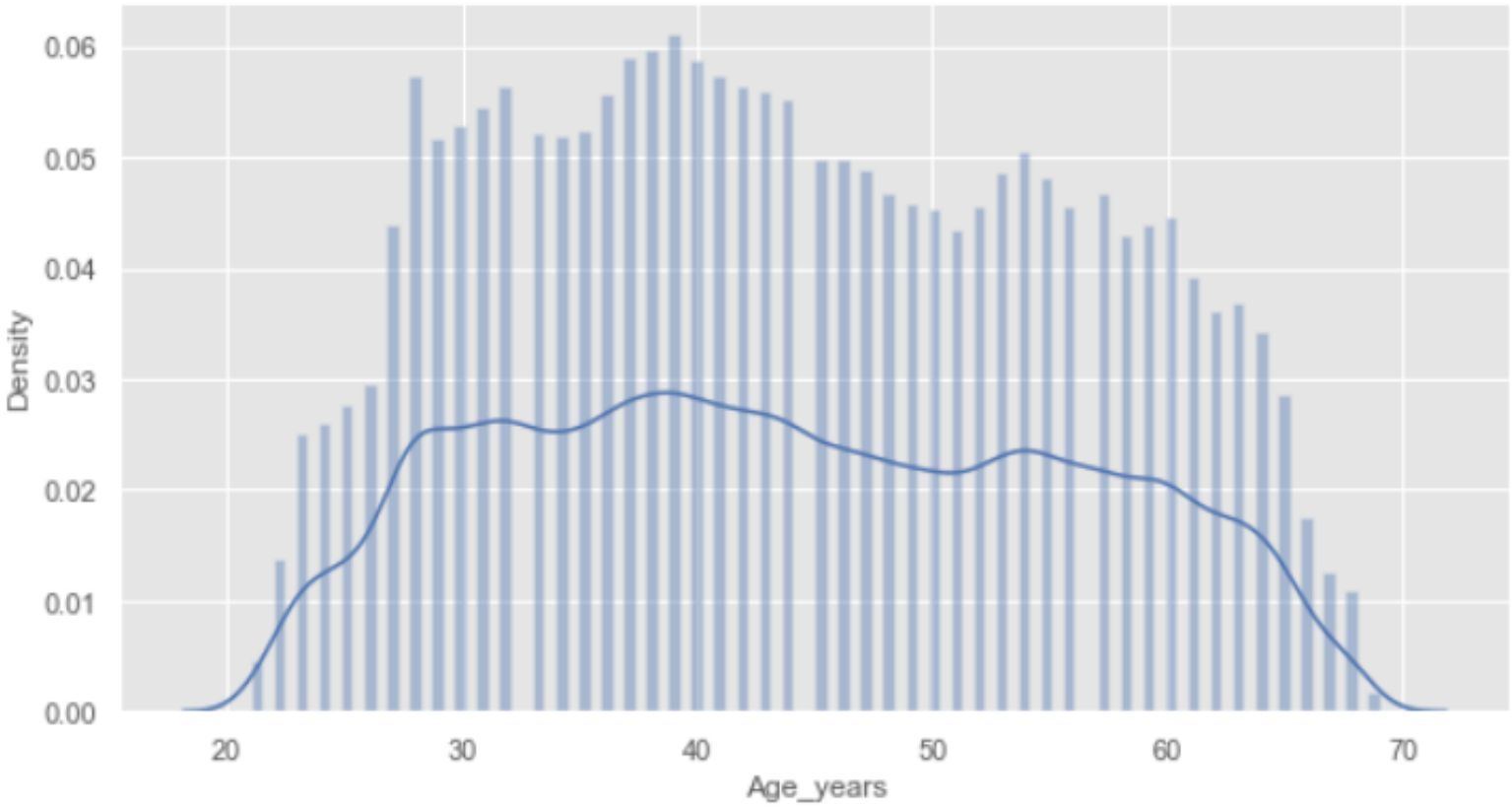


Age group on top which apply for loan

- For this we added 3 more column to dataset
  - TodayDate:
    - Date of current day
  - DateOfBirth:
    - Date of birth calculated from given data
  - Age_years
    - TodayDate – DateOfBirth

This added another angel to observation

Age Wise Analysis

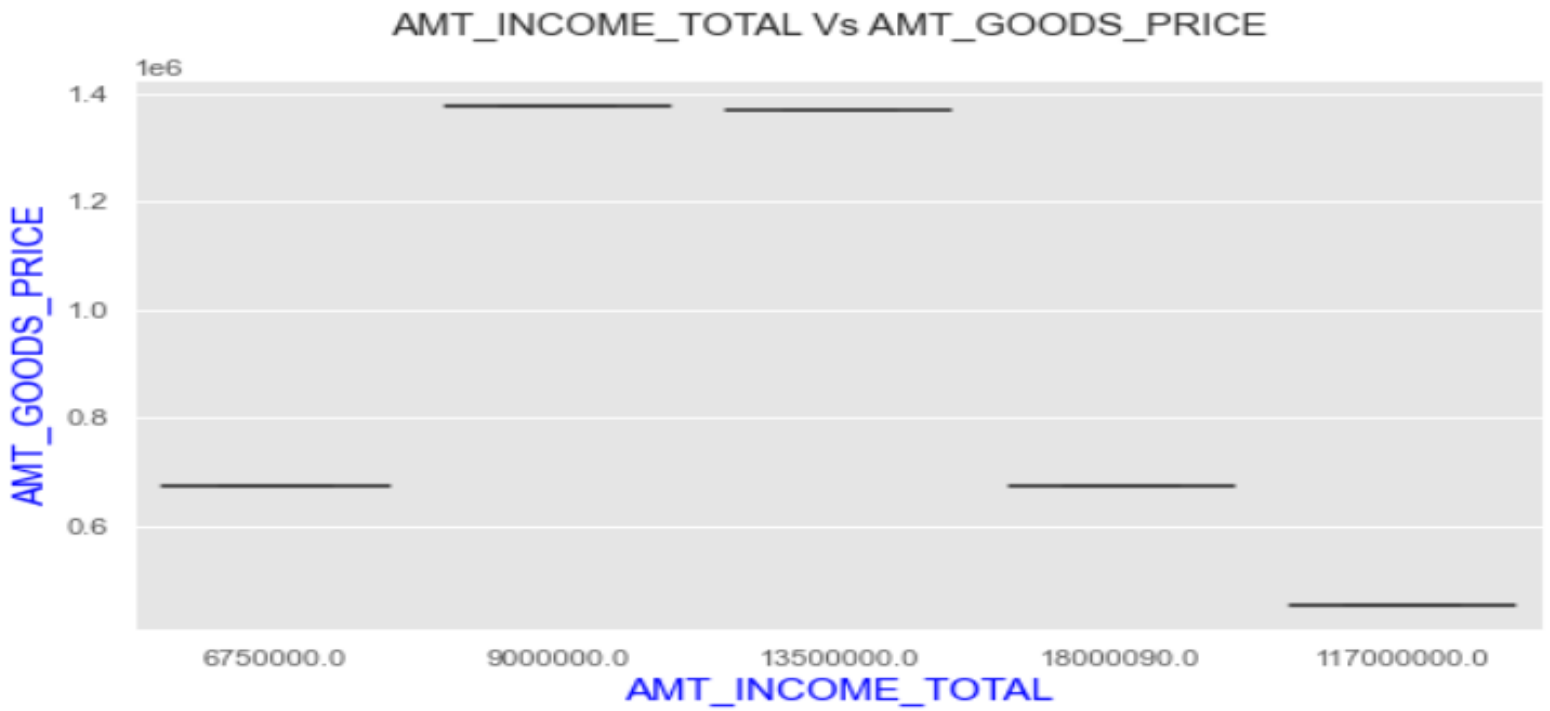


- Customer having 39 years of Age are highest in number (more than 6% of total) who has applied for loan data

Analysis on income of client

- Income in circle is clearly outlier, lets try to find fraud from income

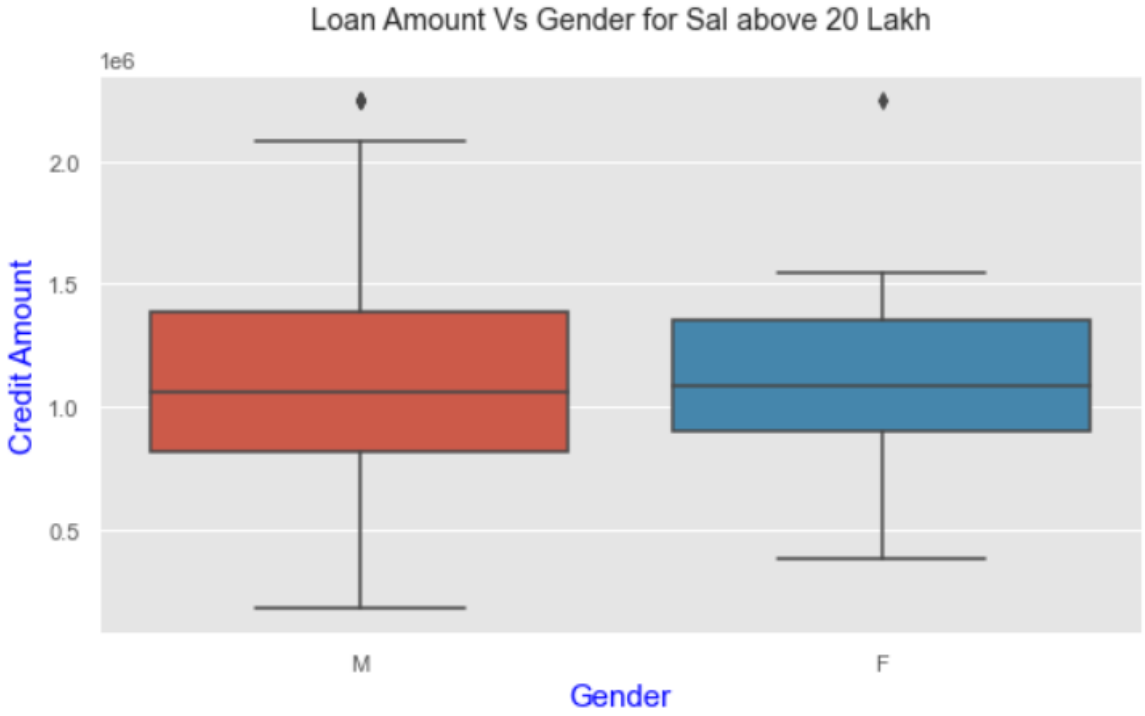
- To get more insight let's loan demand for salary more than 60lakh



- These are some strange cases where customer salary > 60 Lakhs, still they are going for some small loans, which can be checked

Gender VS salary > 20 lakh

- Median of both gender is similar but maximum value for both gender have huge differnce.



TARGET

- Target ==1:
  - Client with payment difficulty
  - Count = 24825
- Target == 0:
  - Other client
  - Count= 282686

There is huge imbalance in target variable.

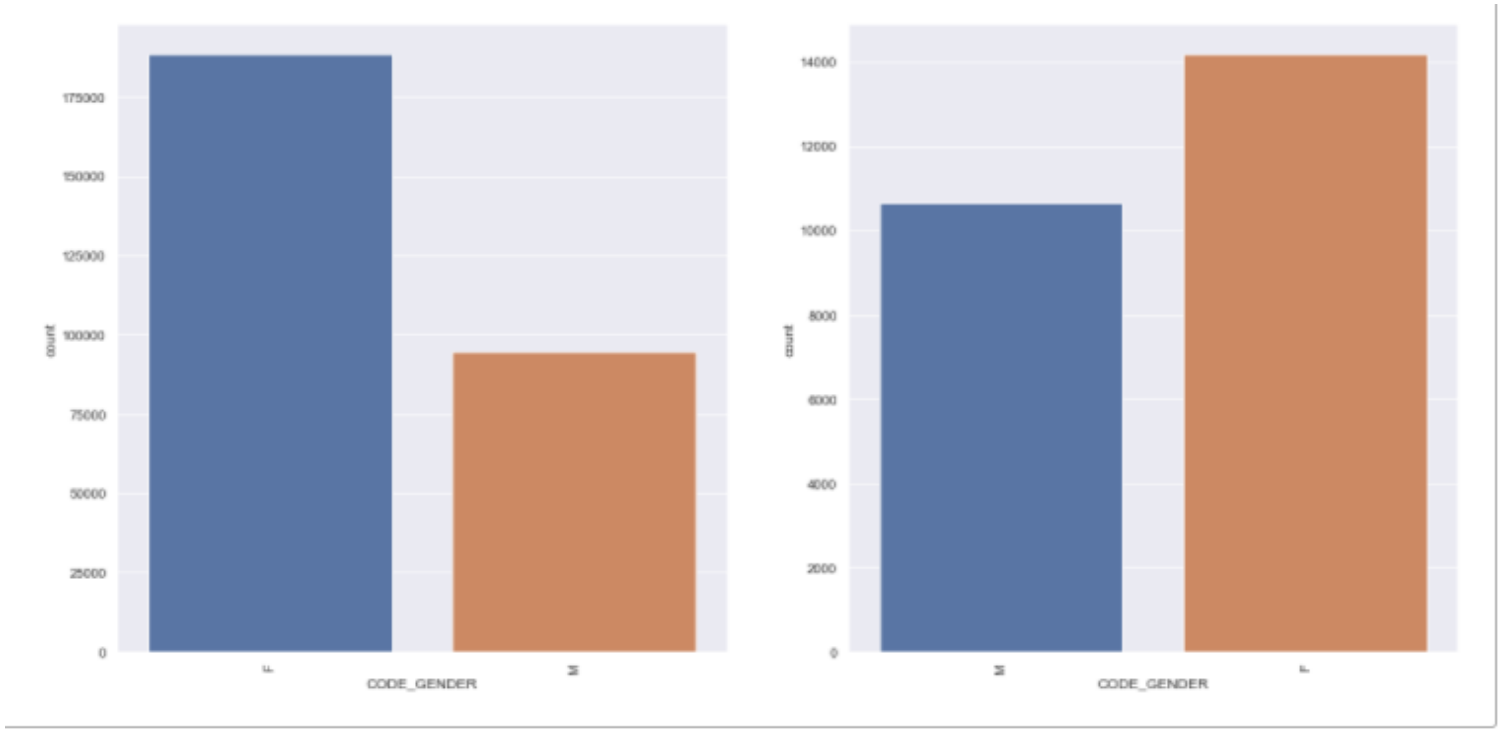
Handling imbalance target variable

- After normalizing count:
  - 0 0.919271
  - 1 0.080729
  - We can observe that maximum loan holders don't have problem in repaying it.
- To resolve this imbalance we will divide target into 2 parts

- target0 = data.loc[data.TARGET == 0]

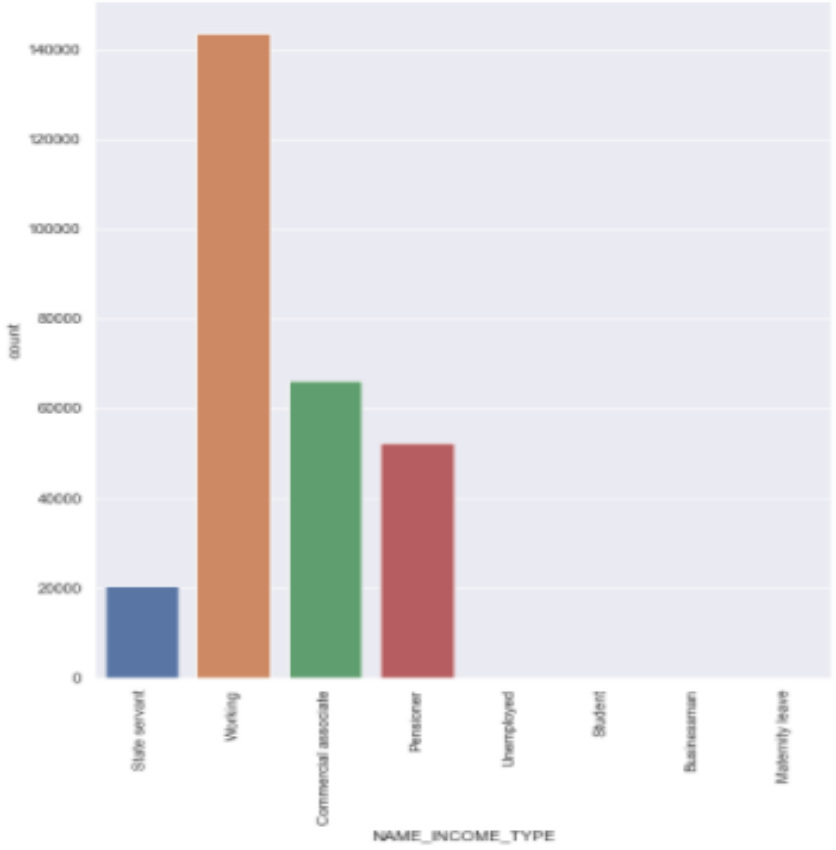
- target1 = data.loc[data.TARGET == 1]

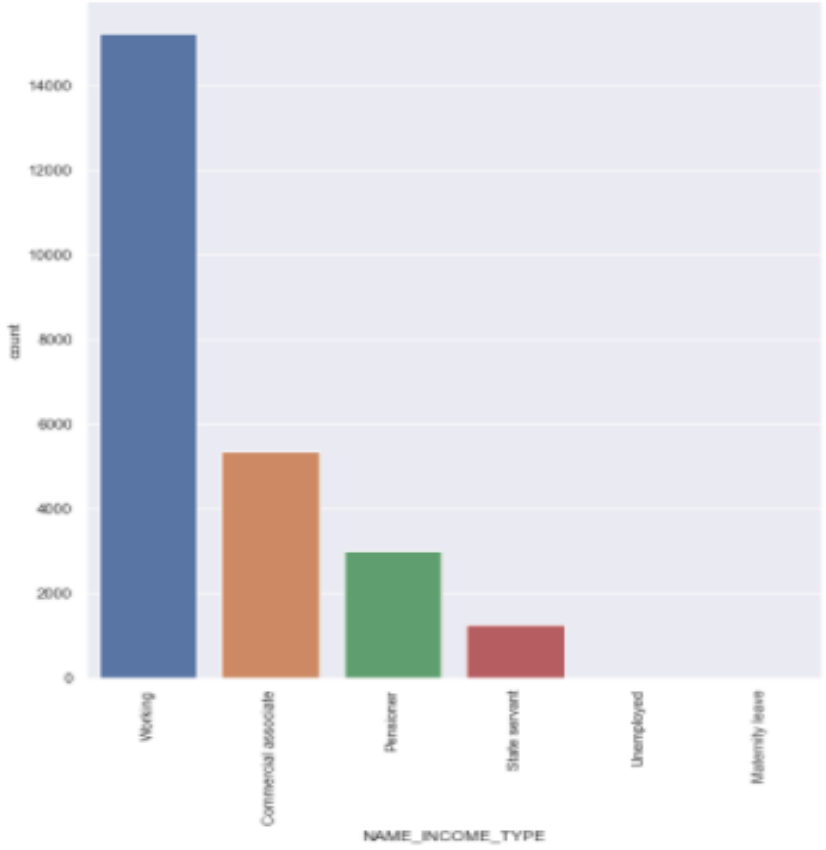
CODE_GENDER VS TARGET0/TARGET1



Comaparing the Payment Difficulties and Non Payment Difficulties on the basis of Gender, we observe that Females are the majority in both the cases although there is an increase in the percentage in Male Payment Difficulties from Non-Payment Difficulties

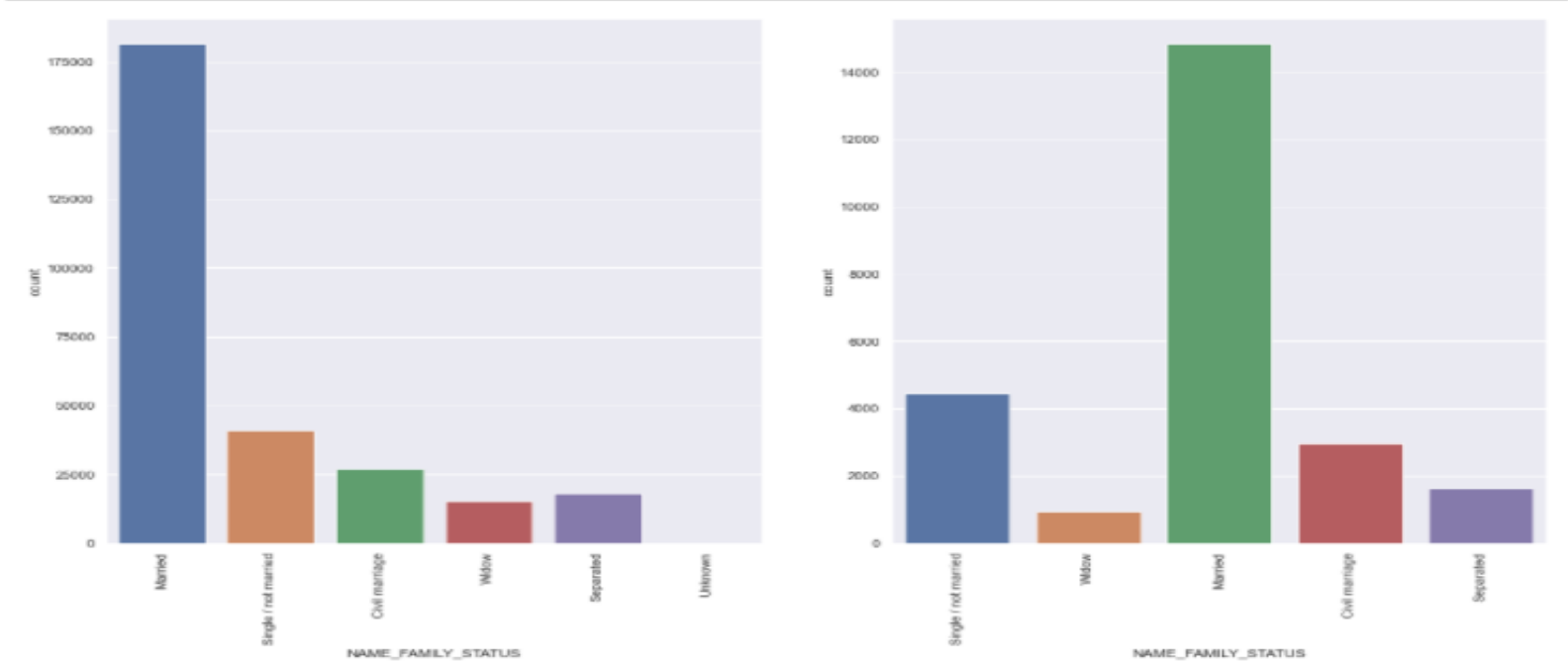
INCOME_SOURCE VS TARGET0/TARGET1





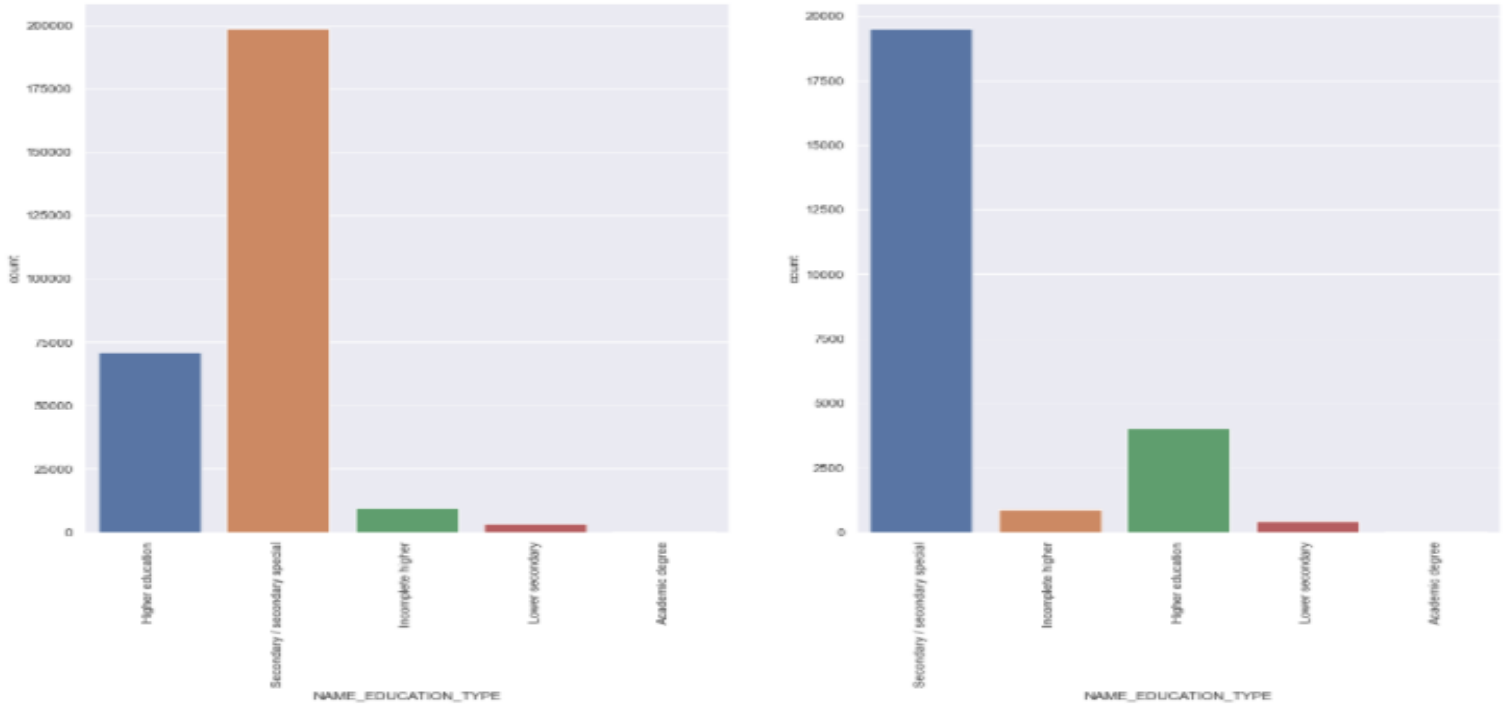
- We observe a decrease in the percentage of Payment Difficulties who are pentioners and an increase in the percentage of Payment Difficulties who are working when compared the percentages of both Payment Difficulties and non-Payment Difficulties.

FAMILY_STATUS VS TARGET0/TARGET1



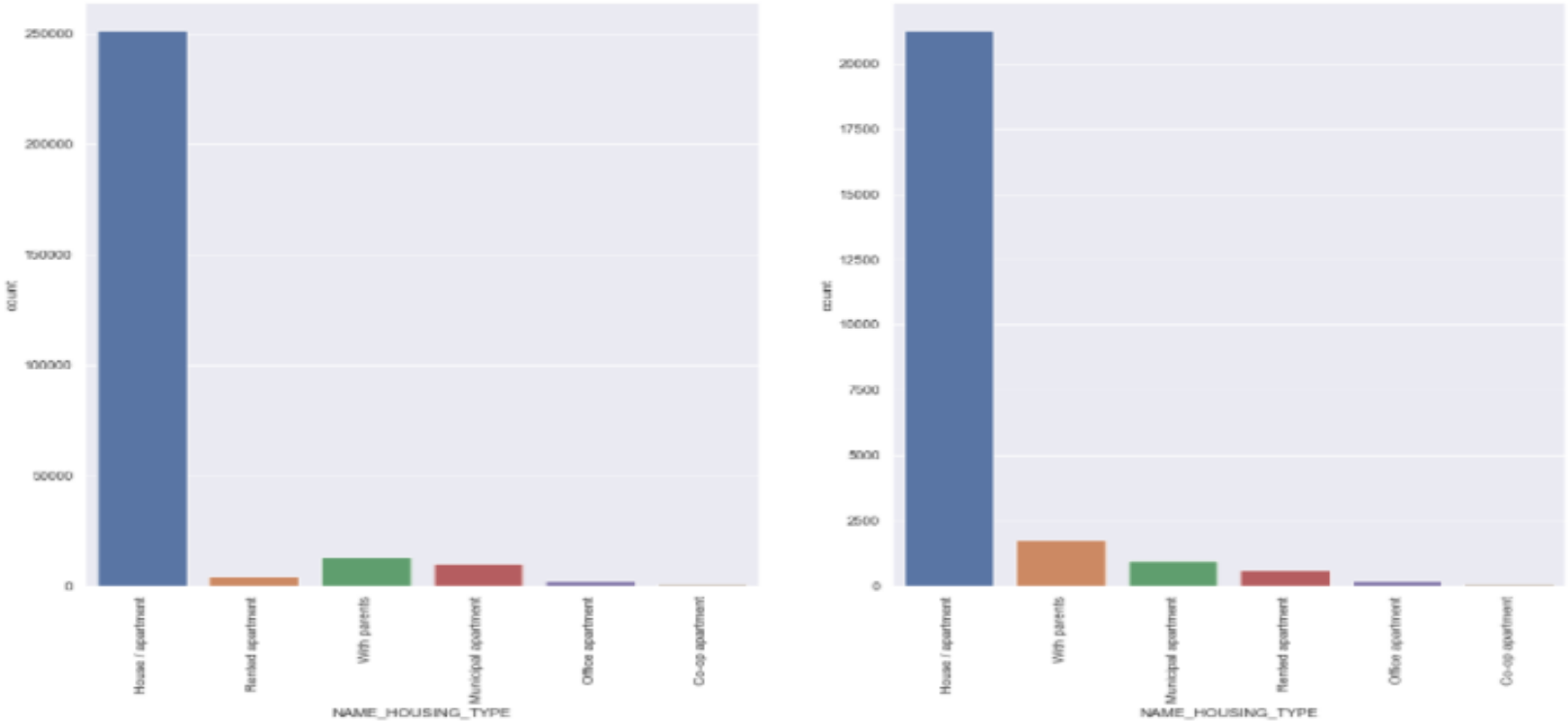
- We observe a decrease in the percentage of married and widowed with Loan Payment Difficulties and an increase in the the percentage of single and civil married with Loan Payment Difficulties when comapred with the percentages of both Loan Payment Difficulties and Loan Non-Payment Difficulties

EDUCATION VS TARGET0/TARGET1



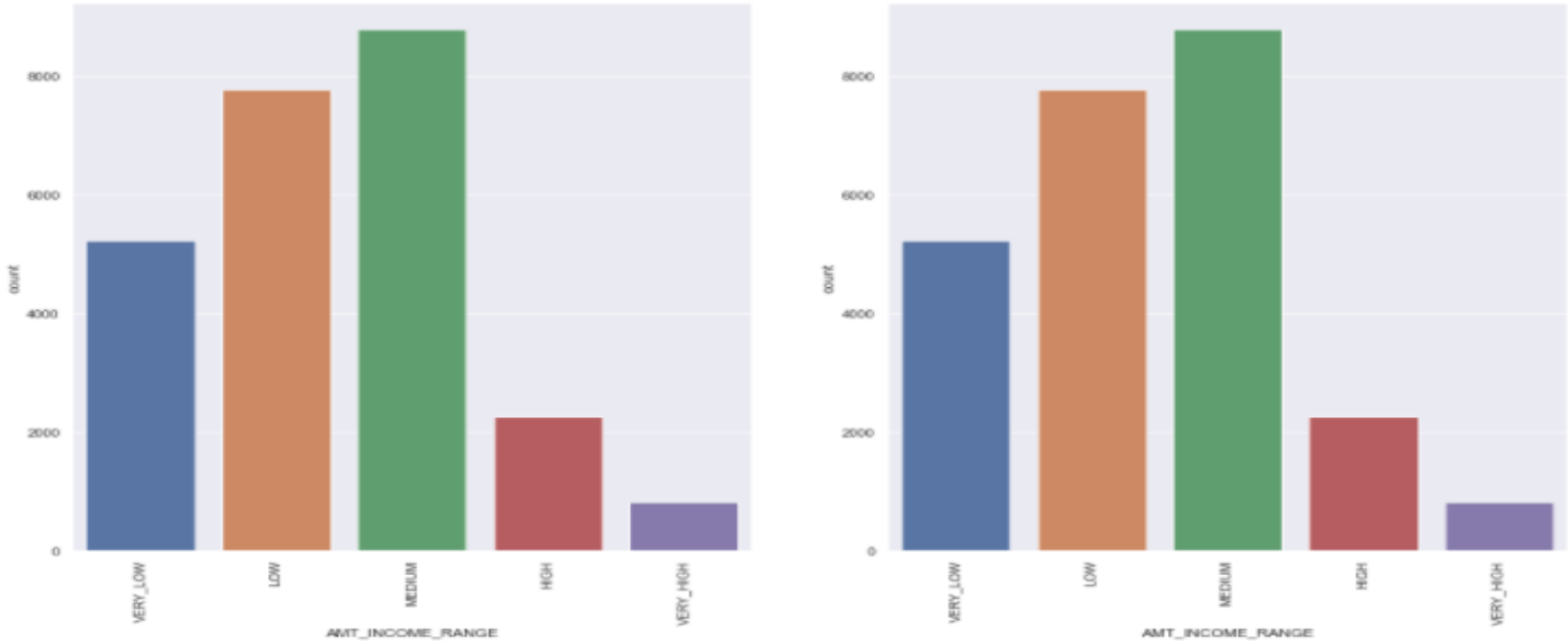
- We observe an increase in percentage of Loan Payment Difficulties whose educational qualifications are secondary/secondary special and a decrease in the percentage of Loan Payment Difficulties who have completed higher education when compared with the percentages of Loan Payment Difficulties and Loan Non-Payment Difficulties

HOUSE_TYPE VS TARGET0/TARGET1



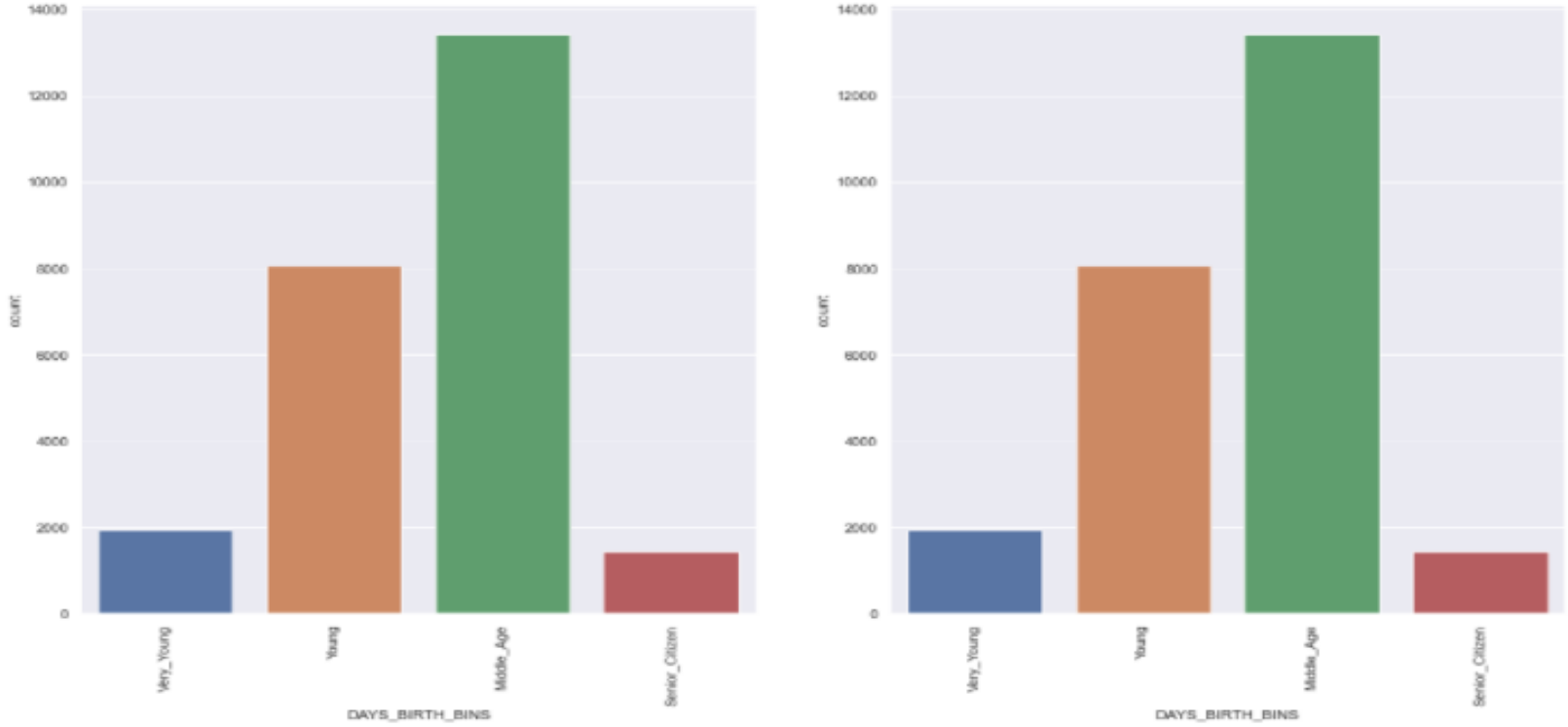
- We observe an increase in percentage of Loan Payment Difficulties whose educational qualifications are secondary/secondary special and a decrease in the percentage of Loan Payment Difficulties who have completed higher education when compared with the percentages of Loan Payment Difficulties and Loan Non-Payment Difficulties

INCOME_RANGE VS TARGET0/TARGET1



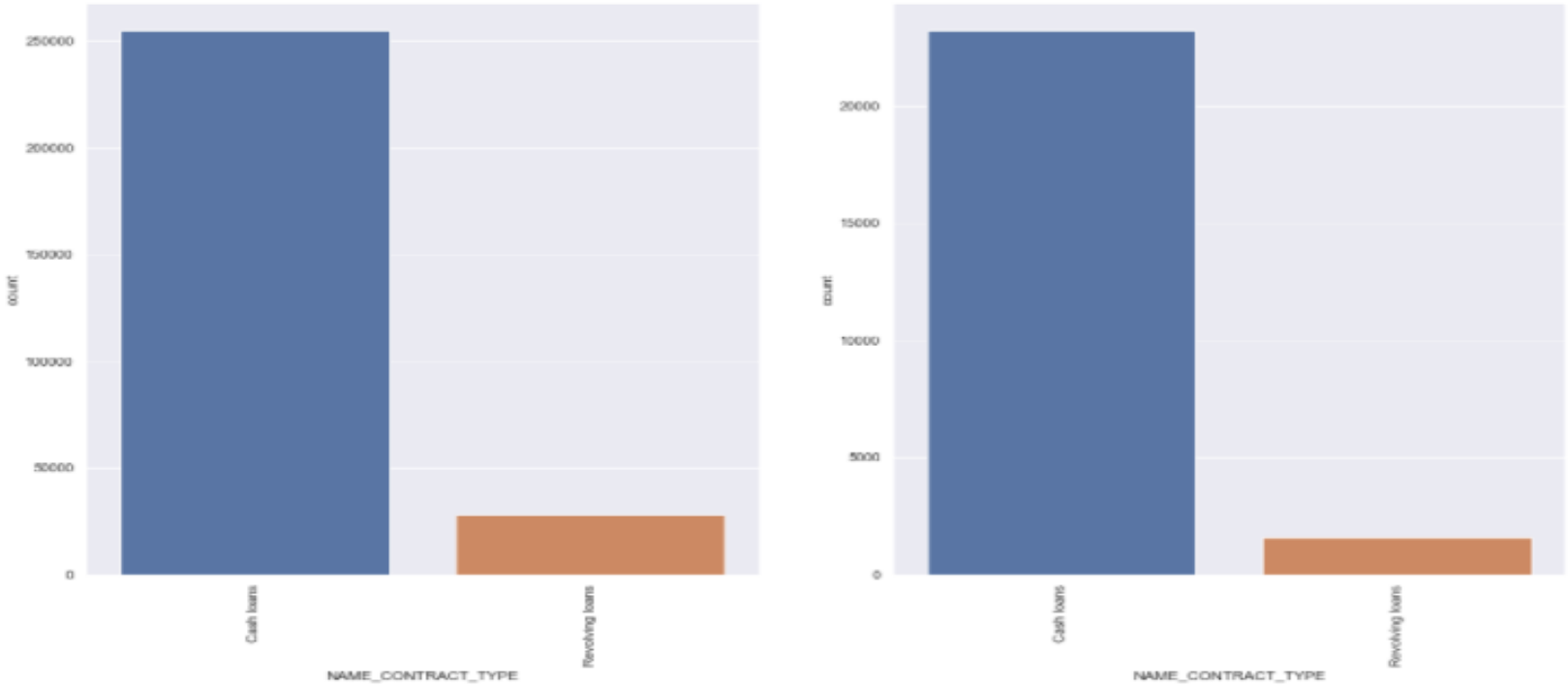
- We observe an increase in the percentage of Loan Payment Difficulties whose income is low when compared with the percentages of Payment Difficulties and Loan-Non Payment Difficulties

AGE VS TARGET0/TARGET1



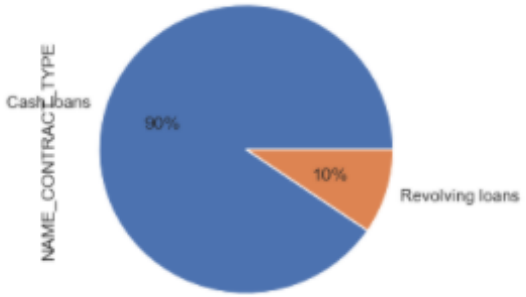
- We observe that there is an increase in the percentage of Loan Payment Difficulties who are young in age when compared to the percentages of Payment Difficulties and Loan-Non Payment Difficulties.

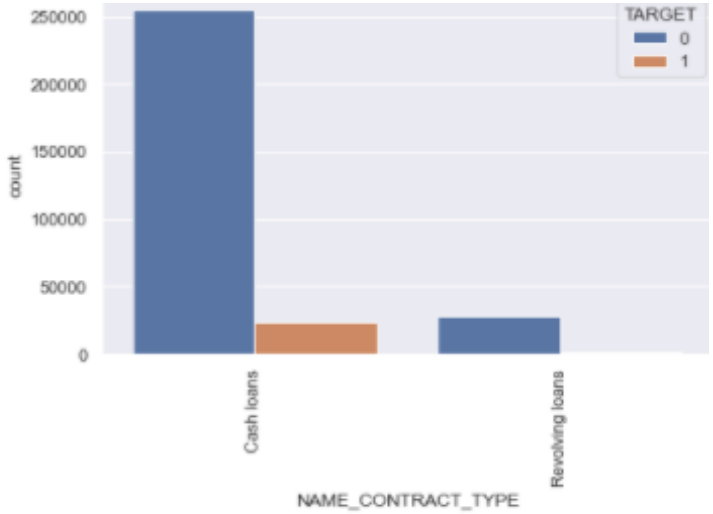
LOAN_TYPE VS TARGET0/TARGET1



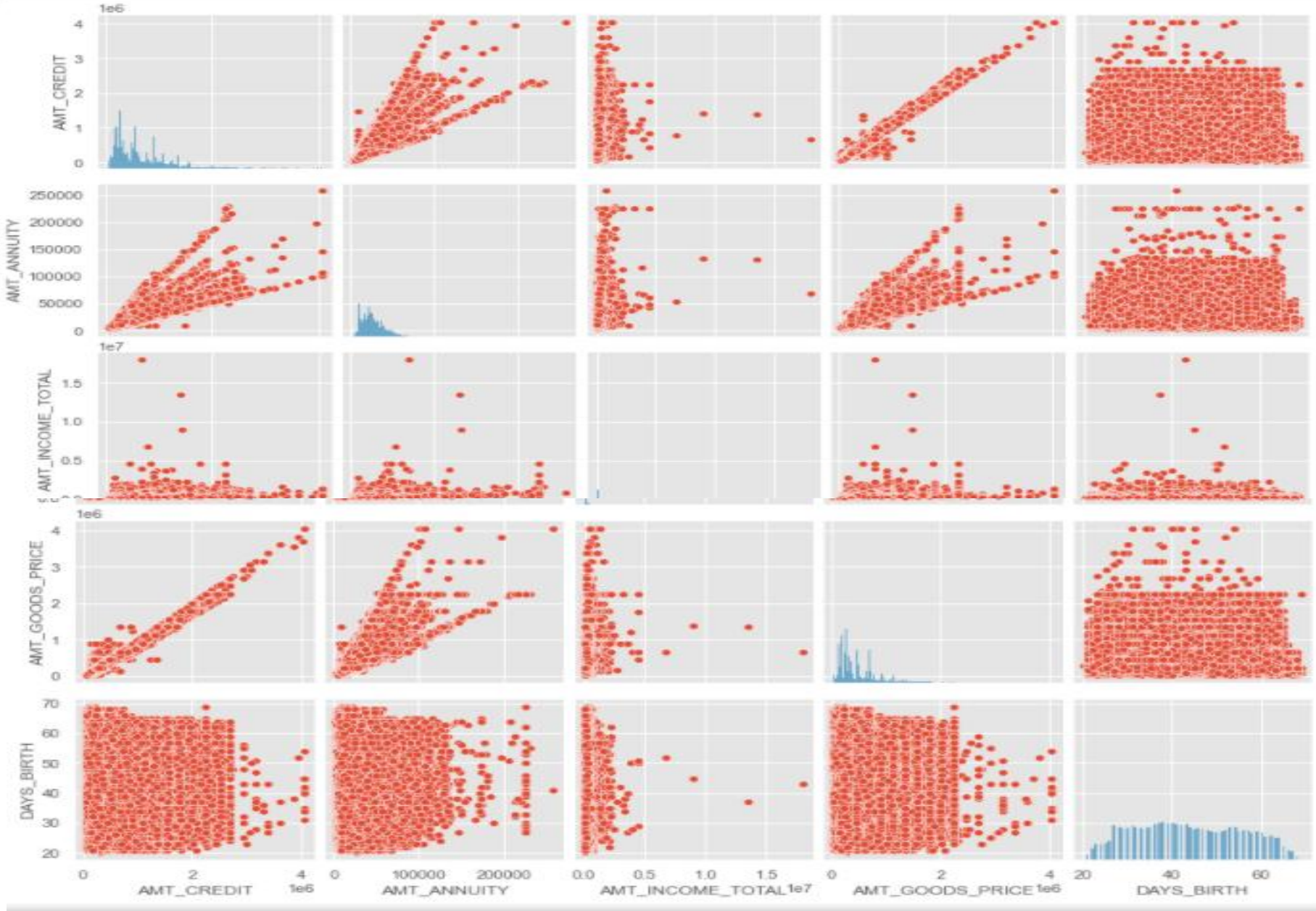
- We can observe that cash loans are preffered by both Loan Payment Difficulties and Loan-Non Payment Difficulties although there is a decrease in the percentage of Payment Difficulties who opt for revolving loans.

Function for all columns in list



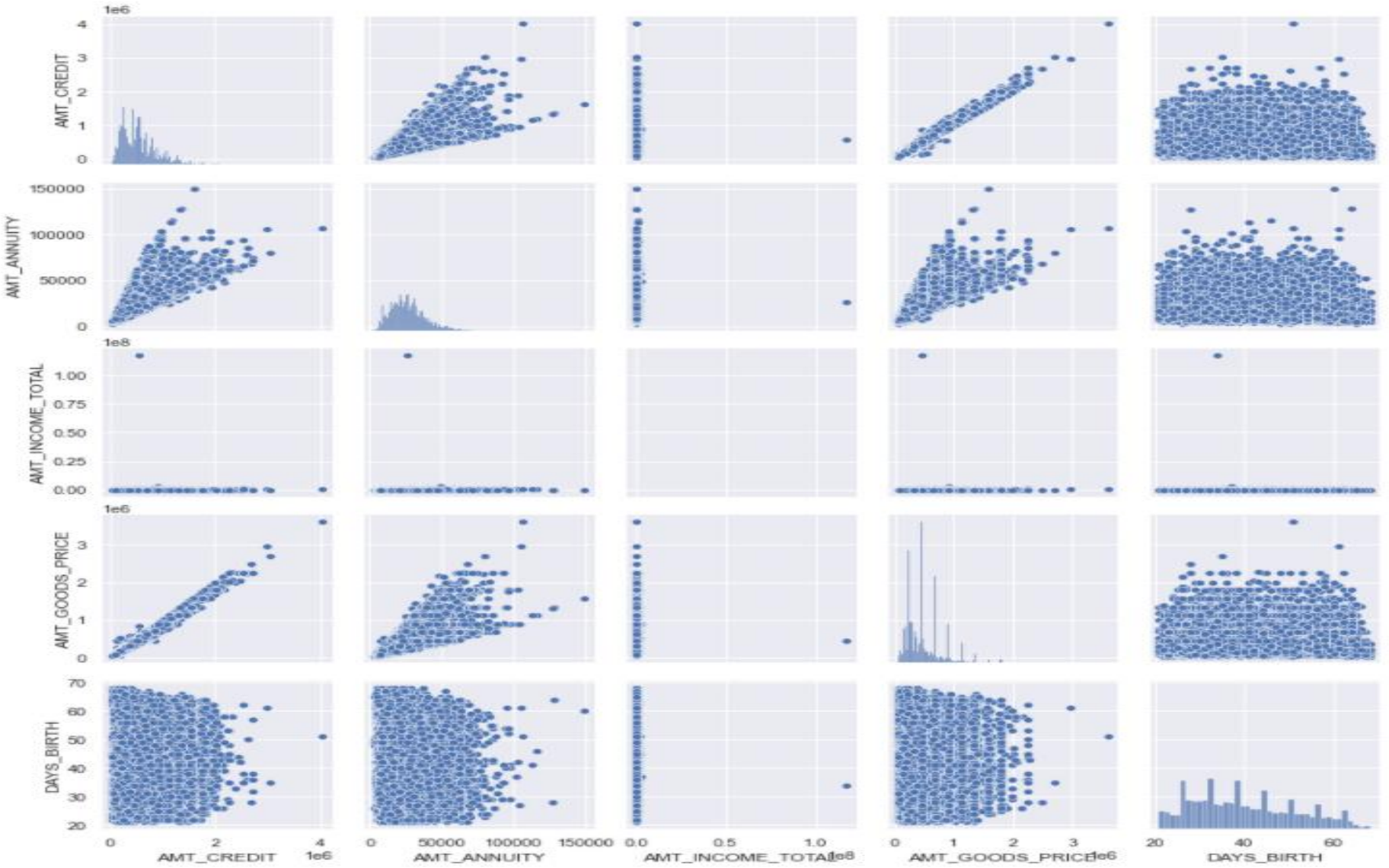


BIVARIENT ANALYSIS ON TARGET0



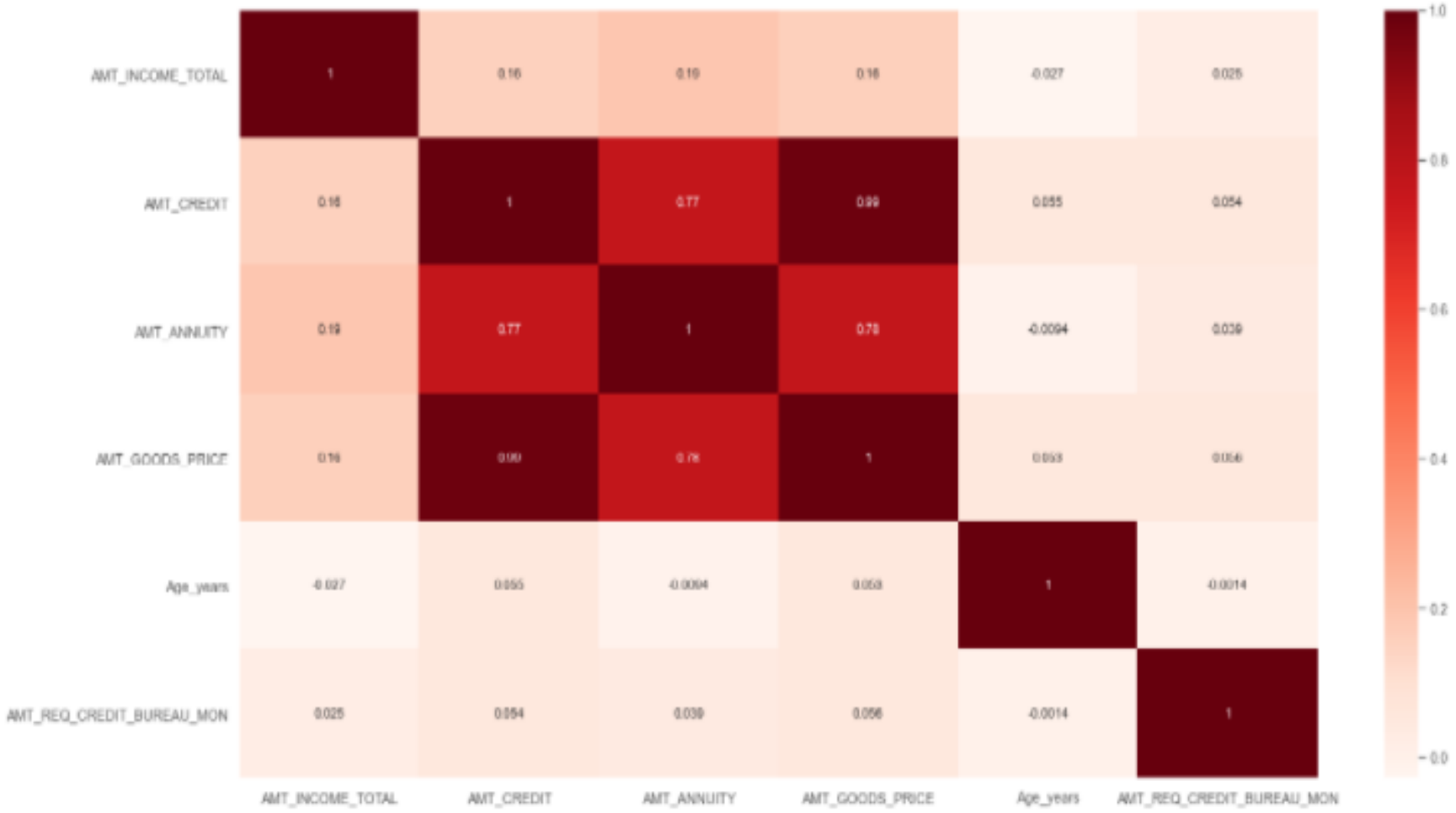
- AMT _CREDIT & AMT_ANNUITY are directly proportion to each other
- AMT_CREDIT & AMT_INCOME_TOTAL less income tends to more credit
- AMT_CREDIT & AMT_GOODS_PRIZE are directly proportional
- AMT_CREDIT & DAYS_BIRTH younger the person tends to take more loan

BIVARIENT ANALYSIS ON TARGET1

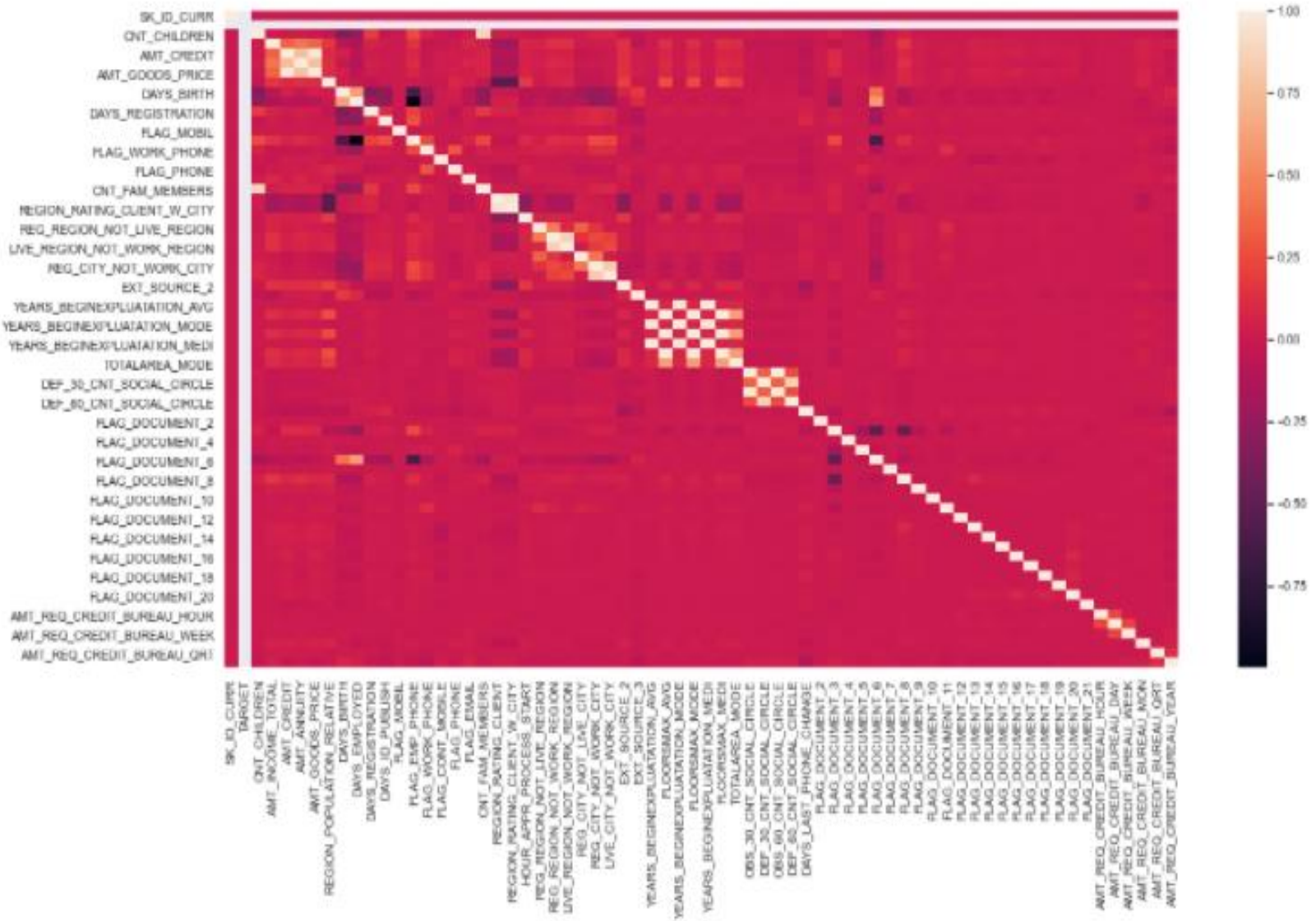


- AMT _CREDIT & AMT_ANNUITY are directly proportion to each other
- AMT_CREDIT & AMT_INCOME_TOTAL less income tends to more credit
- AMT_CREDIT & AMT_GOODS_PRIZE are directly proportional
- AMT_CREDIT & DAYS_BIRTH younger the person tends to take more loan

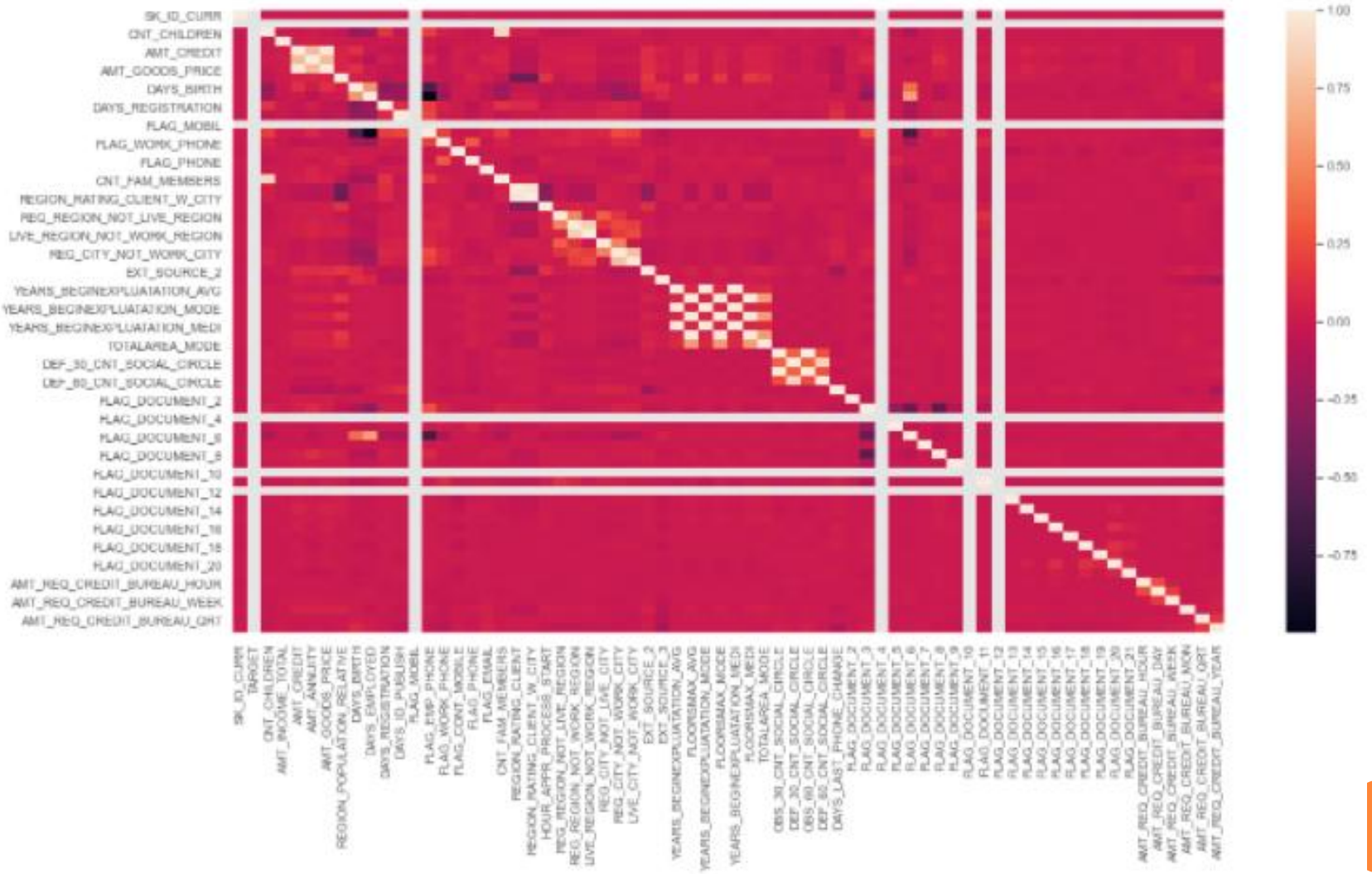
Correlation on AMT_INCOME_TOTAL, AMT_CREDIT, AMT_ANNUITY, AMT_GOODS_PRICE, Age_years, AMT_REQ_CREDIT_BUREAU_MON



HEATMAP ON TARGET0



HEATMAP ON TARGET1

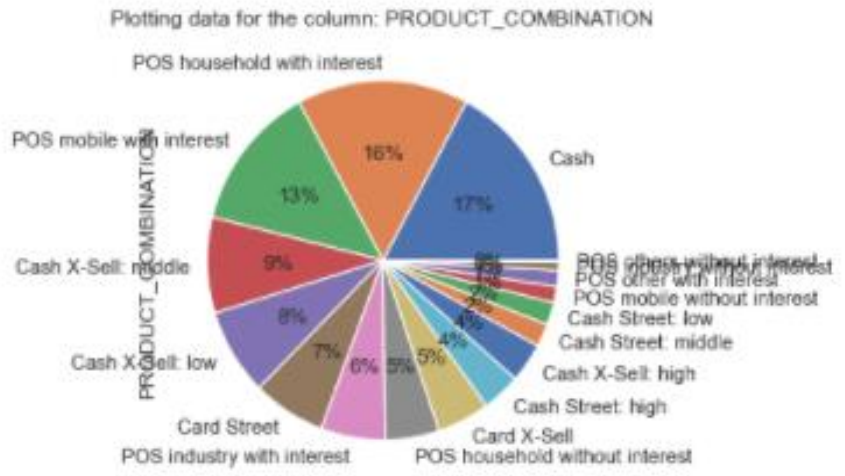


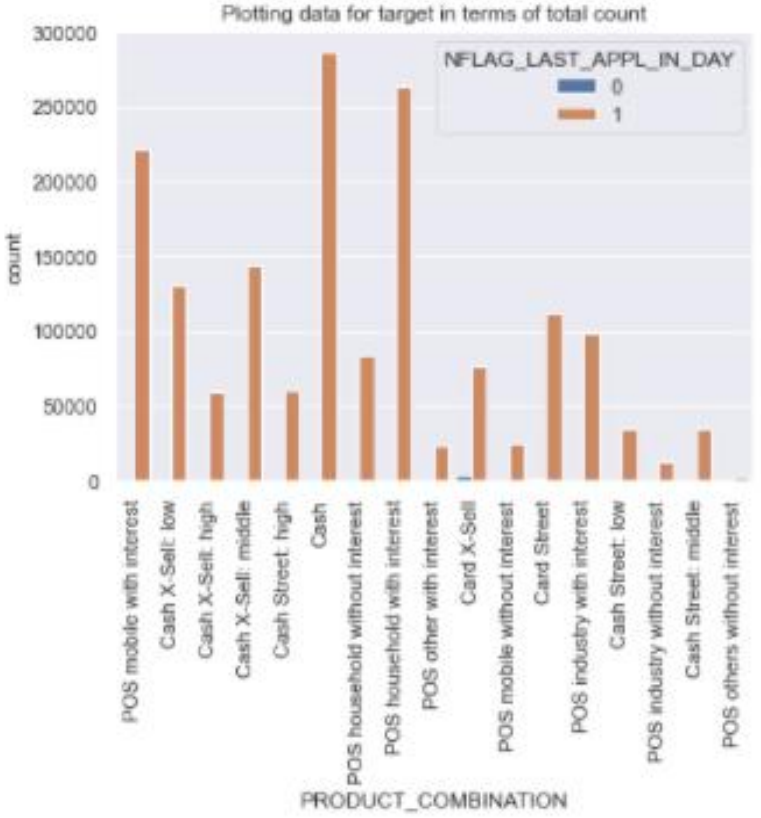
Analysis on previous_application.csv

- After checking null values there are only 2 variable which have null values
- Product_combination and amt_credit
- We analysed it on bases of NFLAG_LAST_APPL_IN_DAY (as a target)

FUNCTION FOR UNIVARIENT ANALYSIS

```
Pltting  PRODUCT_COMBINATION
```





Merging application_data and previous_application

- Both dataframe have common variable SK_ID_CURR
- We have merged both on SK_ID_CURR variable
- Used pandas function merge to merge both dataframe for futher analysis