Date of Submission – 27 September 2022

Credit Assignment

Exploratory Data Analysis

Problem Statement

- This is a case study on two sets of data available on loan applicants provided by a bank
- The two sets of data are <u>1. Current loan application data and 2. Previous loan application data</u>
- The case study aims to find patterns in both of these data sets that can help identify if a loan applicant has difficulty in paying his/her loan payments
- The bank needs data on which parameters have an impact on defaults in payment
- Ultimately the factors driving the results can also help the bank identify patterns and optimise the risk assessment, so that the loans are given to applicants who are more likely to pay them back

Datasets

Three Datasets have been provided as described below -

1. *'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.

2. *'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.

3. *'columns_description.csv'* is data dictionary which describes the meaning of the variables.

My approach in brief

Importing all Libraries

Reading all files

Analysis of **application_data.csv**

Analysis of **previous_application.csv**

Data Cleaning

1. Handling Missing Data
2. Outlier Analysis
3. Checking for inconsistencies in data by segmenting columns based on the datatype

Top 10 Correlations

Merging Data

Segmenting data based on the context of columns

Segmenting data into numerical and categorical data

Data Analysis

Analysed the data based on 3 approaches

1. By checking the distribution of values in each column
2. Checking the distribution of values in each column by segmenting the dataframe into a set of defaulters(TARGET=1) and Non-Defaulters(TARGET=0)
3. Checking the rate of defaulters in columns containing categorical columns

Points to Note

There may be certain inconsistencies in the approach as mentioned below –

1. Outlier analysis for some columns are done in the data analysis part and not in the data cleaning part, because the outliers were identified

2. Some categorical columns that are represented by 1/0 may be analysed along with numerical data

3. I have performed the data analysis via Google Collaboratory, so the files are read directly from my google drive folder

Application Data Analysis

0. Reading the data and Introduction

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	...	FLAG_DOCUMENT_18	FLAG
0	100002	1	Cash loans	M	N	Y	0	202500.0	406597.5	24700.5	...	0	
1	100003	0	Cash loans	F	N	N	0	270000.0	1293502.5	35698.5	...	0	
2	100004	0	Revolving loans	M	Y	Y	0	67500.0	135000.0	6750.0	...	0	
3	100006	0	Cash loans	F	N	Y	0	135000.0	312682.5	29686.5	...	0	
4	100007	0	Cash loans	M	N	Y	0	121500.0	513000.0	21865.5	...	0	

5 rows × 122 columns

Data Cleaning

```
[ ]  # Checking number of columns and rows
     print(app_data.shape)
```

```
(307511, 122)
```

1. I stored the data of the file application_data.csv is stored in data frames ‘app_data’ and ‘app_data_raw’

2. While I used ‘app_data’ to do all the analysis. I did no operation on ‘app_data_raw’ so that I could use the original data whenever needed (For eg – To merge with previous application data)

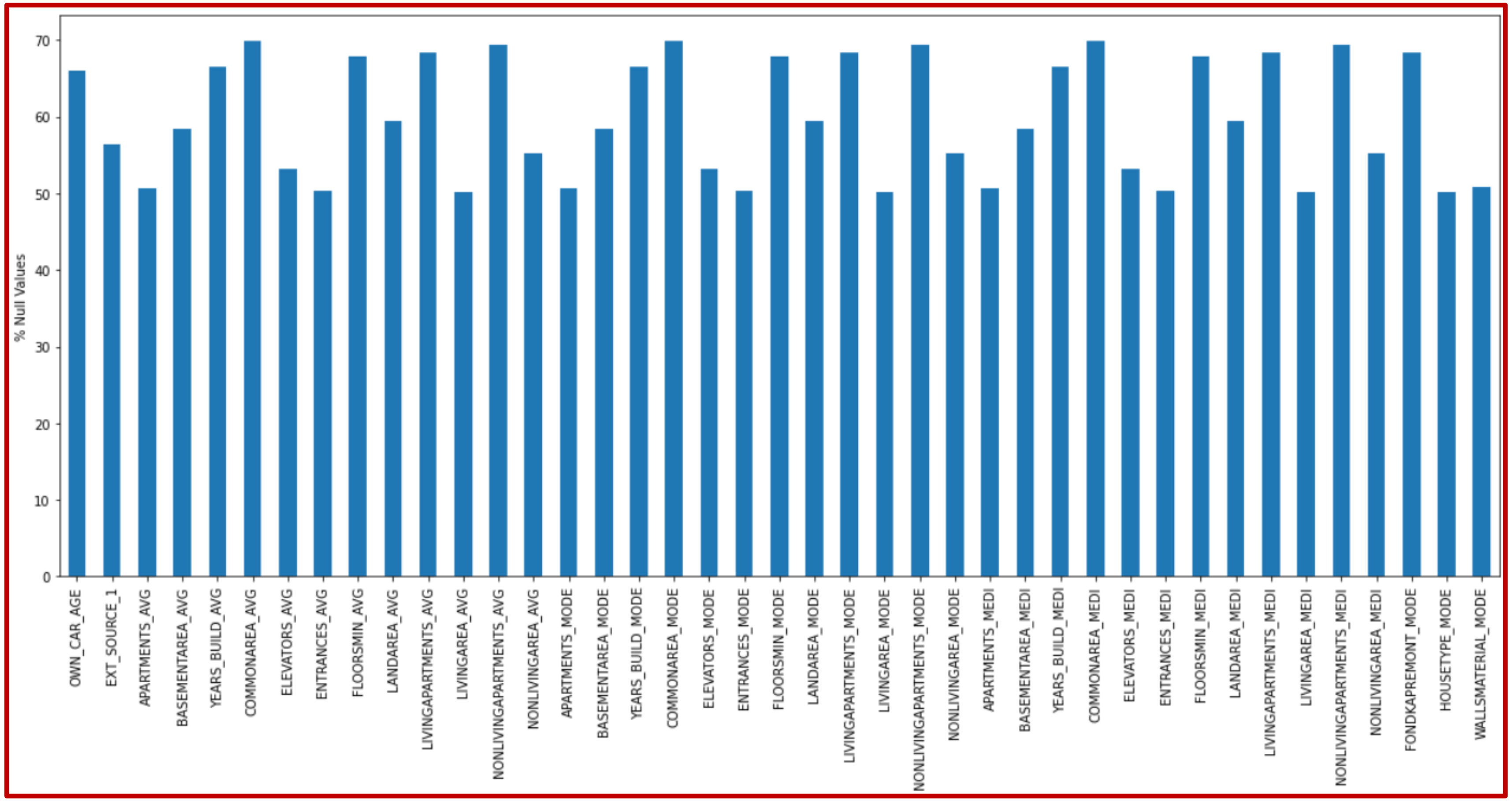
3. Here, since there are 122 columns, I did not identify outliers for all the relevant columns during data cleaning, but only the evident ones. I identified outliers during the analysis and handled the same in the analysis stage

4. There are 307,511 rows

1. Data Cleaning - Handling missing data

Columns with majority null values

- There are too many columns to check the null data for. So I identified only those columns that have more than 50% null data
- Most of this data pertains to housing information of the applicants
- Since we do not know the impact of these columns on the TARGET variable I will not drop them yet



Graph showing % Null values in columns having >50% null values

1. Data Cleaning - Handling missing data

Columns with very few null values

- I checked for columns that have <5% values as null and there were 10 such columns and they had missing data for less than 0.5% of the rows
- I now checked what are the total number of rows that are affected by missing data in these columns. They were just 2980 rows which is less than 1% of the entire data
- I dropped these rows

```
AMT_ANNUITY                   0.003902
AMT_GOODS_PRICE               0.090403
NAME_TYPE_SUITE               0.420148
CNT_FAM_MEMBERS               0.000650
EXT_SOURCE_2                  0.214626
OBS_30_CNT_SOCIAL_CIRCLE      0.332021
DEF_30_CNT_SOCIAL_CIRCLE      0.332021
OBS_60_CNT_SOCIAL_CIRCLE      0.332021
DEF_60_CNT_SOCIAL_CIRCLE      0.332021
DAYS_LAST_PHONE_CHANGE        0.000325
dtype: float64
```

List of columns that have <5% missing data

1. Data Cleaning - Studying the data based on the datatype of the columns

I will now analyse what are the data types and distribution of the same among the column

There are columns with int, float and object data types and the distribution is as follows –

1. Object - 16
2. Int - 41
3. Float - 65

```
# Distribution of datatypes among the columns
app_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 304531 entries, 0 to 307510
Columns: 122 entries, SK_ID_CURR to AMT_REQ_CREDIT_BUREAU_YEAR
dtypes: float64(65), int64(41), object(16)
memory usage: 285.8+ MB
```

Output showing the datatypes and respective number of columns

1. Data Cleaning - Studying the data based on the datatype of the columns

Analysing columns with object datatype (16 Columns)

Now I will check for data cleanliness in columns with object data type to see the following points like –

1. Inconsistencies in categorical values in each columns(Eg- Spelling errors,)

2. Data type misrepresentation(Eg- Float values is shown as object because of ",")

My Observations -

1. There are no repeated values in any column due to spelling errors or any other reasons

2. All the columns are categorical and none are of the datatype object due to error in representing float or int data

3. Missing values are mentioned as 'XNA' in 2 cases - CODE_GENDER and ORGANIZATION_TYPE.

4. I will not be dropping the CODE_GENDER missing value rows because I am not yet sure of the impact/relevance of gender on the target variable. I will also not drop the ORGANIZATION_TYPE missing value rows because they form >15% of all the rows in the data frame and dropping them may affect the overall analysis

1. Data Cleaning - Studying the data based on the datatype of the columns

Analysing columns with int datatype (41 columns)

After looking at the value counts for all the columns using a loop
Observations - 3 types of data here

1. Flags - These columns are categorical having values as either 1 or 0 to denote yes or no
2. Unique ID - each applicant will have a unique ID
3. Continuous data - Numerical data Like count of days and ordered numeric categories like ratings etc.

I will be doing outlier analysis only on the continuous data.
I will be omitting the only unique ID there is in this data frame i.e, SK_ID_CURR.

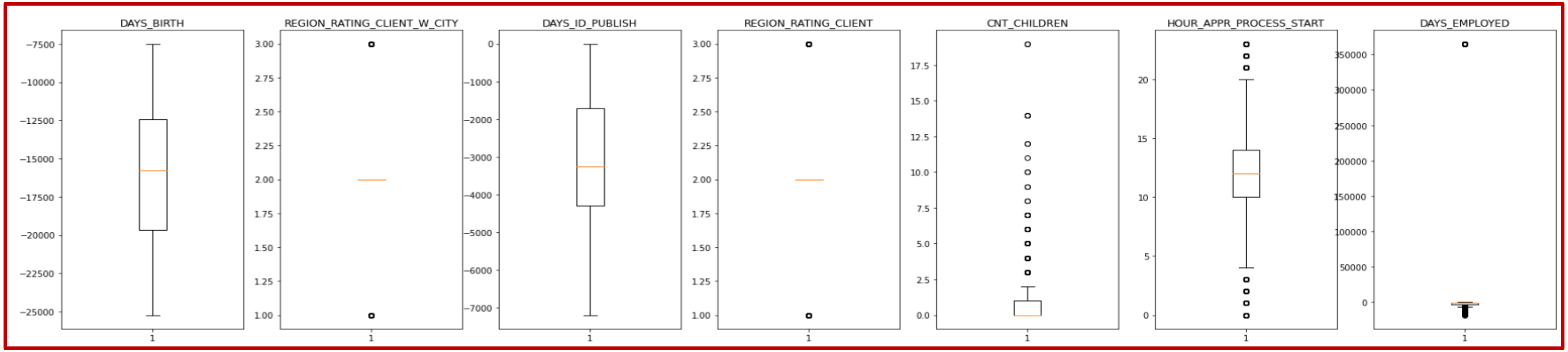
Analysing columns with float datatype (65 Columns)

After looking at the value counts for all the columns using a loop
Observations – The data can be broadly classified as -

1. Columns that contain housing information like apartment area, building area etc.(47 columns)
2. All other data

1. Data Cleaning - Handling Outliers

Outlier Analysis on INT continuous data

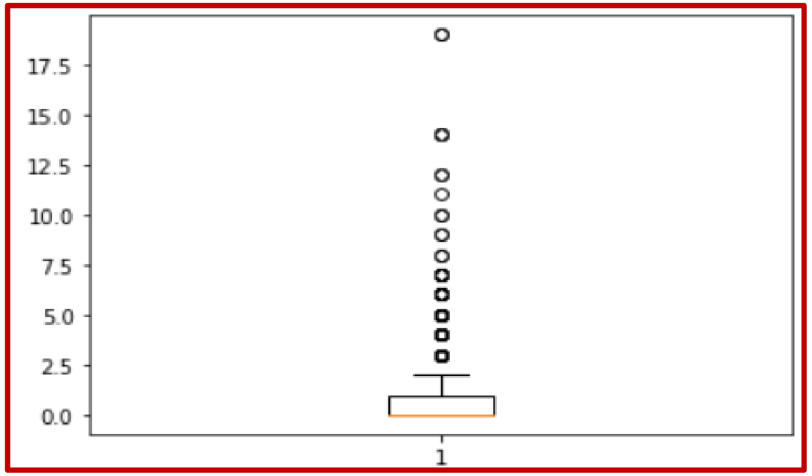


CNT_CHILDREN

- There are many some applicants who have more then 4 children and they form a minor part of the entire data. (~0.4%)
- Hence I will drop all columns having children >4

```
Name: CNT_CHILDREN, dtype: int64
0     70.034578
1     19.873182
2      8.702562
3      1.208416
4      0.139887
5      0.027583
6      0.006896
7      0.002299
14     0.000985
8      0.000657
9      0.000657
12     0.000657
10     0.000657
19     0.000657
11     0.000328
Name: CNT_CHILDREN, dtype: float64
```

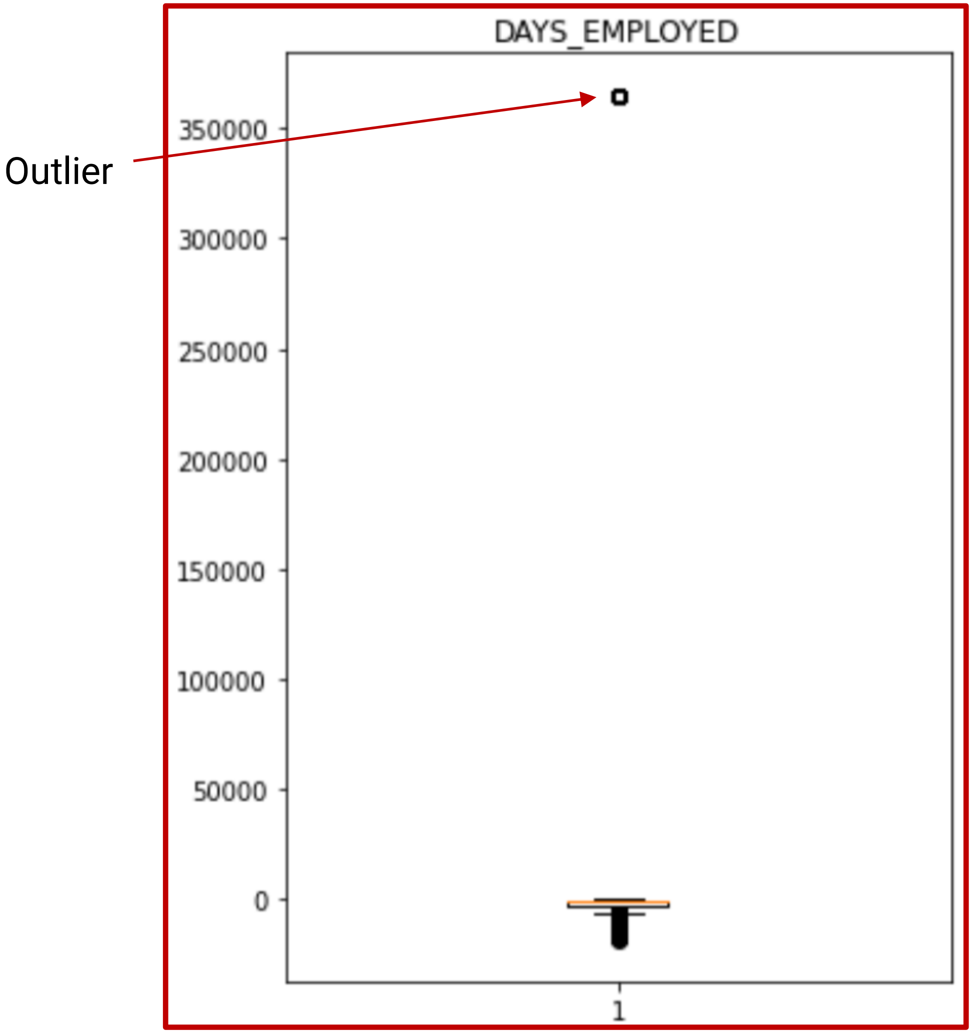
CNT_CHILDREN normalised value counts



1. Data Cleaning - Handling Outliers

DAYS_EMPLOYED

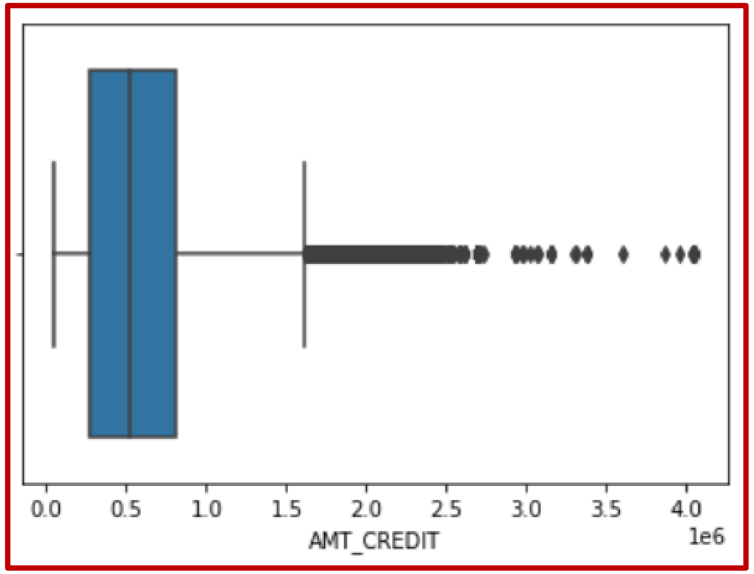
- I could see an outlier in the boxplot of DAYS_EMPLOYED
- After converting the units of the column to years and storing it in YEARS_EMPLOYED column, I could see that around 18% of the applicants have been employed for nearly 1000 years which is logically impossible. Hence, I will be replacing this value(1000 Years) in YEARS_EMPLOYED column with NaN



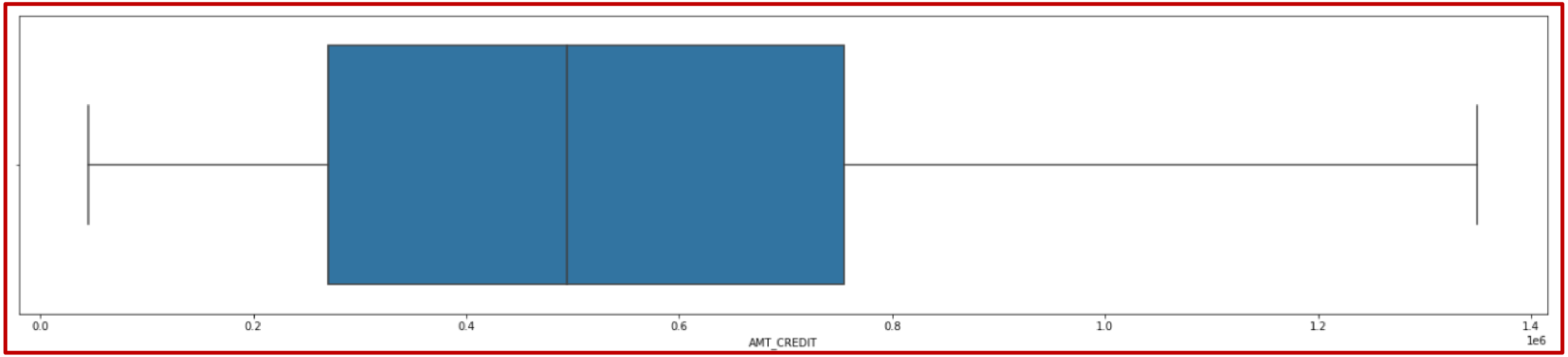
1. Data Cleaning - Handling Outliers

AMT_CREDIT

- There are extreme outliers in the AMT_CREDIT column
- I have capped the value to 95%-ile of the data



Boxplot of AMT_CREDIT

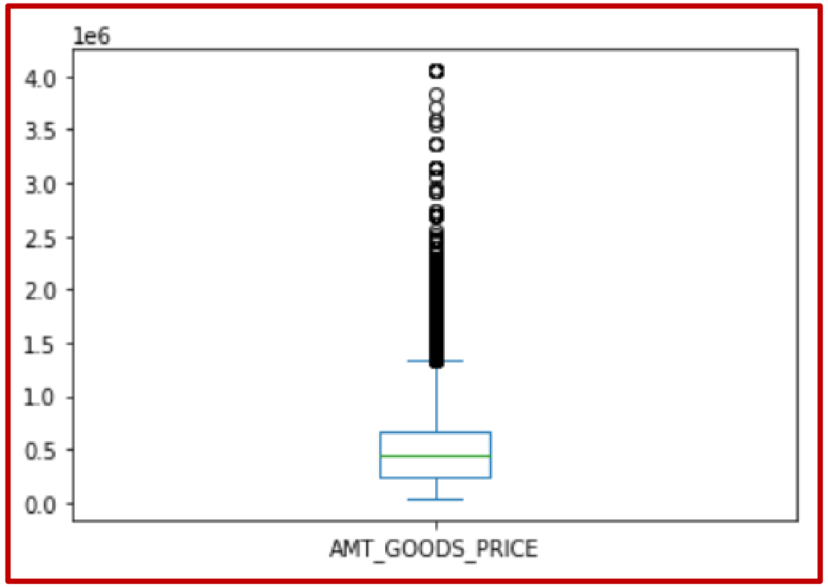


Boxplot of AMT_ CREDIT after capping the values

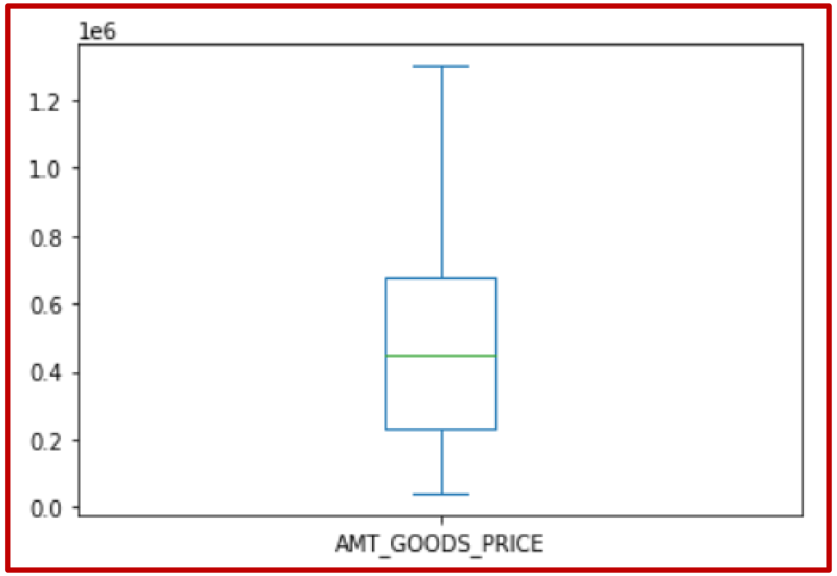
1. Data Cleaning - Handling Outliers

AMT_GOODS_PRICE

- There are extreme outliers in the AMT_GOODS_PRICE column
- I have capped the value to 95%-ile of the data



Boxplot of AMT_GOODS_PRICE

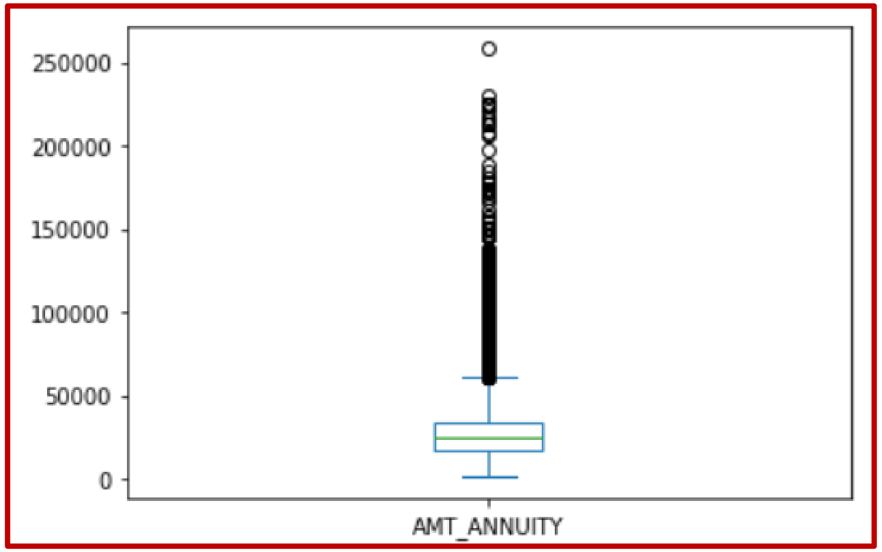


Boxplot of AMT_ GOODS_PRICE after capping the values

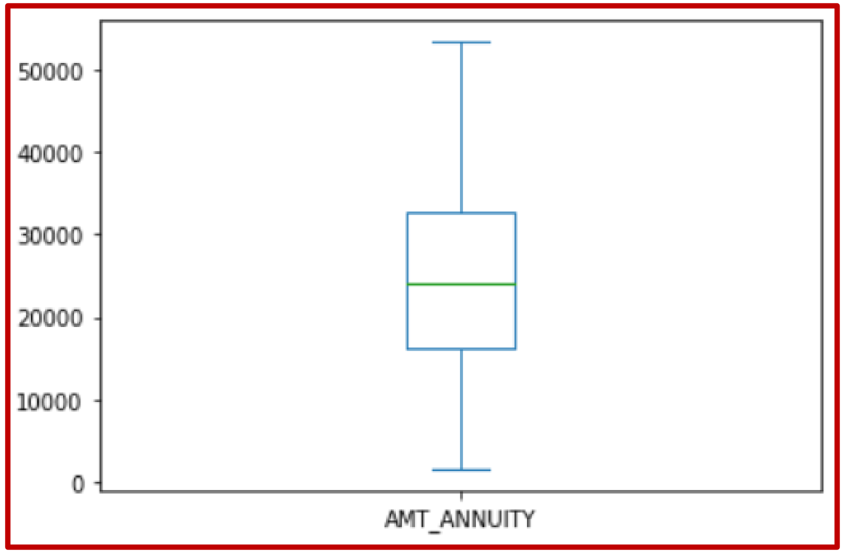
1. Data Cleaning - Handling Outliers

AMT_ANNUITY

- There are extreme outliers in the AMT_ANNUITY column
- I have capped the value to 95%-ile of the data



Boxplot of AMT_ANNUITY

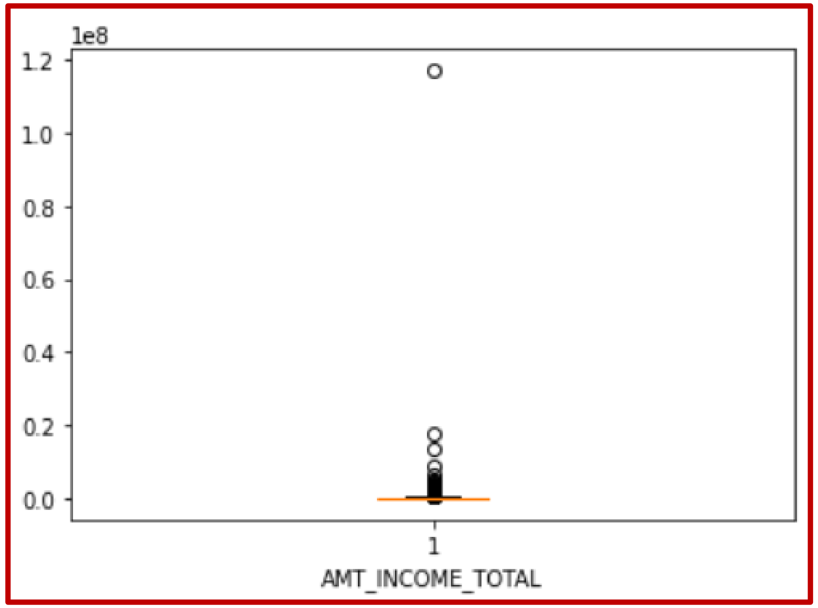


Boxplot of AMT_ ANNUITY after capping the values

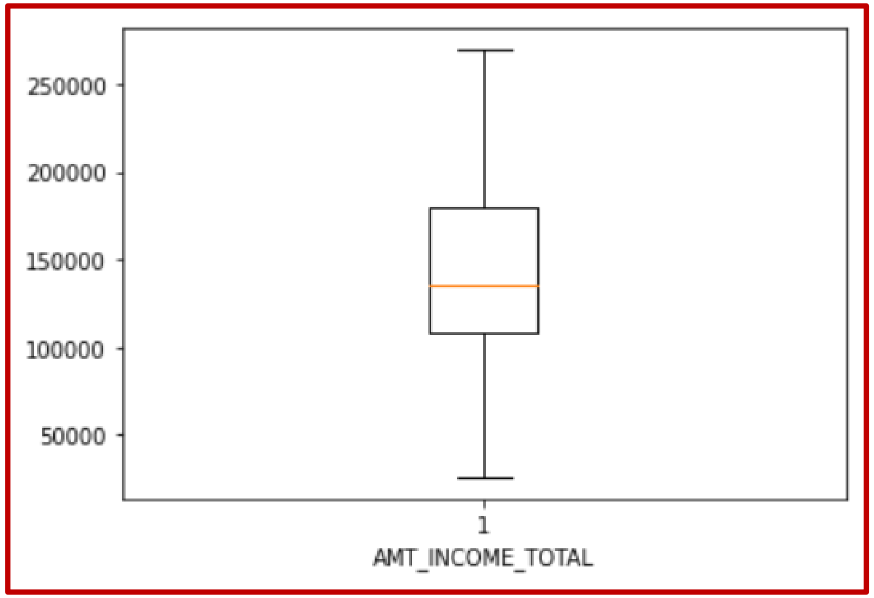
1. Data Cleaning - Handling Outliers

AMT_INCOME_TOTAL

- There are extreme outliers in the AMT_INCOME_TOTAL column
- I have capped the value to 90%-ile of the data, because there is substantial increase in the income after this value



Boxplot of AMT_INCOME_TOTAL



Boxplot of AMT_INCOME_TOTAL after capping the values

2. Top 10 Correlations

Top 10 correlations for Defaulters

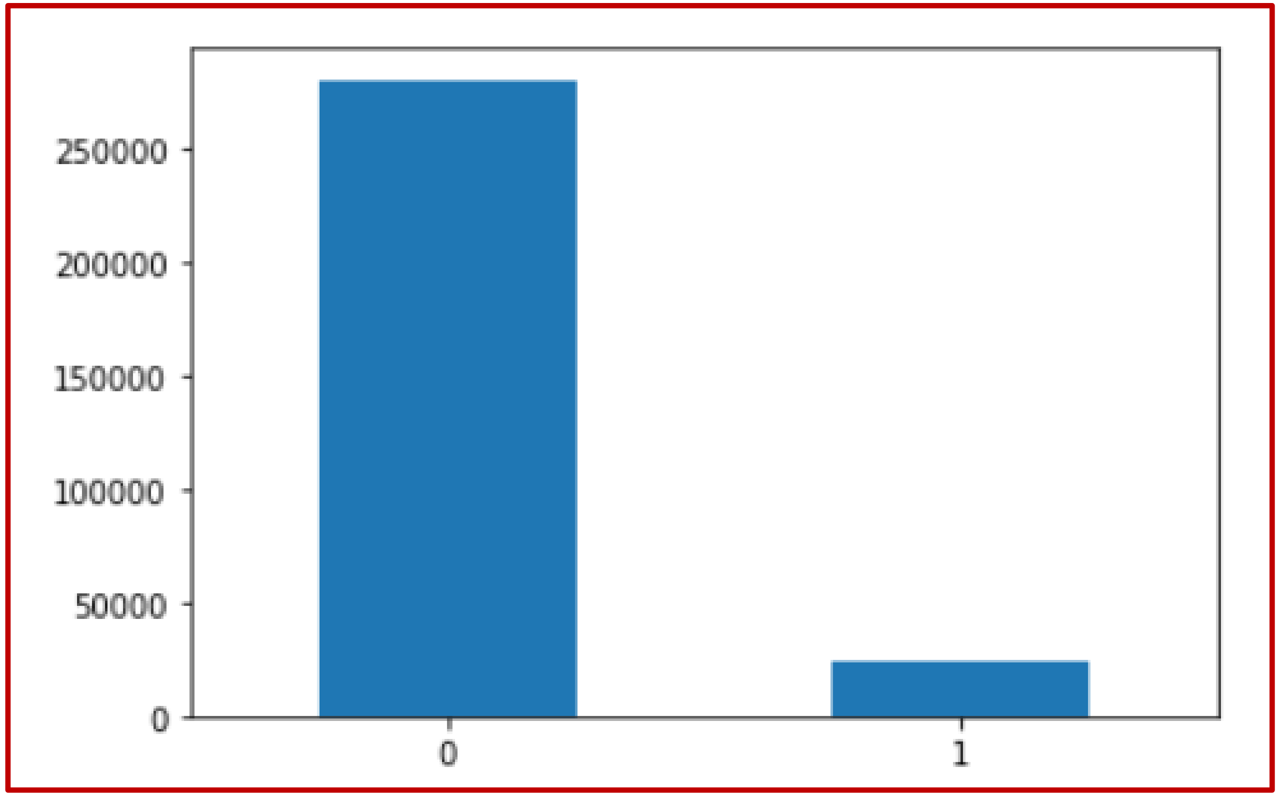
SK_ID_CURR	SK_ID_CURR	1.000000
OBS_30_CNT_SOCIAL_CIRCLE	OBS_60_CNT_SOCIAL_CIRCLE	0.998286
BASEMENTAREA_MEDI	BASEMENTAREA_AVG	0.998205
YEARS_BUILD_MEDI	YEARS_BUILD_AVG	0.998087
COMMONAREA_MEDI	COMMONAREA_AVG	0.998083
NONLIVINGAPARTMENTS_AVG	NONLIVINGAPARTMENTS_MEDI	0.998053
FLOORSMIN_AVG	FLOORSMIN_MEDI	0.997810
LIVINGAPARTMENTS_AVG	LIVINGAPARTMENTS_MEDI	0.997638
FLOORSMAX_MEDI	FLOORSMAX_AVG	0.997189
NONLIVINGAPARTMENTS_MODE	NONLIVINGAPARTMENTS_MEDI	0.997003
ENTRANCES_MEDI	ENTRANCES_AVG	0.996670
dtype: float64		

Top 10 correlations for Non-Defaulters

SK_ID_CURR	SK_ID_CURR	1.000000
YEARS_BUILD_AVG	YEARS_BUILD_MEDI	0.998519
OBS_60_CNT_SOCIAL_CIRCLE	OBS_30_CNT_SOCIAL_CIRCLE	0.998514
FLOORSMIN_AVG	FLOORSMIN_MEDI	0.997215
FLOORSMAX_MEDI	FLOORSMAX_AVG	0.997033
ENTRANCES_AVG	ENTRANCES_MEDI	0.996910
ELEVATORS_AVG	ELEVATORS_MEDI	0.996182
COMMONAREA_MEDI	COMMONAREA_AVG	0.995840
LIVINGAREA_MEDI	LIVINGAREA_AVG	0.995578
APARTMENTS_AVG	APARTMENTS_MEDI	0.995152
BASEMENTAREA_AVG	BASEMENTAREA_MEDI	0.994039
dtype: float64		

3. Data Analysis – Data Imbalance

- Visualising the value count distribution of TARGET column, it is evident that the data is extremely imbalanced.
- Few loan applicants have defaulted on their payments.
- The Imbalance ratio is approximately – **Defaulters(1) : Non-Defaulters(0) = 8.1 : 91.9 (or) 1 : 11**




```
[48]  1 print('Imbalance ratio for TARGET vari

Imbalance ratio for TARGET variable is -

Default : No-Default =  8.1 : 91.9
```

3. Data Analysis – Segmenting based on context of columns

While studying the data during data cleaning, I have categorised the 122 columns in the following way based on the context of the columns –

1. Loan information - Amount, Down payment etc (Unknown)
2. Housing Info (47 Columns)
3. Documents (20 Columns)
4. Region Rating (6 Columns)
5. Credit Bureau information (6 Columns)
6. Social Circle information (4 Columns)
7. External Source rating (3 Columns)
8. Personal Information - flagged and non-flagged (Unknown)

3. Data Analysis – Loan information

1. NAME_CONTRACT _TYPE

- Most of the loans are Cash loans and less than 10% are Revolving loans

```
Revolving loans      9.169035
Cash loans          90.830965
Name: NAME_CONTRACT_TYPE, dtype: float64
```

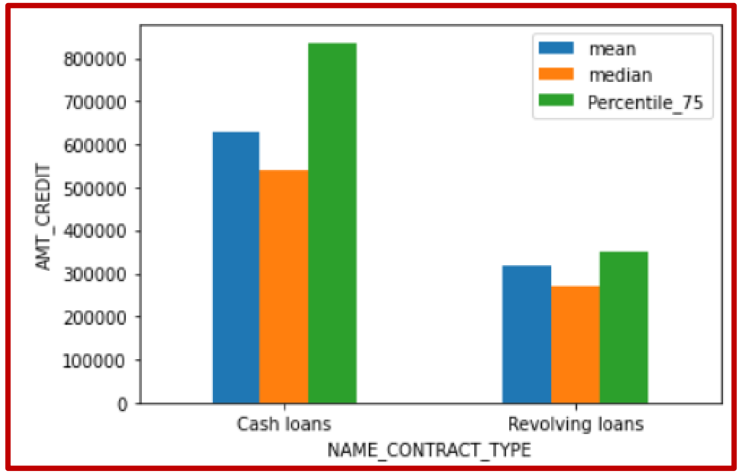
NAME_CONTRACT _TYPE normalised value counts

- Revolving loans have a lesser default rate of 5.5% as against the default rate of 8.35% for Cash loans

```
NAME_CONTRACT_TYPE
Cash loans         8.357143
Revolving loans    5.531869
Name: TARGET, dtype: float64
```

Rate of defaulters for each contract type

- Cash loans have a higher Credit Amount than Revolving loans

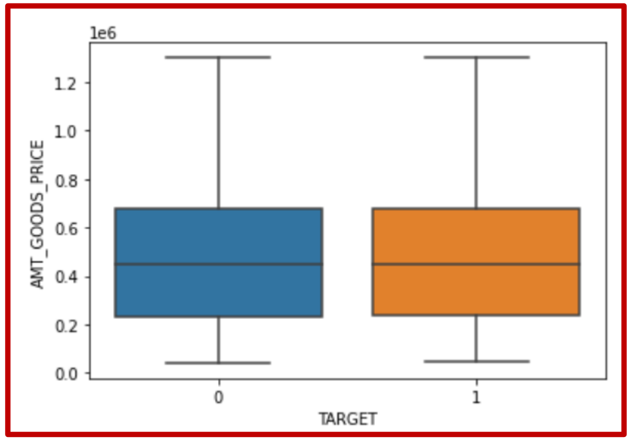


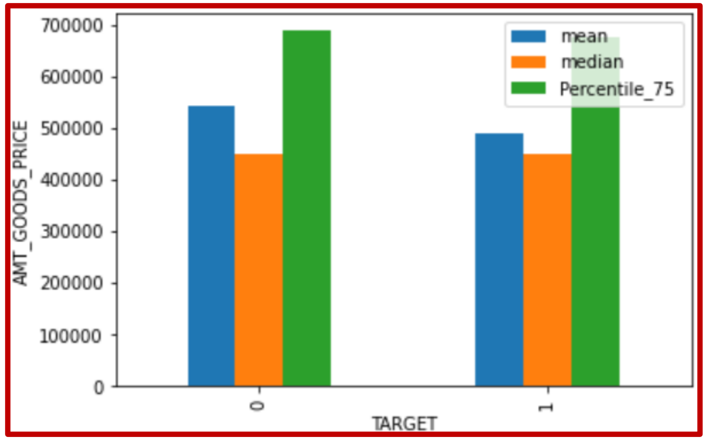
Mean, Median and 75th %-ile of AMT_CREDIT for cash and Revolving loans

3. Data Analysis – Loan information

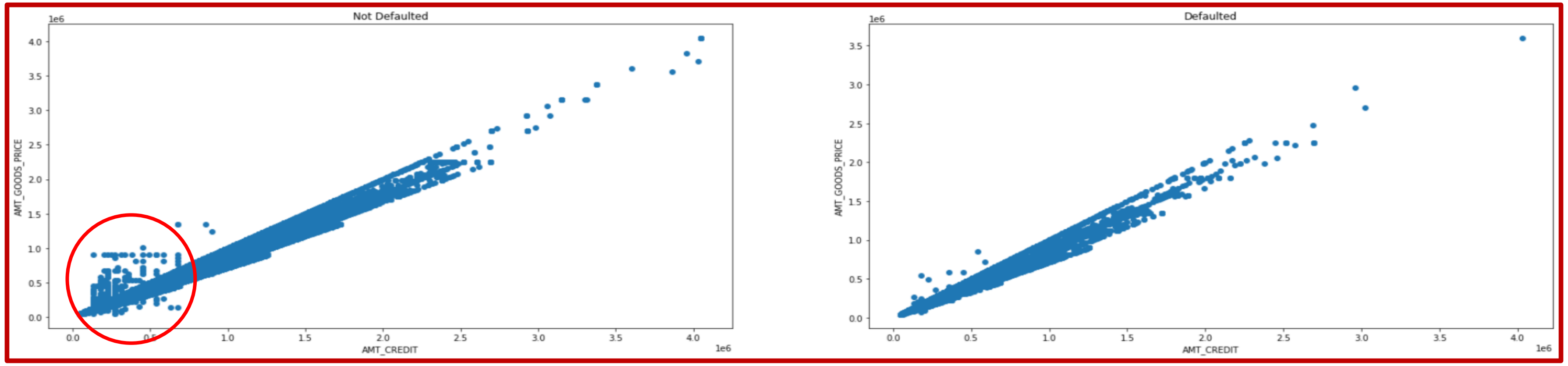
2. AMT_GOODS_PRICE

- There is no obvious evidence of the impact of AMT_GOODS_PRICE on TARGET as seen in the boxplot and the quantiles bar graph





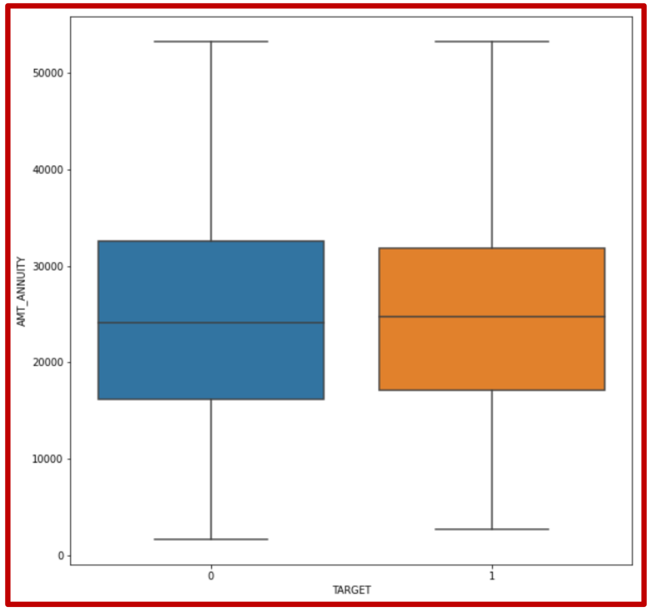
- But when a bivariate analysis is done for AMT_GOODS_PRICE vs AMT_CREDIT for defaulters and non-defaulters, we can see that for lower range of AMT_GOODS_PRICE and AMT_CREDIT , the defaulters are less
- AMT_CREDIT and AMT_GOODS_PRICE have a linear relationship



3. Data Analysis – Loan information

3. AMT_ANNUITY

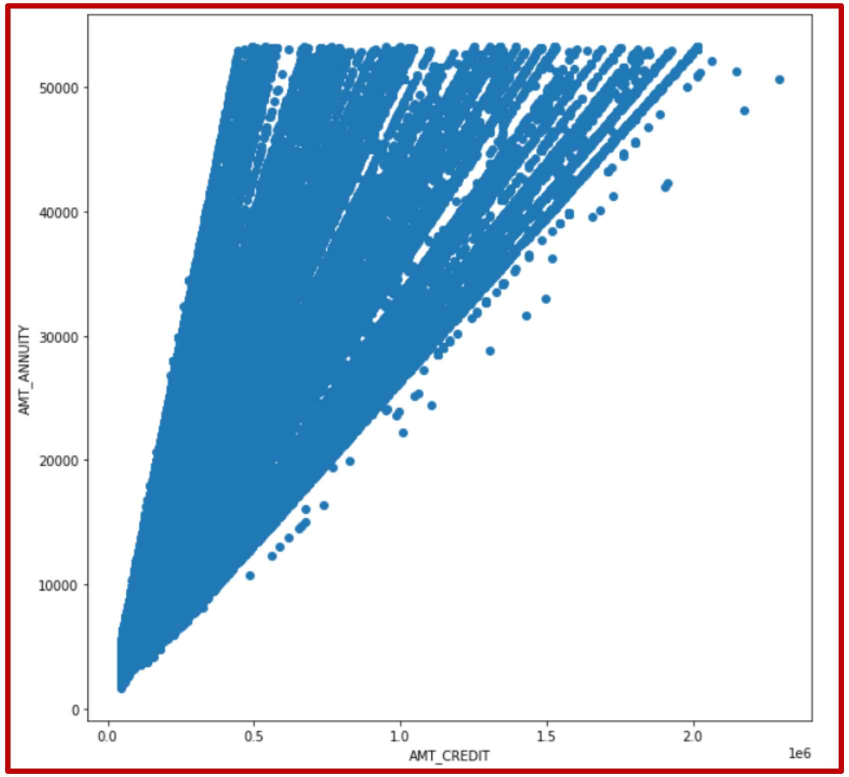
- AMT_ANNUITY has a linear relationship with AMT_CREDIT as seen in the scatter-plot
- But AMT_ANNUITY has no impact on TARGET as seen in the boxplot and quantile values



AMT_ANNUITY boxplot for defaulters and non-defaulters

TARGET	mean	median	Percentile_75
0	25131.307221	24088.5	32602.5
1	25267.350946	24750.0	31887.0

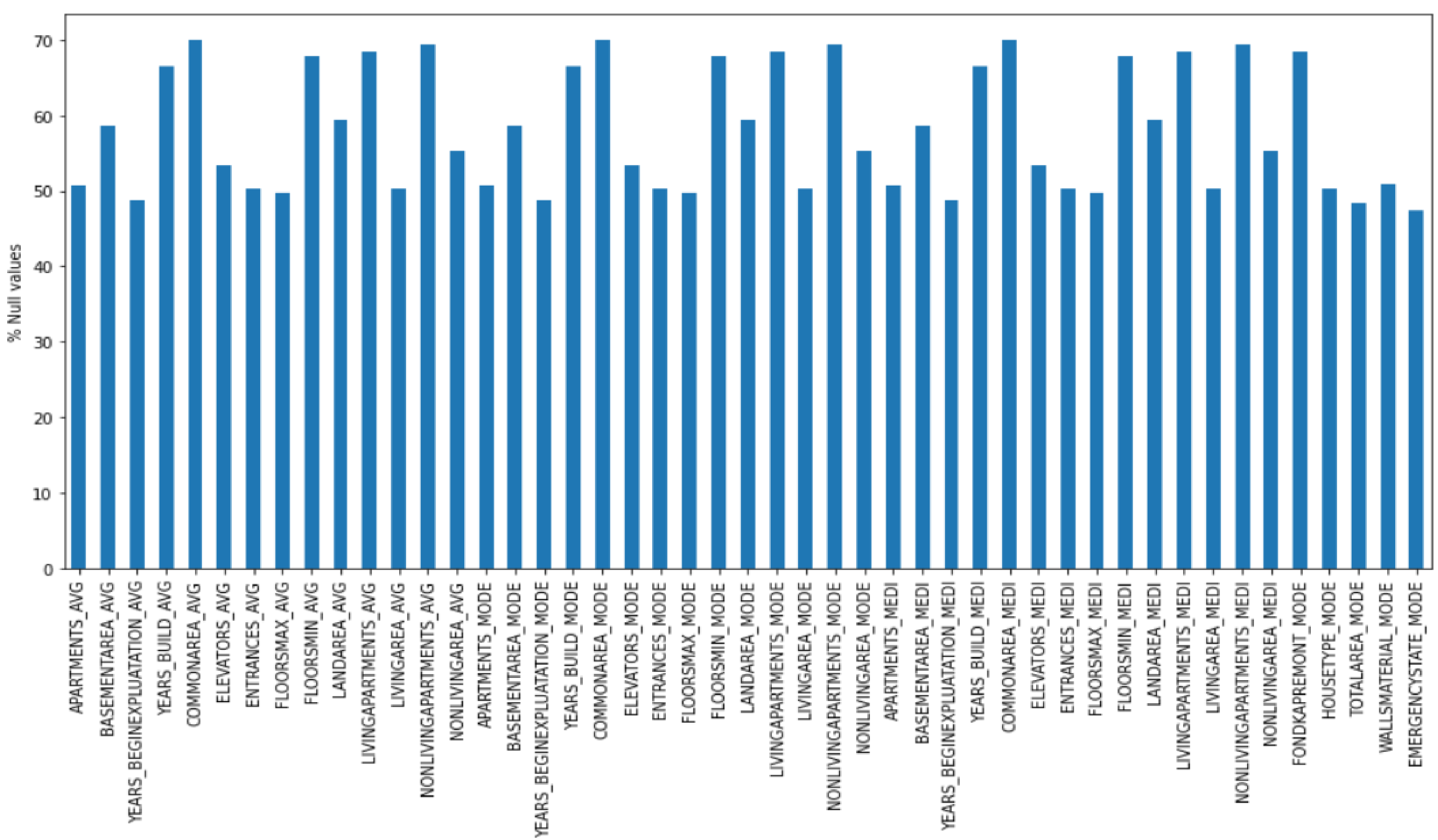
Quantile values for AMT_ANNUITY



Scatter-plot between AMT_CREDIT and AMT_ANNUITY

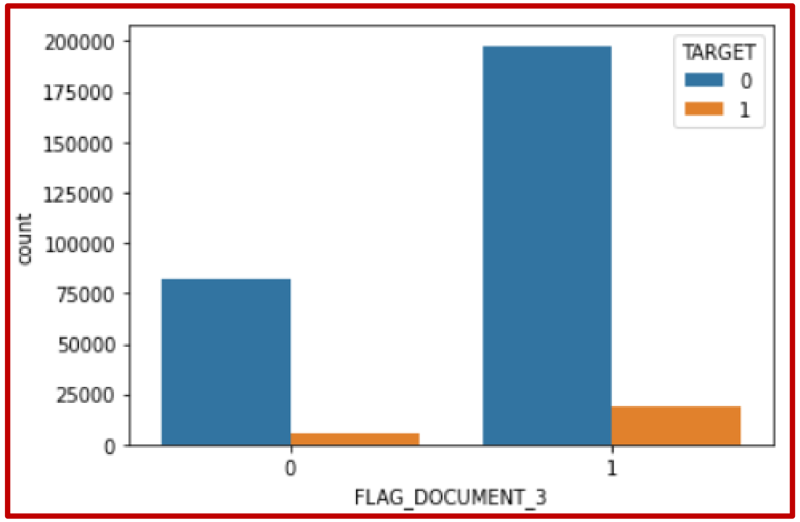
3. Data Analysis – Housing information(47 Columns)

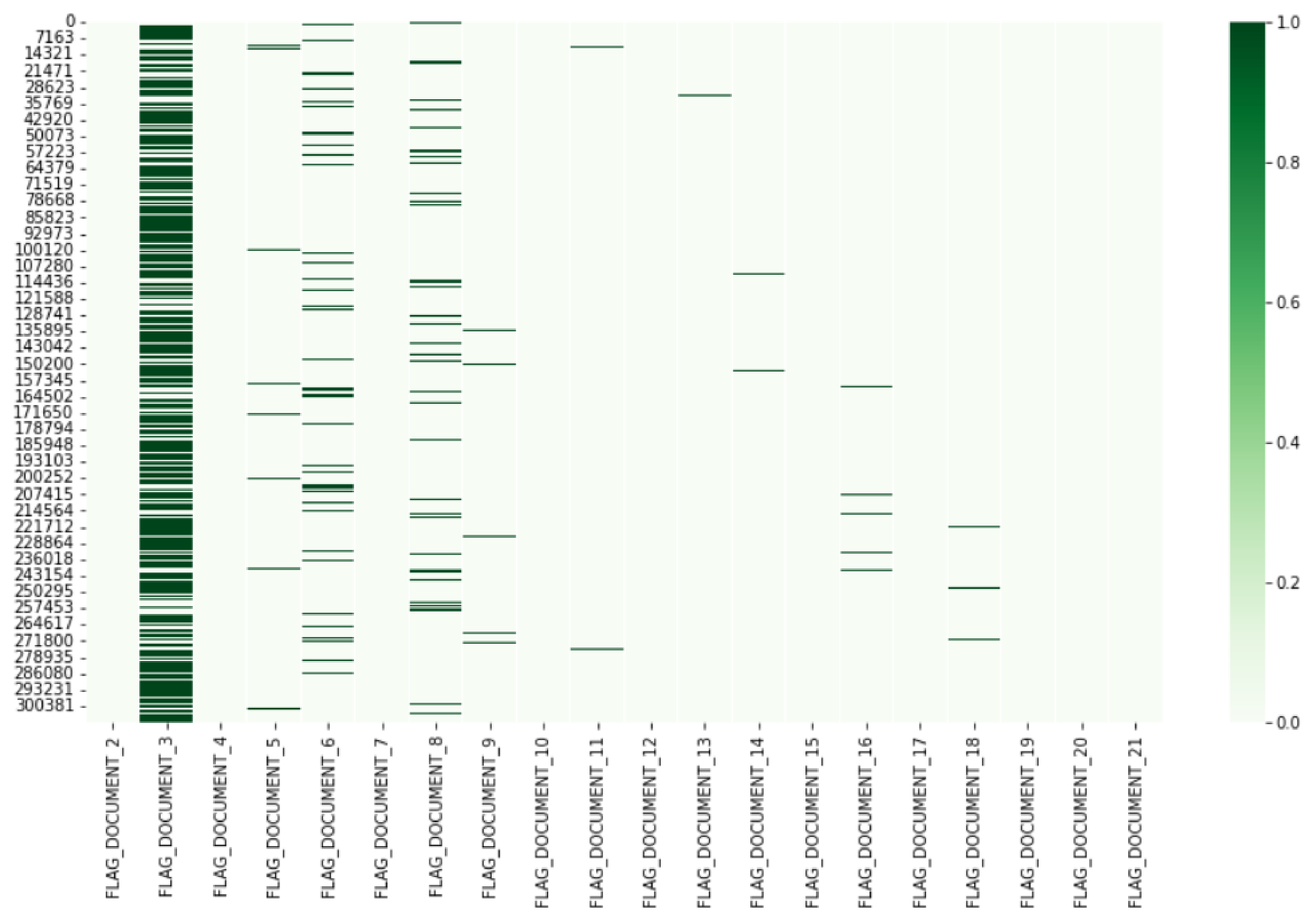
- Percentage null values in these columns range from 47%-70% of the data is missing. Hence I will drop these columns



3. Data Analysis – Documents (20 Columns)

- Heatmap of all documents columns shows that most of the documents were not submitted by the applicants except for DOCUMENT_3
- DOCUMENT_3 has no impact on TARGET as seen in the count plot





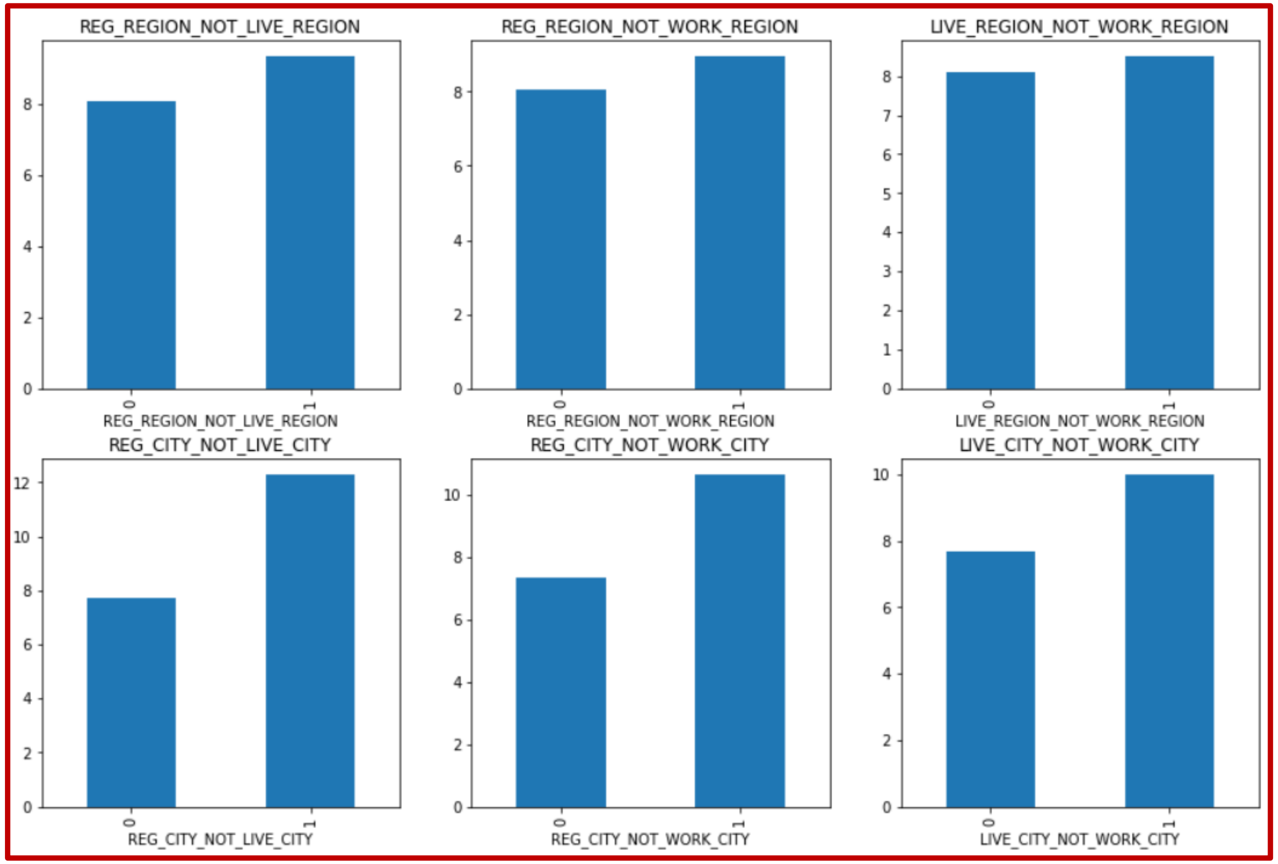
3. Data Analysis – Region Rating (6 Columns)

Flag type columns

- The columns 'REG_REGION_NOT_WORK_REGION', 'REG_REGION_NOT_LIVE_REGION' and 'LIVE_REGION_NOT_WORK_REGION' show almost identical default rates of 8-9% So I will drop these columns

- Columns 'REG_CITY_NOT_WORK_CITY' and 'REG_CITY_NOT_LIVE_CITY' have a higher default rates when the cities are different (Applicant has responded 1)

- I also notice a slight increase in payment defaults when the LIVE_CITY_NOT_WORK_CITY (Living city and working city) are different. But not very significant, so I will drop this as well



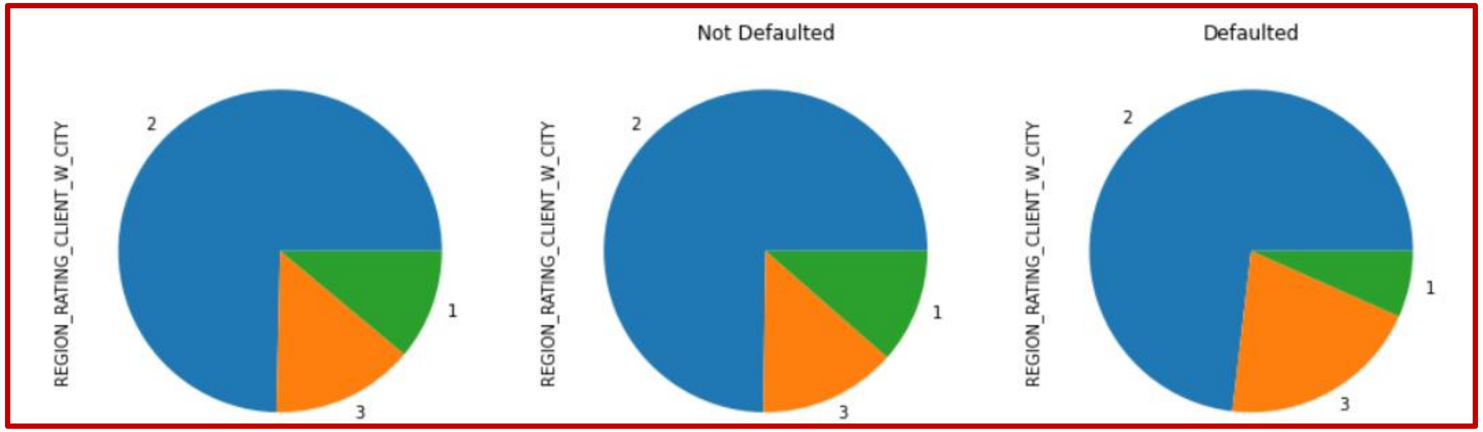
Rate of default vs Region Rating (Flag columns)

3. Data Analysis – Region Rating (6 Columns)

REGION_RATING_CLIENT_W_CITY

- For Rating 2, the trend is similar across both cases and on an overall level

- Within the group of applicants who have defaulted on their payments, I notice that the ratio of applicants in region of rating 1 and 3 slightly moves towards 3, which are the lower rated cities/regions.



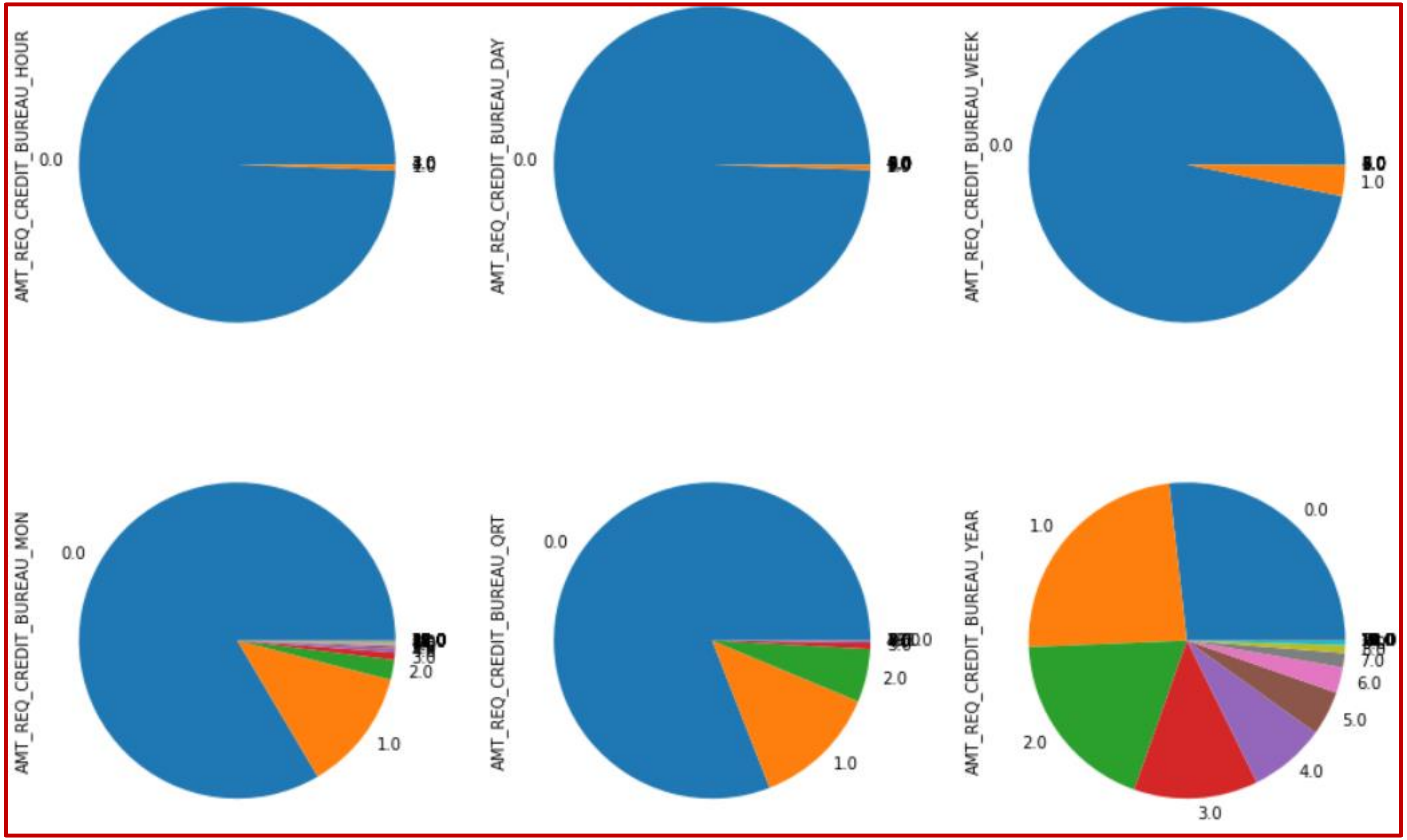
- Region/City with Category 1 has lowest percentage of loan applicants who have had difficulty with their payments

- Region/City with Category 3 has highest percentage of loan applicants who have had difficulty with their payments

- Category 2 falls in between

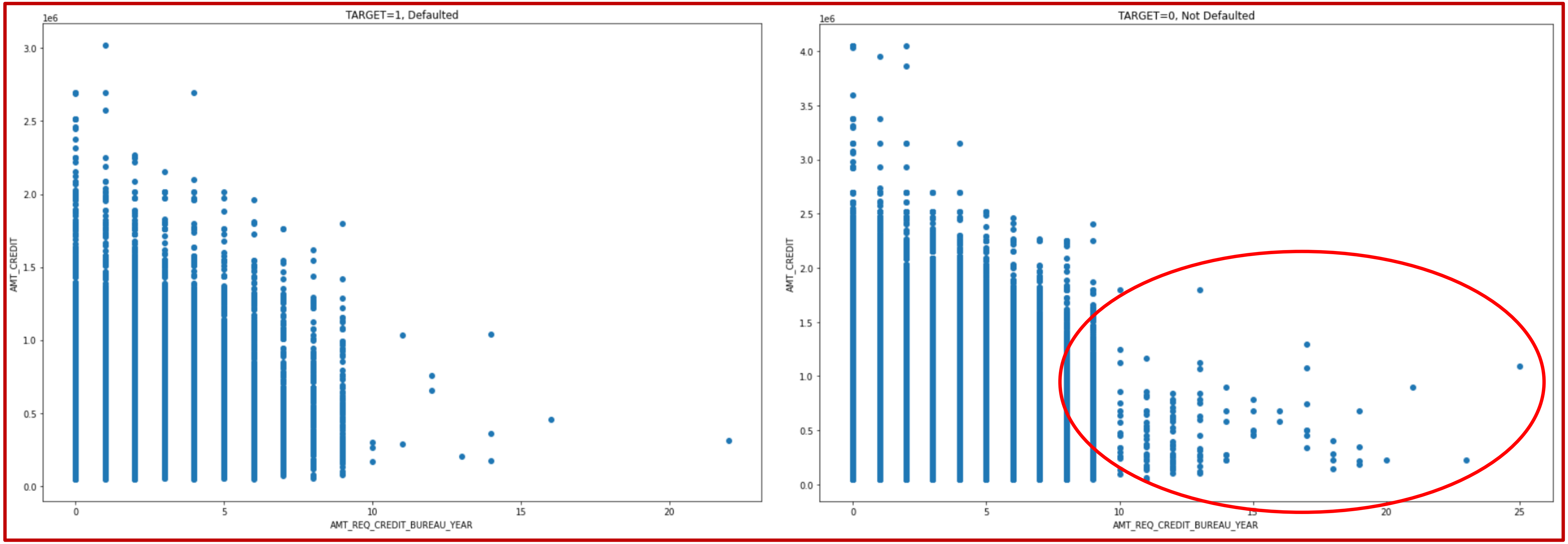
3. Data Analysis – Credit Bureau Information (6 Columns)

- Most responses for AMT_REQ_CREDIT_BUREAU_HOUR and AMT_REQ_CREDIT_BUREAU_DAY is zero(0), hence I will drop these columns



3. Data Analysis – Credit Bureau Information (6 Columns)

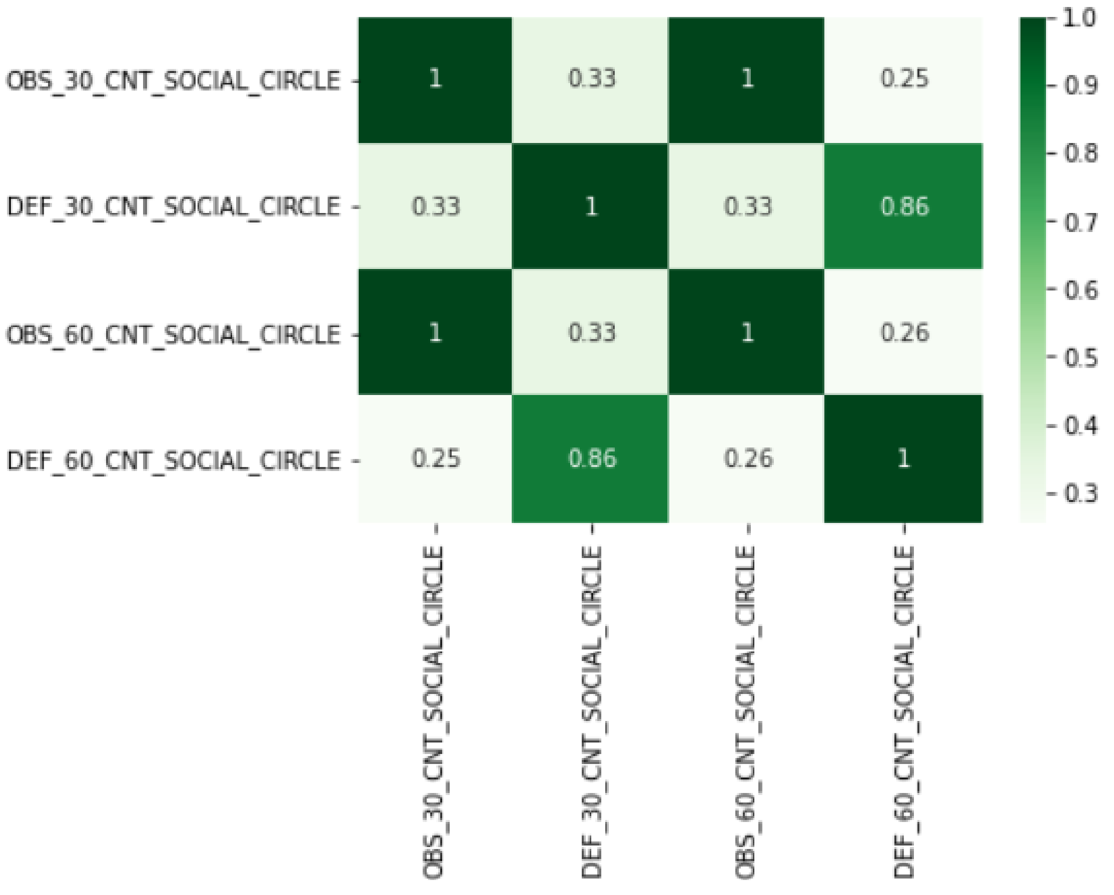
- As AMT_REQ_CREDIT_BUREAU_YEAR increase, the AMT_CREDIT also increases
- Applicants who defaulted did not have more than enquiries 10 enquiries in the past year(Except few)



3. Data Analysis – Social Circle information (4 Columns)

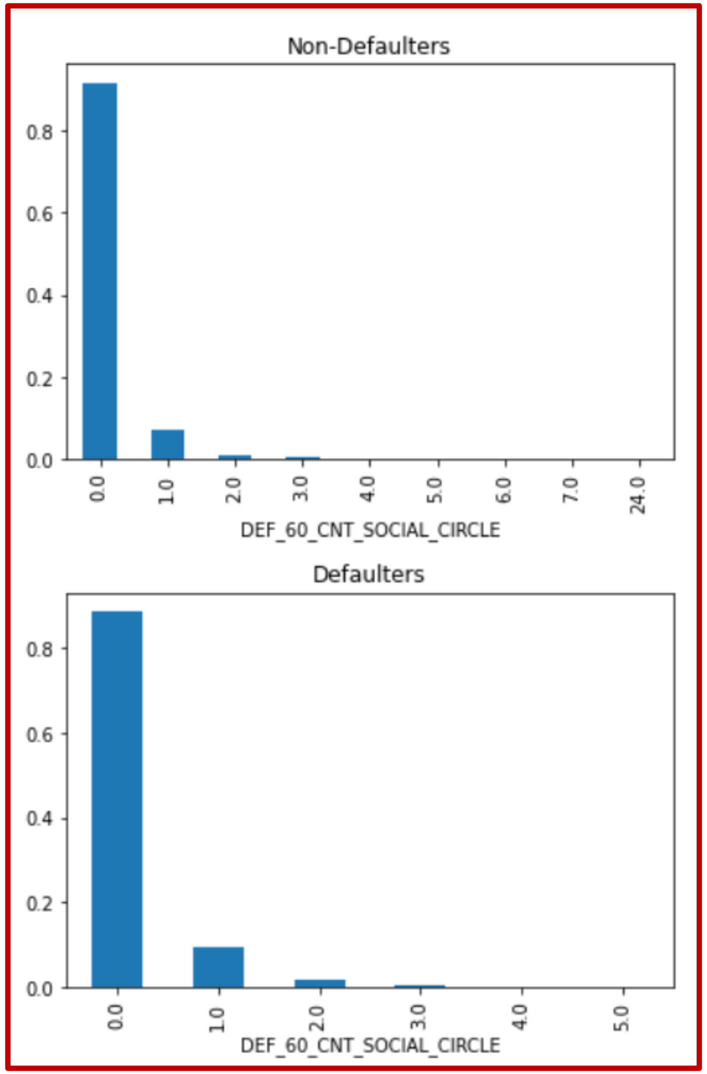
- There is high correlation between
  1. DEF_30_CNT_SOCIAL_CIRCLE and DEF_60_CNT_SOCIAL_CIRCLE
  2. OBS_30_CNT_SOCIAL_CIRCLE and OBS_60_CNT_SOCIAL_CIRCLE

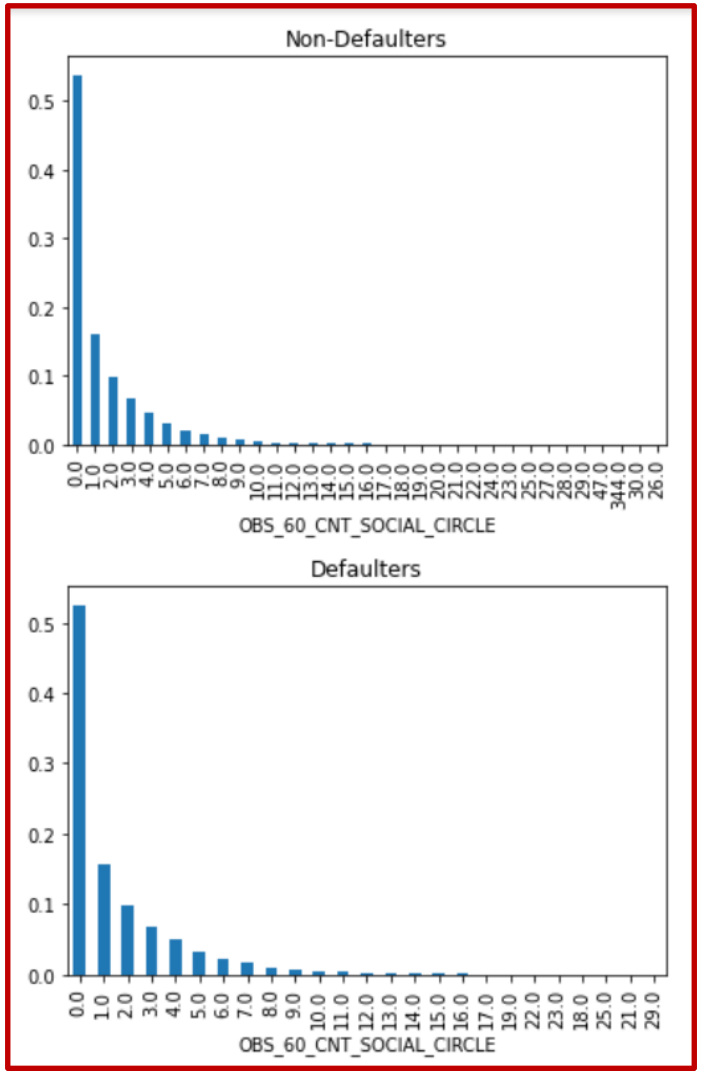
- So I will only analyse DEF_60_CNT_SOCIAL_CIRCLE and OBS_60_CNT_SOCIAL_CIRCLE



3. Data Analysis – Social Circle information (4 Columns)

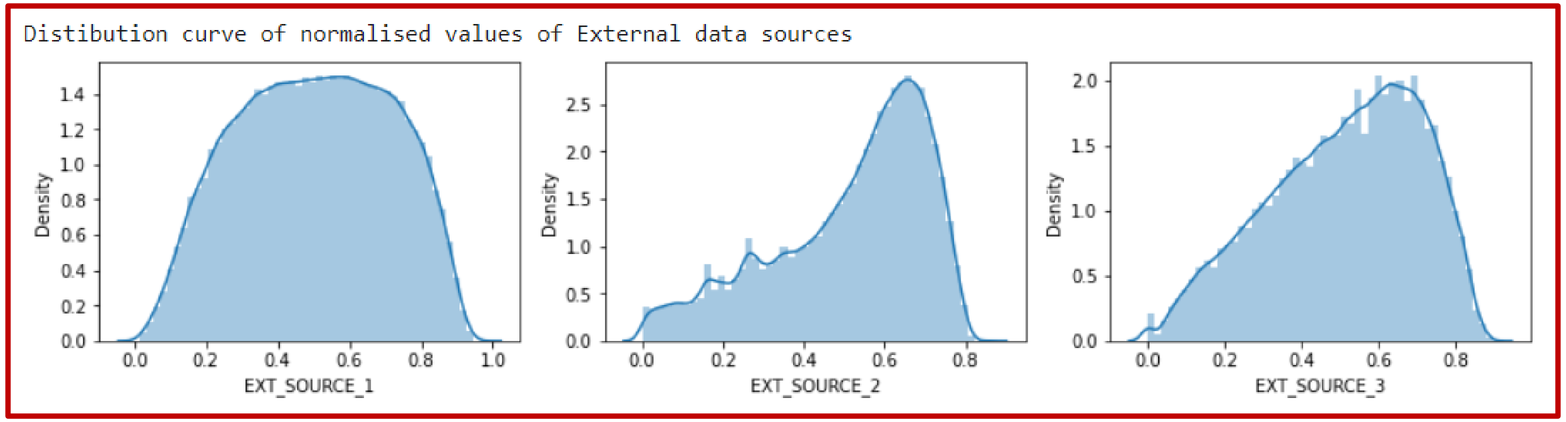
- Trend is same for defaulters and non-defaulters

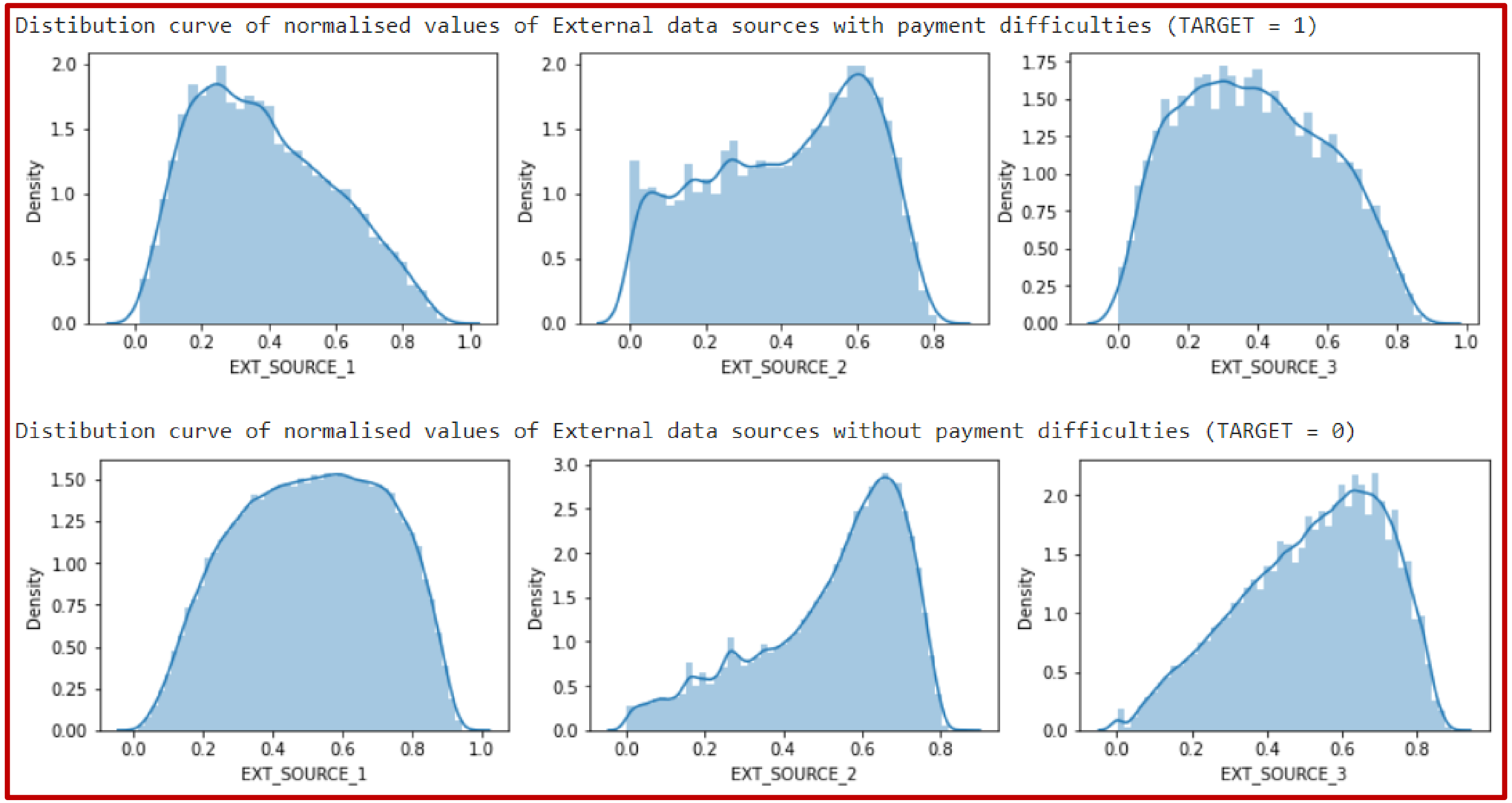




Value counts for Defaulters and Non-Defaulters

3. Data Analysis – External Source Rating (3 Columns)





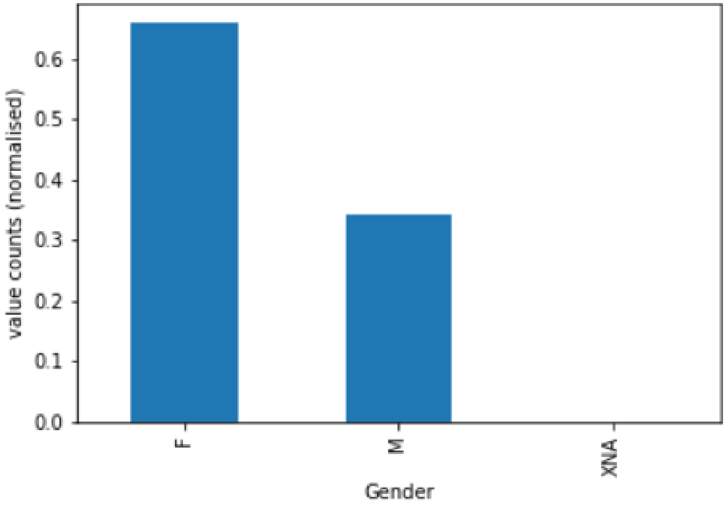
- Based on the score of all three external sources, applicants with a higher score align with the overall trend of score showing that they are less likely to have difficulty with payment
- Whereas, applicants with payment difficulties have a distribution curve that shows an opposite trend of the distribution curve of scores of the external sources
- **Defaulters tend to be lower rating provided by external sources 1 and 3**

3. Data Analysis – Personal information

CODE_GENDER

- Female applicants(65%) are significantly more than male applicants(34%)
- Missing values form a very small part, so I will ignore them
- Male applicants have a higher default rate than female applicants

```
F       65.835975
M       34.162711
XNA      0.001314
Name: CODE_GENDER, dtype: float64
```

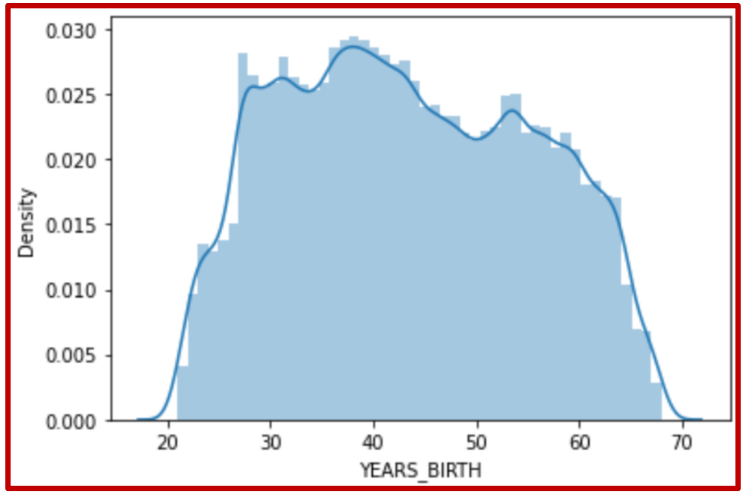


Rate of default vs Gender

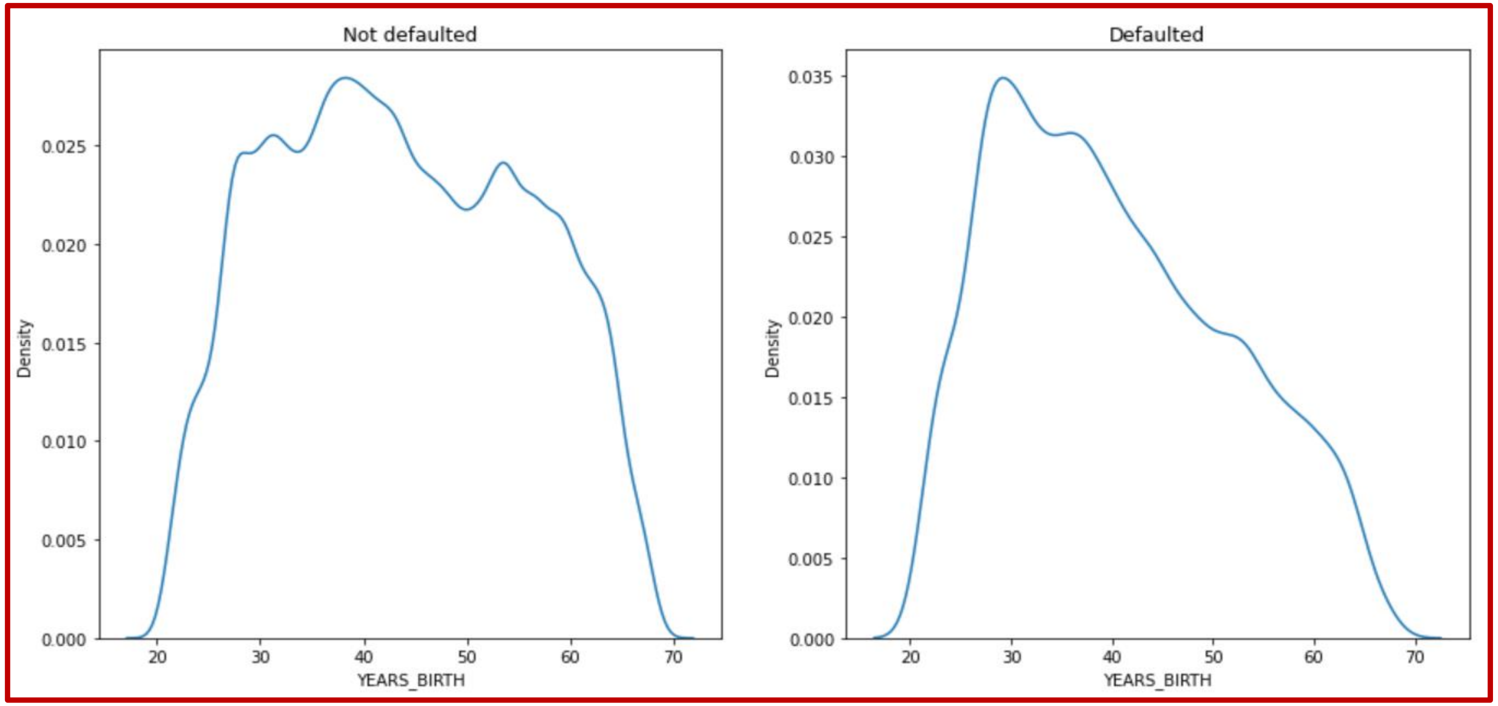
```
Default rates within each gender category are -

CODE_GENDER
F        7.012195
M       10.191071
XNA      0.000000
Name: TARGET, dtype: float64
```

3. Data Analysis – Personal information

YEARS_BIRTH (Age)



- The age of applicants is rather evenly distributed between 30-60 years

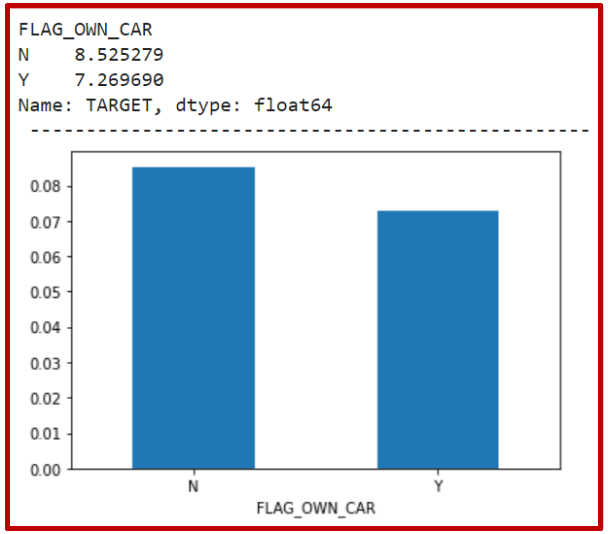


- Applicants of the younger age group tend default on their payments more than their older counterparts. As seen in the second plot(right) titled 'Defaulted', the age of the defaulters is inclined towards the younger applicants

3. Data Analysis – Personal information

FLAG_OWN_CAR

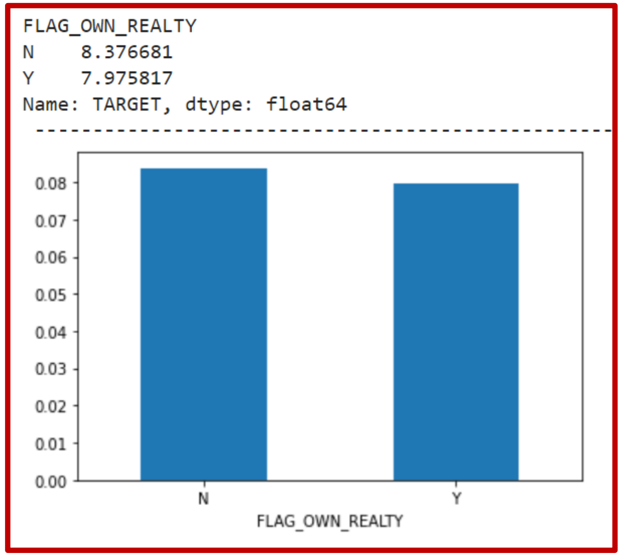
- Applicants who do not own a car have a slightly higher default rate



Rate of default vs FLAG_OWN_CAR

FLAG_OWN_REALTY

- Applicants who do not own Realty have a slightly higher default rate

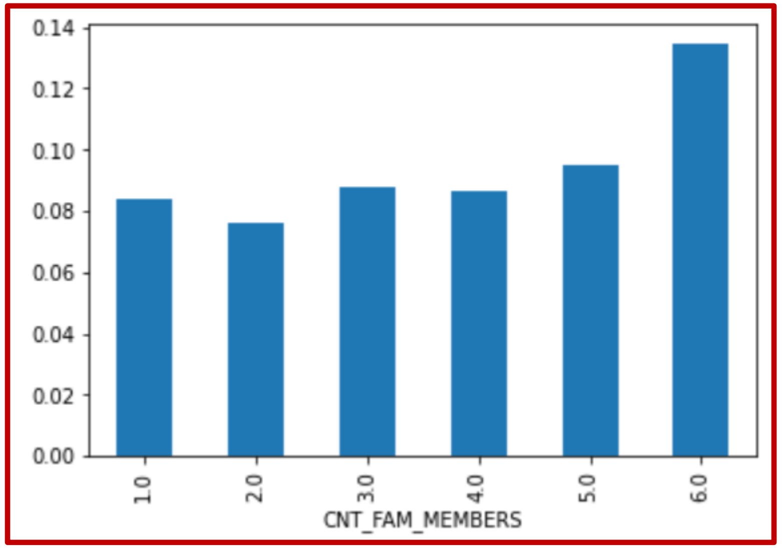


Rate of default vs FLAG_OWN_REALTY

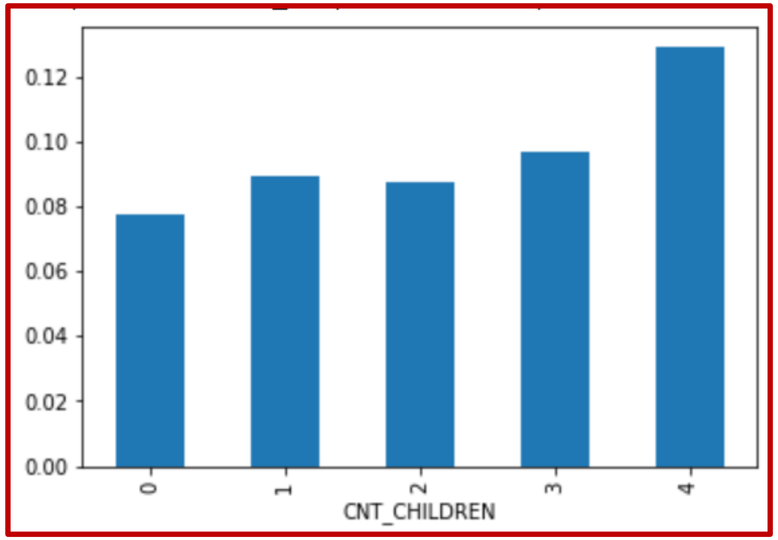
3. Data Analysis – Personal information

Family Details

- More the number of children/family members, higher is the default rate
- 7.73% of Applicant with 0 children have defaulted in payments and this percentage increases as count of children increases from 1-4. 12.9% of Applicant with 4 children have defaulted in payments
- 8.40% of Applicant with 1 Family members have defaulted in payments and this percentage increases as count of family members increases from 1-6. 12.46% of Applicant with 6 Family members have defaulted in payments



Rate of default vs CNT_FAM_MEMBERS

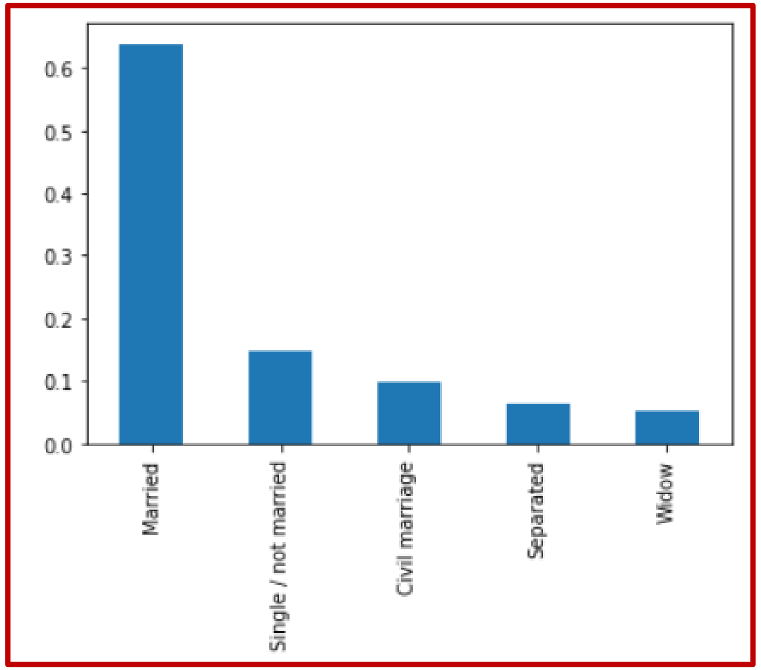


Rate of default vs CNT_CHILDREN

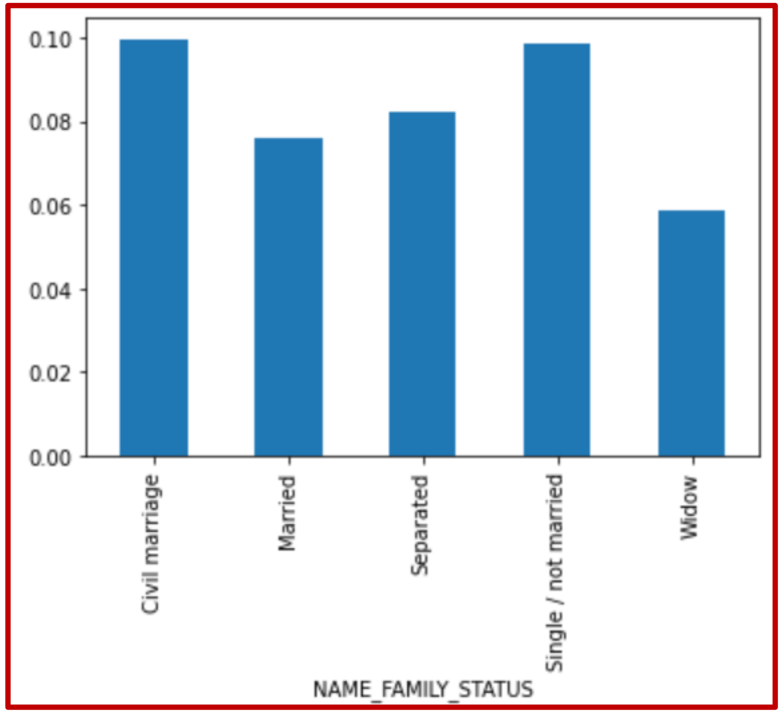
3. Data Analysis – Personal information

Family Details

- Most of the loan applicants are married
- Civil Marriage and Single/Not Married have a slightly higher default rate. Marrier and separated have default rates lesser than that of civil marriage and single/not married. Widow have the lowest default rate



Value count distribution of NAME_FAMILY_STATUS

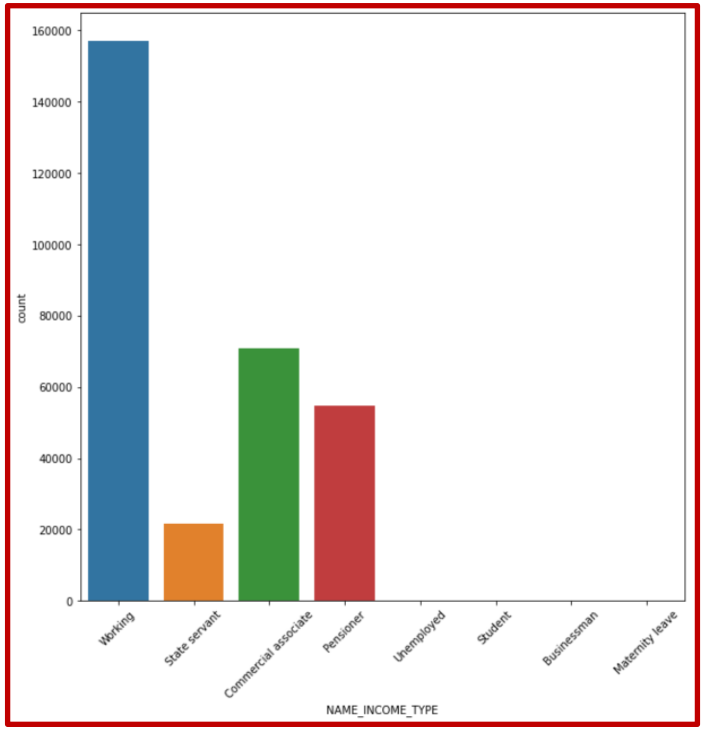


Rate of default vs NAME_FAMILY_STATUS

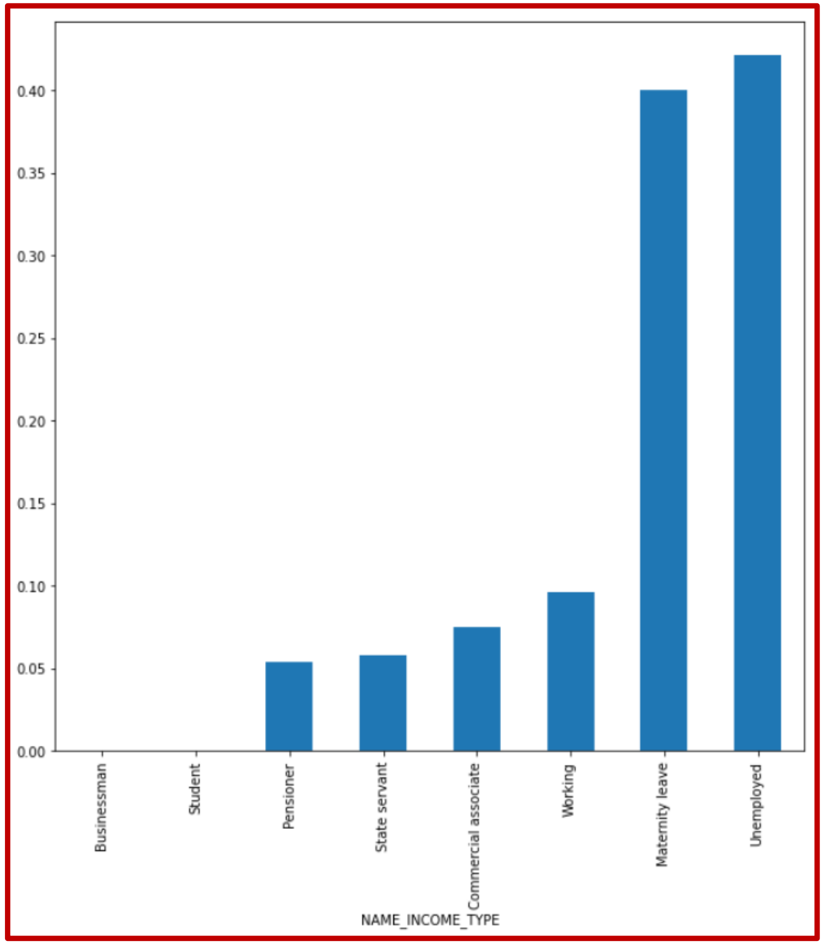
3. Data Analysis – Personal information

Income and Occupation details

- NAME_INCOME_TYPE - Most of the loan applicants are of working income type
- Unemployed, Student, Businessman and Maternity leave are in total <1% of the entire data
- Default rates are highest for unemployed and decreased in the order Maternity>Working>Commercial associate>State Servant> Pensioner
- There is no pattern evident in the occupation type and organisation type columns to derive any inferences



Value count distribution of NAME_INCOME_TYPE



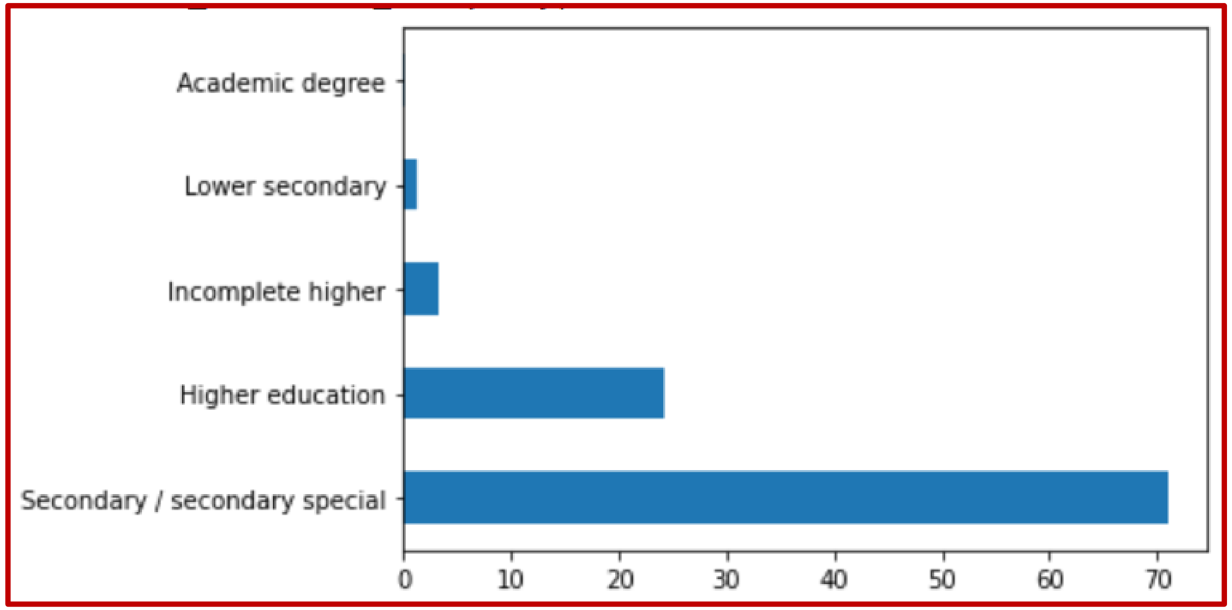
Rate of default vs NAME_INCOME_TYPE

3. Data Analysis – Personal information

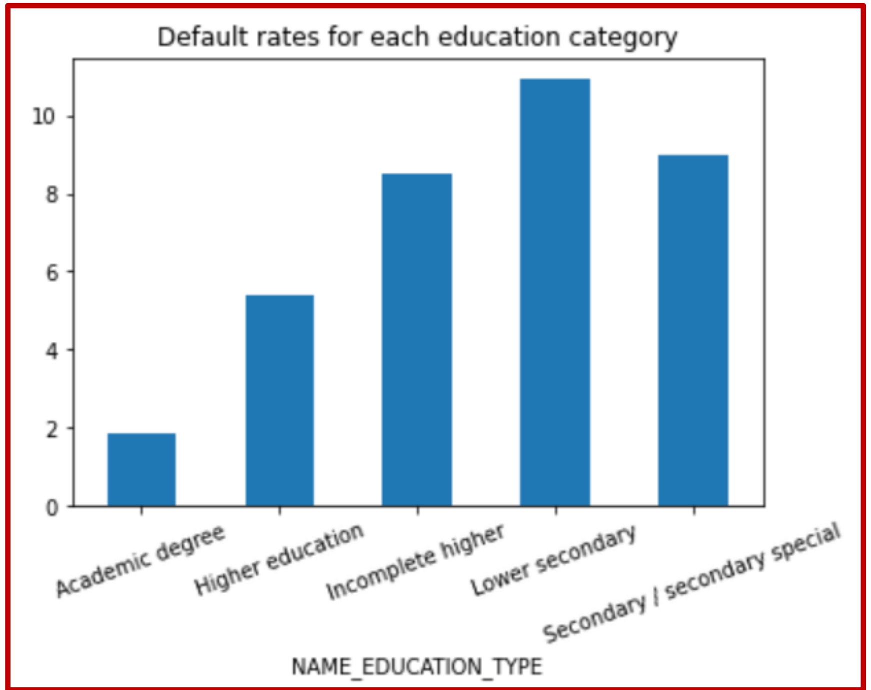
Education details

- Negligible percentage of applicants have an Academic degree
- 71% of applicants have completed Secondary / secondary special education
- 24% of applicants have completed Higher education
- Very few applicants are from the group that have incomplete higher or just completed lower education

I will neglect Lower secondary and Academic degree in further analysis since there is no substantial data



Value counts distribution for NAME_EDUCATION_TYPE



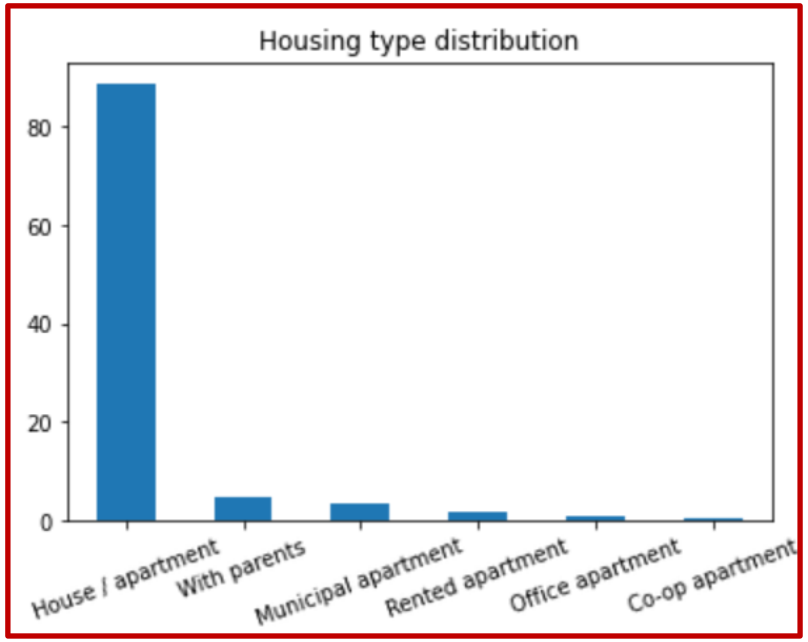
Higher the level the information, less is the default rate

- Applicants who have completed only Lower secondary education show the highest default rates (~10%)
- Applicants who have completed Secondary / secondary special or incomplete higher education also have a high default rate of 8-9%
- Higher education applicants have a low default rate and Academic degree applicants even more

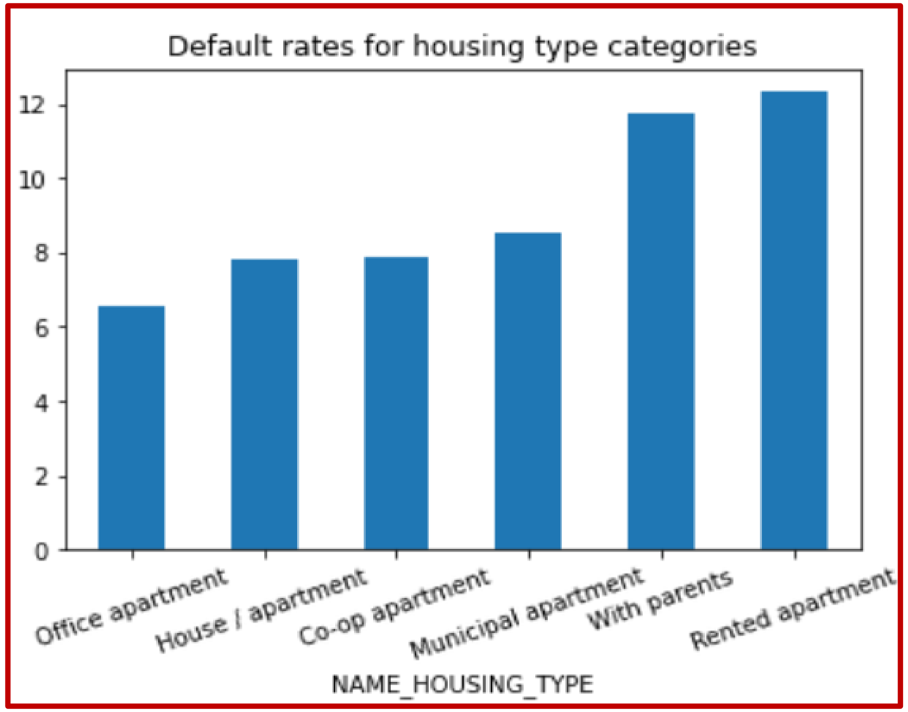
3. Data Analysis – Personal information

NAME_HOUSING_TYPE

- Most applicants(89%) have their own house/apartment or live in a co-op apartment, and other categories have a substantially low percentage of applicants
- Around 6% of applicants live in housing provided by a 3rd party like parents or government or rented
- A very small percentage of applicants live in office provided apartments



Value counts distribution for NAME_HOUSING_TYPE



- Applicants living with parents or in rented apartments have a high default rate of ~12%
- Applicants living in Municipal apartments have a default rate of 8.5%
- The applicants of other categories have a lesser default rate of ~7%
- Applicants living in office apartments have the lowest default rate of 6.5%

4. Summary

- The data contains 122 variables (columns) including the TARGET variable
- The data is extremely imbalanced. The number of defaulters is very less compared to the entire population (1 in 11)

1. Loan Information –

- Although the Revolving loans have a lesser default rate of 5.5%, its contribution to the entire data is very less to form analysis
- AMT_CREDIT and AMT_GOODS_PRICE have a linear relationship and for lower range of these variables, the defaulters are less

2. Housing Information –

- All the columns have 47%-70% data missing, hence dropped

3. Documents –

- Documents were mostly not submitted and there is no impact of these columns on the TARGET variable

4. Region Rating –

- Applicants who belong to REGION_RATING_CLIENT_W_CITY rated as 1 have the lowest rate of default (4.86%) and those rated 3 have the highest rate of default (11.44%)

5. Credit Bureau Information –

- Most responses for these columns is zero(0), hence I will drop these columns. Except AMT_REQ_CREDIT_BUREAU_YEAR
- AMT_REQ_CREDIT_BUREAU_YEAR and AMT_CREDIT have a linear relation
- Applicants who defaulted did not have more than enquiries 10 enquiries in the past year(Except few)

6. Social Circle Information –

- There is no impact of these columns on TARGET

7. External Source Rating –

- Defaulters tend to be of lower rating provided by external sources 1 and 3

8. Personal Information –

- **CODE_GENDER**: Male applicants have a higher default rate than female applicants
- **YEARS_CODE_GENDER: BIRTH**: Applicants of the younger age group tend default on their payments more than their older counterparts
- **FLAG_OWN_CAR/FLAG_OWN_REALTY:** Applicants who do not own a car/Realty have a slightly higher default rate
- **Family Details:** More the number of children/family members, higher is the default rate
- **Income and Occupation: -** Most of the loan applicants are of working income type. Default rates are highest for unemployed and decreased in the order Maternity>Working>Commercial associate>State Servant> Pensioner
- **Education:** Applicants who have completed only Lower secondary education show the highest default rates (~10%)
- **Housing Type:** Applicants living with parents or in rented apartments have a high default rate of ~12%

4. Summary

The TARGET variable is highly influenced by the following variables

- Gender - CODE_GENDER
- Family details - CNT_CHILDREN, CNT_FAM_MEMBERS
- Income and occupation - NAME_INCOME_TYPE
- Education - NAME_EDUCATION_TYPE
- External Source Rating – EXT_SOURCE_1, EXT_SOURCE_2

Previous Application Data Analysis

0. Reading the data and introduction

- The data in previous_data.csv is read and stored in data frame ‘prev_data’
- Previous data contains 37 columns out of the data type breakdown is -

1. Float - 15 columns
2. Int - 6 columns
3. Object - 16 columns

- There are 1670214 rows
- There is no need for the columns 'SK_ID_PREV'. Drop this column
- There are inconsistencies in the number of non-null values of each column

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1670214 entries, 0 to 1670213
Data columns (total 37 columns):
 #   Column                       Non-Null Count    Dtype
---  ------                       --------------    -----
 0   SK_ID_PREV                   1670214 non-null  int64
 1   SK_ID_CURR                   1670214 non-null  int64
 2   NAME_CONTRACT_TYPE           1670214 non-null  object
 3   AMT_ANNUITY                  1297979 non-null  float64
 4   AMT_APPLICATION              1670214 non-null  float64
 5   AMT_CREDIT                   1670213 non-null  float64
 6   AMT_DOWN_PAYMENT             774370 non-null   float64
 7   AMT_GOODS_PRICE              1284699 non-null  float64
 8   WEEKDAY_APPR_PROCESS_START   1670214 non-null  object
 9   HOUR_APPR_PROCESS_START      1670214 non-null  int64
 10  FLAG_LAST_APPL_PER_CONTRACT  1670214 non-null  object
 11  NFLAG_LAST_APPL_IN_DAY       1670214 non-null  int64
 12  RATE_DOWN_PAYMENT            774370 non-null   float64
 13  RATE_INTEREST_PRIMARY        5951 non-null     float64
 14  RATE_INTEREST_PRIVILEGED     5951 non-null     float64
 15  NAME_CASH_LOAN_PURPOSE       1670214 non-null  object
 16  NAME_CONTRACT_STATUS         1670214 non-null  object
 17  DAYS_DECISION                1670214 non-null  int64
 18  NAME_PAYMENT_TYPE            1670214 non-null  object
 19  CODE_REJECT_REASON           1670214 non-null  object
 20  NAME_TYPE_SUITE              849809 non-null   object
 21  NAME_CLIENT_TYPE             1670214 non-null  object
 22  NAME_GOODS_CATEGORY          1670214 non-null  object
 23  NAME_PORTFOLIO               1670214 non-null  object
 24  NAME_PRODUCT_TYPE            1670214 non-null  object
 25  CHANNEL_TYPE                 1670214 non-null  object
 26  SELLERPLACE_AREA             1670214 non-null  int64
 27  NAME_SELLER_INDUSTRY         1670214 non-null  object
 28  CNT_PAYMENT                  1297984 non-null  float64
 29  NAME_YIELD_GROUP             1670214 non-null  object
 30  PRODUCT_COMBINATION          1669868 non-null  object
 31  DAYS_FIRST_DRAWING           997149 non-null   float64
 32  DAYS_FIRST_DUE               997149 non-null   float64
 33  DAYS_LAST_DUE_1ST_VERSION    997149 non-null   float64
 34  DAYS_LAST_DUE                997149 non-null   float64
 35  DAYS_TERMINATION             997149 non-null   float64
 36  NFLAG_INSURED_ON_APPROVAL    997149 non-null   float64
dtypes: float64(15), int64(6), object(16)
memory usage: 471.5+ MB
```

Information on ‘pred_data’ columns

1. Data Cleaning - Handling missing data

Columns with majority null values

- Over 99% the data in Columns 'RATE_INTEREST_PRIVILEGED' and 'RATE_INTEREST_PRIMARY' have null values. Hence drop these columns
- There are two columns which have less than 1% missing values - AMT_CREDIT and PRODUCT_COMBINATION. I will drop these columns as well

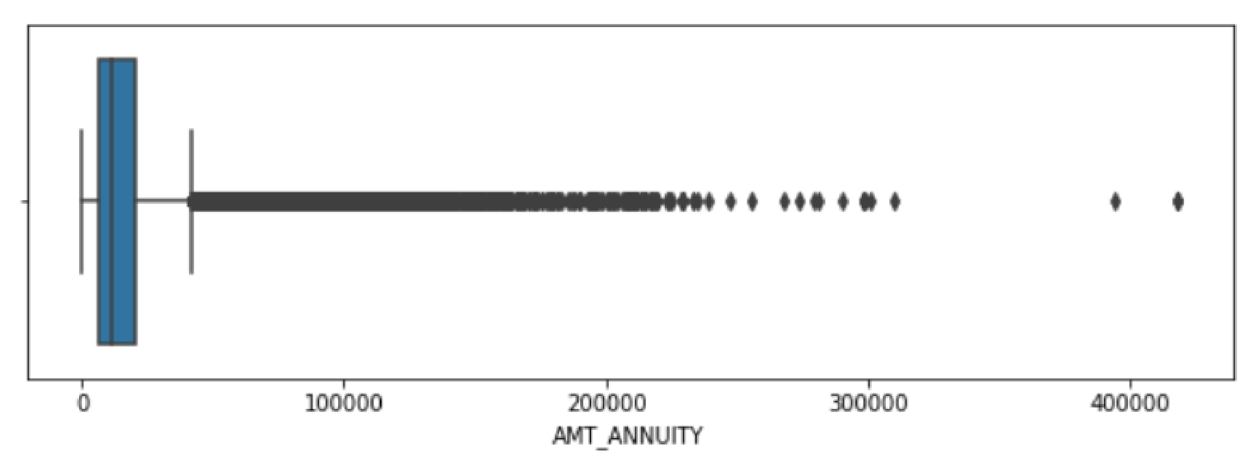
```
RATE_INTEREST_PRIMARY          99.643698
RATE_INTEREST_PRIVILEGED       99.643698
AMT_DOWN_PAYMENT               53.636480
RATE_DOWN_PAYMENT              53.636480
NAME_TYPE_SUITE                49.119754
DAYS_FIRST_DRAWING             40.298129
DAYS_FIRST_DUE                 40.298129
DAYS_LAST_DUE_1ST_VERSION      40.298129
DAYS_LAST_DUE                  40.298129
DAYS_TERMINATION               40.298129
NFLAG_INSURED_ON_APPROVAL      40.298129
AMT_GOODS_PRICE                23.081773
AMT_ANNUITY                    22.286665
CNT_PAYMENT                    22.286366
PRODUCT_COMBINATION             0.020716
AMT_CREDIT                      0.000060
dtype: float64
```

Graph showing % Null values in columns having >0% null values

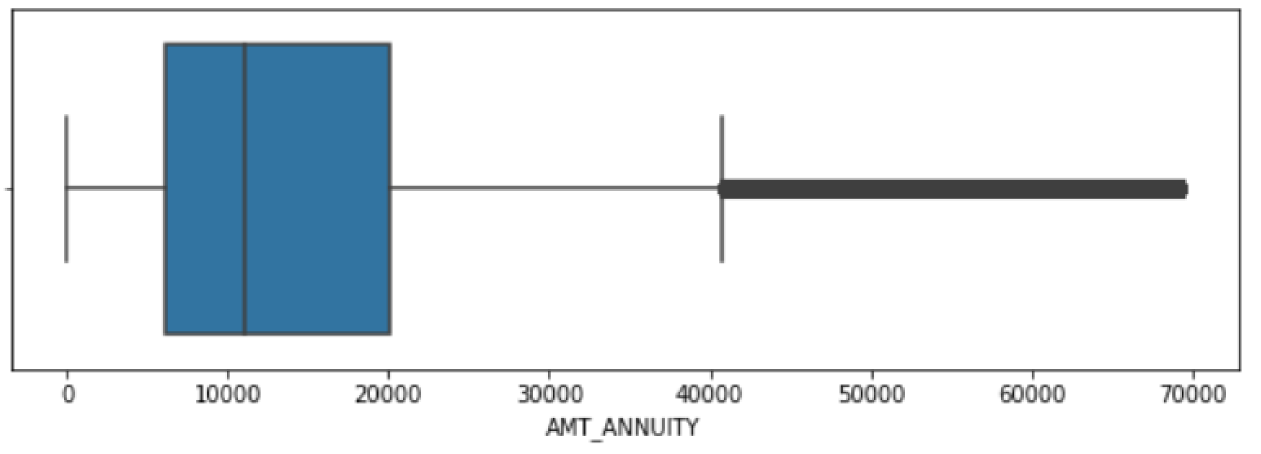
1. Data Cleaning - Handling Outliers

AMT_ANNUITY

- There are extreme outliers in the AMT_ANNUITY column
- I have capped the value to 99% Percentile of the data



Boxplot of AMT_ANNUITY

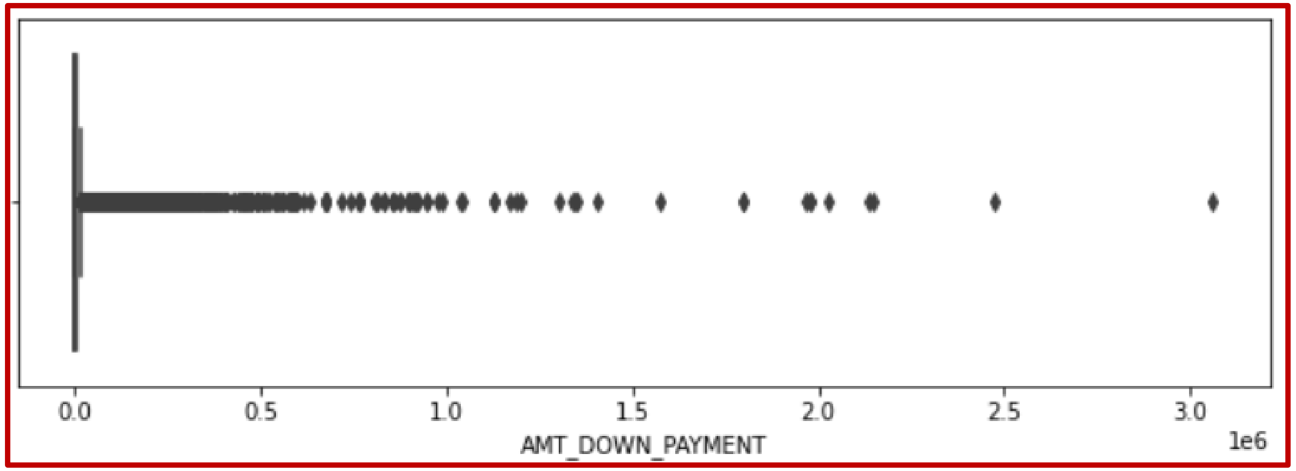


Boxplot of AMT_ANNUITY after capping the values

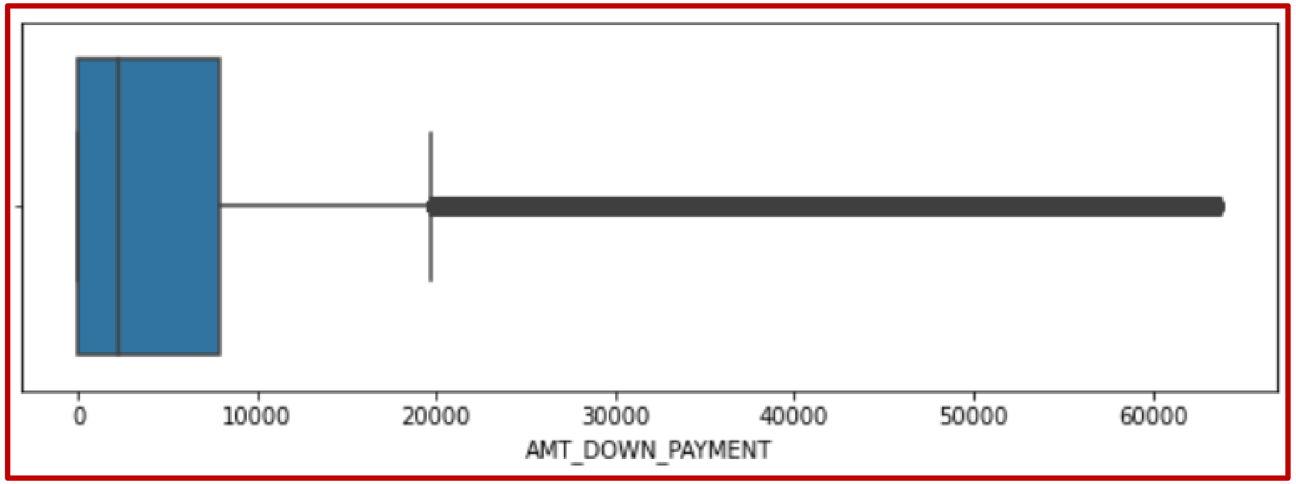
1. Data Cleaning - Handling Outliers

AMT_DOWN_PAYMENT

- There are extreme outliers in the AMT_DOWN_PAYMENT column
- I have capped the value to 99% Percentile of the data



Boxplot of AMT_DOWN_PAYMENT



Boxplot of AMT_ DOWN_PAYMENT after capping the values

2. Merging data

Merging data from app_data_raw into prev_data

- There are 8 common columns between the data frames 'app_data_raw' and 'prev_data' –

1. NAME_TYPE_SUITE
2. AMT_GOODS_PRICE
3. AMT_ANNUITY
4. NAME_CONTRACT_TYPE
5. WEEKDAY_APPR_PROCESS_START
6. SK_ID_CURR
7. AMT_CREDIT
8. HOUR_APPR_PROCESS_START

- Only 2 columns would be required from app_data_raw - SK_ID_CURR and TARGET. The key identifier being SK_ID_CURR

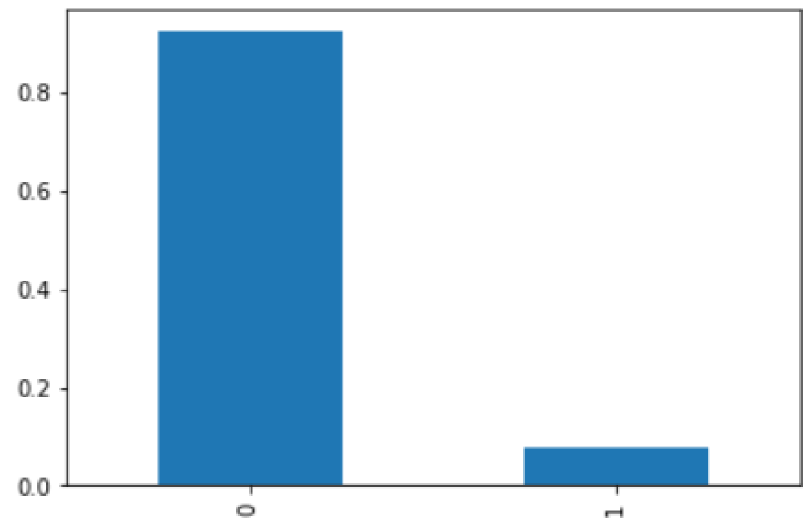
- The data frames are merged and stored in 'merged_data'

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1429841 entries, 0 to 1429840
Data columns (total 35 columns):
 #   Column                       Non-Null Count    Dtype
---  ------                       --------------    -----
 0   SK_ID_CURR                   1429841 non-null  int64
 1   TARGET                       1429841 non-null  int64
 2   NAME_CONTRACT_TYPE           1413387 non-null  object
 3   AMT_ANNUITY                  1106482 non-null  float64
 4   AMT_APPLICATION              1413387 non-null  float64
 5   AMT_CREDIT                   1413387 non-null  float64
 6   AMT_DOWN_PAYMENT             664161 non-null   float64
 7   AMT_GOODS_PRICE              1094176 non-null  float64
 8   WEEKDAY_APPR_PROCESS_START   1413387 non-null  object
 9   HOUR_APPR_PROCESS_START      1413387 non-null  float64
 10  FLAG_LAST_APPL_PER_CONTRACT  1413387 non-null  object
 11  NFLAG_LAST_APPL_IN_DAY       1413387 non-null  float64
 12  RATE_DOWN_PAYMENT            664161 non-null   float64
 13  NAME_CASH_LOAN_PURPOSE       1413387 non-null  object
 14  NAME_CONTRACT_STATUS         1413387 non-null  object
 15  DAYS_DECISION                1413387 non-null  float64
 16  NAME_PAYMENT_TYPE            1413387 non-null  object
 17  CODE_REJECT_REASON           1413387 non-null  object
 18  NAME_TYPE_SUITE              719029 non-null   object
 19  NAME_CLIENT_TYPE             1413387 non-null  object
 20  NAME_GOODS_CATEGORY          1413387 non-null  object
 21  NAME_PORTFOLIO               1413387 non-null  object
 22  NAME_PRODUCT_TYPE            1413387 non-null  object
 23  CHANNEL_TYPE                 1413387 non-null  object
 24  SELLERPLACE_AREA             1413387 non-null  float64
 25  NAME_SELLER_INDUSTRY         1413387 non-null  object
 26  CNT_PAYMENT                  1106487 non-null  float64
 27  NAME_YIELD_GROUP             1413387 non-null  object
 28  PRODUCT_COMBINATION          1413387 non-null  object
 29  DAYS_FIRST_DRAWING           852595 non-null   float64
 30  DAYS_FIRST_DUE               852595 non-null   float64
 31  DAYS_LAST_DUE_1ST_VERSION    852595 non-null   float64
 32  DAYS_LAST_DUE                852595 non-null   float64
 33  DAYS_TERMINATION             852595 non-null   float64
 34  NFLAG_INSURED_ON_APPROVAL    852595 non-null   float64
dtypes: float64(17), int64(2), object(16)
memory usage: 392.7+ MB
```

Information about the merged data frame

3. Data Analysis – Data Imbalance

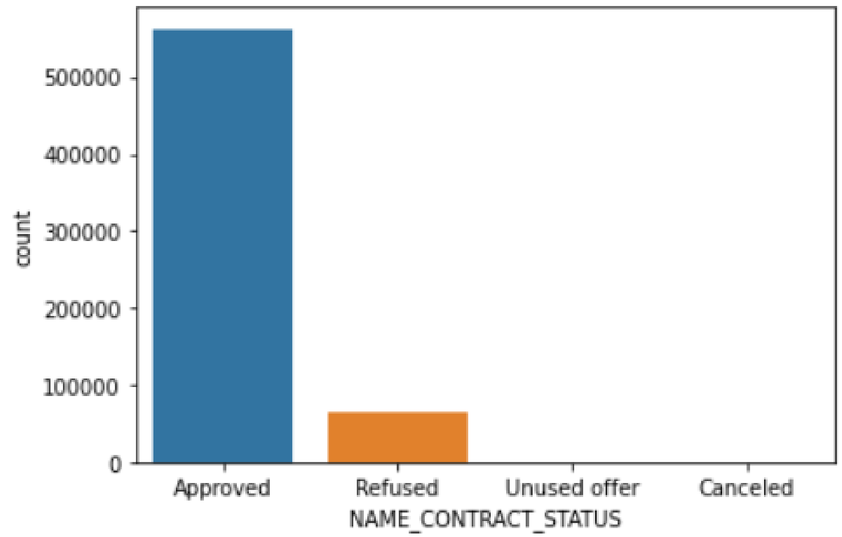
Data imbalance for TARGET




```
---------------------------------------------------------------
0    92.385143
1     7.614857
Name: TARGET, dtype: float64
```

- Data imbalance ratio between **Defaulters and Non-defaulters is nearly 7.61 : 92.38 (or) 1 : 12.13**

Data imbalance for NAME_CONTRACT_STATUS




```
------------------------------------------------------------
Approved        89.500997
Refused         10.325808
Unused offer     0.124484
Canceled         0.048711
Name: NAME_CONTRACT_STATUS, dtype: float64
```

- Data is highly imbalanced. Most of the applicants were either Approved or Rejected. Less than 1% were cancelled and unused offer

3. Data Analysis

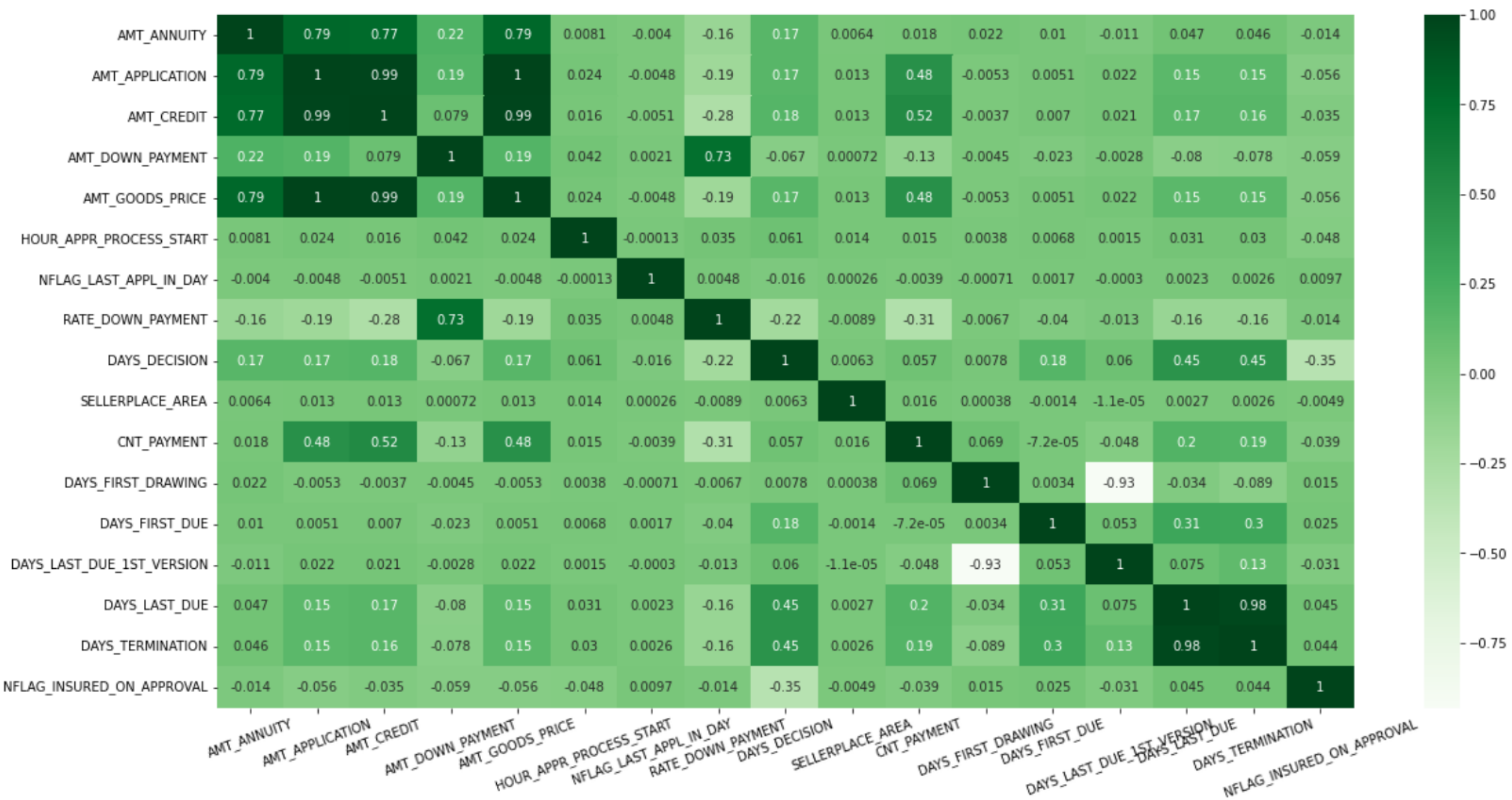
I have segmented the data into 2 parts –

1. **Numerical Data**
2. **Categorical Data**

3. Data Analysis – Numerical Data

Heatmap to check collinearity

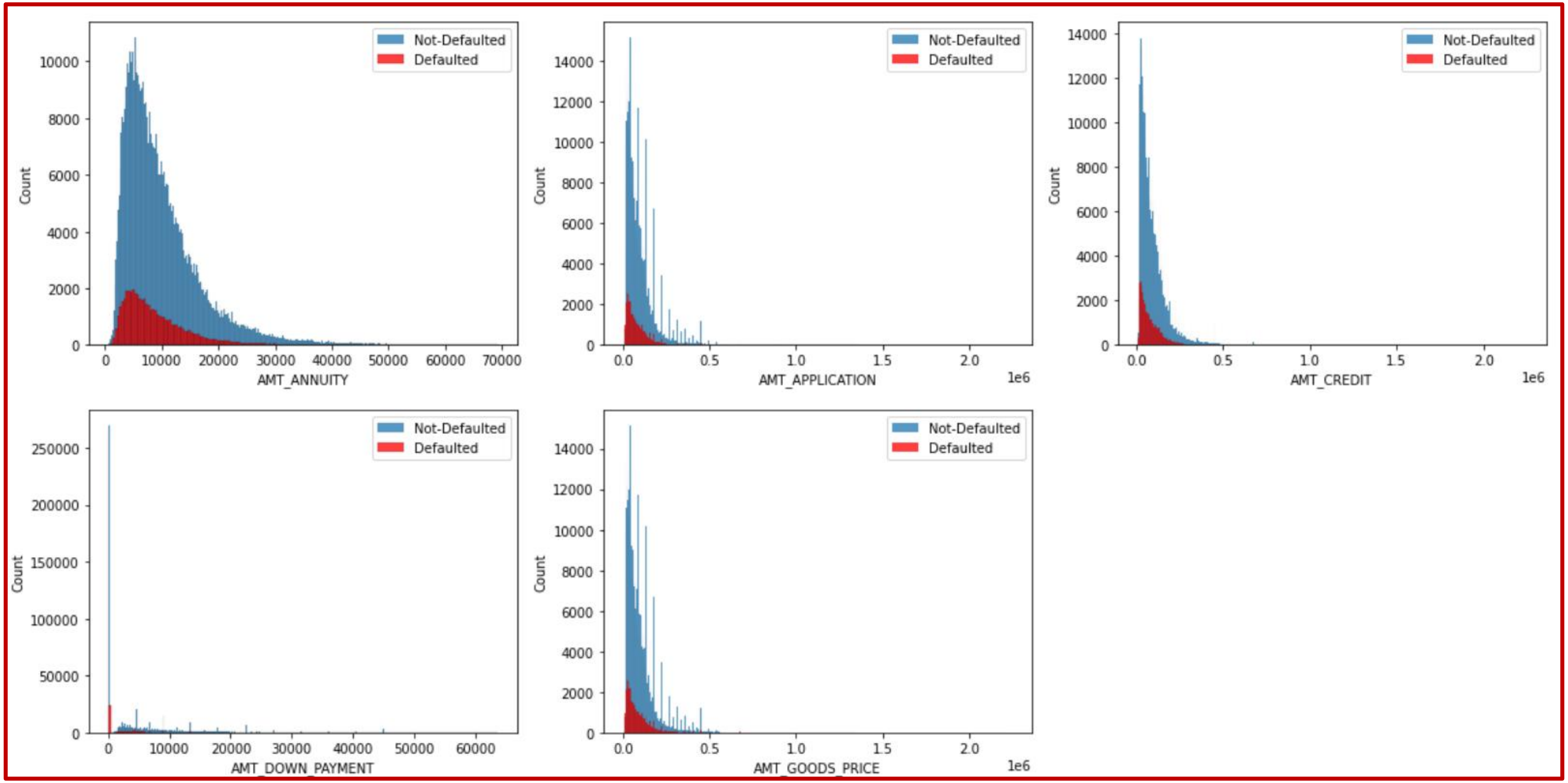
- AMT_ANNUITY, AMT_APPLICATION, AMT_CREDIT and AMT_GOODS_PRICE are highly correlated
- DAYS_TERMINATION and DAYS_LAST_DUE are highly correlated



3. Data Analysis – Numerical Data

Amount Related Columns

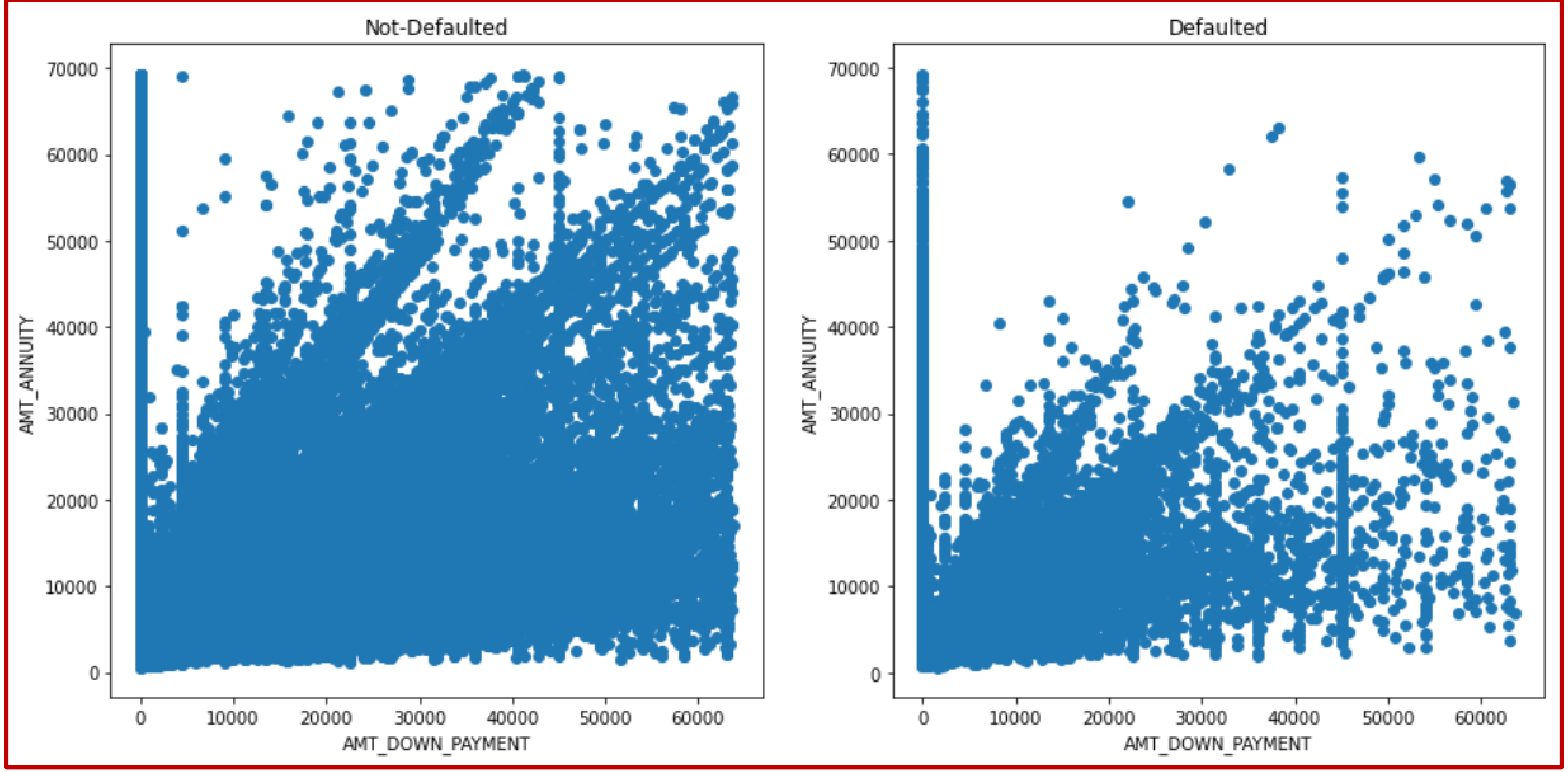
- Number of applicants are less for large loan amounts (AMT_CREDIT) which is highly correlated with the other columns



3. Data Analysis – Numerical Data

Amount Related Columns - **AMT_ DOWN_PAYMENT vs AMT_ANNUITY**

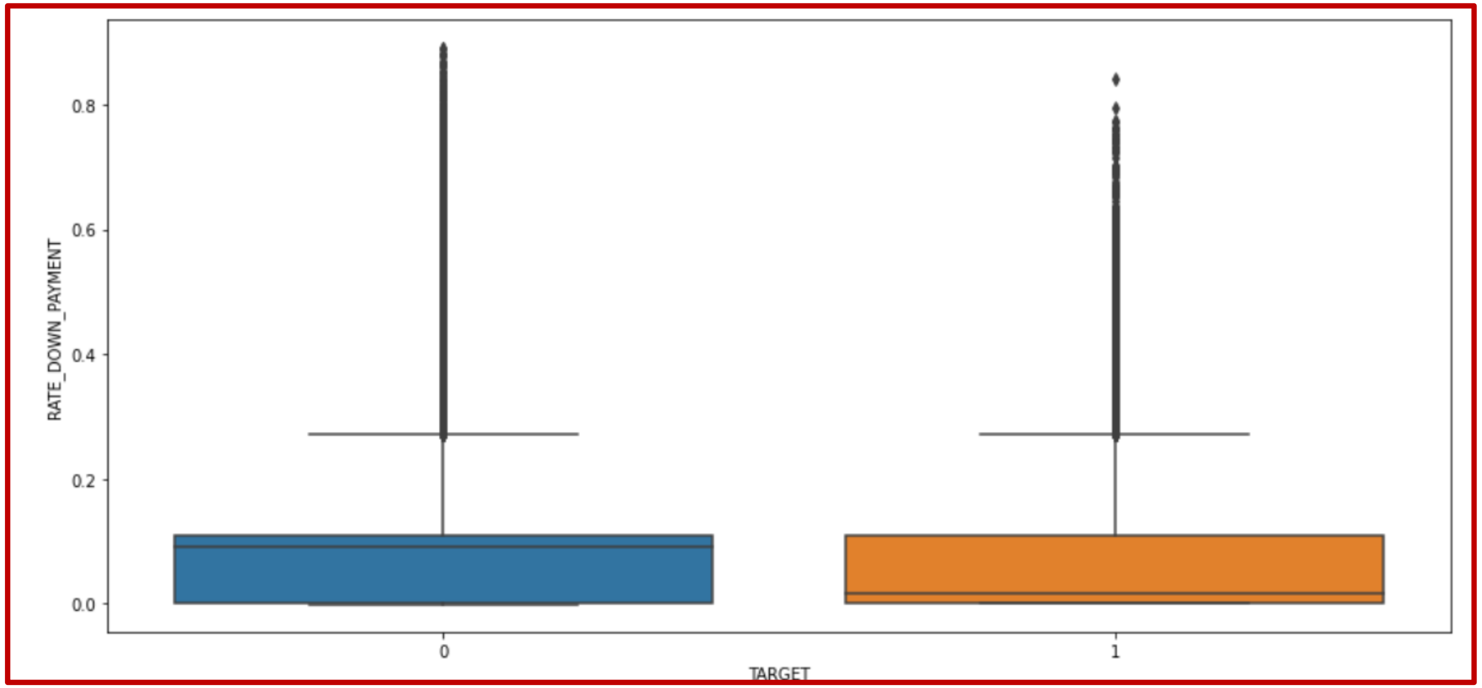
- There are less defaulters for higher values of AMT_DOWN_PAYMENT and AMT_ANNUITY



3. Data Analysis – Numerical Data

Amount Related Columns - **RATE_DOWN_PAYMENT**

- Median of defaulted group is higher. For lower rate of down payments, the cases of default are high.
- For lower rate of down payment, the rate of default is high
- Cannot comment on very high down payment rates(>0.5 or 50%) as these values are outliers



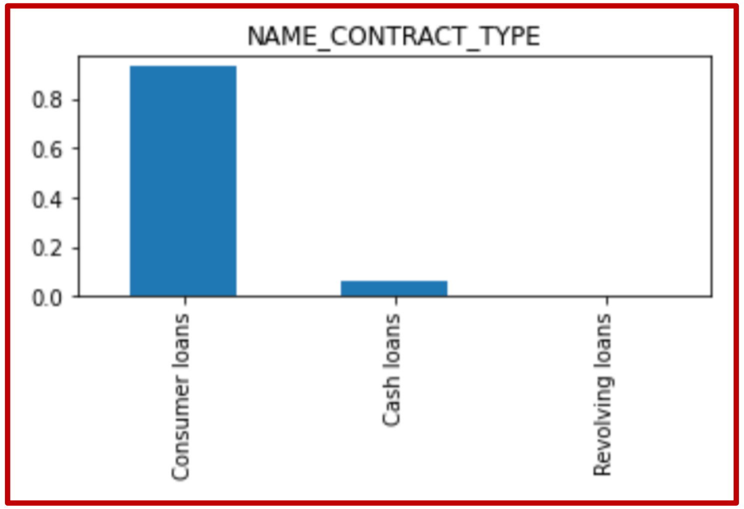
Boxplot for RATE_DOWN_PAYMENT for Defaulters and Non-Defaulters

Rate of default vs RATE_DOWN_PAYMENT_BINS

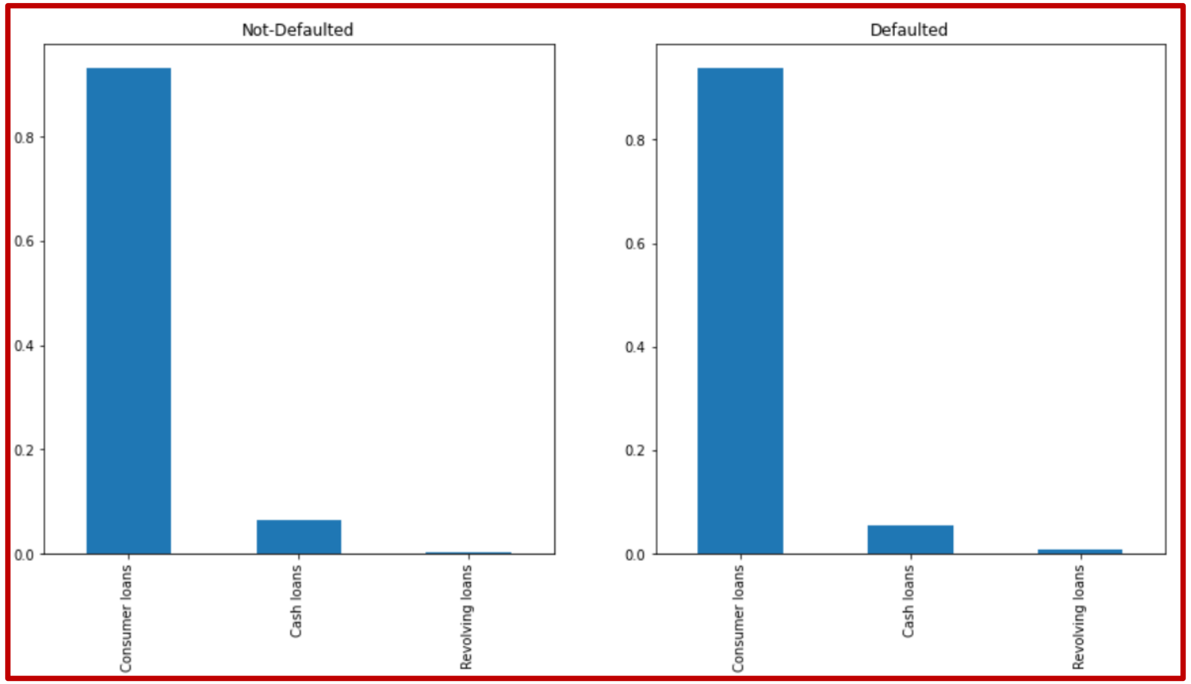
3. Data Analysis – Categorical Data

NAME_CONTRACT_TYPE

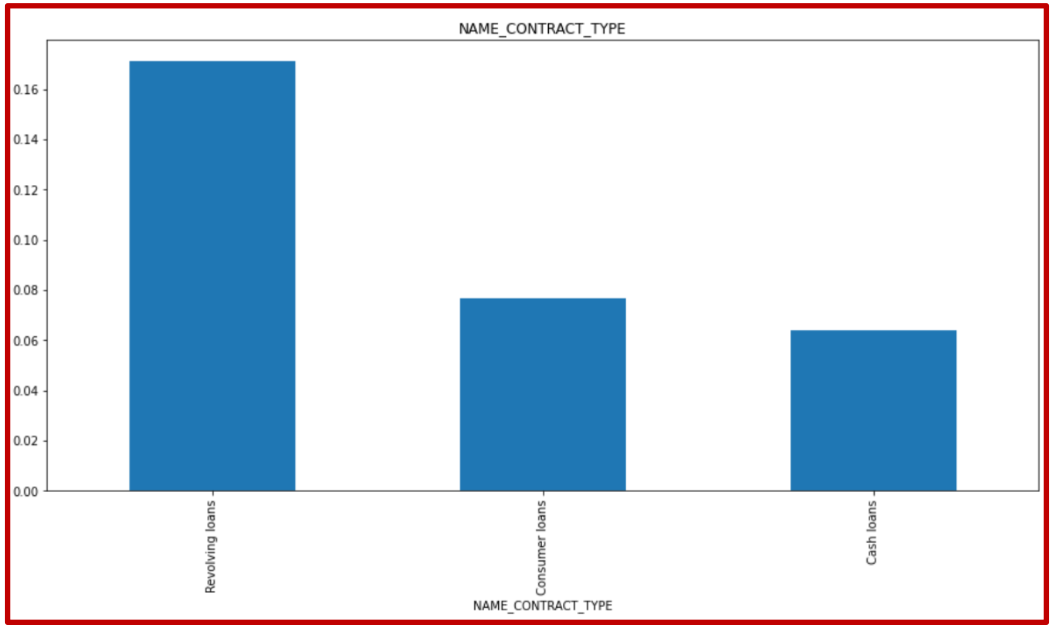
- Majority of previous loans are consumer loans
- Similar trend in both segments of TARGET
- Default rate is highest for Revolving Loans> consumer loans> Cash loans



Value count distribution



Value counts for Defaulters and Non-Defaulters

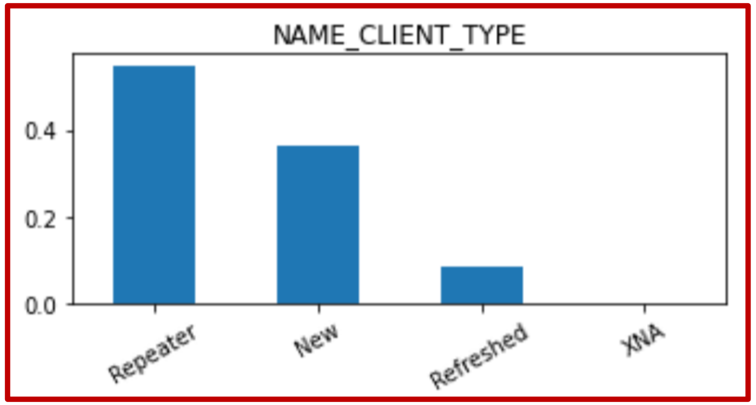


Rate of default

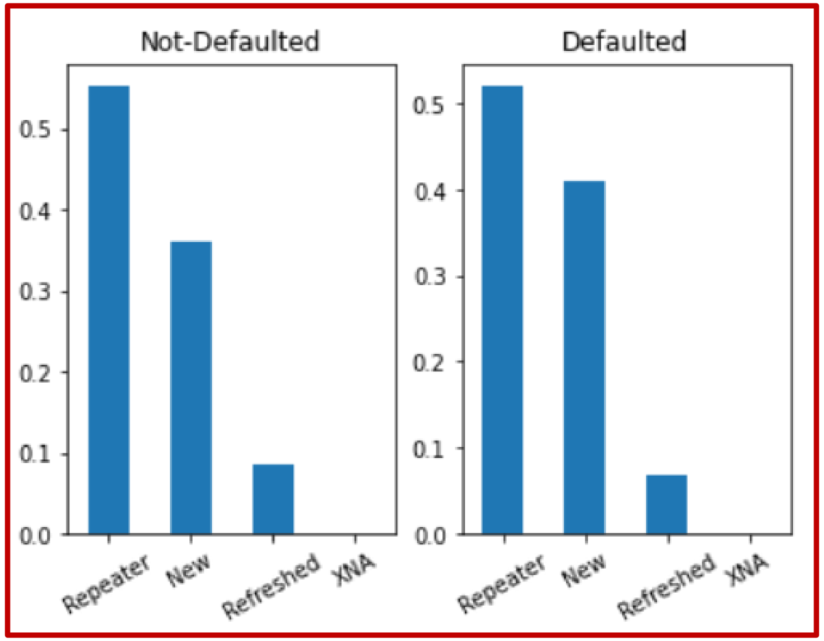
3. Data Analysis – Categorical Data

NAME_CLIENT_TYPE

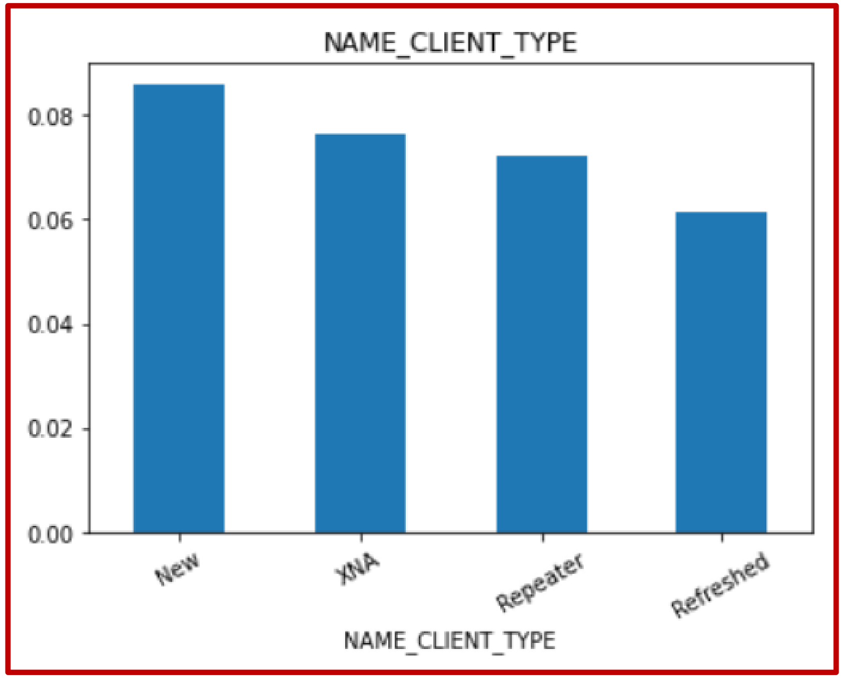
- Repeater is the most common client type and Refreshed is the least common
- New Client type are slightly more in the defaulter group
- Slight difference in default rates. New clients have the highest and Refreshed clients have the lowest



Value count distribution



Value counts for Defaulters and Non-Defaulters

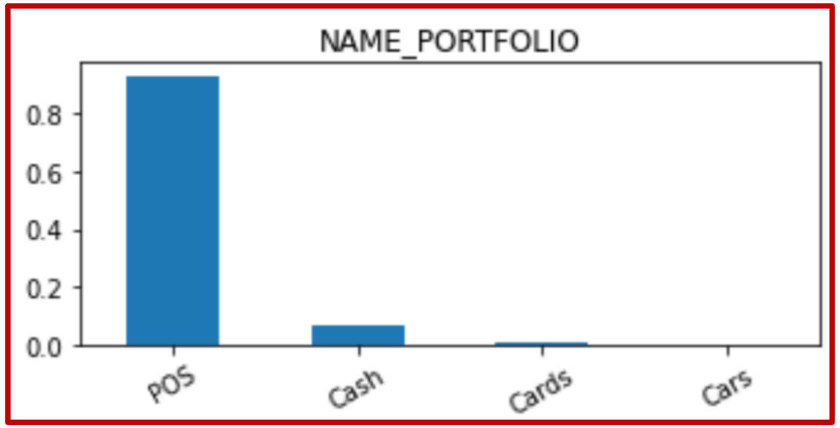


Rate of default

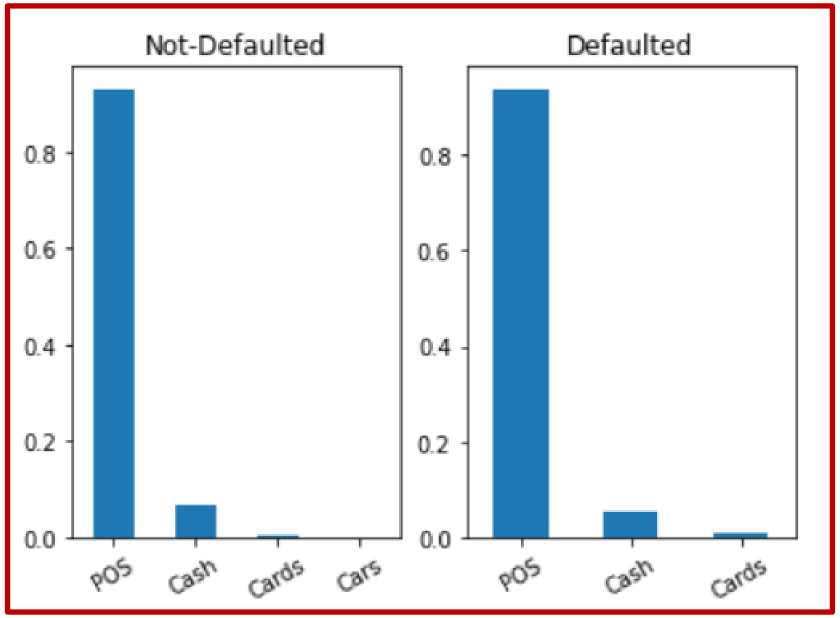
3. Data Analysis – Categorical Data

NAME_PORTFOLIO

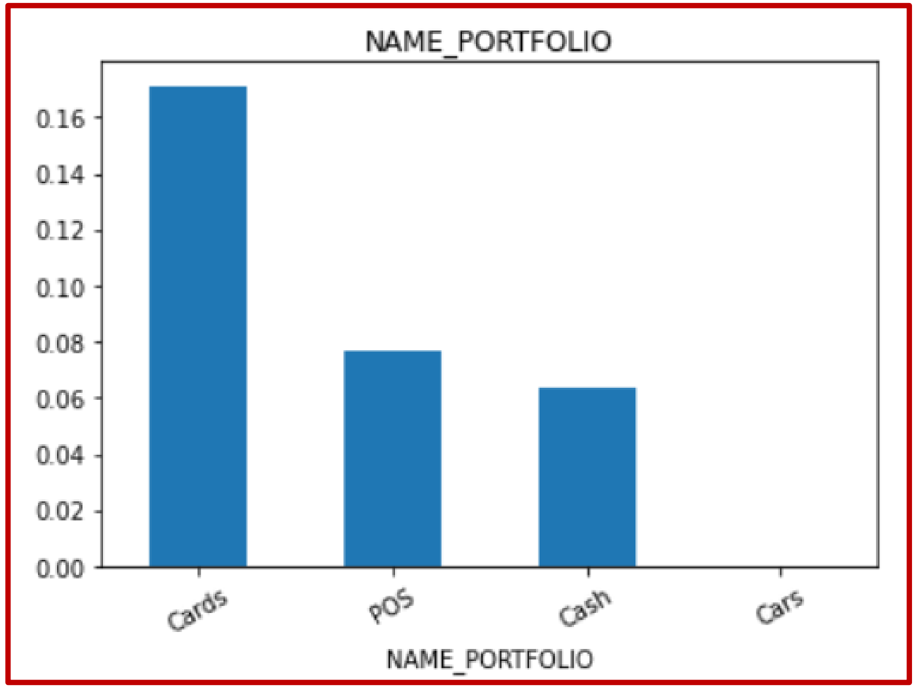
- Most of the Loan Portfolios are POS for previous loans(>80%)
- Similar trend in both segments of TARGET
- Card portfolio has a higher rate of defaulters than other portfolios(17%).



Value count distribution



Value counts for Defaulters and Non-Defaulters

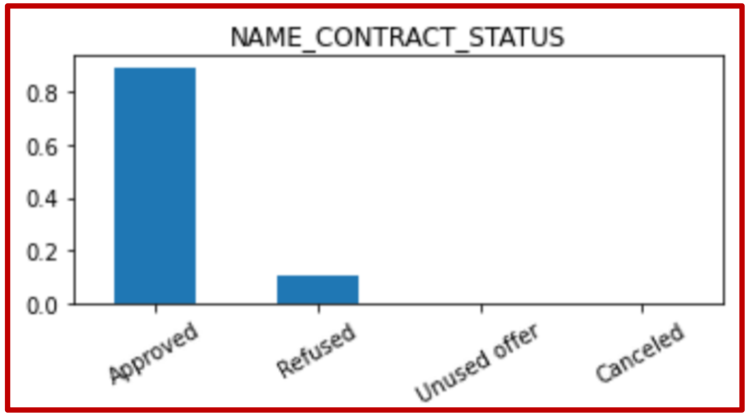


Rate of default

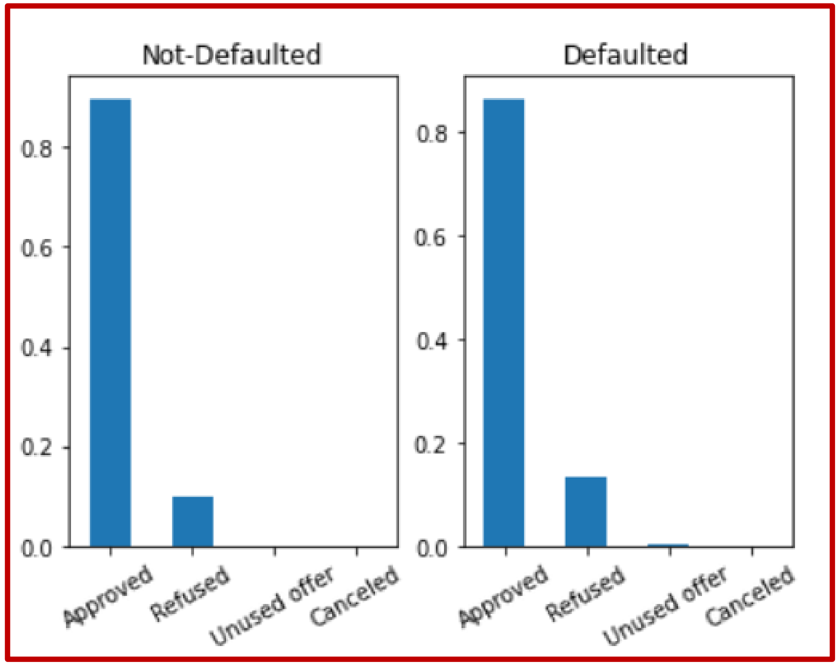
3. Data Analysis – Categorical Data

NAME_CONTRACT_STATUS

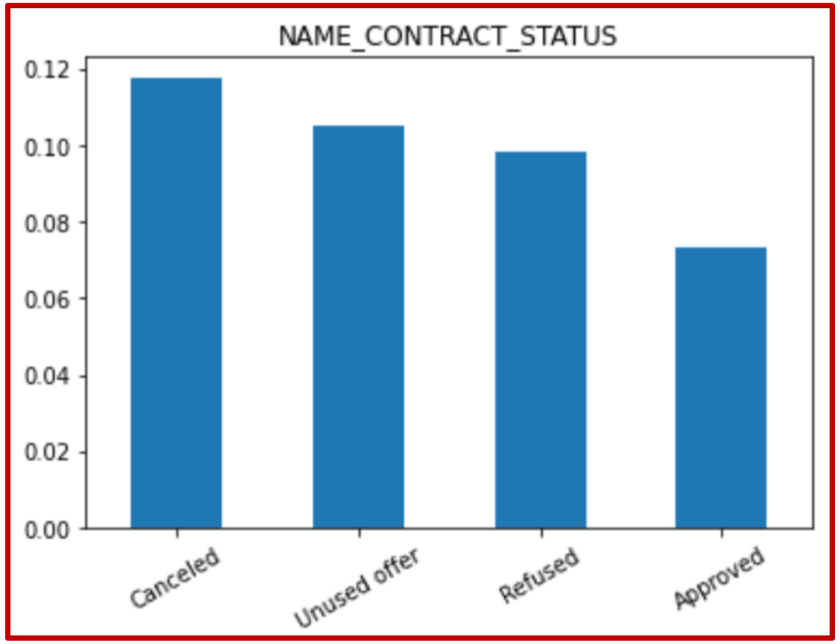
- 89% of the applications were approved, while 10% were refused. A very small portion of them were cancelled or unused
- The ratio of approved applicants for previous loans who have defaulted is slightly less than the ones who have not defaulted
- Applicants whose Previous loan was cancelled have the highest rate of defaulters(11%) but their count is very low so cannot infer anything here.
- Approved applicants have the lowest rate of defaulters(7.35%)



Value count distribution



Value counts for Defaulters and Non-Defaulters



Rate of default

4. Summary

- The data contains 37 variables (columns) including the NAME_CONTRACT_STATUS variable
- The data is extremely imbalanced. 89% of the applicants' loan applications were approved and 10% were refused. The remaining were unused or canceled

<u>Numerical Data –</u>

Amount Related Columns:
- Amount related columns - AMT_ANNUITY, AMT_APPLICATION, AMT_CREDIT and AMT_GOODS_PRICE are highly correlated

- There are less defaulters for higher values of AMT_DOWN_PAYMENT and AMT_ANNUITY

RATE_DOWN_PAYMENT: Median of defaulted group is higher. For lower values of RATE_DOWN_PAYMENT, the cases of default are high.

<u>Categorical Data –</u>

NAME_CONTRACT_TYPE: Default rate is highest for Revolving Loans> consumer loans> Cash loans

NAME_CLIENT_TYPE: New clients have the highest defaulter rates, and Refreshed clients have the lowest

NAME_CONTRACT_STATUS :
- 89% of the applications were approved, while 10% were refused. A very small portion of them were cancelled or unused
- The ratio of approved applicants for previous loans who have defaulted is slightly less than the ones who have not defaulted
- Applicants whose Previous loan was cancelled have the highest rate of defaulters(11%) but their count is very low so cannot infer anything here.
- Approved applicants have the lowest rate of defaulters(7.35%)