CREDIT EDA CASE STUDY

Presented by:

Abhinav Mohan Agarwal

Purpose

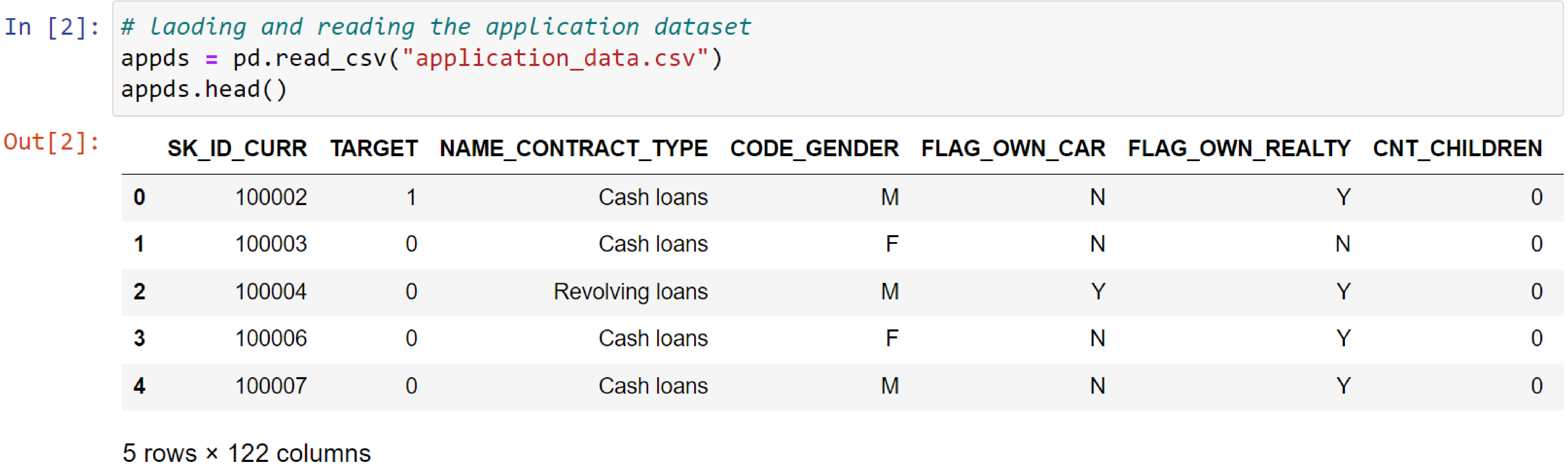
This Case study helps us to apply EDA on real life scenario, it also gives us an idea on risk analytics in the field of banking and allows us to understand what steps should be taken to minimize the risk in the financial sector

Introduction

- In this case study we are given three datasets :
- *application_data.csv* - contains all the information of the client at the time of application. The data is about whether a client has payment difficulties.
- *previous_application.csv* - contains information about the client's previous loan data. It contains the data on whether the previous application had been Approved, Cancelled, Refused or Unused offer.
- *columns_description.csv* - data dictionary which describes the meaning of the variables.

Data loading

- First we load the application_data.csv for the analysis

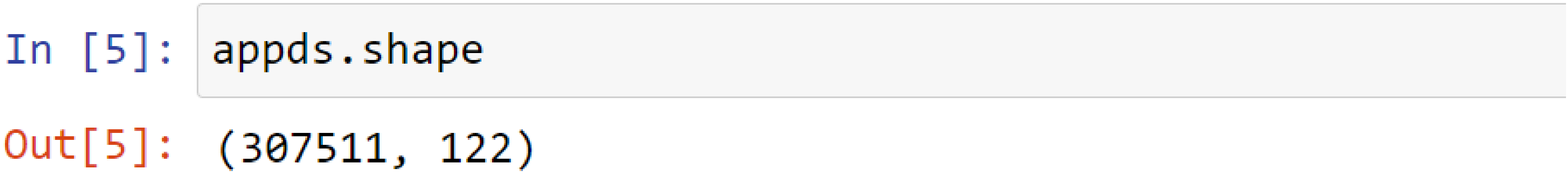


```
In [2]: # laoding and reading the application dataset
        appds = pd.read_csv("application_data.csv")
        appds.head()
```

Out[2]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN
0	100002	1	Cash loans	M	N	Y	0
1	100003	0	Cash loans	F	N	N	0
2	100004	0	Revolving loans	M	Y	Y	0
3	100006	0	Cash loans	F	N	Y	0
4	100007	0	Cash loans	M	N	Y	0

5 rows × 122 columns

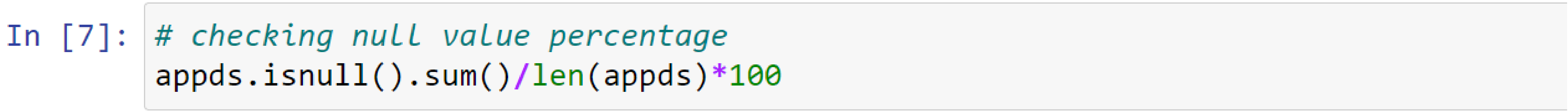


```
In [5]: appds.shape

Out[5]: (307511, 122)
```

- On applying .shape to the data we see there are 307511 rows and 122 columns
- now from here we do data handling i.e. checking the null values and removing the same

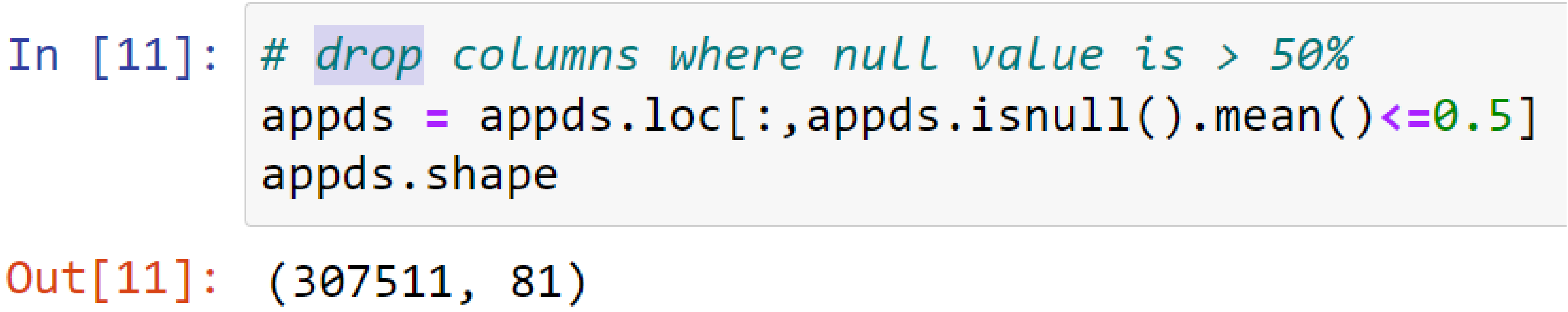
Data Cleaning

- Data cleaning is done for better analysis for the data and to remove irrelevant or null values which would alter the analysis done and give a false/wrong analysis
- So , for this we find the irrelevant cols and the cols having null values



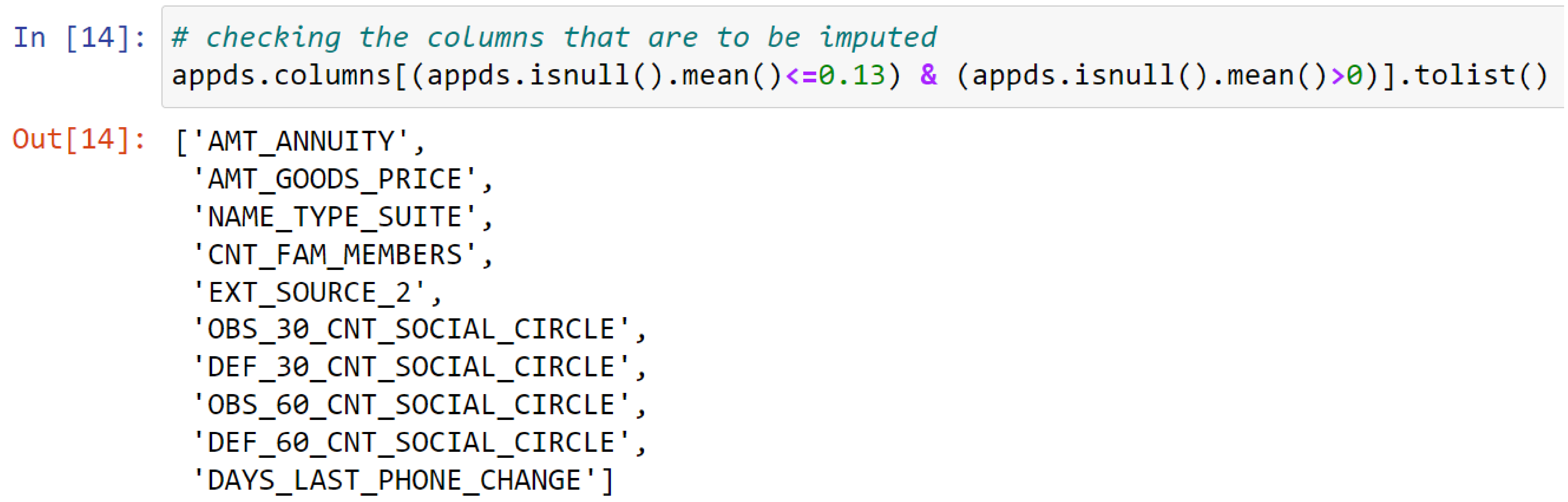
```
In [7]: # checking null value percentage
        appds.isnull().sum()/len(appds)*100
```

- From this we find the null value percentage and remove those having null value >50% and drop those cols.



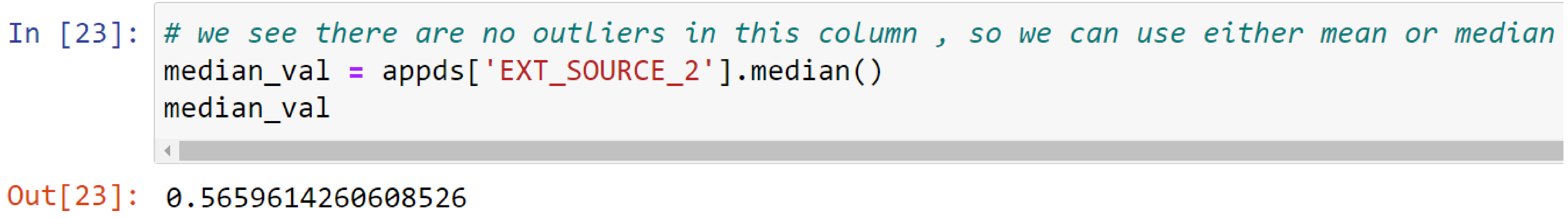
```
In [11]: # drop columns where null value is > 50%
         appds = appds.loc[:,appds.isnull().mean()<=0.5]
         appds.shape

Out[11]: (307511, 81)
```

- Here we see that the cols have dropped to 81 , now we do imputation in cols where there are null values >13% , imputation can be done with Mean , Median or Mode depending on the type of value we have (numerical or categorical)



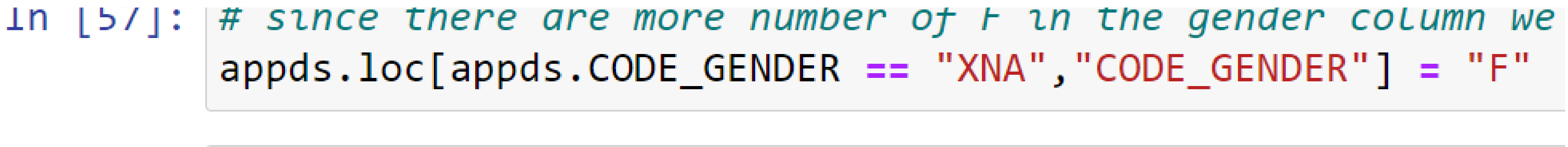
```
In [14]: # checking the columns that are to be imputed
         appds.columns[(appds.isnull().mean()<=0.13) & (appds.isnull().mean()>0)].tolist()

Out[14]: ['AMT_ANNUITY',
          'AMT_GOODS_PRICE',
          'NAME_TYPE_SUITE',
          'CNT_FAM_MEMBERS',
          'EXT_SOURCE_2',
          'OBS_30_CNT_SOCIAL_CIRCLE',
          'DEF_30_CNT_SOCIAL_CIRCLE',
          'OBS_60_CNT_SOCIAL_CIRCLE',
          'DEF_60_CNT_SOCIAL_CIRCLE',
          'DAYS_LAST_PHONE_CHANGE']
```

- Here we can see the cols that are needed for imputation , we check the values for the same using Mean , Median and mode but don't do as it goes beyond the objective of the case study.



```
In [23]: # we see there are no outliers in this column , so we can use either mean or median
         median_val = appds['EXT_SOURCE_2'].median()
         median_val

Out[23]: 0.5659614260608526
```

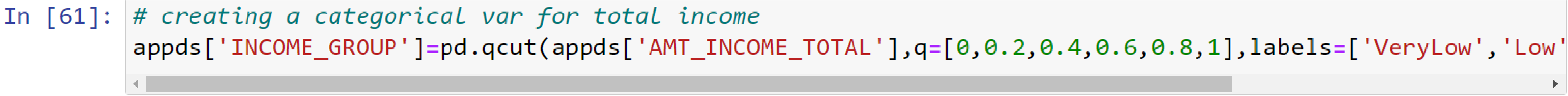
Now , we check the Gender cols where there is XNA (not available) , here we perform mean and which goes in favour of F since there are more F genders and fill XNA with F



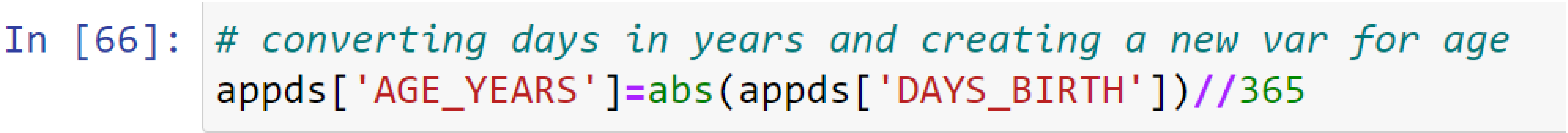
```
In [57]: # since there are more number of F in the gender column we
         appds.loc[appds.CODE_GENDER == "XNA","CODE_GENDER"] = "F"
```

Binning variable

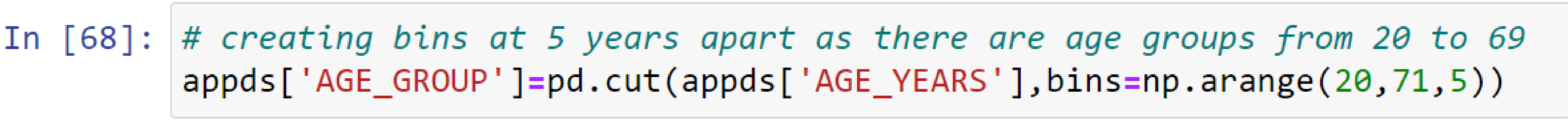
- we perform binning for a better clarity while doing analysis on 2 variables 'AMT_INCOME_TOTAL' to 'INCOME_GROUP' after binning and on 'DAYS_BIRTH'
- for 'DAYS_BIRTH' first we convert the day in years and get 'AGE_YEARS' and then we bin that to 'AGE_GROUP'.



```
In [61]: # creating a categorical var for total income
         appds['INCOME_GROUP']=pd.qcut(appds['AMT_INCOME_TOTAL'],q=[0,0.2,0.4,0.6,0.8,1],labels=['VeryLow','Low'
```



```
In [66]: # converting days in years and creating a new var for age
         appds['AGE_YEARS']=abs(appds['DAYS_BIRTH'])//365
```

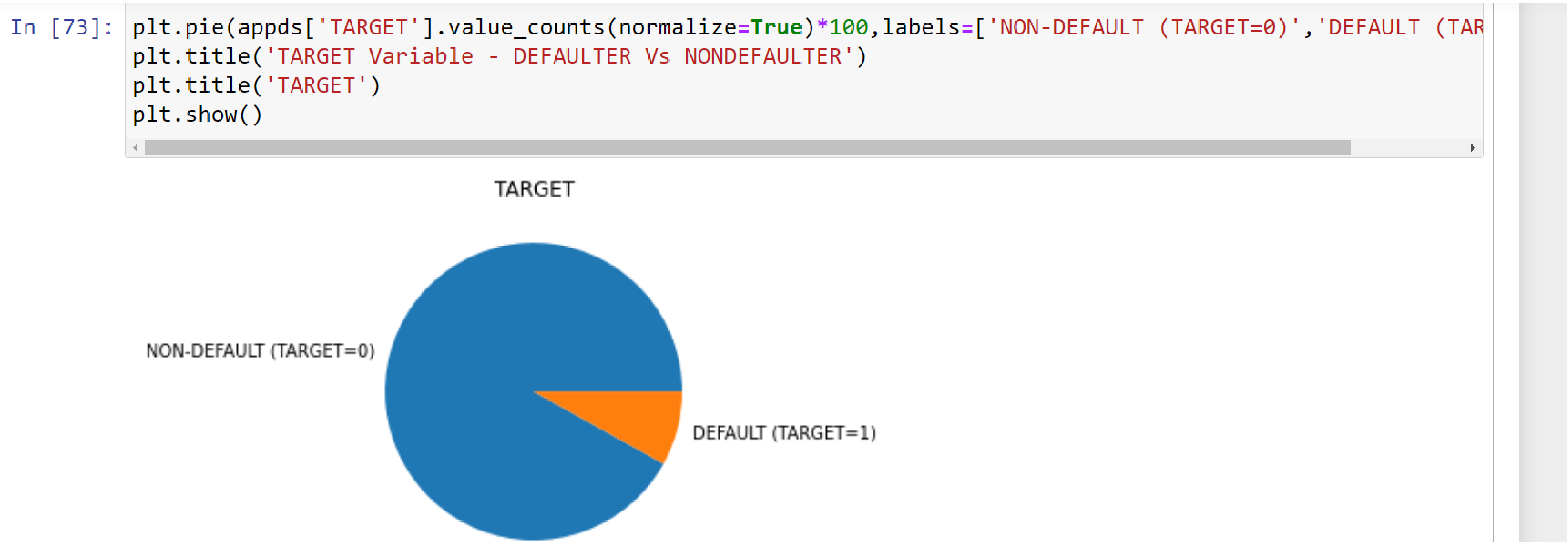


```
In [68]: # creating bins at 5 years apart as there are age groups from 20 to 69
         appds['AGE_GROUP']=pd.cut(appds['AGE_YEARS'],bins=np.arange(20,71,5))
```

- Here the 'INCOME_GROUP' is binned as ['VeryLow','Low','Medium','High','VeryHigh']
- The 'AGE_GROUP' is binned from 20-71 having a difference of 5 np.arange(20,71,5))

Imbalance Check

- Here we do analysis on the TARGET variable i.e. either 0 or 1 (Non-Defaulter and Defaulter)
- helping us in understanding to whom we give the loan or not.

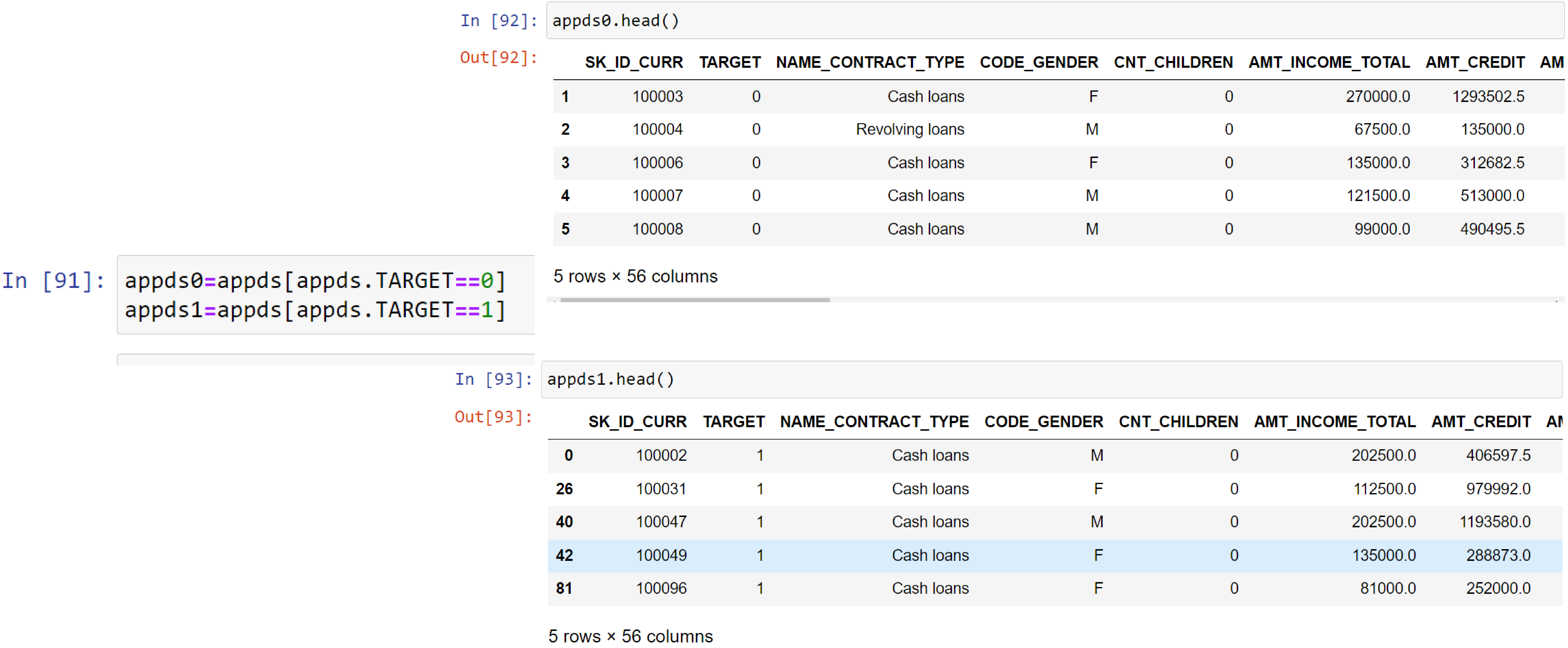


```
In [73]: plt.pie(appds['TARGET'].value_counts(normalize=True)*100,labels=['NON-DEFAULT (TARGET=0)','DEFAULT (TAR
         plt.title('TARGET Variable - DEFAULTER Vs NONDEFAULTER')
         plt.title('TARGET')
         plt.show()
```

- here we can see from the pie chart that around 92% people are non defaulters and 8% people are defaulters.

Creating 2 dataset wrt the TARGET variable

- now we create 2 Datasets appds0 and appds1 for TARGET as 0 and 1 respectively



```
In [92]: appds0.head()
```

Out[92]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AM
1	100003	0	Cash loans	F	0	270000.0	1293502.5	
2	100004	0	Revolving loans	M	0	67500.0	135000.0	
3	100006	0	Cash loans	F	0	135000.0	312682.5	
4	100007	0	Cash loans	M	0	121500.0	513000.0	
5	100008	0	Cash loans	M	0	99000.0	490495.5	

5 rows × 56 columns

```
In [91]: appds0=appds[appds.TARGET==0]
         appds1=appds[appds.TARGET==1]
```

```
In [93]: appds1.head()
```

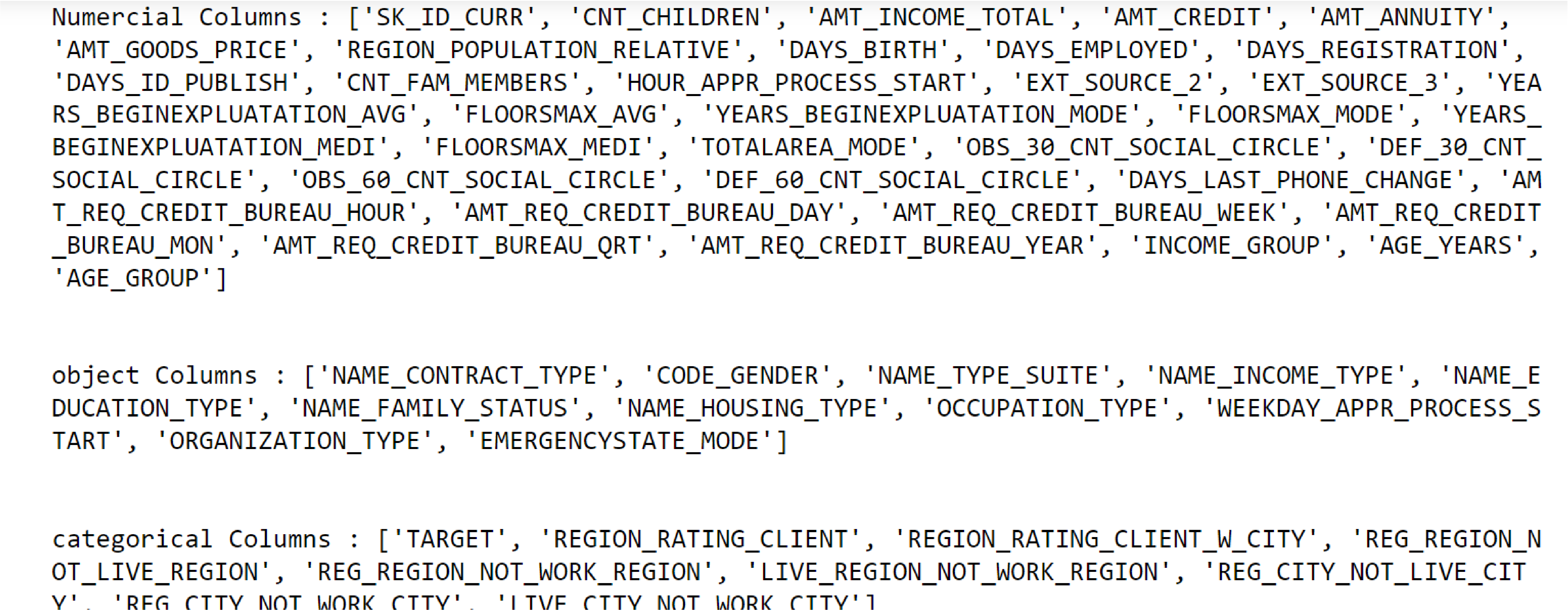
Out[93]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AI
0	100002	1	Cash loans	M	0	202500.0	406597.5	
26	100031	1	Cash loans	F	0	112500.0	979992.0	
40	100047	1	Cash loans	M	0	202500.0	1193580.0	
42	100049	1	Cash loans	F	0	135000.0	288873.0	
81	100096	1	Cash loans	F	0	81000.0	252000.0	

5 rows × 56 columns

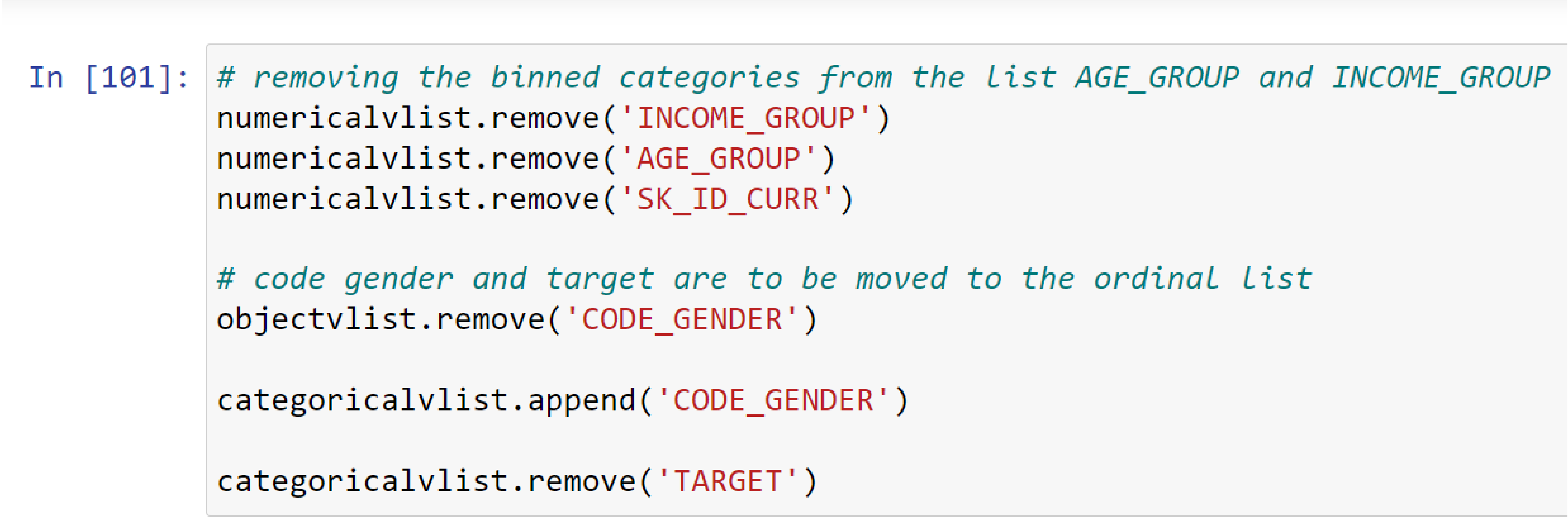
Univariate Analysis

- for univariate analysis we create 3 list from the dataset(appds)
  - Categorical List
  - Object List
  - Numerical List



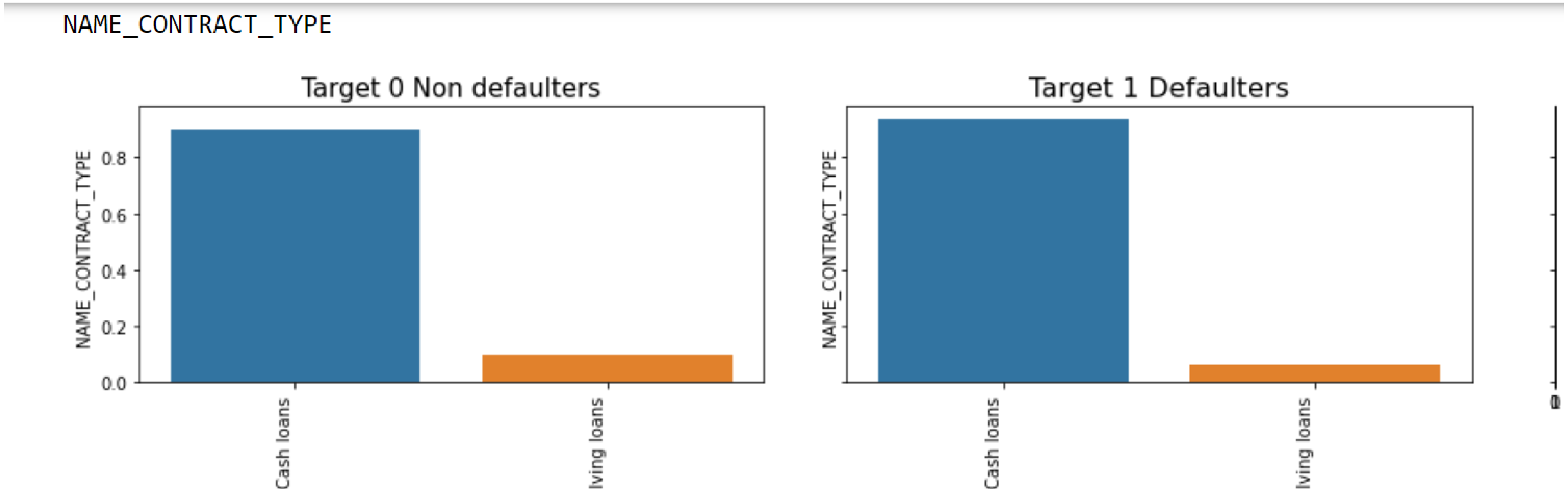
```
Numercial Columns : ['SK_ID_CURR', 'CNT_CHILDREN', 'AMT_INCOME_TOTAL', 'AMT_CREDIT', 'AMT_ANNUITY',
'AMT_GOODS_PRICE', 'REGION_POPULATION_RELATIVE', 'DAYS_BIRTH', 'DAYS_EMPLOYED', 'DAYS_REGISTRATION',
'DAYS_ID_PUBLISH', 'CNT_FAM_MEMBERS', 'HOUR_APPR_PROCESS_START', 'EXT_SOURCE_2', 'EXT_SOURCE_3', 'YEA
RS_BEGINEXPLUATATION_AVG', 'FLOORSMAX_AVG', 'YEARS_BEGINEXPLUATATION_MODE', 'FLOORSMAX_MODE', 'YEARS_
BEGINEXPLUATATION_MEDI', 'FLOORSMAX_MEDI', 'TOTALAREA_MODE', 'OBS_30_CNT_SOCIAL_CIRCLE', 'DEF_30_CNT_
SOCIAL_CIRCLE', 'OBS_60_CNT_SOCIAL_CIRCLE', 'DEF_60_CNT_SOCIAL_CIRCLE', 'DAYS_LAST_PHONE_CHANGE', 'AM
T_REQ_CREDIT_BUREAU_HOUR', 'AMT_REQ_CREDIT_BUREAU_DAY', 'AMT_REQ_CREDIT_BUREAU_WEEK', 'AMT_REQ_CREDIT
_BUREAU_MON', 'AMT_REQ_CREDIT_BUREAU_QRT', 'AMT_REQ_CREDIT_BUREAU_YEAR', 'INCOME_GROUP', 'AGE_YEARS',
'AGE_GROUP']


object Columns : ['NAME_CONTRACT_TYPE', 'CODE_GENDER', 'NAME_TYPE_SUITE', 'NAME_INCOME_TYPE', 'NAME_E
DUCATION_TYPE', 'NAME_FAMILY_STATUS', 'NAME_HOUSING_TYPE', 'OCCUPATION_TYPE', 'WEEKDAY_APPR_PROCESS_S
TART', 'ORGANIZATION_TYPE', 'EMERGENCYSTATE_MODE']


categorical Columns : ['TARGET', 'REGION_RATING_CLIENT', 'REGION_RATING_CLIENT_W_CITY', 'REG_REGION_N
OT_LIVE_REGION', 'REG_REGION_NOT_WORK_REGION', 'LIVE_REGION_NOT_WORK_REGION', 'REG_CITY_NOT_LIVE_CIT
Y', 'REG_CITY_NOT_WORK_CITY', 'LIVE_CITY_NOT_WORK_CITY']
```

- these list have binned categories too so they are removed from the list



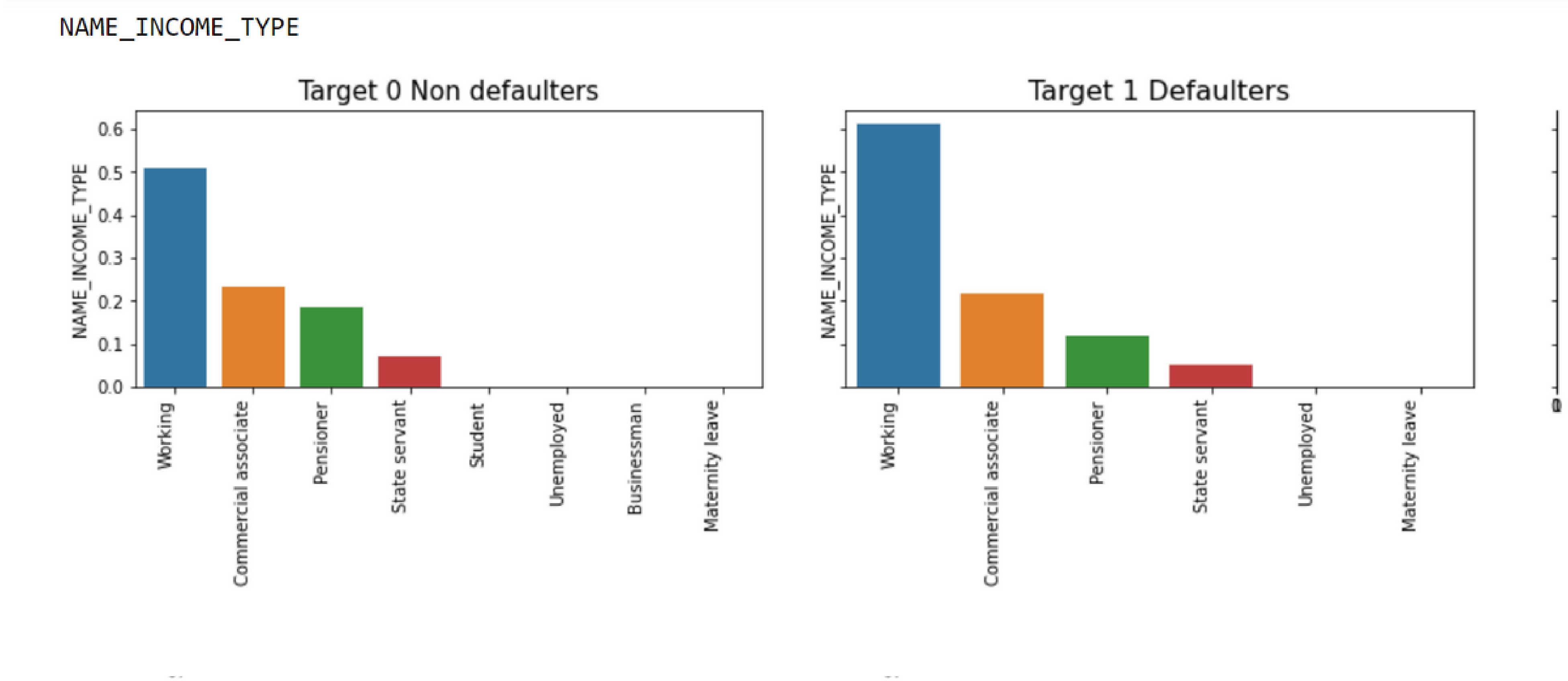
```
In [101]: # removing the binned categories from the list AGE_GROUP and INCOME_GROUP
          numericalvlist.remove('INCOME_GROUP')
          numericalvlist.remove('AGE_GROUP')
          numericalvlist.remove('SK_ID_CURR')

          # code gender and target are to be moved to the ordinal list
          objectvlist.remove('CODE_GENDER')

          categoricalvlist.append('CODE_GENDER')

          categoricalvlist.remove('TARGET')
```

- now we do univariate analysis on the object list with the help of for loop in the list and create graphs for each col.

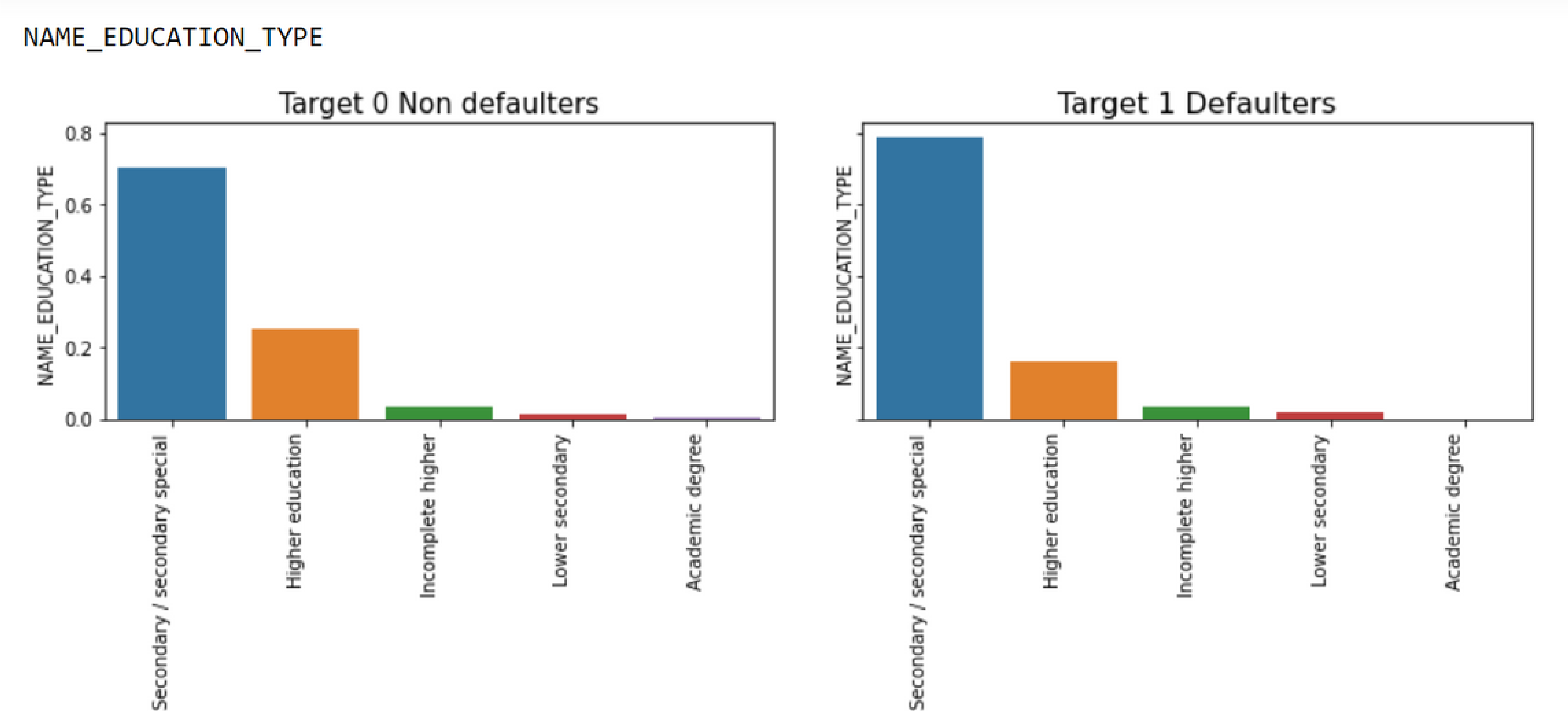


```
NAME_CONTRACT_TYPE
```

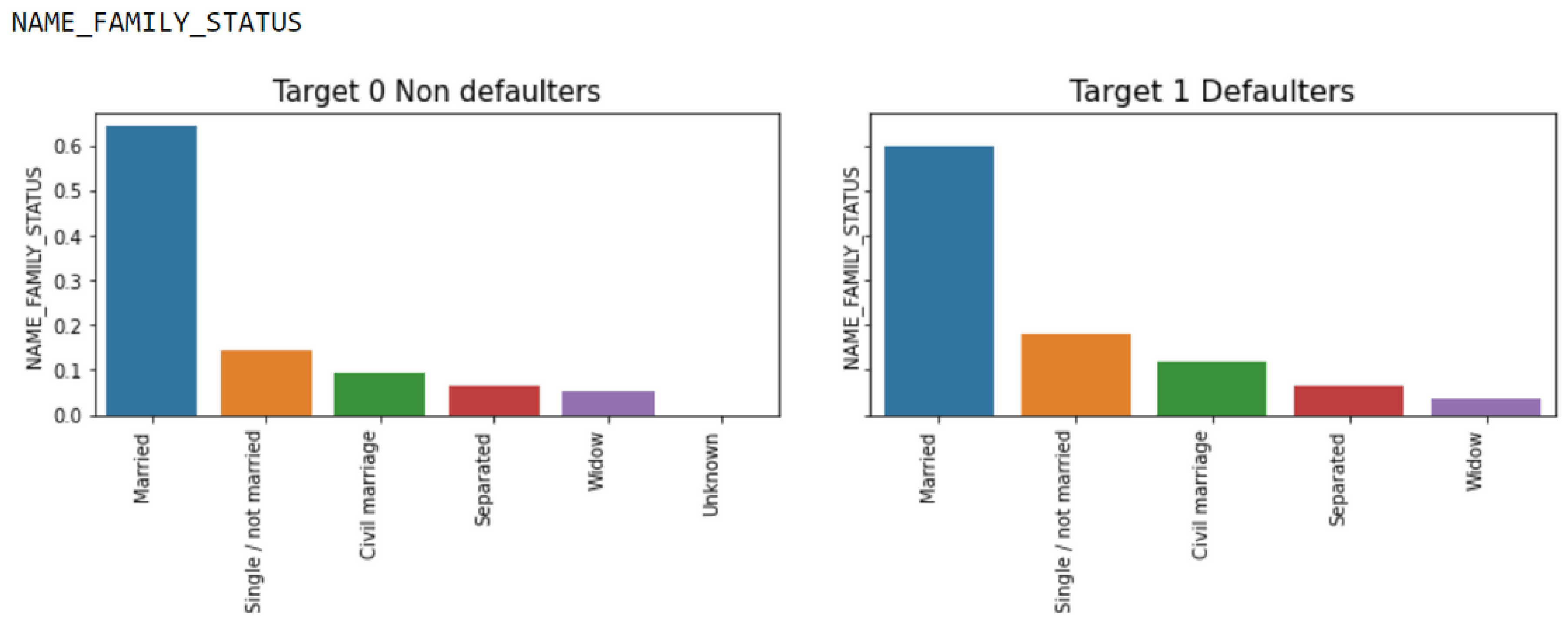
NAME_INCOME_TYPE



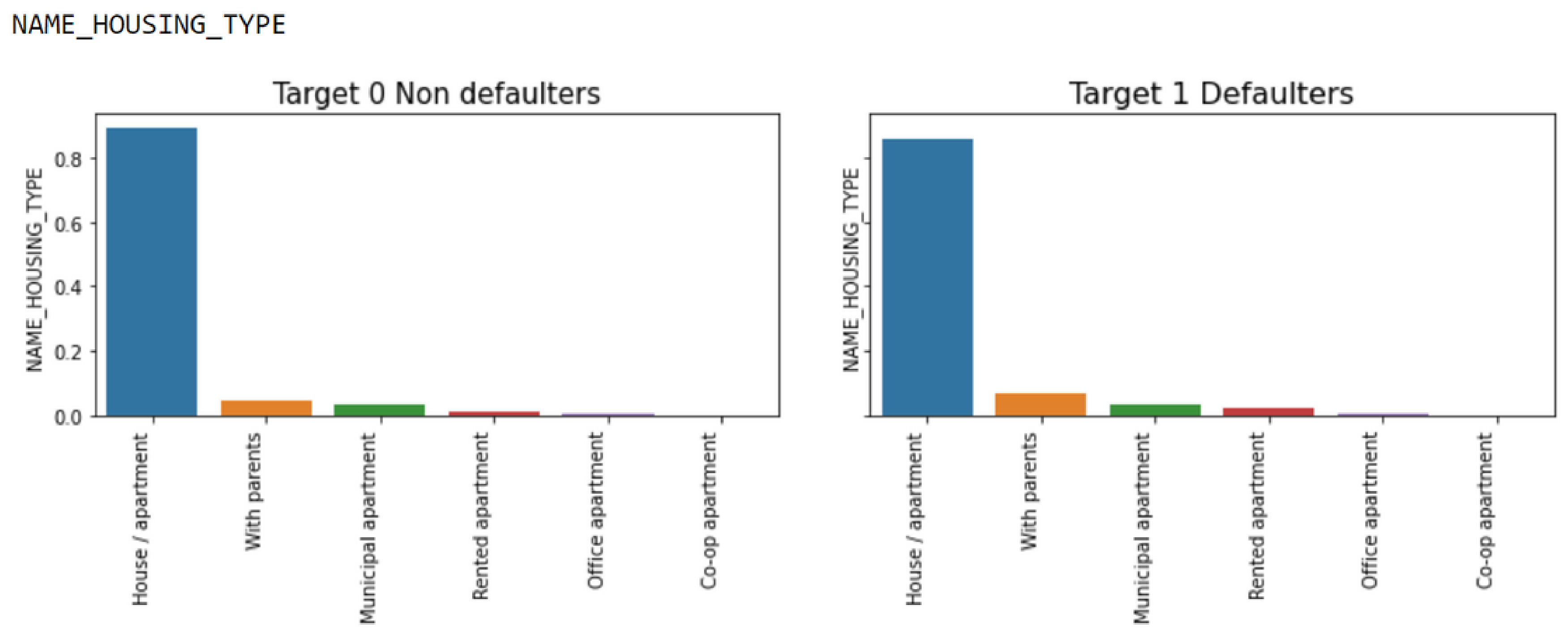
NAME_EDUCATION_TYPE



NAME_FAMILY_STATUS



NAME_HOUSING_TYPE

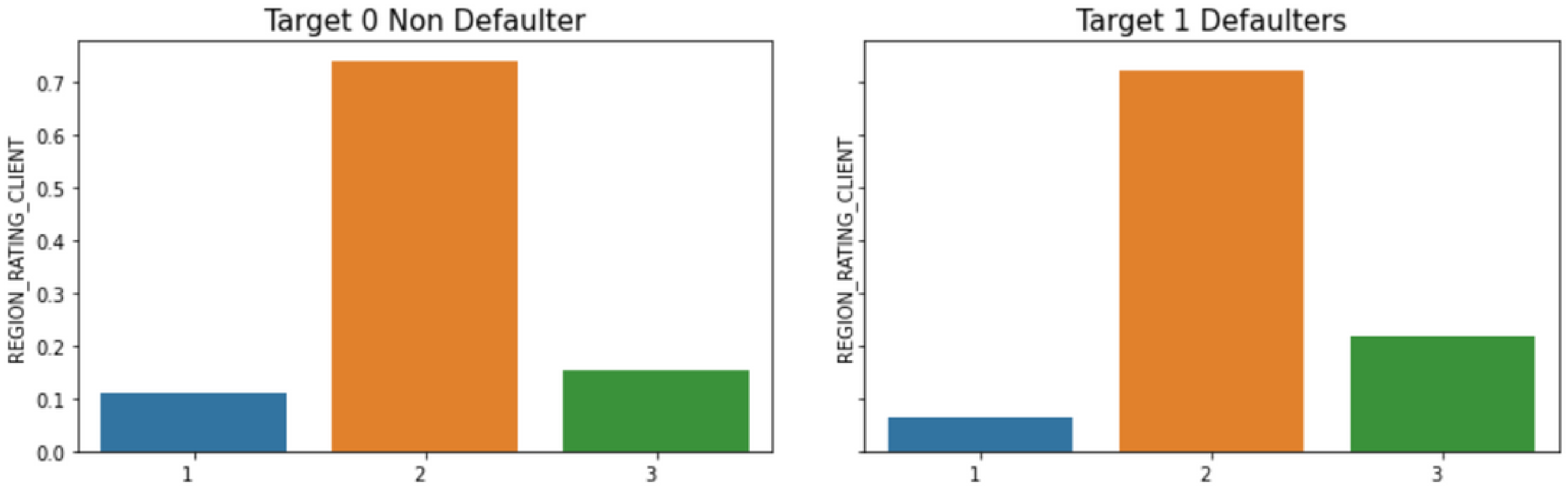


...and more

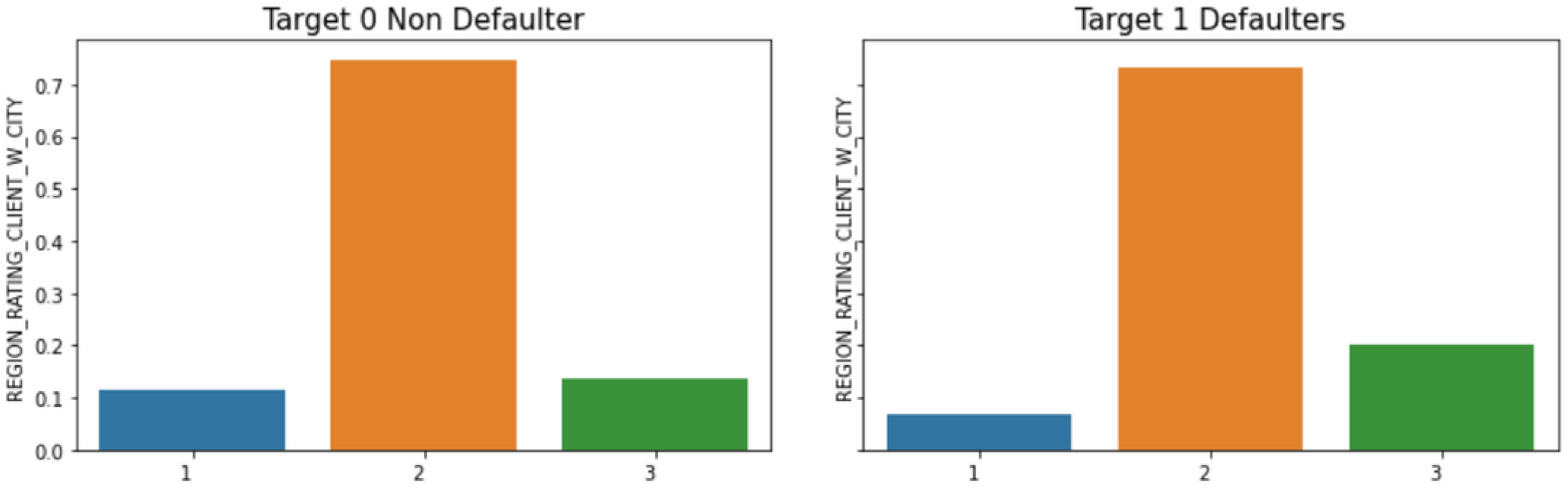
we do the same thing for the remaining 2 lists i.e. Categorical and numerical list

For Categorical List

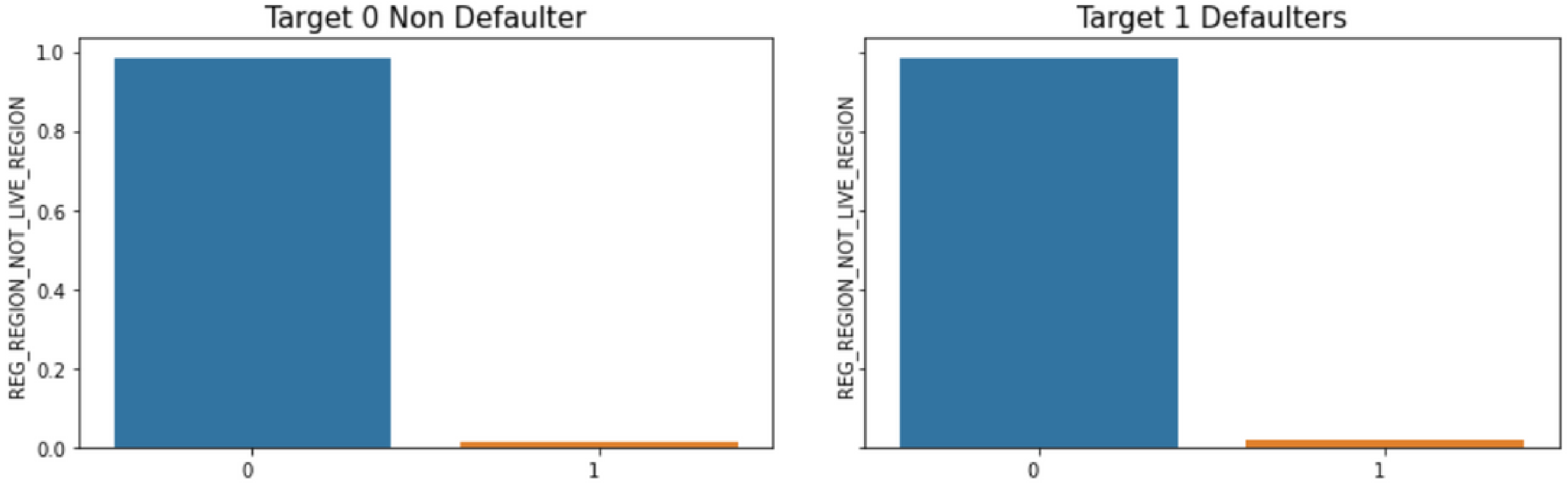
REGION_RATING_CLIENT



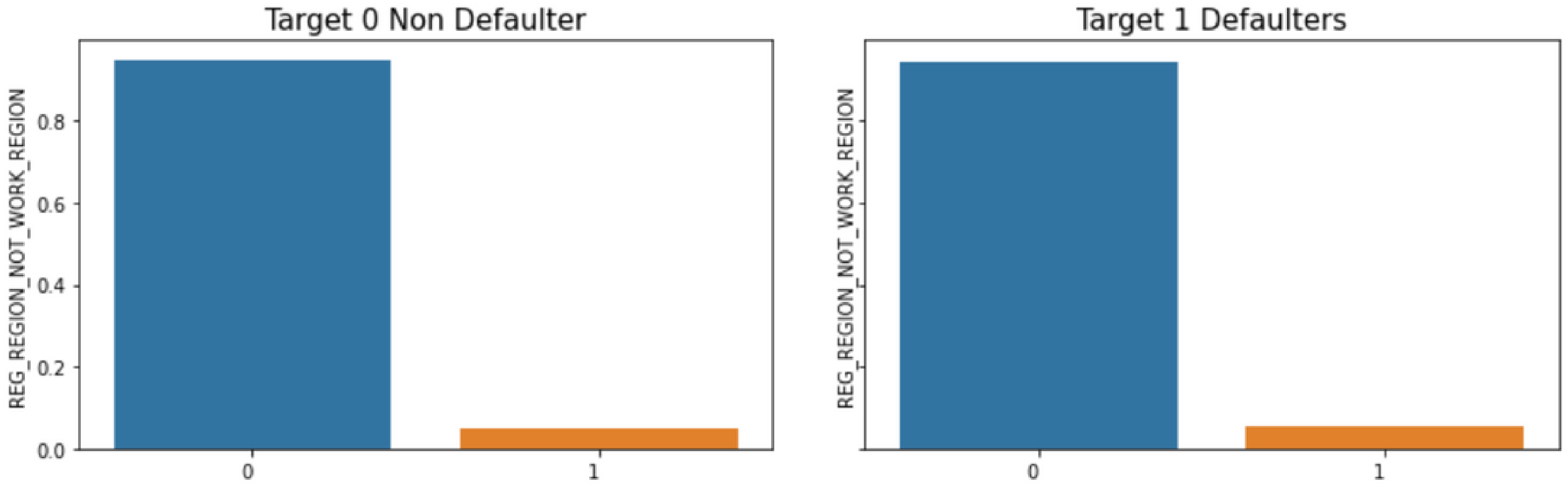
REGION_RATING_CLIENT_W_CITY



REG_REGION_NOT_LIVE_REGION



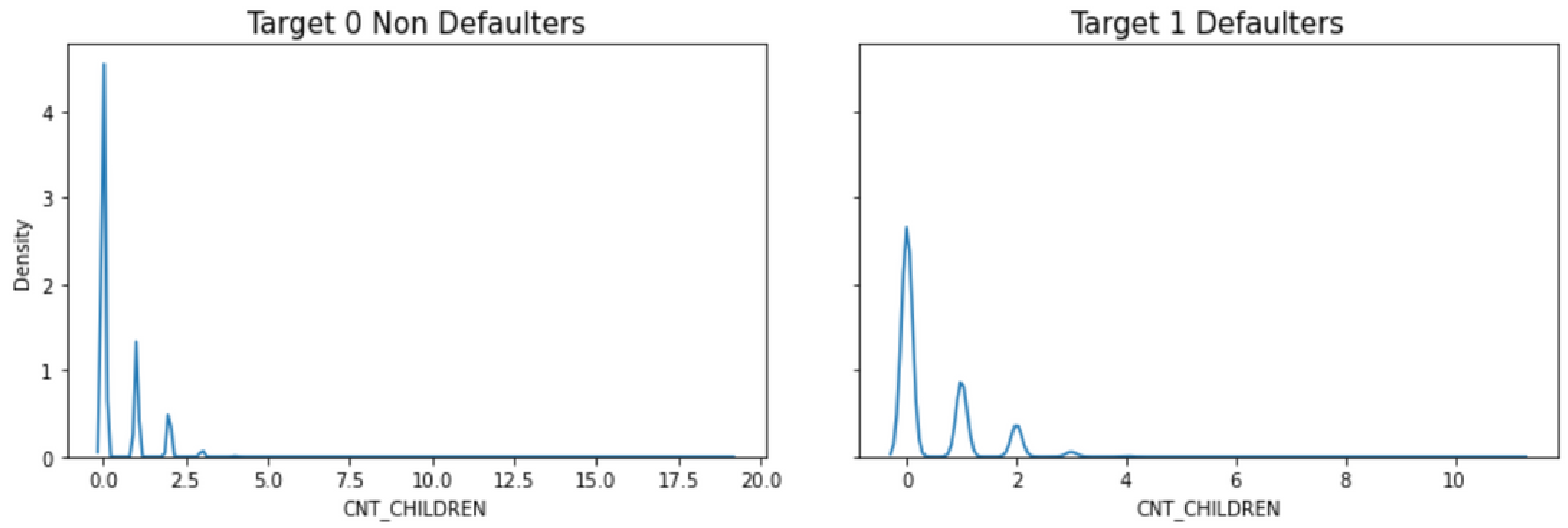
REG_REGION_NOT_WORK_REGION



...and more

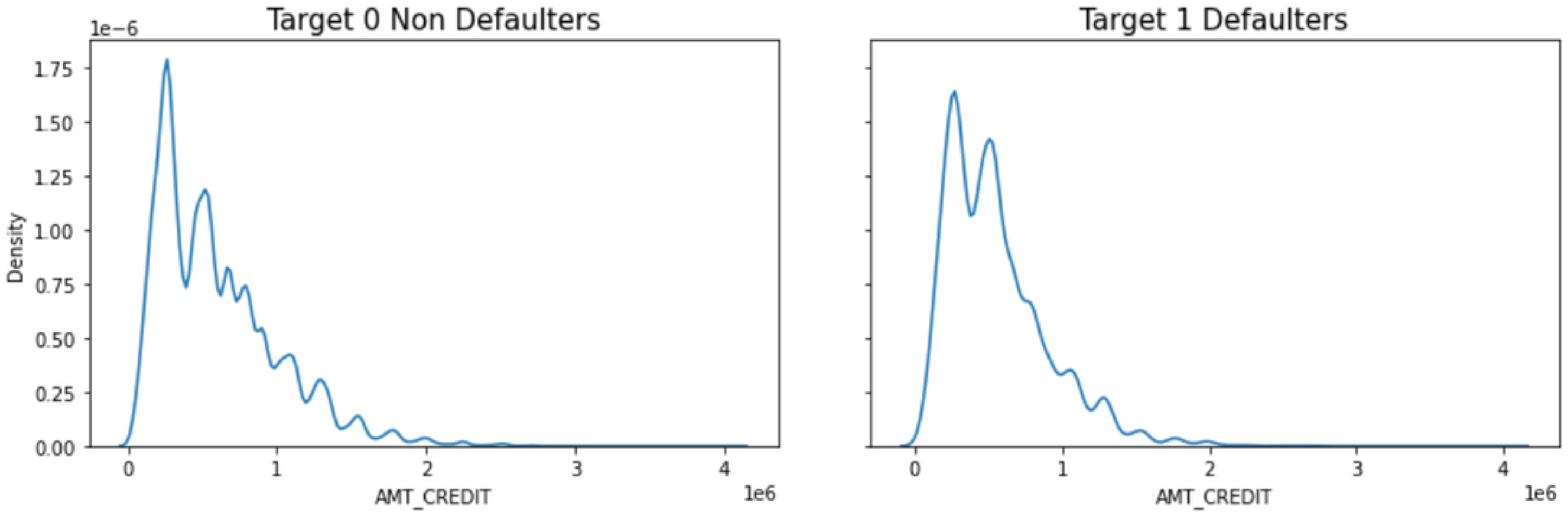
Numerical List

CNT_CHILDREN


Target 0 Non Defaulters

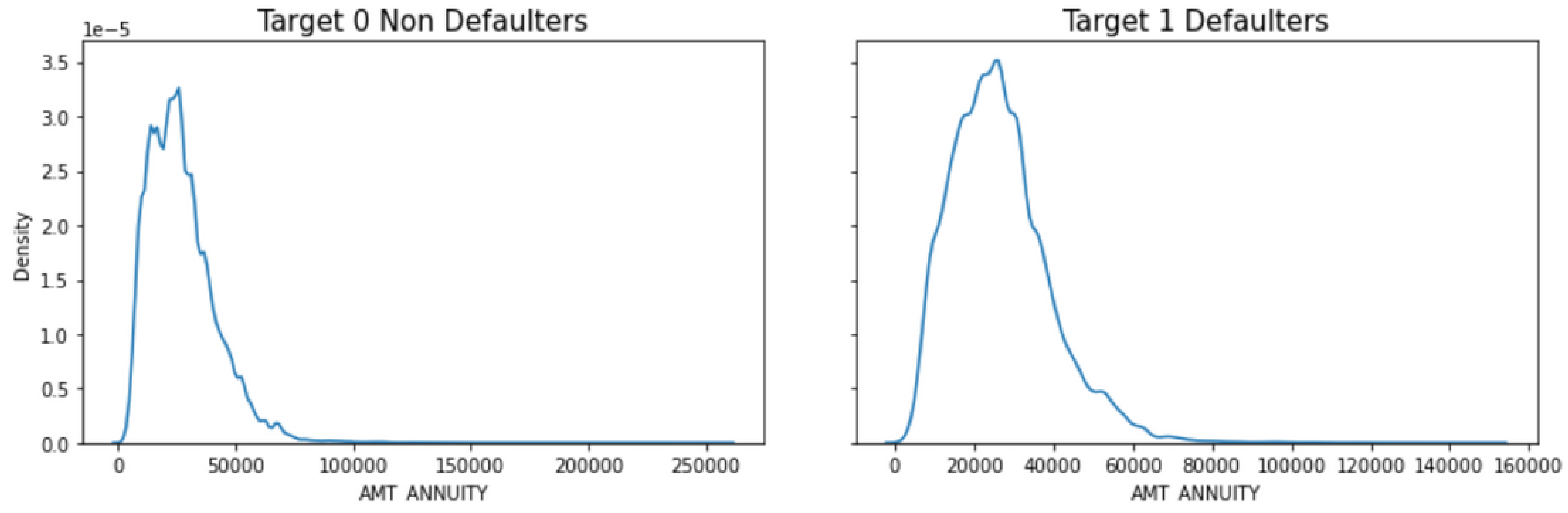
Target 1 Defaulters

AMT_CREDIT


Target 0 Non Defaulters

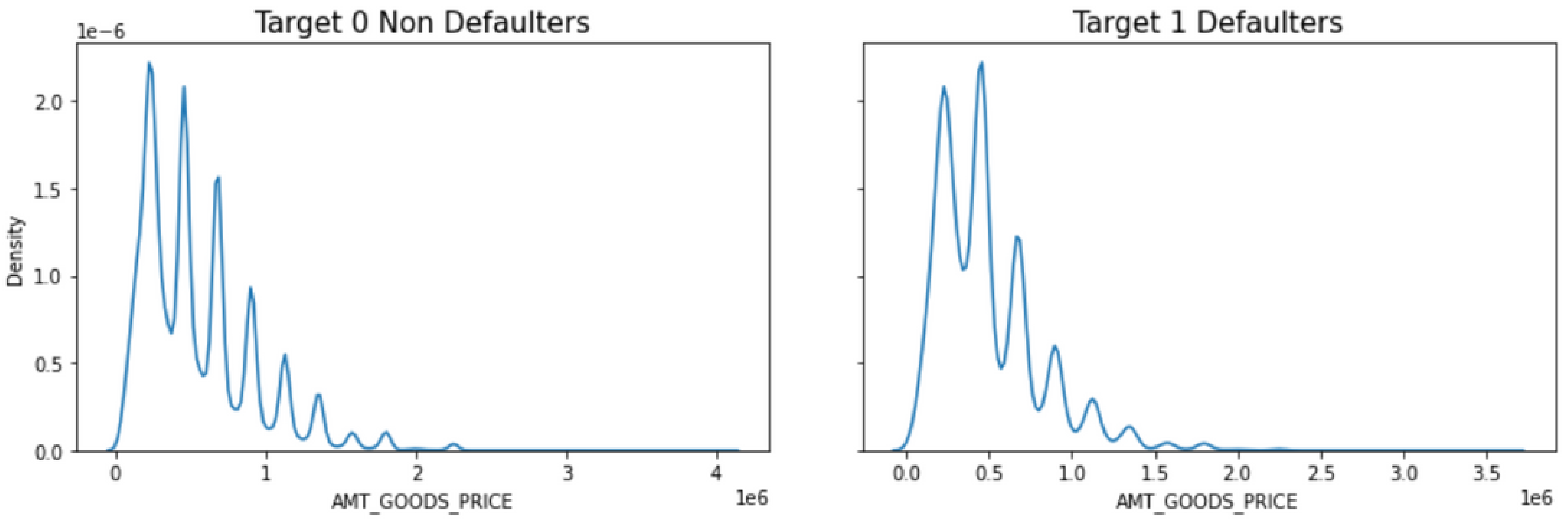
Target 1 Defaulters

AMT_ANNUITY


Target 0 Non Defaulters

Target 1 Defaulters

AMT_GOODS_PRICE

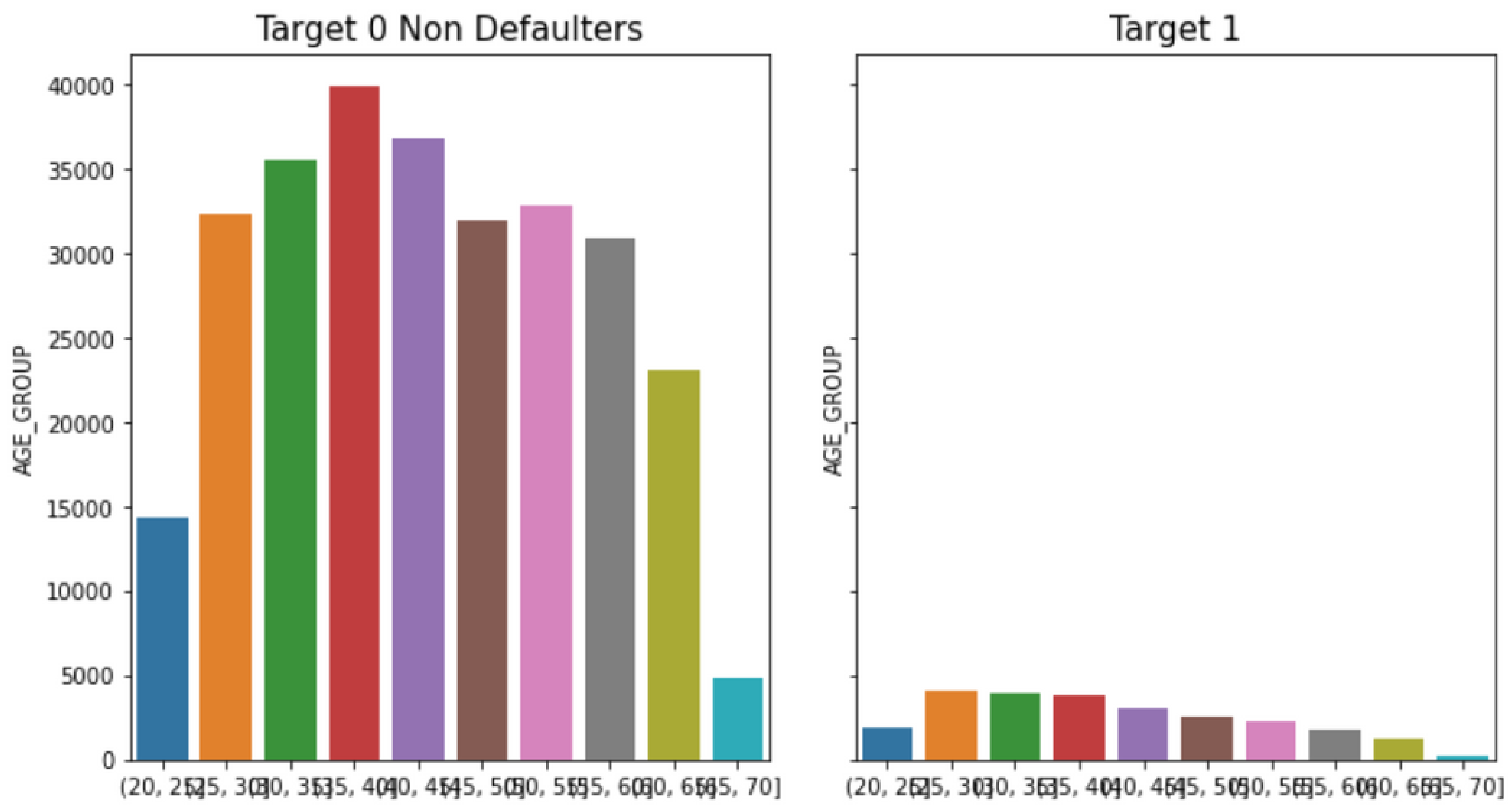

Target 0 Non Defaulters

Target 1 Defaulters

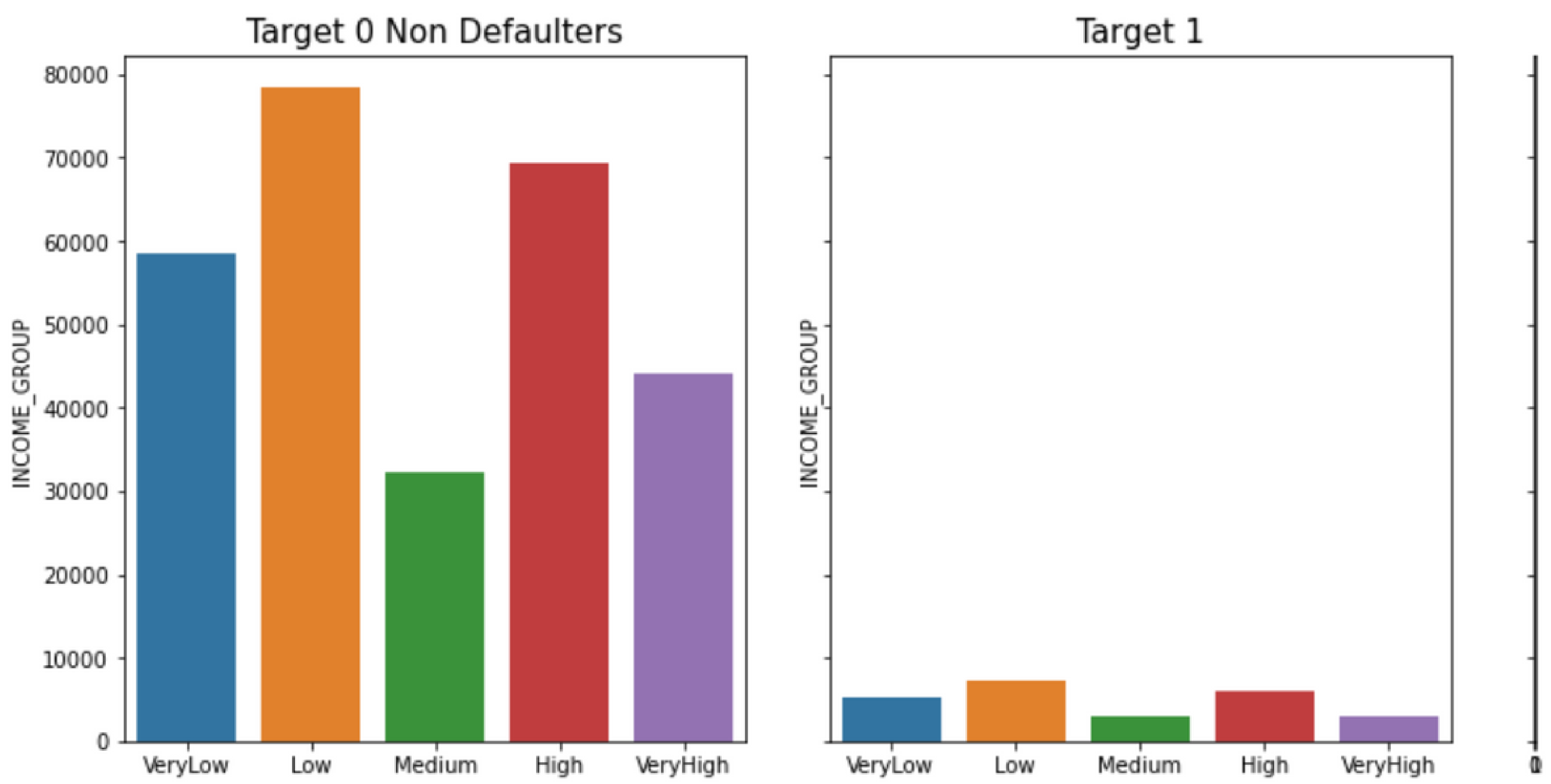
...and more

We do the same for binned categories as they were removed from the list(s) created

AGE_GROUP



INCOME_GROUP



- From the Object list analysis we notice :
    - NAME_EDUCATION_TYPE - Secondary Education applicants have submitted more loan applications than other applicants in both Target 0 and Target 1.The majority of applicants with secondary income have defaulted payments. Needs more investigation

- NAME_CONTRACT_TYPE - The majority of the company's portfolio consists of Cash Loans. 85% for Target 0 and nearly 95% for Target 1
- NAME_INCOME_TYPE - Working income types are 50% working in the scenario of Target 0 and 60% working in the case of Target 1.
- NAME_FAMILY_STATUS - Nearly 60% of married applicants have fallen behind (defaulted) on their payments
- NAME_HOSUING_TYPE - In Target 0 and Target 1 applications, "House/apartment" is where 85–90% of them are staying. proving that this is not a factor that can affect a payment default.
- OCCUPATION_TYPE - 50% of defaulters are drivers, core employees, salespeople, and laborers. The biggest percentage of applicants are also laborers.
- ORGANIZATION_TYPE - 40% of defaulters are self-employed individuals and business entities of type 3. This category also includes those who take out the most loans.

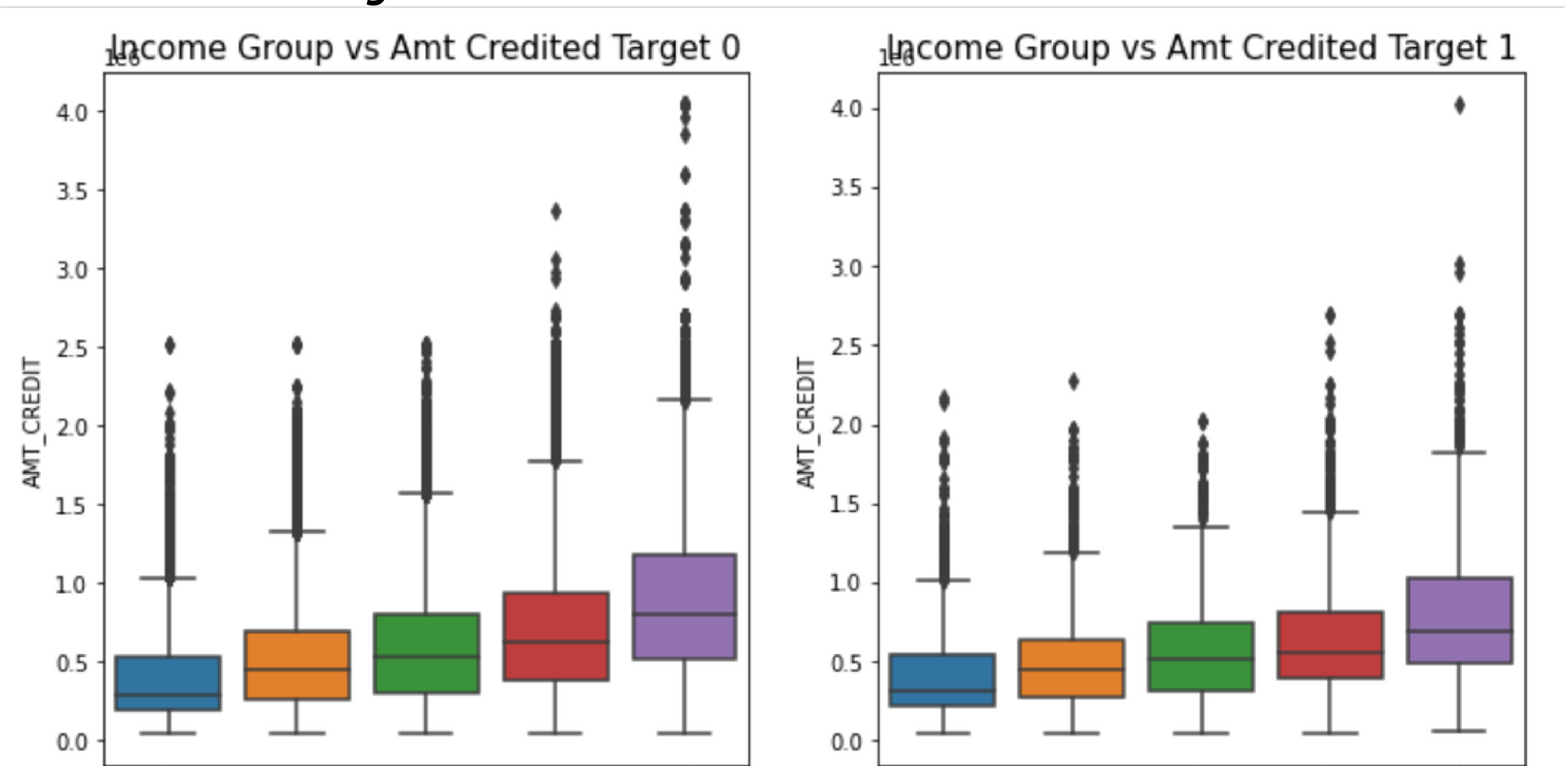
- for the categorical list we analysed that:
    - REG_CITY_NOT_LIVE_CITY, REG_CITY_NOT_WORK_CITY -For 1, the default ratio is greater, making it distinct from a permanent address.
    - REG_REGION_NOT_LIVE_REGION, REGION_NOT_WORK_REGION, VE_REGION_NOT_WORK_REGION - Target 1 is quite low and doesn't appear to have an impact on the default rate for both Target 0 and Target 1 out of Region.
    - CODE_GENDER - F to M in Target 0 is a ratio of 2.3, while M to F is a ratio of 1.3. showing that MEN default more often than Women

- for the numerical list it is analysed that:
    - AGE_IN_YEARS - Target 1 has a higher 30-year density, showing that younger people default more.
    - YEARS_EMPLOYED - has a huge number of rows with inaccurate data, hence the data representation is incorrect
    - EXT_SOURCE_2 - is obvious that TARGET 0 has a higher density of higher scores
    - TARGET 1 have less number of cars.

- for binned category we analysed that:
    - AGE_GROUP - 35-45 are more common in tARGET 0. Target 1- 25-25 have a larger proportion. Age appears to influence default.
    - INCOME_GROUP - The medium income category has a higher representation in Targets 0 and 1.

Bivariate Analysis

- we first do bivariate analysis for INCOME_GROUP and AMT_CREDIT

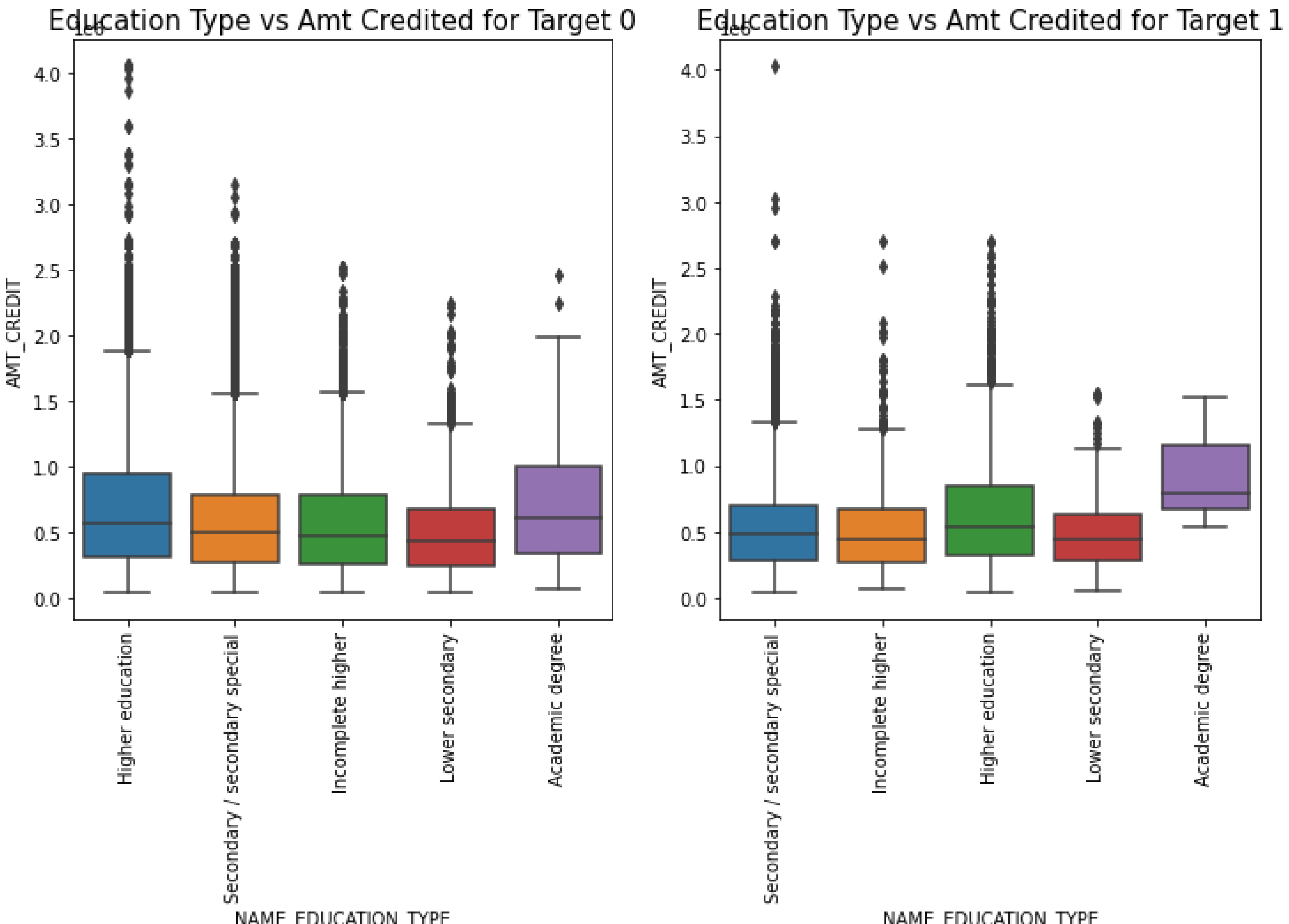


Income Group vs Amt Credited Target 0

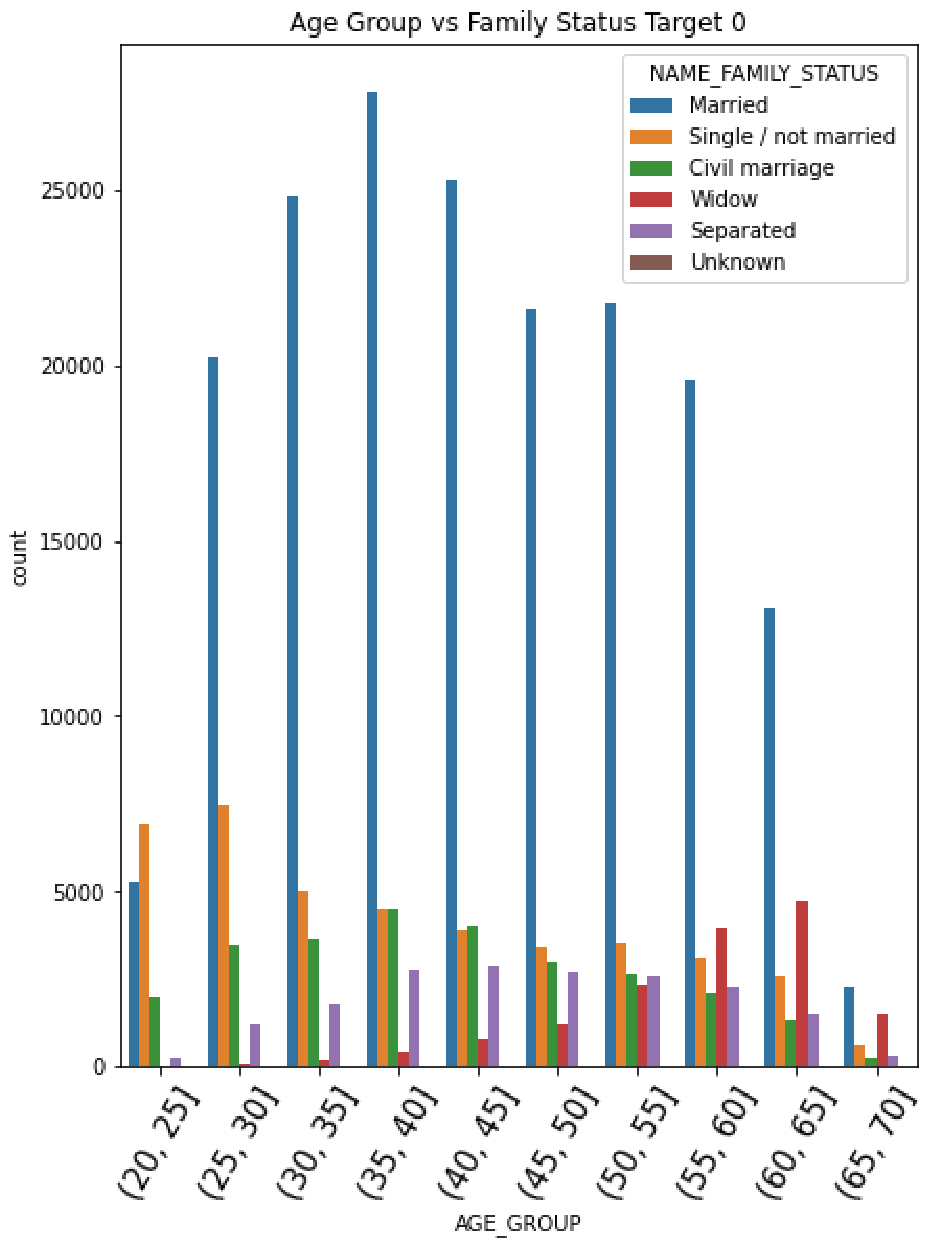
Income Group vs Amt Credited Target 1

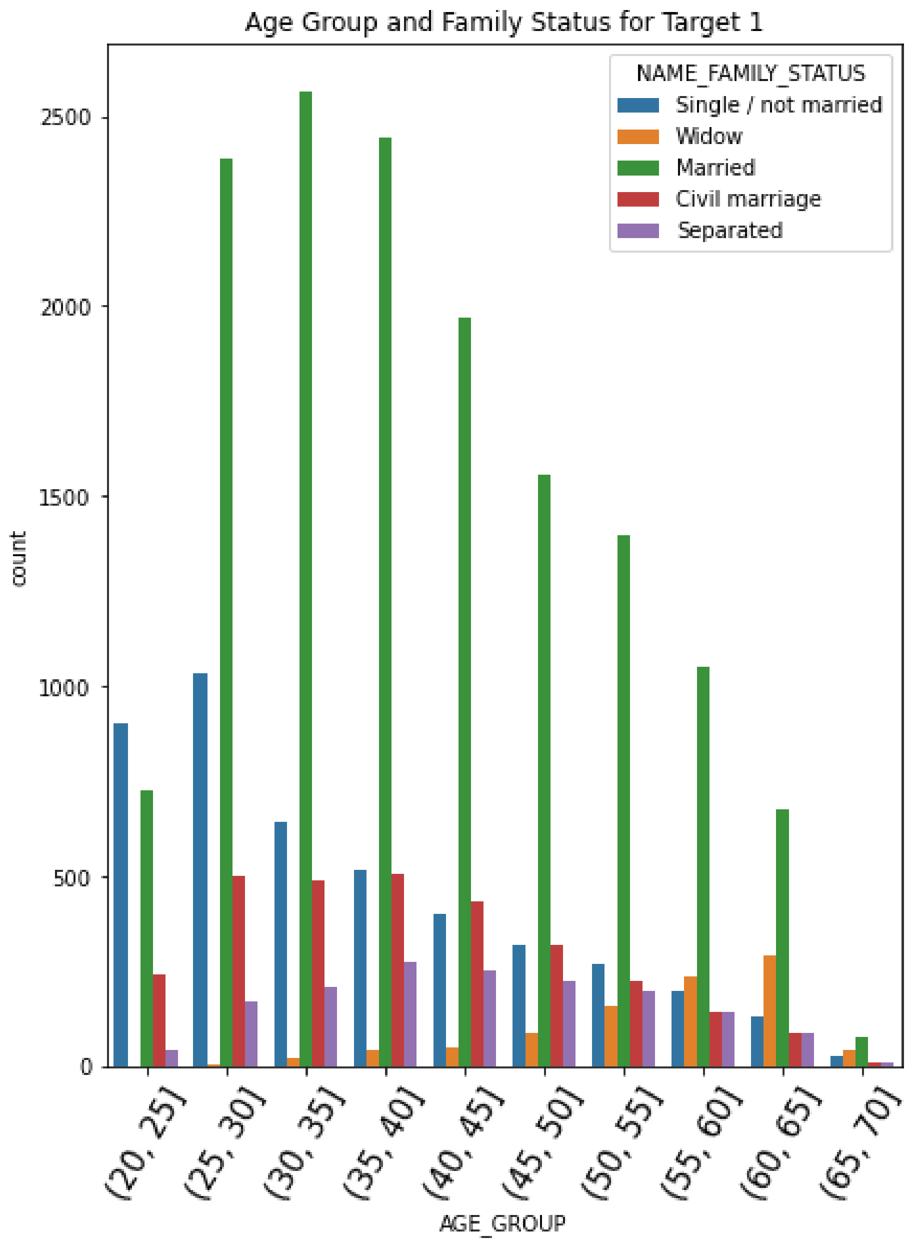
- here we analyse
    - We might conclude that, despite the fact that the medium income group receives the greatest number of loans. The default value per loan is largest in the High income group, as is the AMT_CREDIT. The financial institution's loan book may suffer as a result of a larger sum not being paid back.

- Bivariate analysis on EDUCATION_TYPE and AMT_CREDIT



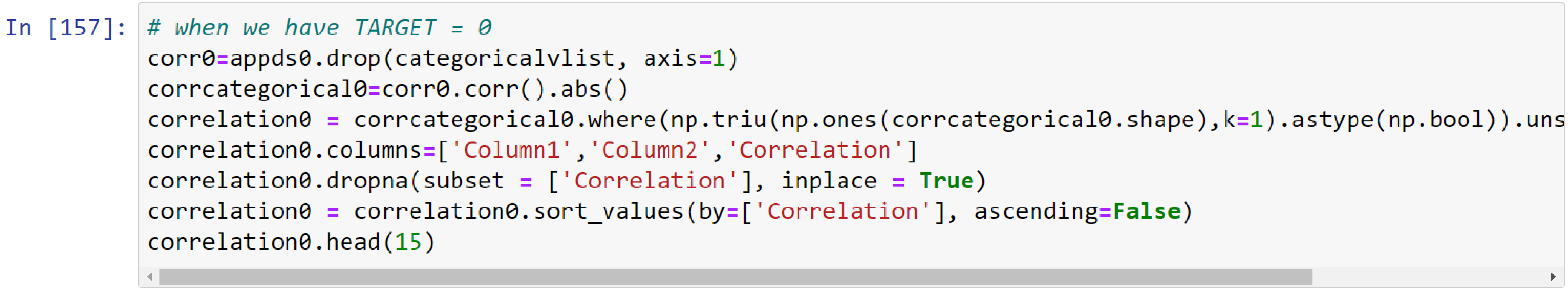
- bivariate analysis on AGE_GROUP and FAMILY_STATUS for TARGET



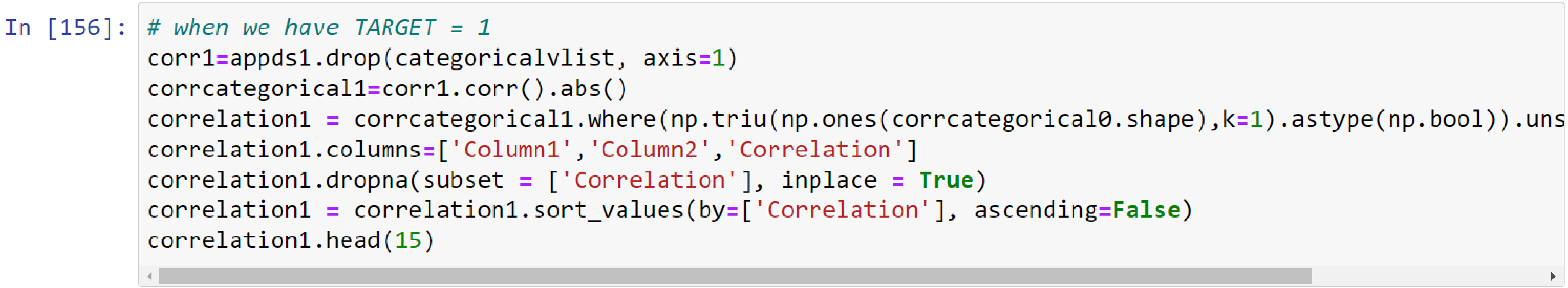


- here we can say that :
  - The greatest group of applicants experiencing payment difficulties are married applicants aged 25-35 and 35-45.

Correlation

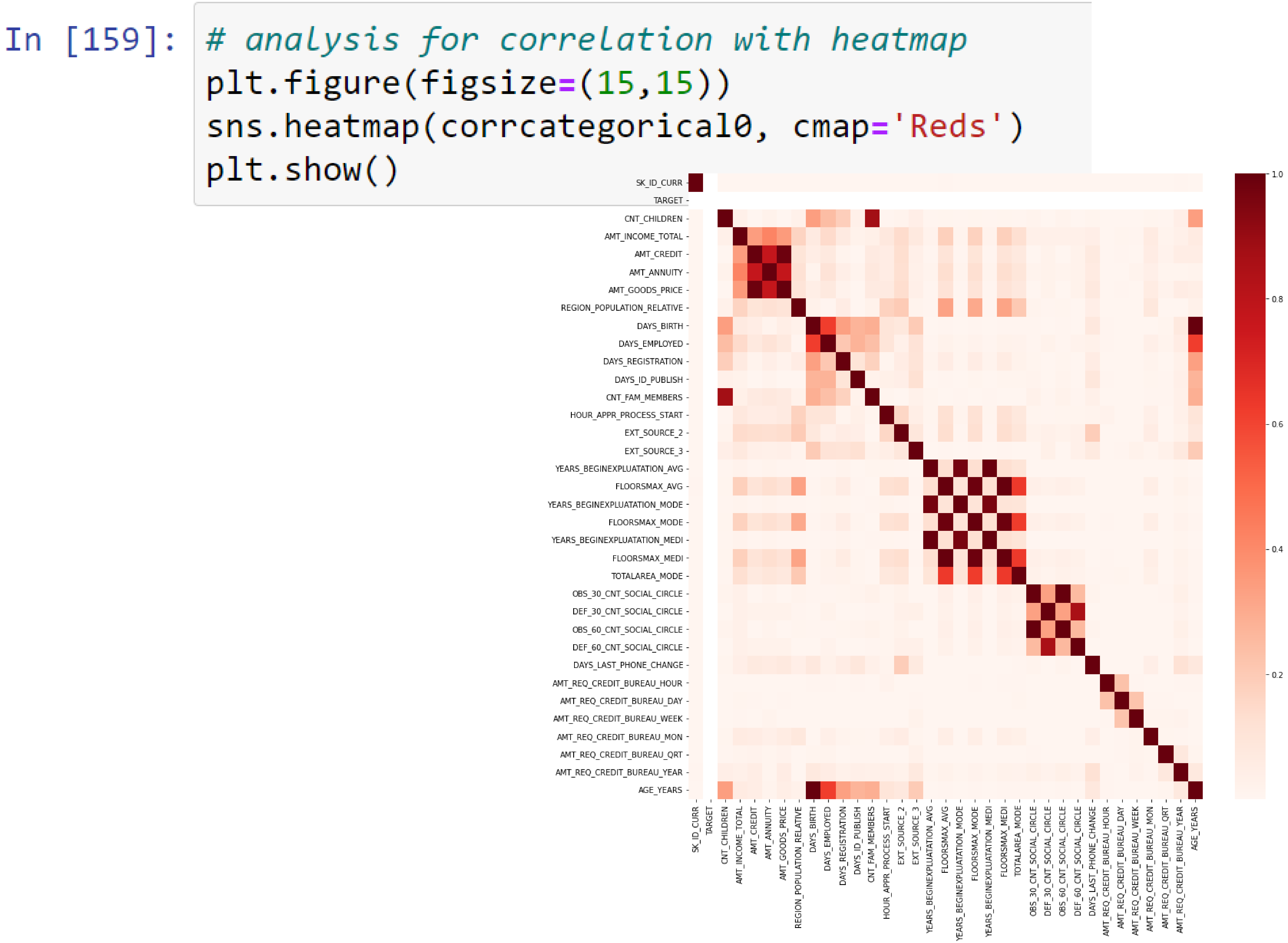


```
In [157]: # when we have TARGET = 0
          corr0=appds0.drop(categoricalvlist, axis=1)
          corrcategorical0=corr0.corr().abs()
          correlation0 = corrcategorical0.where(np.triu(np.ones(corrcategorical0.shape),k=1).astype(np.bool)).uns
          correlation0.columns=['Column1','Column2','Correlation']
          correlation0.dropna(subset = ['Correlation'], inplace = True)
          correlation0 = correlation0.sort_values(by=['Correlation'], ascending=False)
          correlation0.head(15)
```



```
In [156]: # when we have TARGET = 1
          corr1=appds1.drop(categoricalvlist, axis=1)
          corrcategorical1=corr1.corr().abs()
          correlation1 = corrcategorical1.where(np.triu(np.ones(corrcategorical0.shape),k=1).astype(np.bool)).uns
          correlation1.columns=['Column1','Column2','Correlation']
          correlation1.dropna(subset = ['Correlation'], inplace = True)
          correlation1 = correlation1.sort_values(by=['Correlation'], ascending=False)
          correlation1.head(15)
```

- this gives us the top 15 correlation for TARGET as 0 and TARGET as 1
- for a better understanding we plot a heatmap



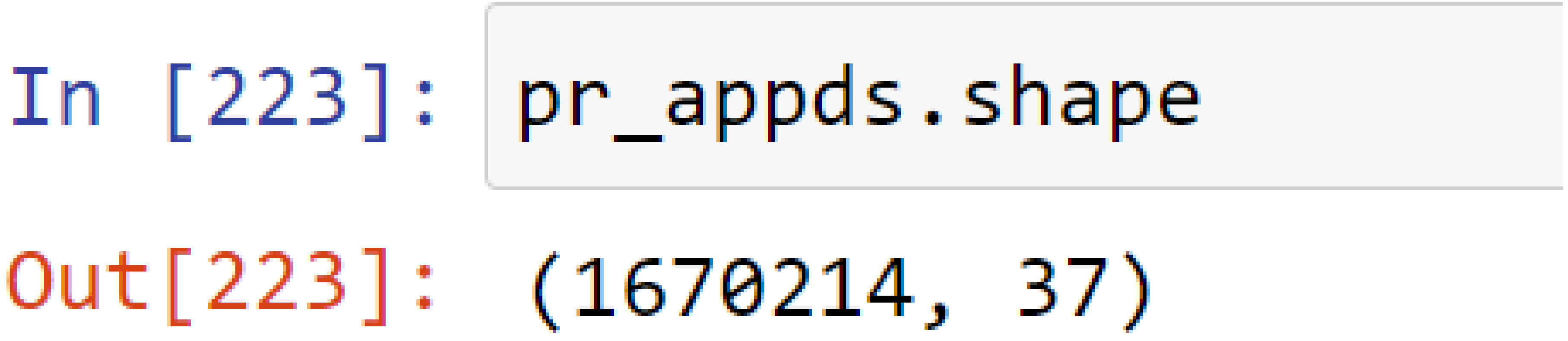
```
In [159]: # analysis for correlation with heatmap
          plt.figure(figsize=(15,15))
          sns.heatmap(corrcategorical0, cmap='Reds')
          plt.show()
```

- from the heat map we can conclude that:
    - AMT_CREDIT and AMT_GOODS_PRICE
    - CNT_FAM_MEMBERS and CNT_CHILDREN
    - AMT_ANNUITY and AMT_CREDIT

  all have good correlation

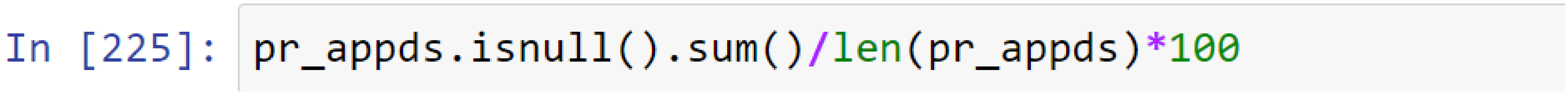
Previous Application

- Here we take *previous_application.csv* for our understanding of the data and to perform a better analysis . This data gives us knowledge of the previous loans given and what was the outcome of the same.
- We do the same as in *application_data.csv* , we load and read the data and similarly handle the null values and outliers.



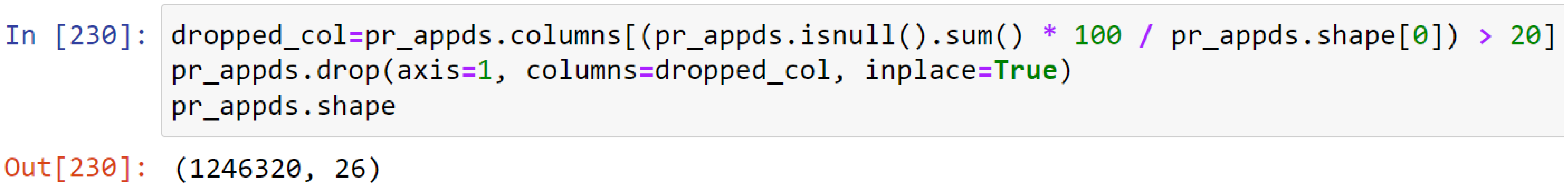
```
In [223]: pr_appds.shape

Out[223]: (1670214, 37)
```

- Here we can see that the data has 1670214 rows and 37 columns
- In the same way now we check for the percentage null values



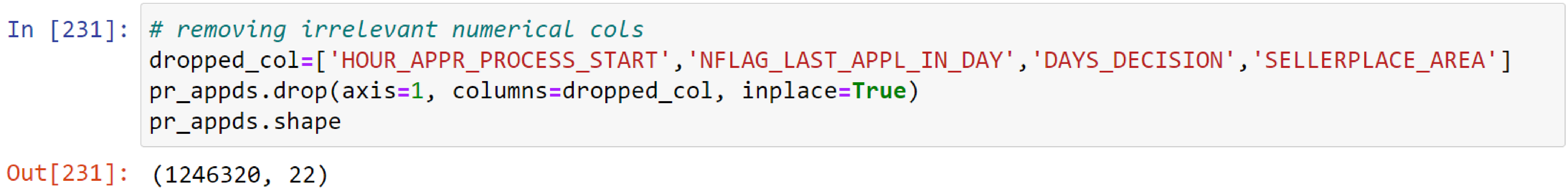
```
In [225]: pr_appds.isnull().sum()/len(pr_appds)*100
```

- on doing that we first drop the null values rows from AMT_ANNUITY and AMT_GOOD_PRICE , we don't remove the entire columns as they are further needed to for analysis
- Now we remove the cols where null value is >20%



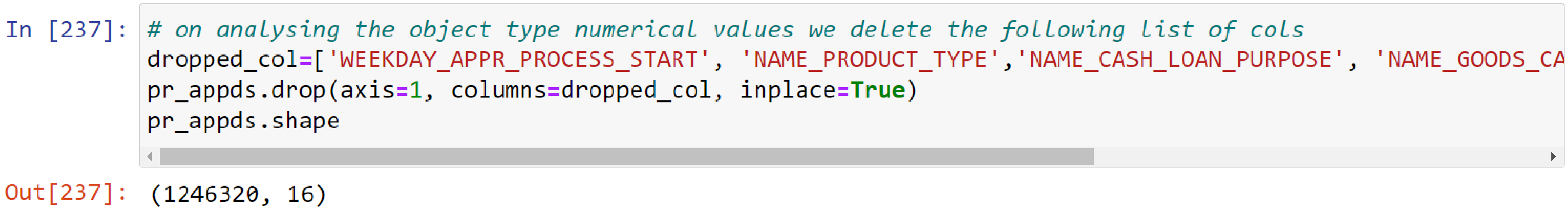
```
In [230]: dropped_col=pr_appds.columns[(pr_appds.isnull().sum() * 100 / pr_appds.shape[0]) > 20]
          pr_appds.drop(axis=1, columns=dropped_col, inplace=True)
          pr_appds.shape

Out[230]: (1246320, 26)
```

- now we also remove the irrelavant numerical columns which won't be used for analysis



```
In [231]: # removing irrelevant numerical cols
          dropped_col=['HOUR_APPR_PROCESS_START','NFLAG_LAST_APPL_IN_DAY','DAYS_DECISION','SELLERPLACE_AREA']
          pr_appds.drop(axis=1, columns=dropped_col, inplace=True)
          pr_appds.shape

Out[231]: (1246320, 22)
```

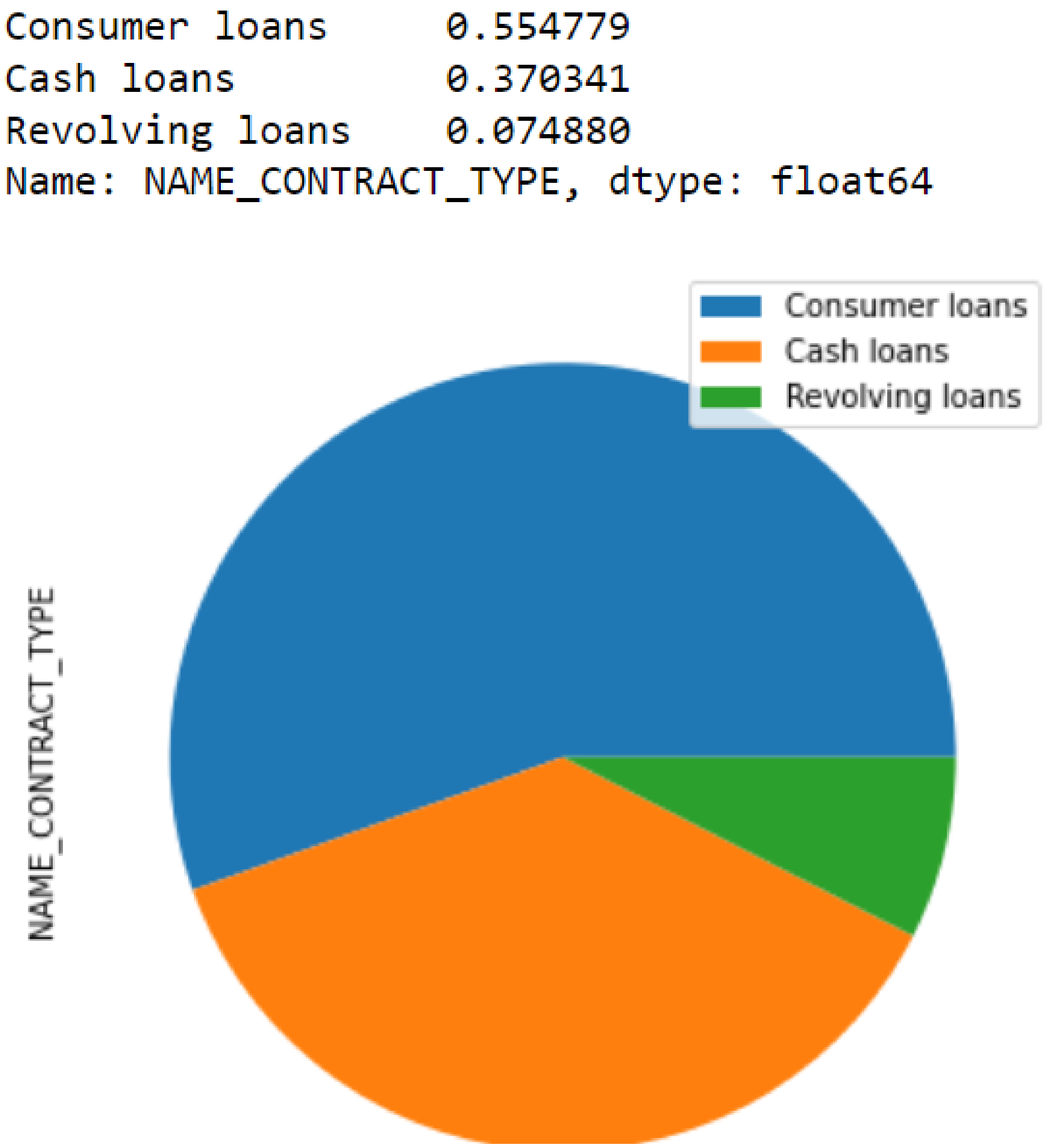
- after removing the irrelevant and null value cols we are left with 1246320 rows and 22 cols.
- now we perform analysis on object type variables and remove the irrelevant cols from the object type after doing so we are left with 1246320 rows and 16 cols on which we perform univariate and bivariate analysis.



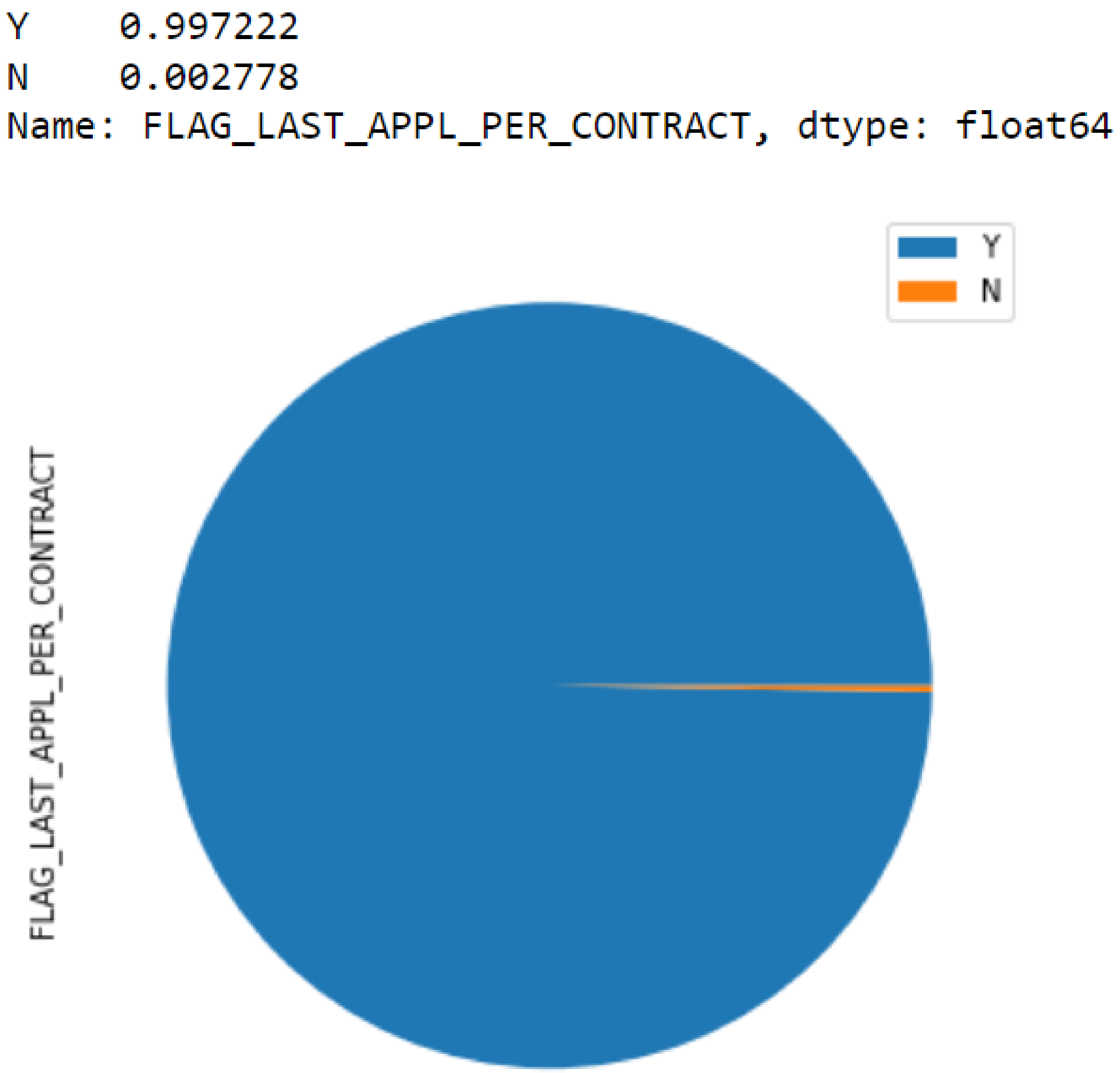
```
In [237]: # on analysing the object type numerical values we delete the following list of cols
          dropped_col=['WEEKDAY_APPR_PROCESS_START', 'NAME_PRODUCT_TYPE','NAME_CASH_LOAN_PURPOSE', 'NAME_GOODS_CA
          pr_appds.drop(axis=1, columns=dropped_col, inplace=True)
          pr_appds.shape

Out[237]: (1246320, 16)
```

Univariate Analysis

- we perform univariate analysis using pie-chart on all object type variables and get this:

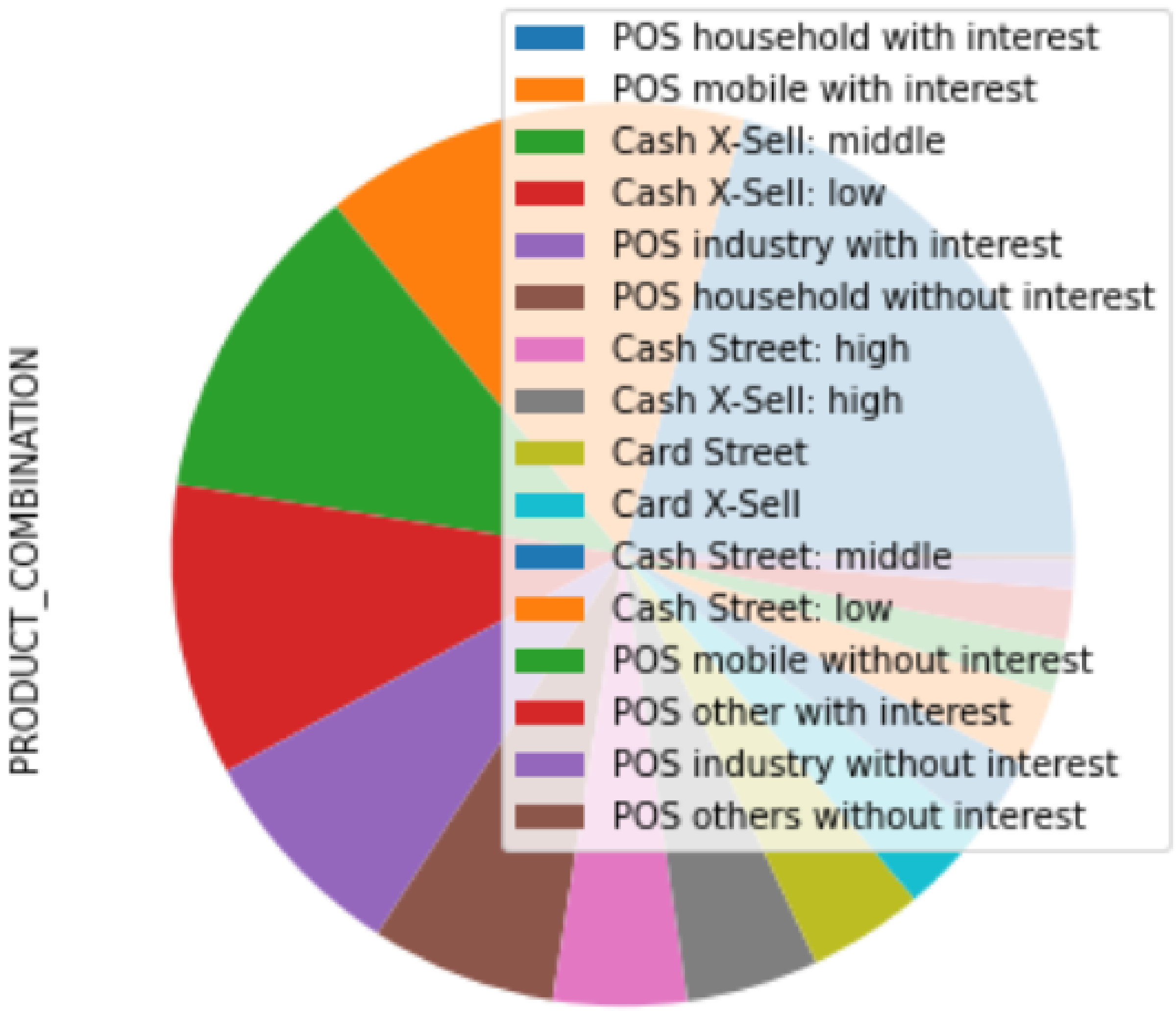


```
Consumer loans     0.554779
Cash loans         0.370341
Revolving loans    0.074880
Name: NAME_CONTRACT_TYPE, dtype: float64
```



```
Y    0.997222
N    0.002778
Name: FLAG_LAST_APPL_PER_CONTRACT, dtype: float64
```

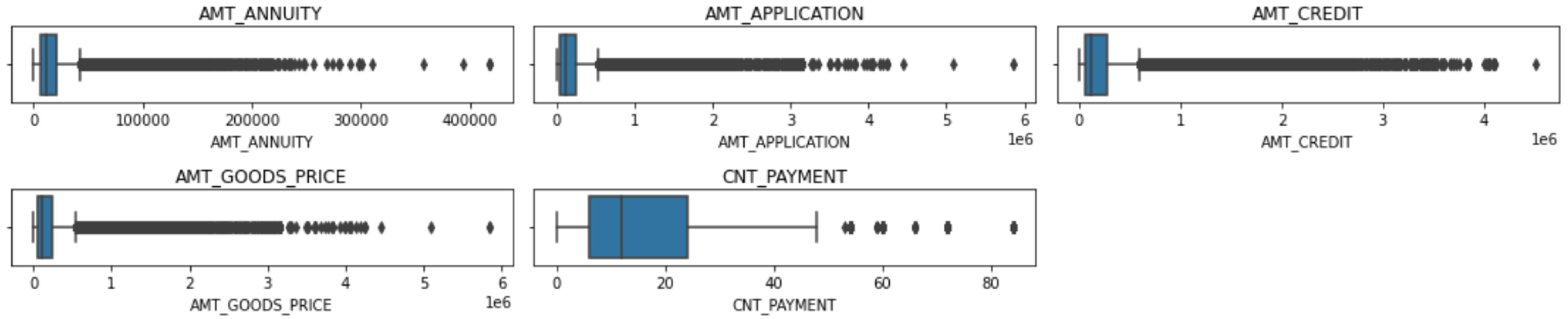
```
POS others without interest        0.002036
Name: PRODUCT_COMBINATION, dtype: float64
```



...and more

- from here we analyse that:
    - The name seller industry has 37% XNA values, with consumer electronics coming in second at 30%.
    - There is a different form of loan in this dataframe called a consumer loan that wasn't there in the application data frame. Consumer loans account for 55% of all lending. Recurrent loans made up the majority of borrowing at 37%.

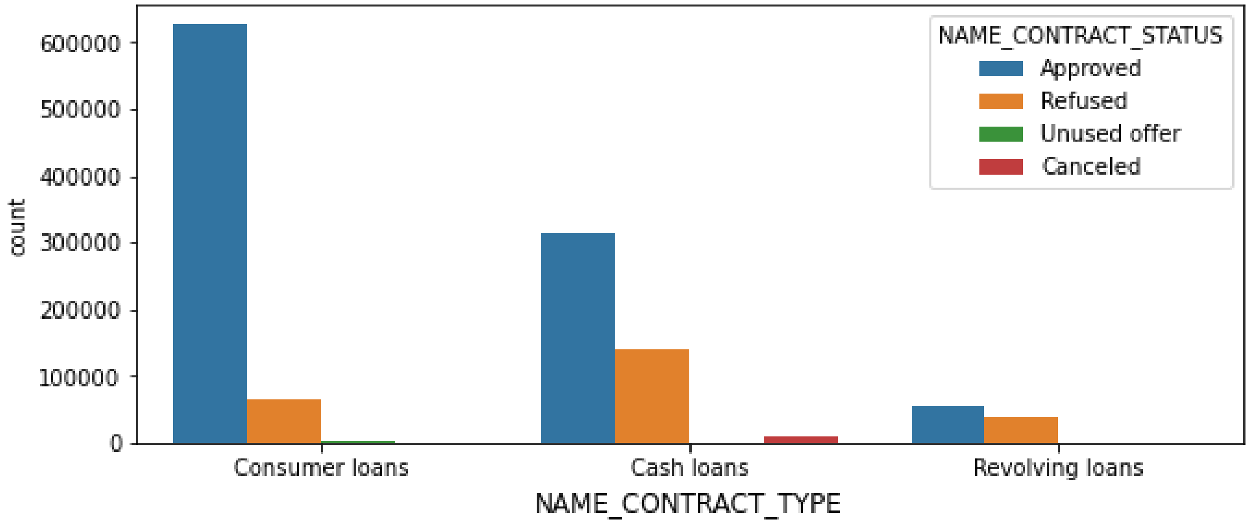
- univariate analysis where data type is float

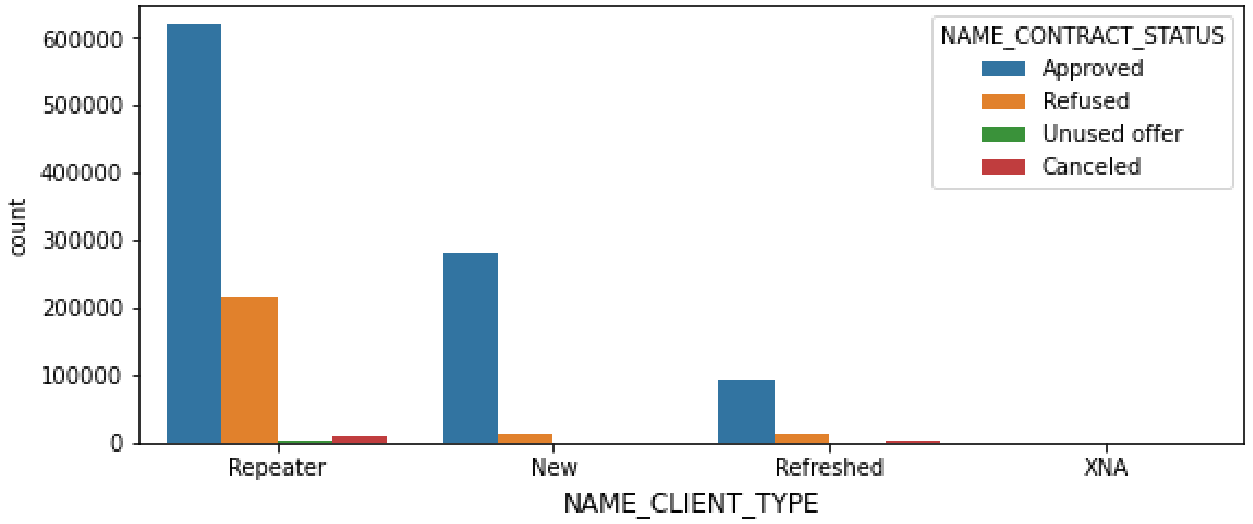


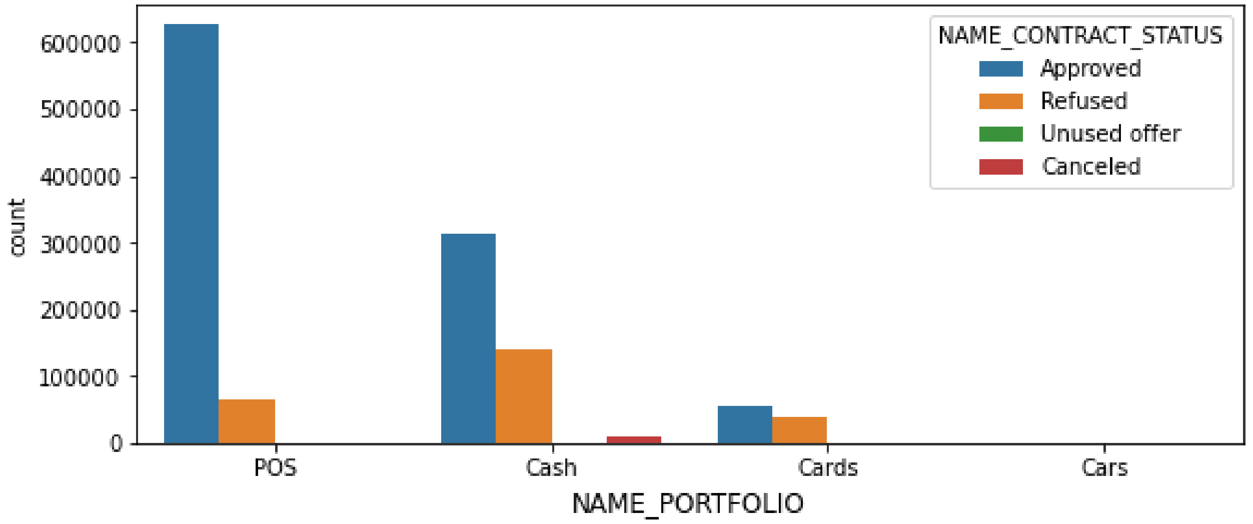
- the percentage of outliers is higher in the numerical variable

Bivariate Analysis

- for this we create a list having bivariate variables
    - bivar_cols=['NAME_CONTRACT_TYPE','NAME_CLIENT_TYPE','NAME_PORTFOLIO']



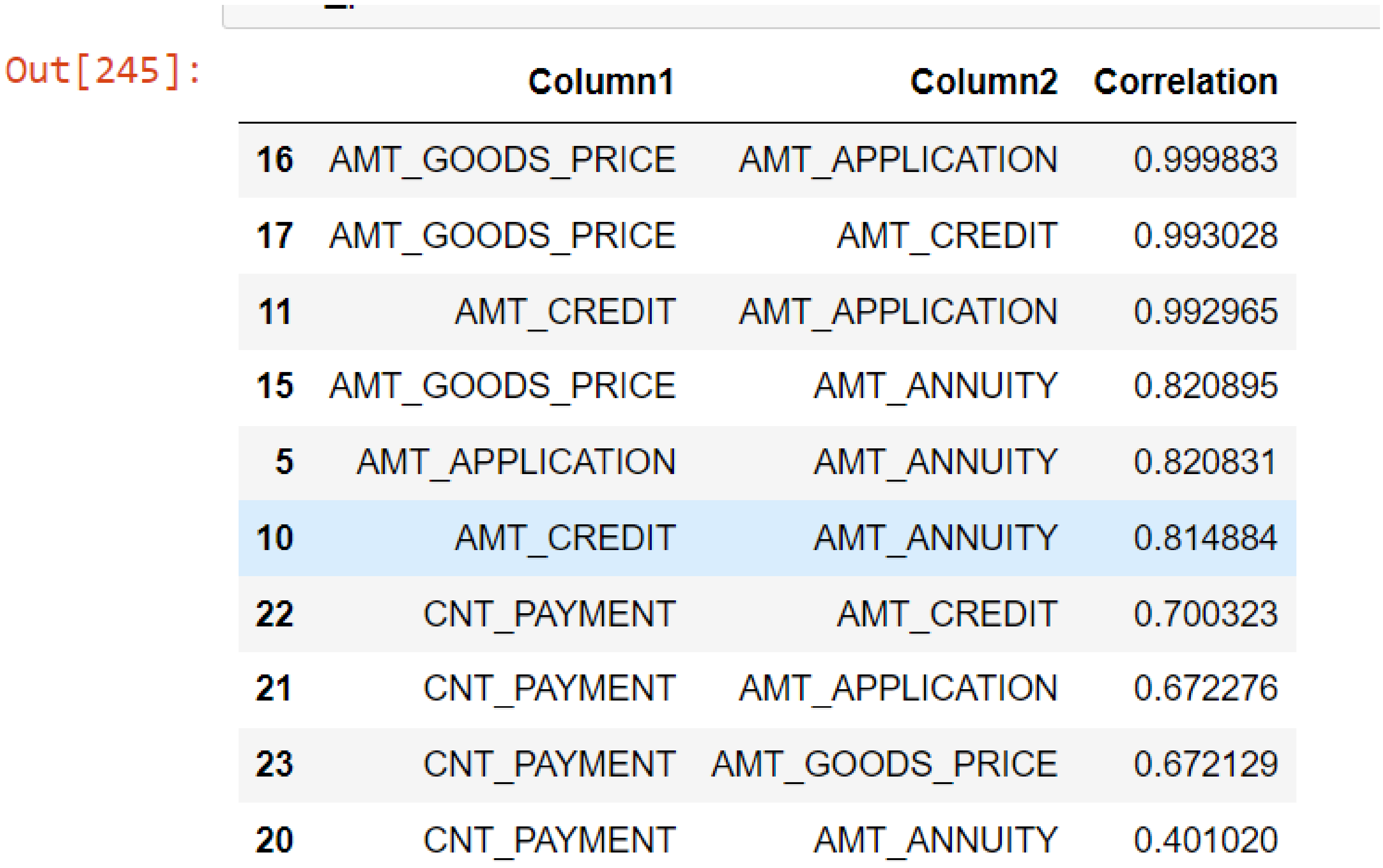




- we observer that
  - In every authorized, rejected, underused, and cancelled category, the bank has additional repeaters.
  - Compared to consumer loans, more cash loans have been turned down.
  - In contrast to consumer loans, it appears that there have been no revoked cash loans.
  - More cash loans have been turned down than POS transactions, which appear to be consumer loans comparable to point 2.

Correlation

- for this we ignore SK_ID_CURR and SK_ID_PREV cols and perform check correlation

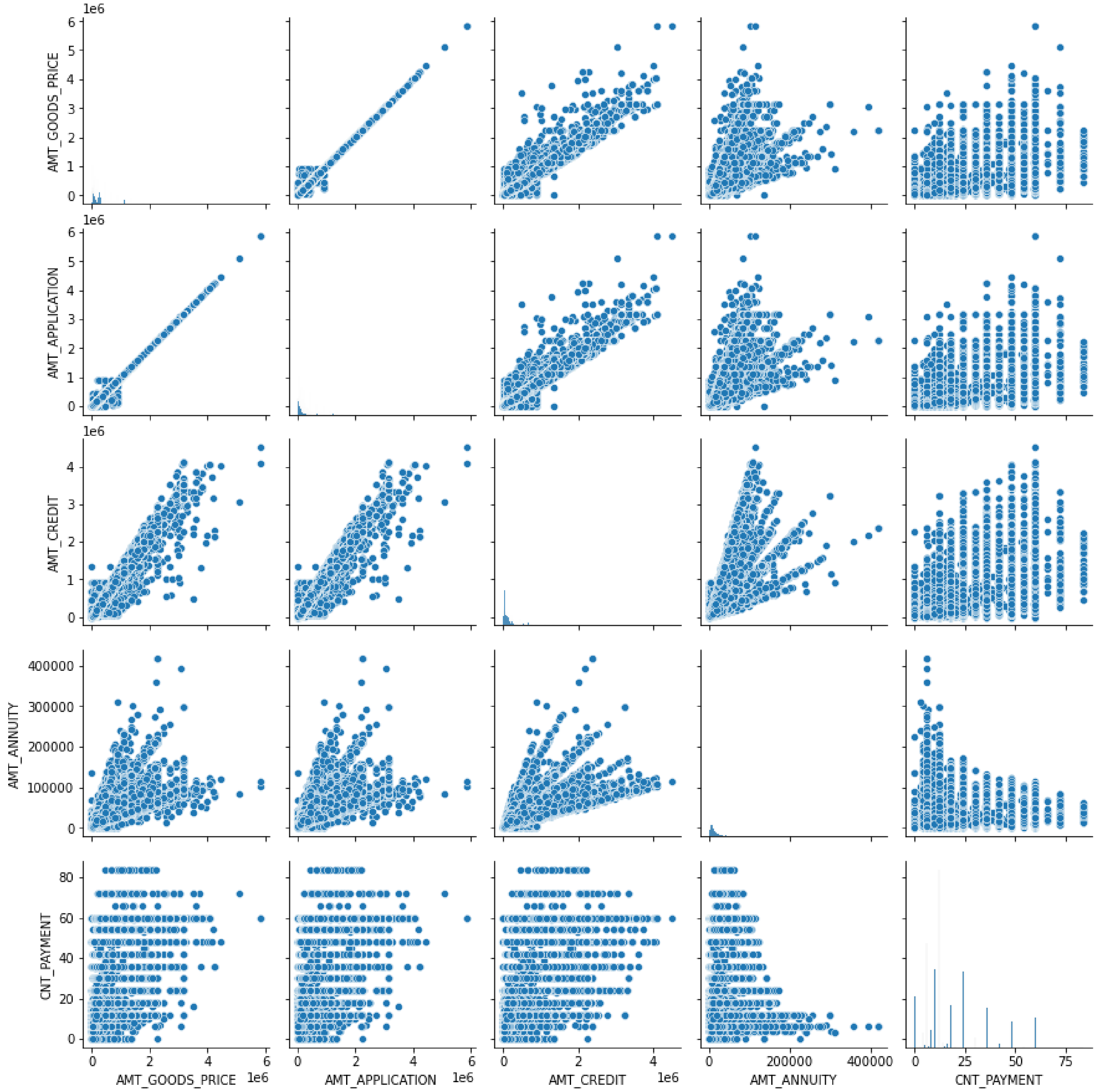


Out[245]:

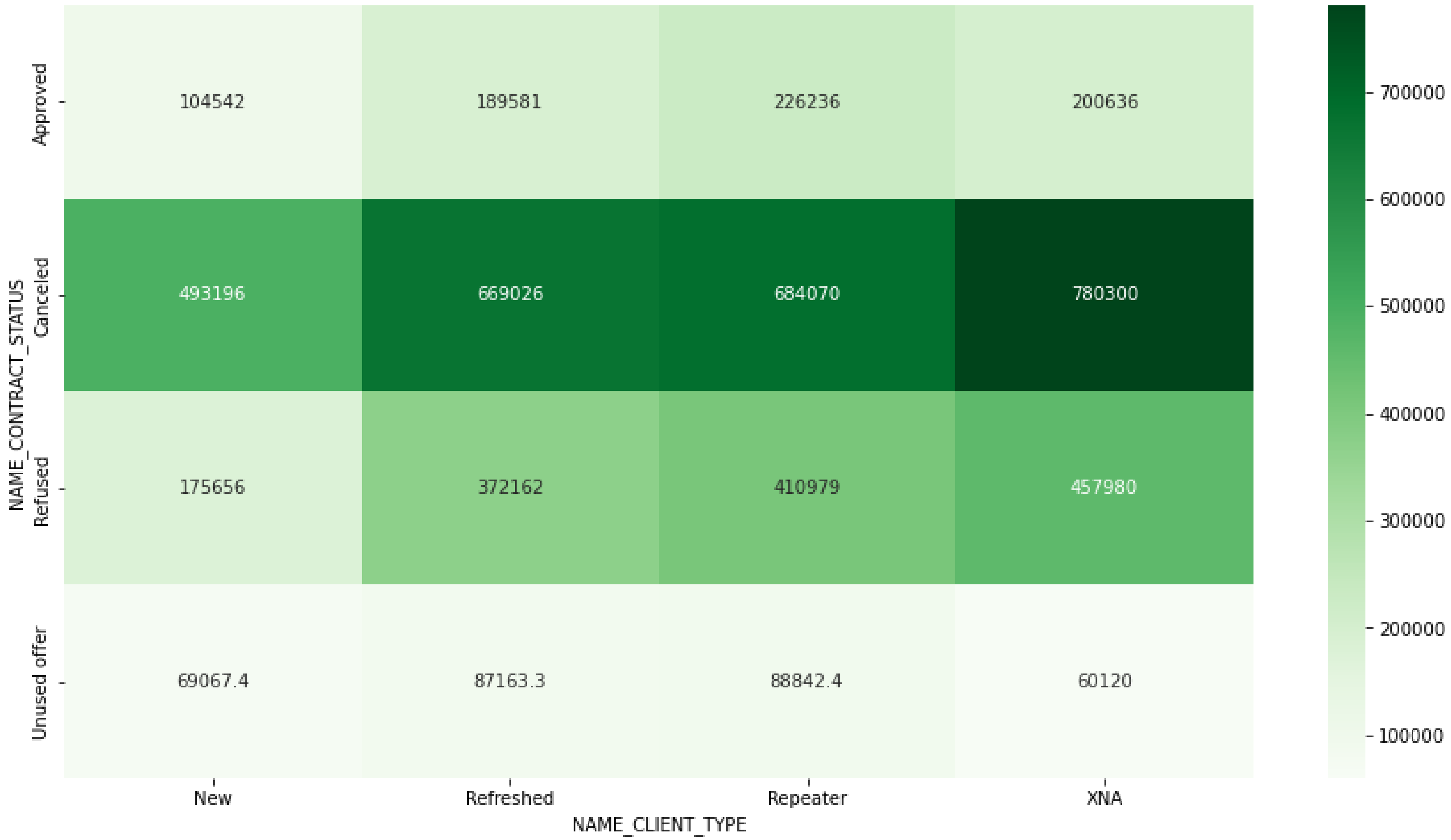
	Column1	Column2	Correlation
16	AMT_GOODS_PRICE	AMT_APPLICATION	0.999883
17	AMT_GOODS_PRICE	AMT_CREDIT	0.993028
11	AMT_CREDIT	AMT_APPLICATION	0.992965
15	AMT_GOODS_PRICE	AMT_ANNUITY	0.820895
5	AMT_APPLICATION	AMT_ANNUITY	0.820831
10	AMT_CREDIT	AMT_ANNUITY	0.814884
22	CNT_PAYMENT	AMT_CREDIT	0.700323
21	CNT_PAYMENT	AMT_APPLICATION	0.672276
23	CNT_PAYMENT	AMT_GOODS_PRICE	0.672129
20	CNT_PAYMENT	AMT_ANNUITY	0.401020

- correlation is observed in these cols
  - ['AMT_GOODS_PRICE','AMT_APPLICATION','AMT_CREDIT','AMT_ANNUITY','CNT_PAYMENT']

- on these cols we create a pairplot for correlation understanding
- from this we observe that:
  - AMT_GOODS_PRICE, AMT_ANNUITY, and AMT_APPLICATION all have high correlations as would be predicted. Loan requirements will increase as the value of the good being purchased increases, and this trend will undoubtedly continue.
  - Ideal would have been for column CNT_Payment and AMT_credit to be highly correlated, meaning that the longer the loan duration, the bigger the credit. However, no such correlation is apparent.
  - AMT_Credit to AMT_GOOD_PRICE is comparable. The correlation is strong as well.



Multivariate Analysis



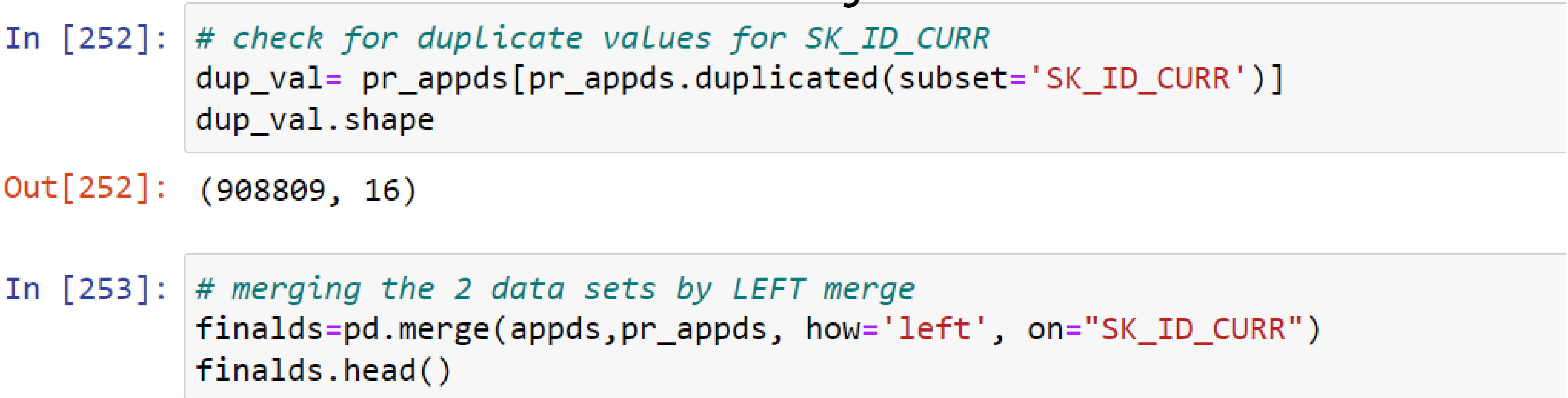
- we get this heatmap when we do analysis over NAME_CONTRACT_STATUS, NAME_CLIENT_TYPE and AMT_APPLICATION
- we observe that:
    - Low application amount for an unused offer

- on plotting the same for NAME_CONTRACT_STATUS,NAME_CLIENT_TYPE and AMT_CREDIT , we observe that:
    - Rejected offer Low credit amount. This could be the cause of a customer's failure to use

- and similarly with AMT_GOODS_PRICE, we observe that:
    - All rejected and cancelled cases have commodities with a higher value than other categories.

Merging two datasets

- we merge the 2 datasets for the final analysis for defaulters and non defaulters

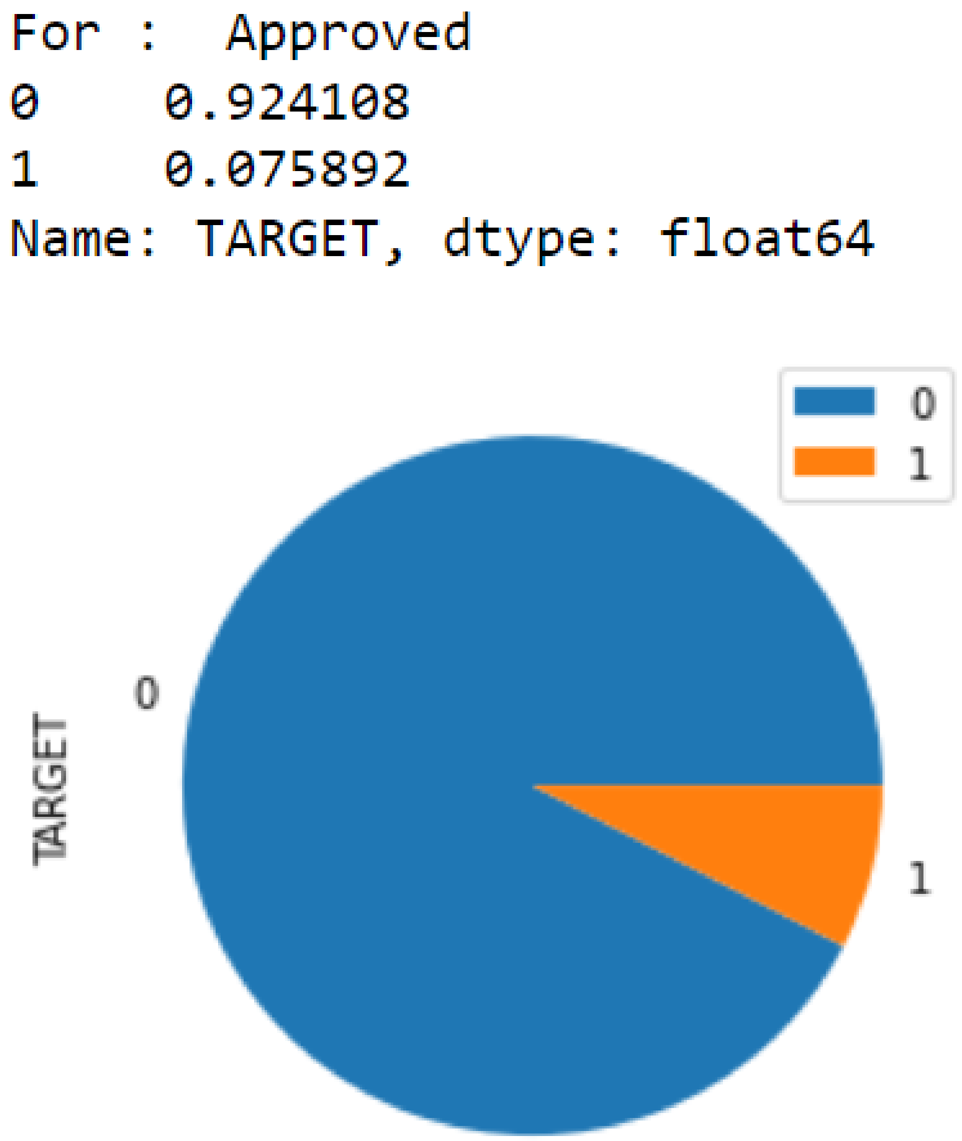


```
In [252]: # check for duplicate values for SK_ID_CURR
          dup_val= pr_appds[pr_appds.duplicated(subset='SK_ID_CURR')]
          dup_val.shape

Out[252]: (908809, 16)

In [253]: # merging the 2 data sets by LEFT merge
          finalds=pd.merge(appds,pr_appds, how='left', on="SK_ID_CURR")
          finalds.head()
```

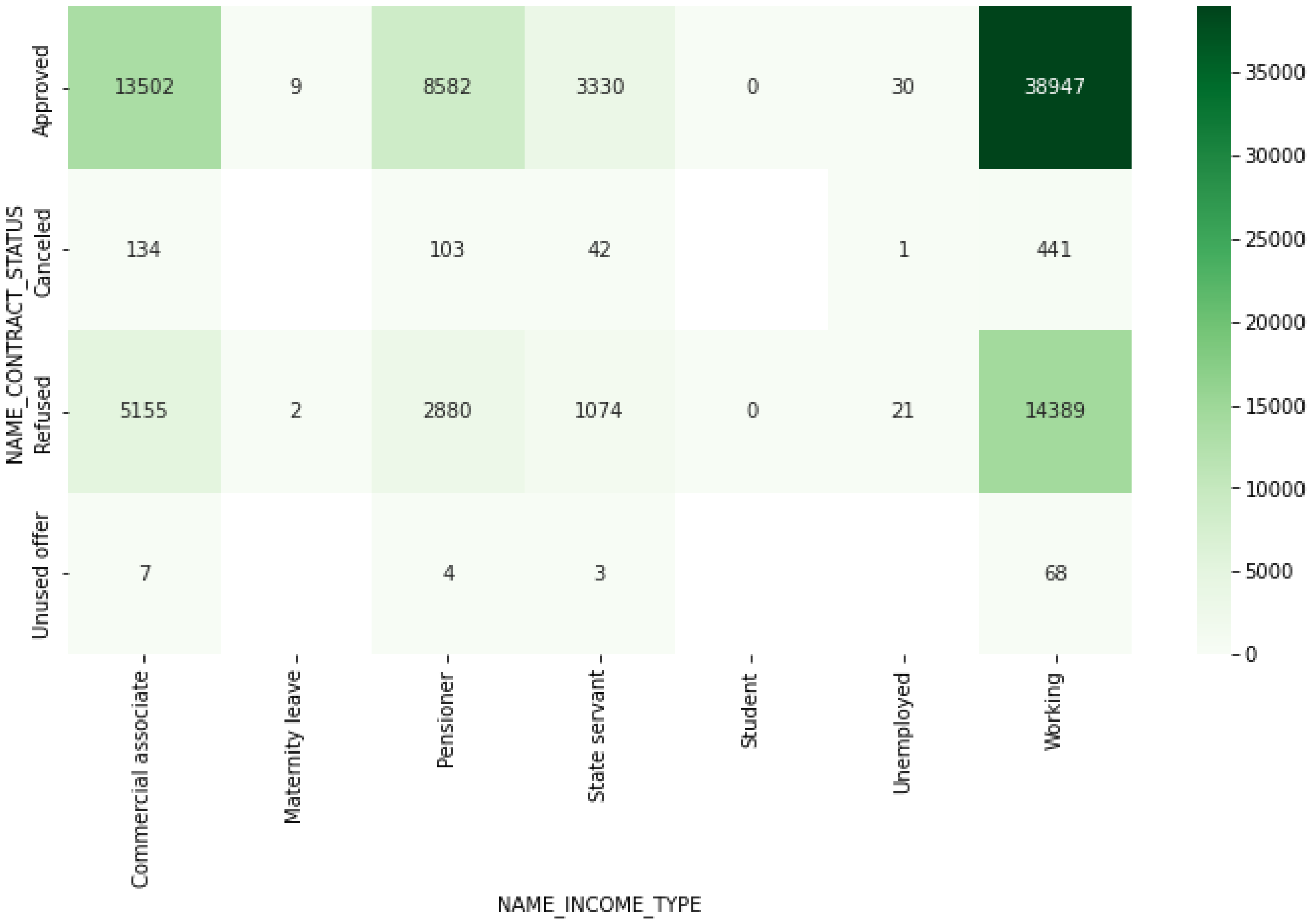
- we do analysis on the unique values of NAME_CONTRACT_STATUS by plotting a pie-chart ['Approved', 'Refused', nan, 'Canceled', 'Unused offer']

- we get this pie-chart for Approved and similarly for refused , canceled and unused offer.

- Now we map a heat map for NAME_CONTRACT_STATUS and NAME_INCOME_TYPE , we do this so that we can understand the who all are approved and rejected and so on based on their income.

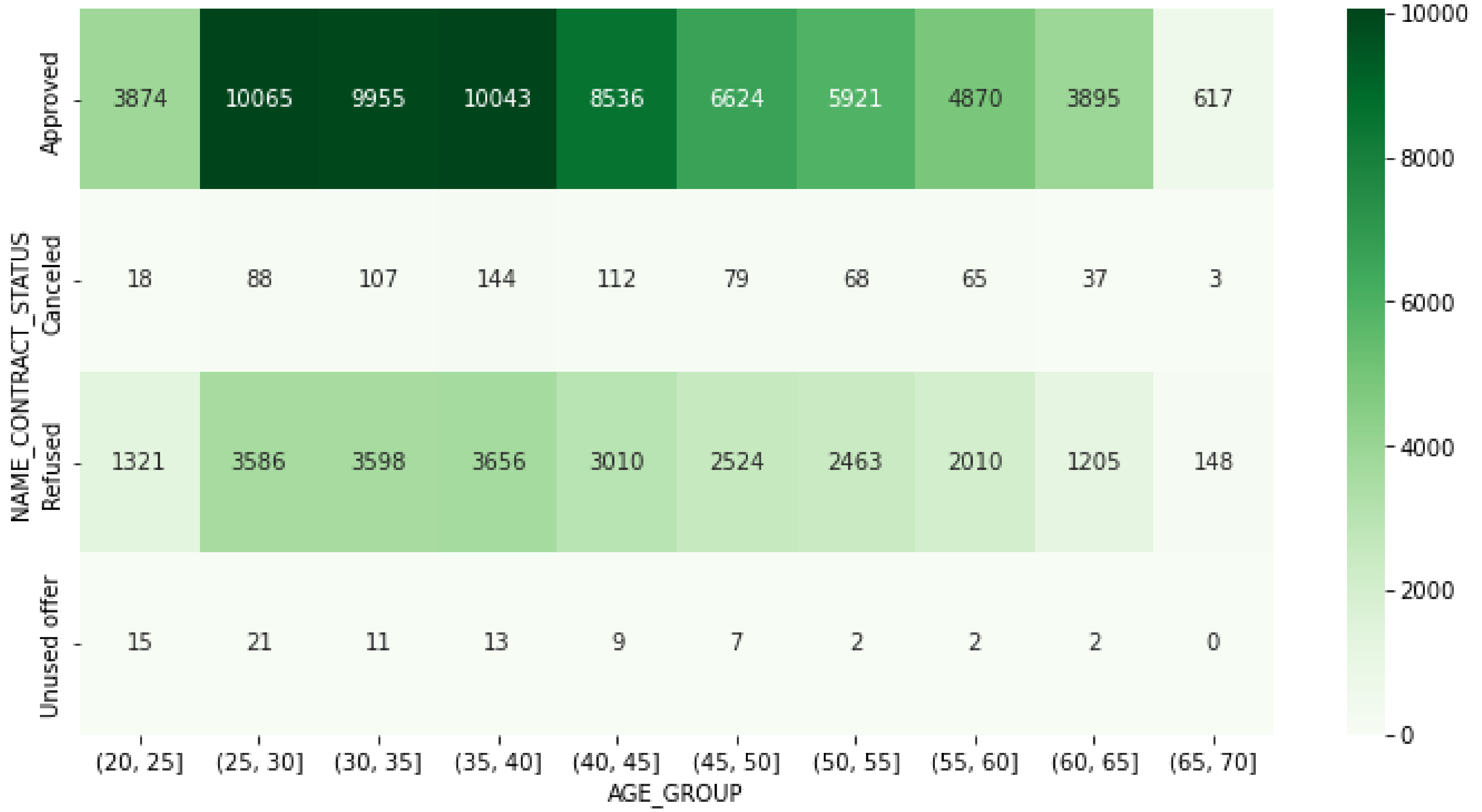


```
For :  Approved
0    0.924108
1    0.075892
Name: TARGET, dtype: float64
```



- we observe this:
  - The majority of working applicants with approved status have defaulted.
  - 14,389 working-class applicants who had been rejected before have now defaulted.

- we plot the same for AGE_GROUP and get this heatmap and observe this:
  - Age groups 25–35 and 35–45 with approved loans have higher default rates



- we finallly do a analysis to check defaulters in the approved cases from these cols:
  - ['INCOME_GROUP','AGE_GROUP','CODE_GENDER','NAME_INCOME_TYPE','OCCUPATION_TYPE',"ORGANIZATION_TYPE"]
- after which we can see the defaulters in the approved case
- *Now from this we can say that defaulters can be identified and the following factors should be checked first before granting a loan :*
  - Don't have high income rather have medium income
  - the Age factor would range from(25-35] and (35-45]
  - Female's are mostly non defaulters so person could be a Male
  - employment status should be unemployed
  - job type would be - Drivers ,Labourers, Salesman
  - person may not have own home
  - It is concerning that prior applications with Refused, Cancelled, and Unused loans also experienced default. This shows that the financial institution had rejected or cancelled a prior application but approved the present one and is now in default on both.