Capstone Project 1: Data Wrangling

Section 1. Data aggregation

In this section we briefly discuss treatment of variables in each of the 6 datasets provided by Home Credit.

1. Bureau data: This dataset contains all of the client's previous credits provided by other financial institutions that were reported to Credit Bureau (CB), such as number of days past due on CB credit at the time of application. For each current application ID, there are as many rows as number of credits the client had in Credit Bureau before the application date. That means each current application ID corresponds to multiple rows, and each row represents a different Bureau ID. This dataset has a total of 17 columns.

   At each row (Bureau ID) level:

   1) Drop CREDIT_CURRENCY column as the variable is not informative, 99% are values of 1.
   2) CREDIT_TYPE has 15 categories and many have very few counts, consolidate any type that are not in the largest 2 categories into a new category called "Loan", as these are all related to some type of loans such as car loans, Microloan and so on. Only keep 3 final categories: Consumer Credit, Credit Card, Loan. Then use one hot encoding to create 3 dummy variables corresponding to each credit types.
   3) Consolidate any non-active status in column CREDIT_ACTIVE into 'Closed' status. Then use one hot encoding to create 2 dummy variables corresponding to each status, active or closed.

   At each current application ID level:

   1) Count total number of Bureau ID’s for each current application ID.
   2) Sum over each of the credit type dummy variables to get the total number of consumer credit, credit card and loan for each current application ID.
   3) Sum over each of the status dummy variables to get the total number of closed and active statuses for each current application ID.
   4) Variables named starting with “AMT” represent current amount of credits, debts, limits of credit cards, or max / sum of amount overdue shown on the Bureau, aggregate these variables using sum to capture the total amount for each current application ID.
   5) Variables named starting with “DAYS” represent at the time of application, how many days since Bureau credit ended, or remaining days of Bureau credit, or how many days before current application did client apply for Bureau credit, or days overdue and so on. Using max or min or both summary statistics to aggregate these columns is reasonable.
   6) Use max to aggregate CREDIT_DAY_OVERDUE to capture the maximum days overdue on CB for the same current application ID.

7) Use max CNT_CREDIT_PROLONG to capture the maximum times was the Credit Bureau credit prolonged for the same current application ID.

2. Credit card balance data: This dataset provides the credit card balance information for previous application IDs in each month prior to current application. Variables include credit card limit during the month, amount drawing at ATM during the month, etc. for each previous application ID. Each current application ID corresponds to multiple previous IDs, and each previous ID corresponds to multiple rows and each row represents the month relative to current application, where -1 meaning the month prior to current application. This dataset has a total of 23 columns.

   At each row (MONTHS_BALANCE) level:

   1) Variable NAME_CONTRACT_STATUS has 7 categories and very few counts in many categories. To reduce the number of categories, create a new dummy variable "STATUS_ACTIVE" with only 2 categories, value 1 meaning active, 0 meaning inactive.
   2) Variable SK_DPD_DEF means number of days past due during each month in the past before the current application. Since over 97.5% of the values are 0, and only less than 0.5% of the values are >= 1, create a new dummy variable to group all the > 0 values into 1 category. Similar treatment is used on variable SK_DPD.

   At each unique current and previous application ID combination level:

   1) Variables named starting with “AMT” are amount of credit card balances, credit limit, ATM drawings, payments etc. in each of the past months before the current application. Aggregate each “AMT” variable using either average, maximum or sum over the past months.
   2) Variables named starting with “CT” are number of drawings in each of past months before the current application. Aggregate each “CT” variable by sum to get the total number of drawings over the past months.
   3) Use max function to aggregate the dummy variable created from STATUS_ACTIVE, to represent whether the record is active within each unique combination of current application ID and previous application ID.
   4) Sum over the dummy variables created from SK_DPD_DEF to get the total number of records having SK_DPD_DEF > 0 within each unique combination of current application ID and previous application ID. Similar treatment for variable SK_DPD.

   At each current application ID level:

   All variables are aggregated again using either the sum or average function, to get the summarized value within each unique current application ID, over all previous application IDs.

3. Pos Cash balance data: This dataset provides the monthly balance snapshots of previous POS (point of sales) and cash loans that the applicant had with Home Credit. Similar to

the credit card balance dataset, each current application ID corresponds to multiple previous IDs, and each previous ID corresponds to multiple rows and each row represents the month relative to current application, where -1 meaning the month prior to current application. This dataset has a total of 8 columns.

At each row (MONTHS_BALANCE) level:

1) Since NAME_CONTRACT_STATUS has 9 categories and very few counts in some categories, to reduce the number of categories, only keep Active and Completed categories, all other statuses are consolidated into 1 category called 'Other'.

At each unique current and previous application ID combination level:

1) Further reduce the number of categories in NAME_CONTRACT_STATUS by creating a new dummy column to represent completed contracts (value 1). All others will be considered as non-completed (value 0).
2) Variable CNT_INSTALMENT means term of previous credit. Create 2 columns to keep the max and min term of previous credit within each unique combination of current application ID and previous application ID.
3) Variable CNT_INSTALMENT_FUTURE represents Installments left to pay on the previous credit. Create 2 columns to keep the max and min.
4) Treatment for variable SK_DPD and SK_DPD_DEF are similar to the credit card balance dataset above.

At each current application ID level:

All variables are aggregated again using either the sum or min, max function, to get the summarized value within each unique current application ID, over all previous application IDs.

4. Installment payment data: This dataset provides past payment data for the previously disbursed credits in Home Credit. Each current application ID corresponds to multiple previous application IDs and each previous application ID contains multiple rows, where one row for every payment that was made plus one row each for missed payment. One row is equivalent to one payment of one installment OR one installment corresponding to one payment. This dataset has a total of 8 columns.

   At each unique current and previous application ID combination level:

   1) Keep the min and max of variable NUM_INSTALMENT_VERSION and NUM_INSTALMENT_NUMBER. The version number signifies payment parameter changes; however, no additional information is available to further engineer this feature.
   2) DAYS_INSTALMENT >= DAYS_ENTRY_PAYMENT meaning the payment is made prior to the due date, which is on time payments, otherwise it would be a late payment. Aggregate by counting number of late payments.

3) Keep average AMT_INSTALMENT and average AMT_PAYMENT across different rows.

   At each current application ID level:

   Logic of aggregation at SK_ID_CURR level are the same as those used in aggregating at the SK_ID_PREV level described above.

5. Previous application data: This dataset contains information about all previous applications for Home Credit loans of clients who have loans in the current application. Each current application ID corresponds to multiple rows which represent one previous application ID. This dataset has a total of 38 columns.

   1) For variables named starting with “AMT” or “RATE”, such as AMT_ANNUITY, AMT_CREDIT and so on, aggregate by summing up the total annuity, total amount of requests, total credit, down payment etc. information over all previous applications. Get the maximum down payment and interest rate etc. from all of the previous applications.
   2) For categorical variables named starting with “NAME” and a few others, first do one hot encoding at each row level, then sum at SK_ID_CURR level to get the total number of instances over all previous application IDs.
   3) For variables named starting with “DAYS” such as the first and last due days of each previous application, get the min and max of days (relative to current application date) over all previous applications.
   4) For columns with high cardinality, first group some categories with fewer frequencies, then do one hot encoding at each row level, and finally, sum at SK_ID_CURR level to get the total counts that previous application appeared in that category, within the same current application ID. As an example, variable "NAME_CASH_LOAN_PURPOSE" has over 20 categories, group the categories whose frequency are <= 20000 to simplify the variable into 4 categories. Then use one hot encoding to create 4 dummy variables representing the presence of each of the 4 categories. Finally, sum up at current application ID level to get the total number of counts that belong to each of the 4 categories.

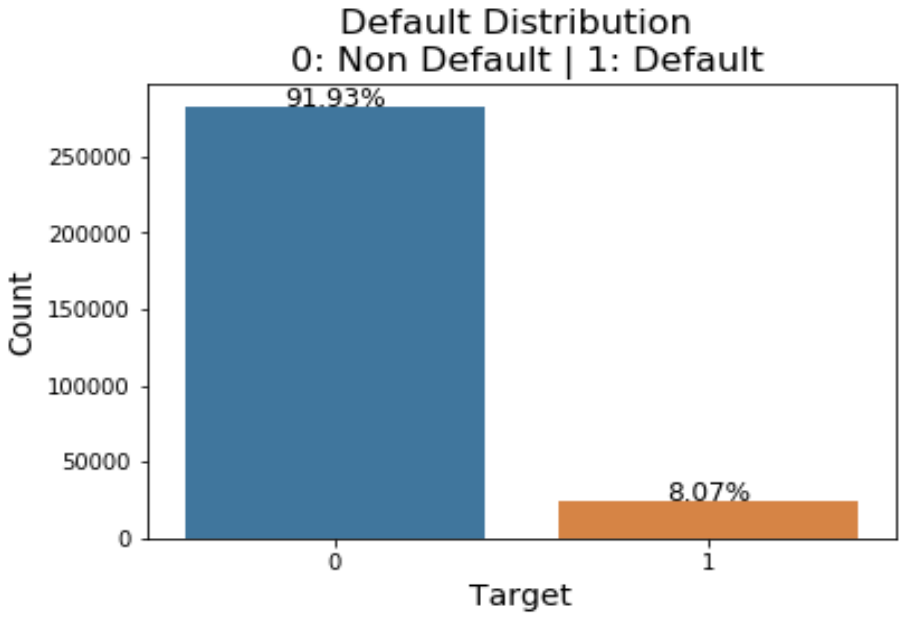
6. Current application training data: This is the main table with static information for all current applications. No data cleaning is done at this point for this dataset. Merge this table with all of the 5 aggregated tables created above to get one combined dataset. Section 2 talks about EDA and further preprocessing of the combined dataset such as checking variable correlations and treating missing values.

Section 2. Data cleaning, exploration and missing value imputations

At the end of the last section, all datasets coming out of 6 sources are combined into one table, and categorical variables only exist in the main lead application table, as the categorical variables in all 5 other tables are all aggregated and summarized into numeric variables using one hot

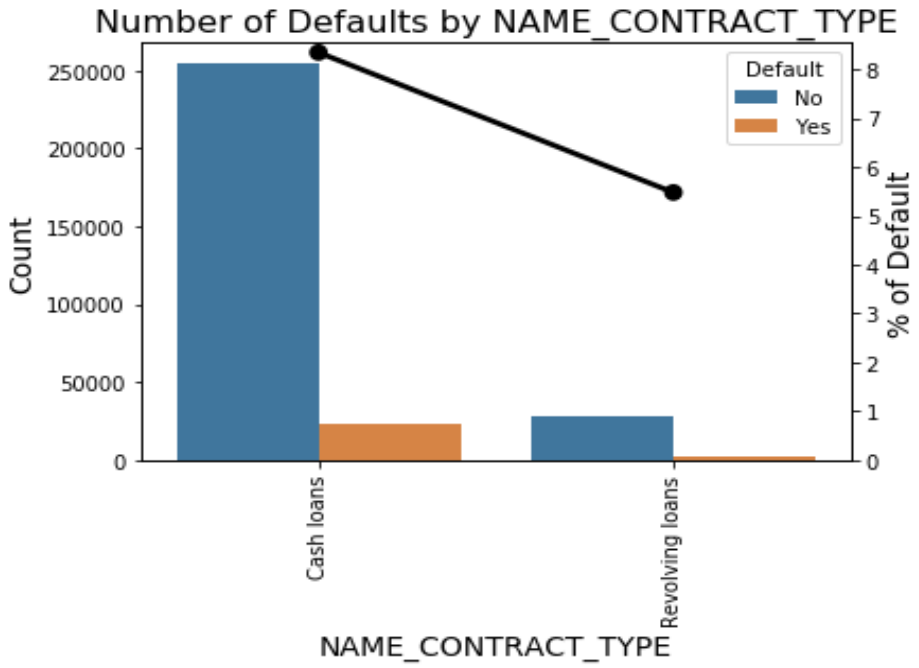
encoding and summary functions. Further preprocessing of the combined table will be done in this section, mainly treating and exploring the categorical and numeric variables in the lead table. First of all, columns with over 60% missing values are deleted.

1) The nature of default data is highly imbalanced, as shown in the following graph, overall default percentage is 8.07% in the training set.

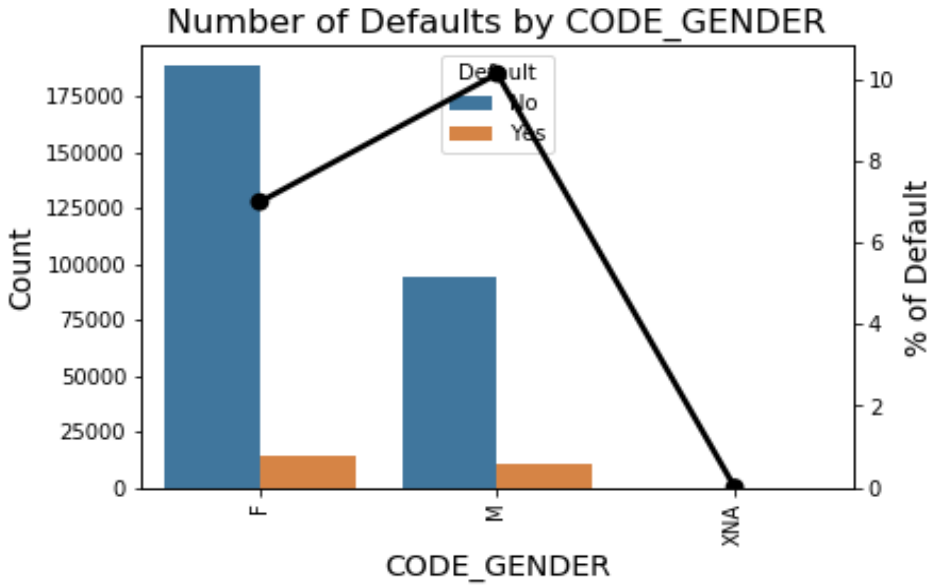


2) Exploration of categorical variables. Looking at the default rate within each category in a categorical variable helps visualizing which variables distinguish default vs. non default records better. Due to the vast number of categorical variables in the dataset, here we only show a few.

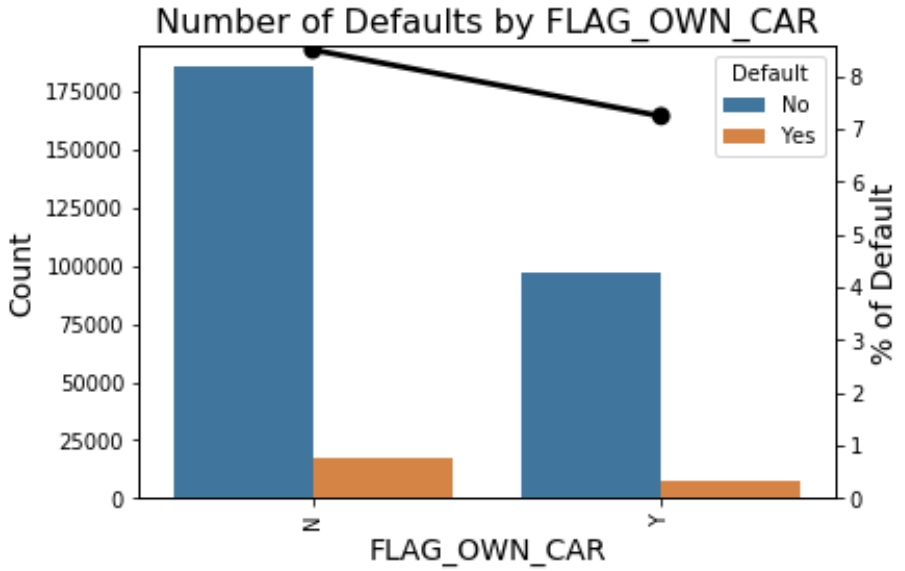
    a. Contract type. Revolving loans group appears to have a lower default rate than the Cash loans group. A hypothesis t-test can be done to test if the difference in default rate is statistically significant.



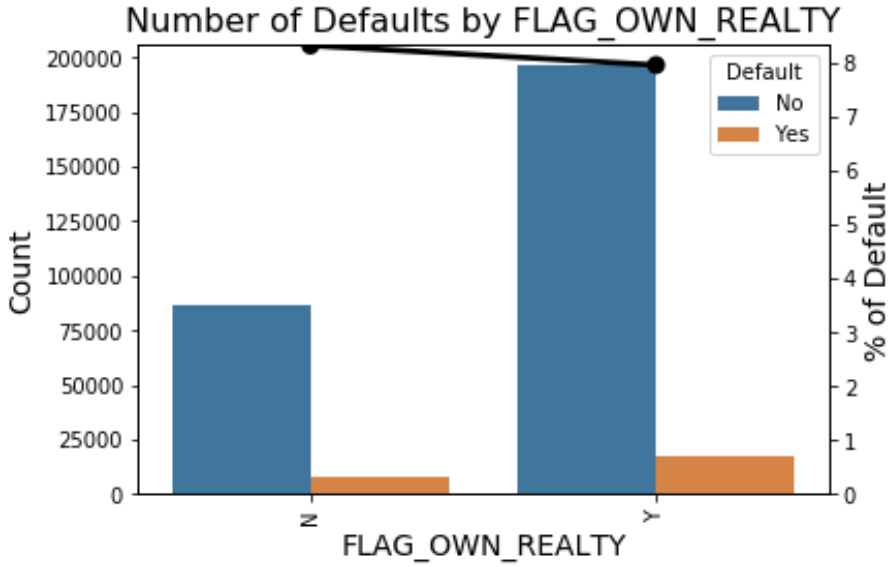
b. Gender. Notice that there are 3 gender groups, we will remove the 4 records having gender = XNA. It can be seen from the graph that majority of the applicants are female, and the default rate within female applicants is lower than that in the male group.



c. Own a car or not. Majority of the applicants do not own a car, default rate within applicants who own a car is 7.24%, whereas the default rate in the other group is 8.5%. A hypothesis t-test can be done to conclude if the difference is statistically significant or not.

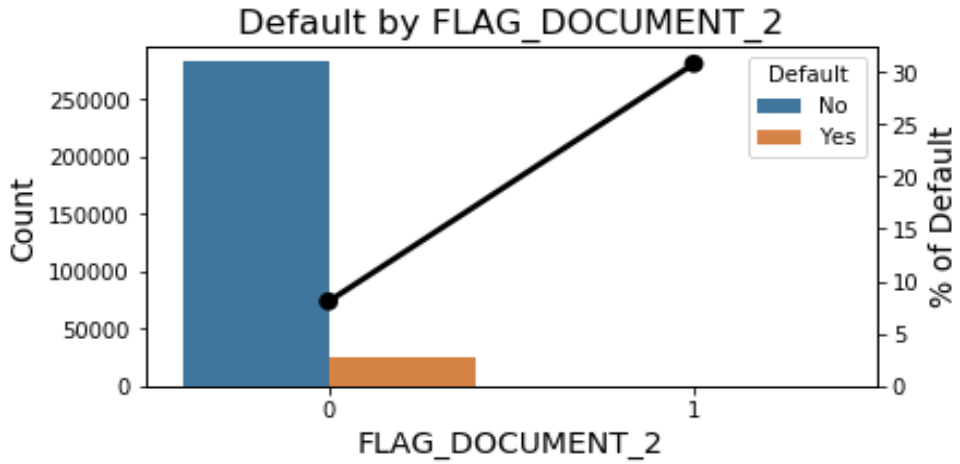


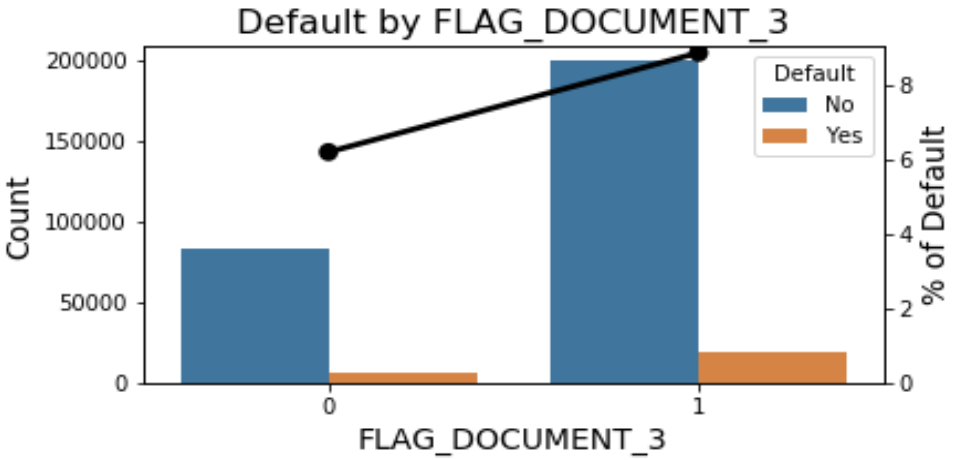
d. Own realty or not. Majority of the applicants own realty; however, the default rate is pretty close in the 2 groups, which may indicate the variable does not have strong power in separating default loans from non-default loans.

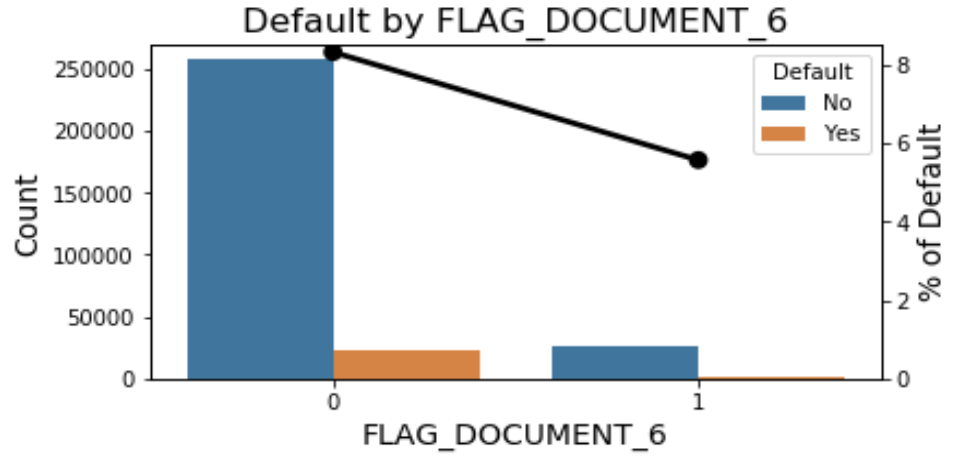


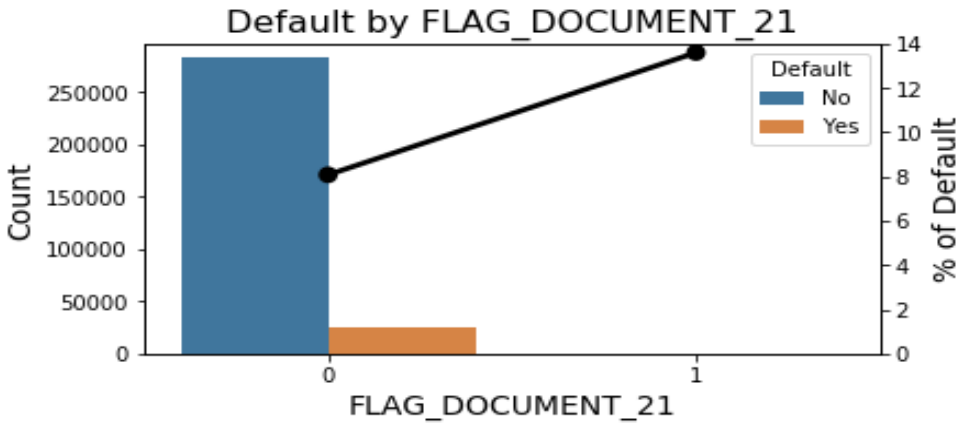
e. If documents 2 ~ 21 are provided by the applicant during the application. There are no detailed descriptions of what each document is, we simply check the distribution of each document variable to see how they distinguish default from non-default applications. Here we only display the graphs of a few document variables. Except for document 3 where a greater number of applicants provided the document, for all the rest documents, majority of the applicants did not provide. For most of the documents, the default rate in the group that provided the document is lower than the

group that didn’t provide. But there are a few documents that show the opposite behavior.

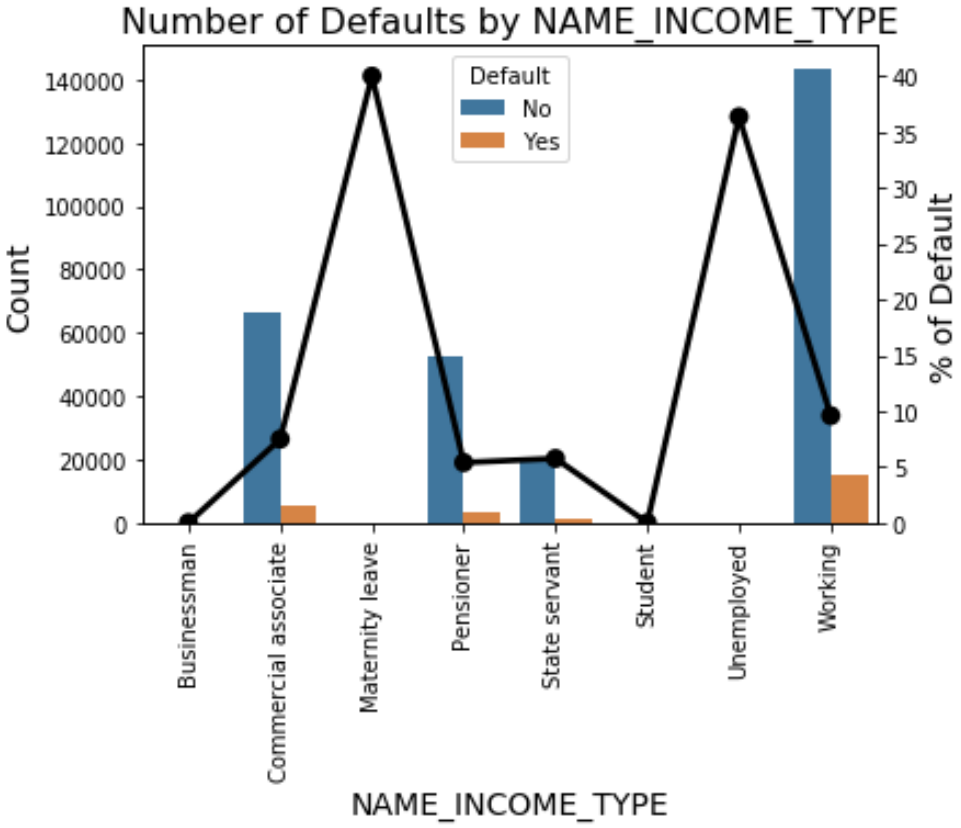




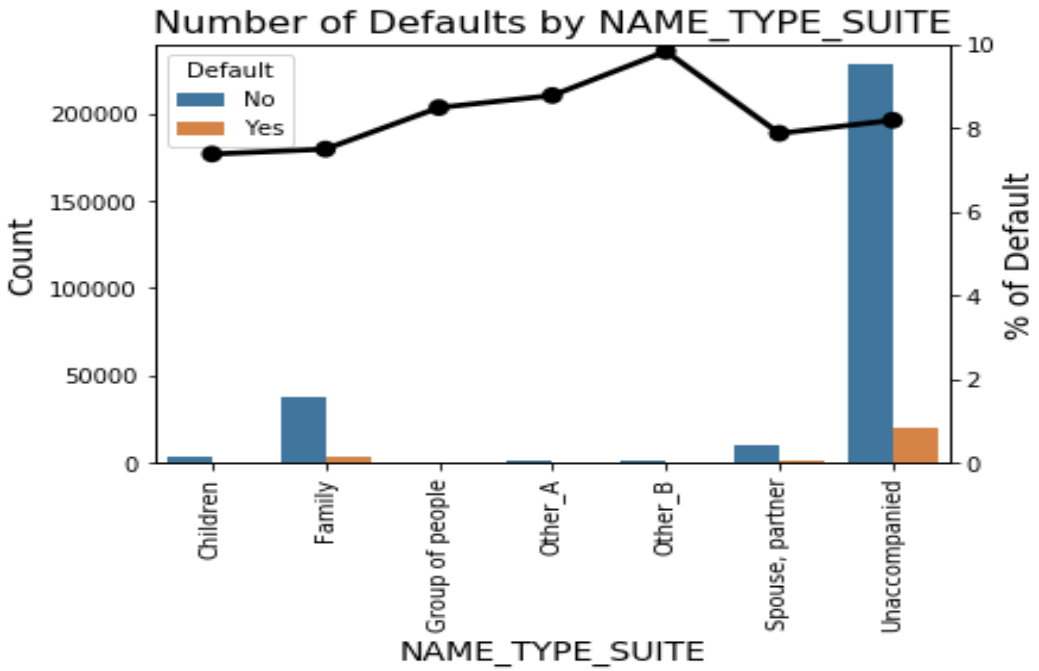


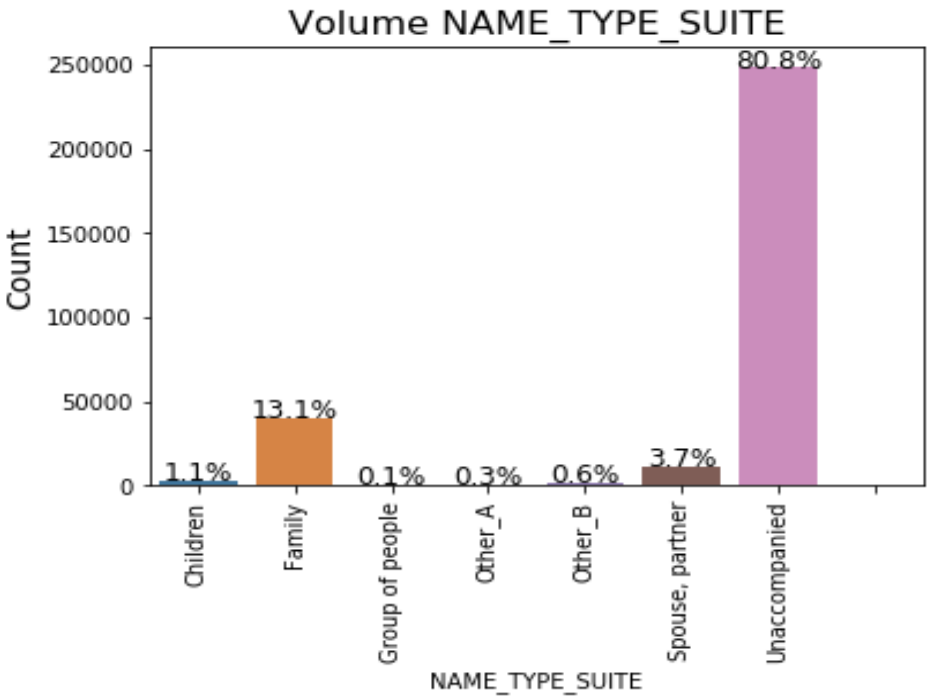


f. It can be seen from below that applicants who are on maternity leave or who are unemployed have the highest default rate. This make sense as these applicants may not have consistent regular income to pay the loan.

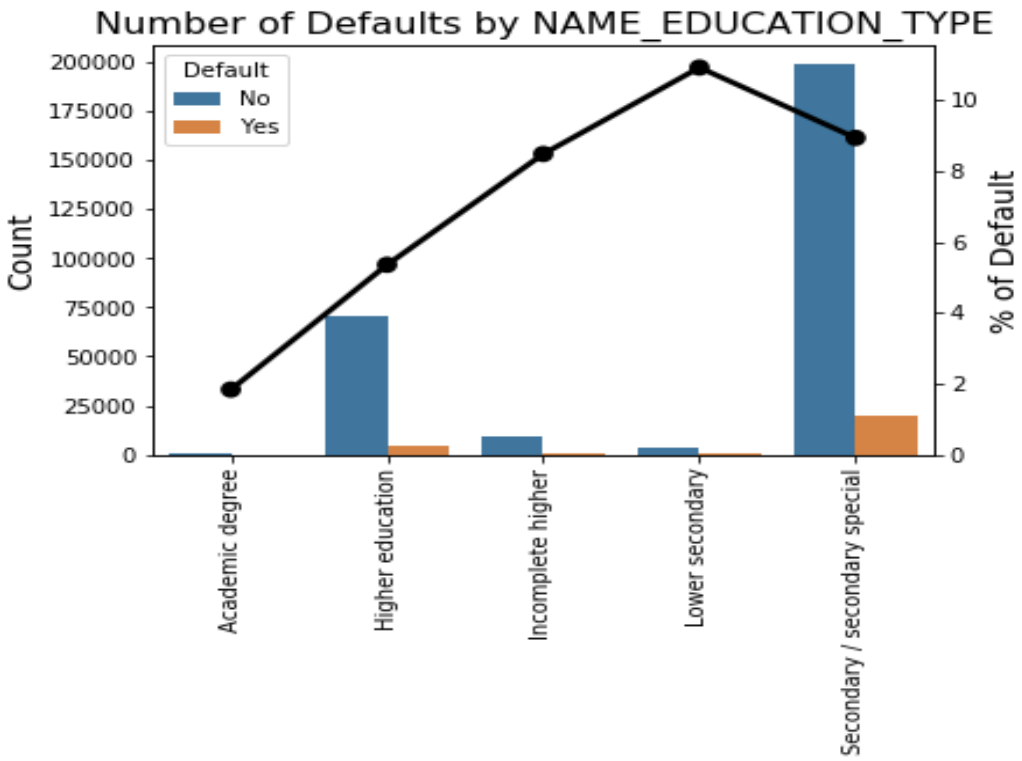


g. Who accompanied applicant at the time of application. In general, this variable does not provide much separation power. Majority of the applicants came alone without any accompaniment. Notice that there are ‘nan’ values in the variable that need to be coded as a new category ‘UKN’ indicating unknown.

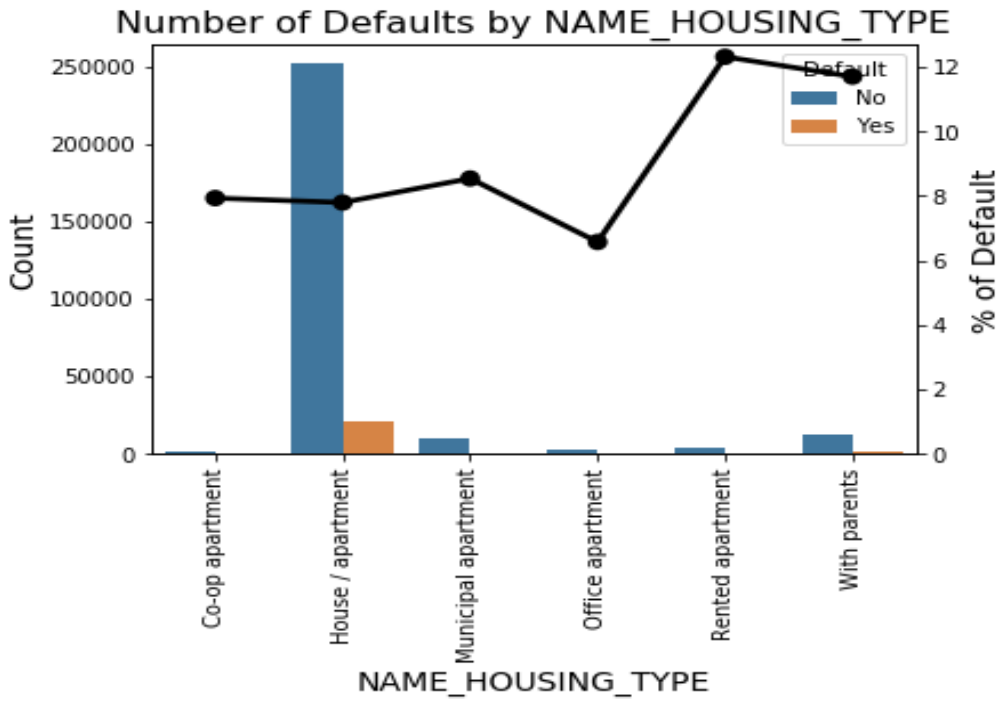




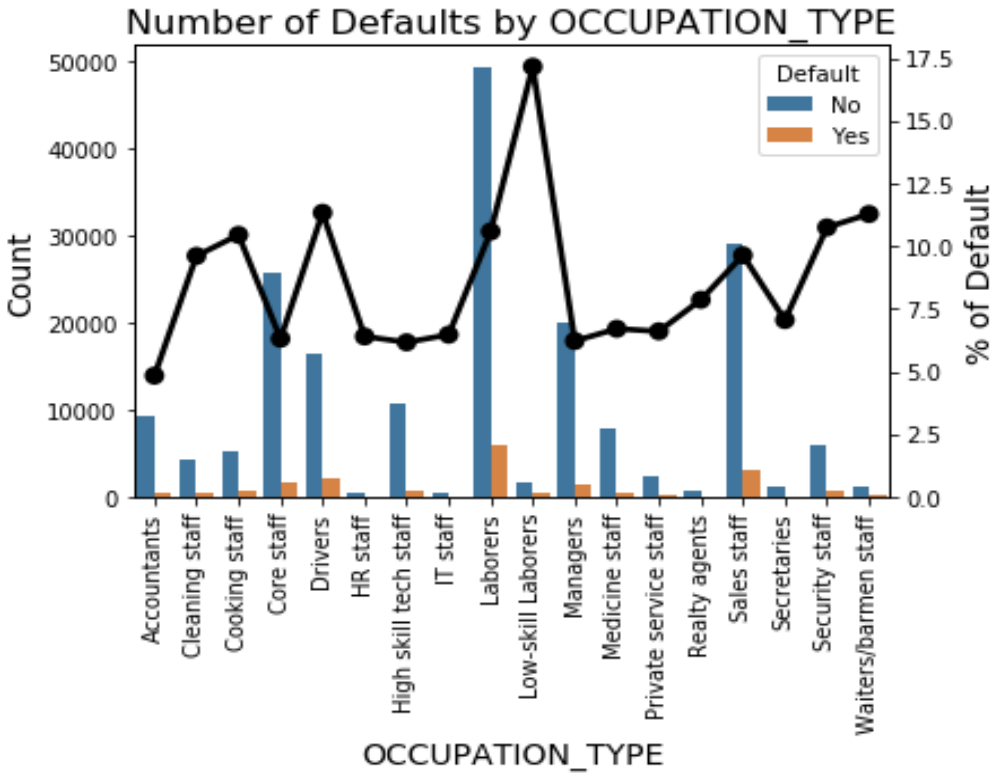
h. Education level. There is a rank ordering in default rate by education level. In general, applicants with low secondary education has the highest default rate. As we can imagine education level is correlated with type of job the applicant can take and highly correlated with income, which is essential to repayment of a loan.

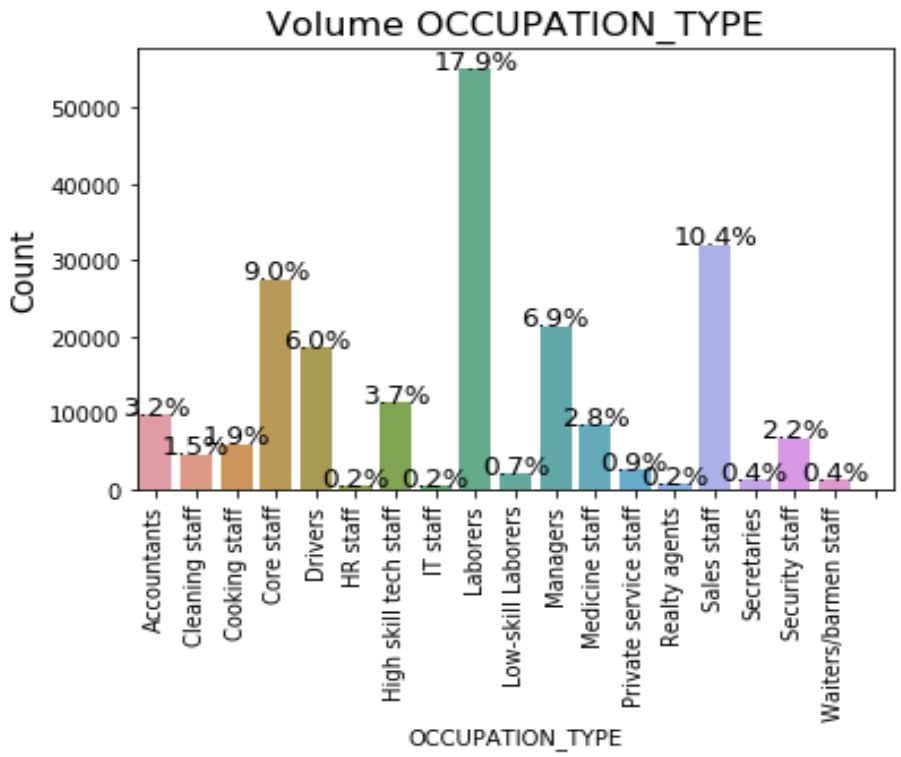


i. This variable again is highly correlated with the financial stability of the applicant. People who rent a house or live with parents may not have a regular income, thus having difficulty repaying the loan.

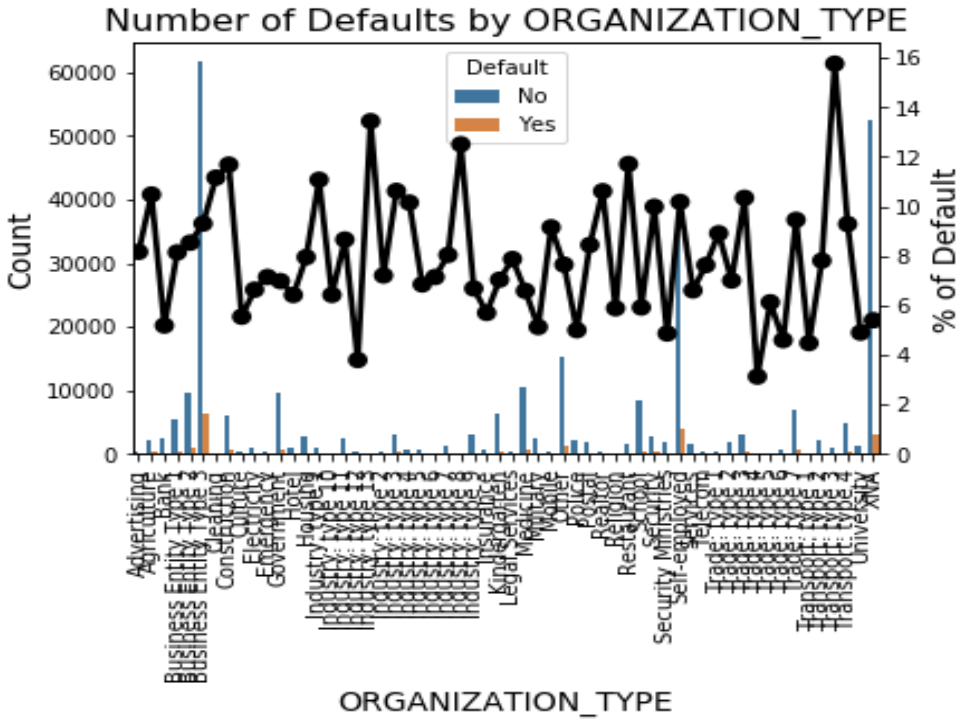


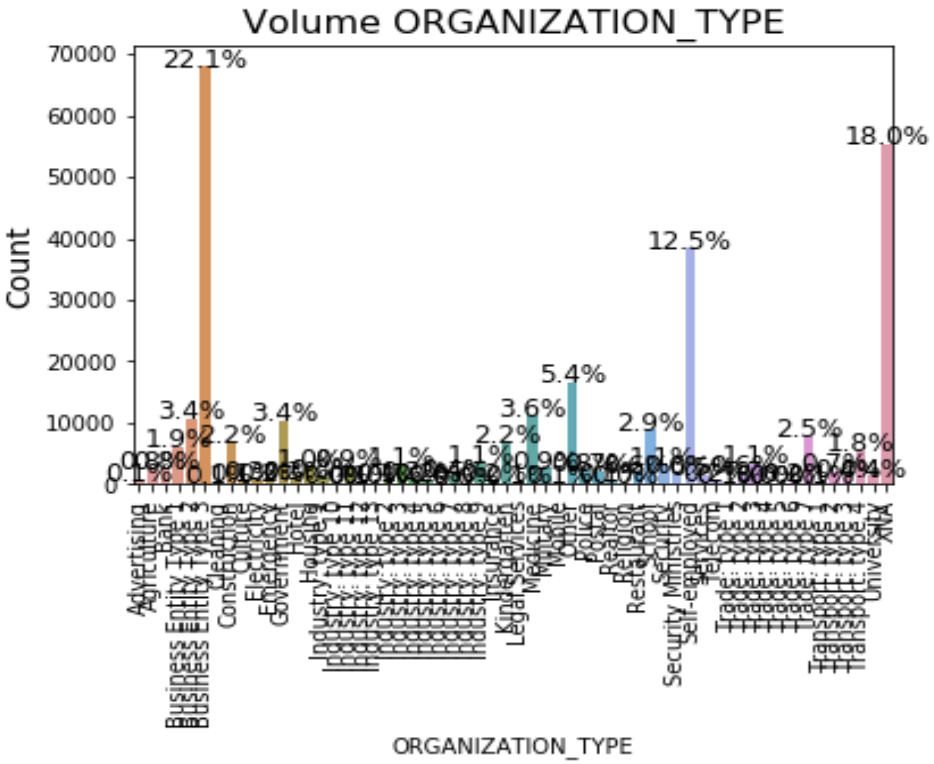
j. Occupation type. This variable has over 15 categories, we group the categories having <= 10000 counts into 1 group. Also notice ‘nan’ values that needs to be coded into a new category.



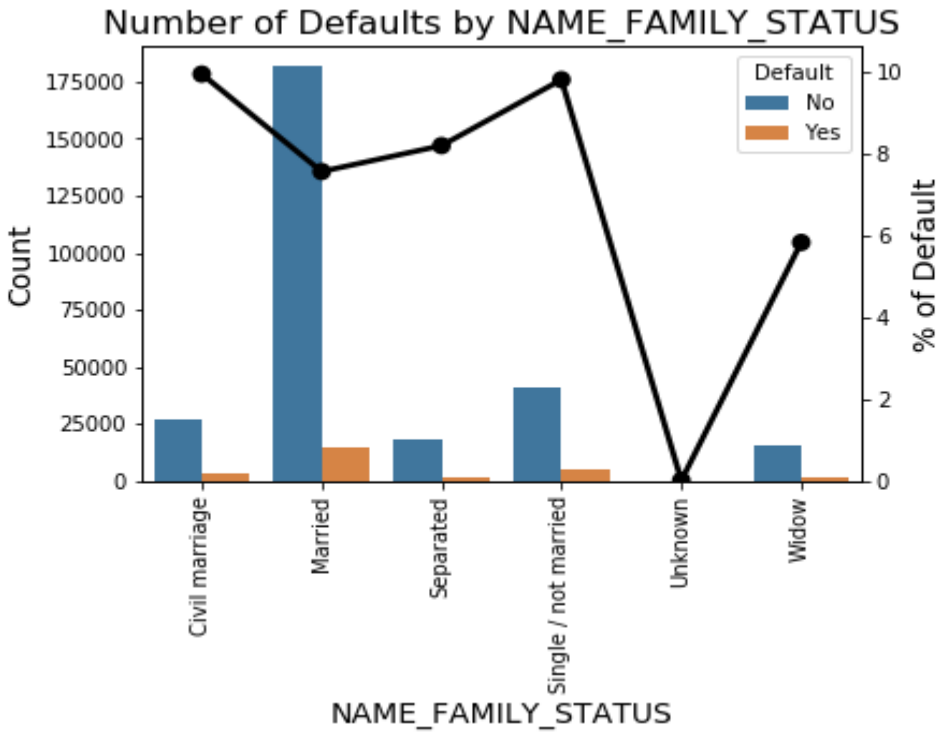


k. Variable organization type contains more than 20 categories and ‘nan’ values in the column, group the categories with < 30000 counts into 1 group and code ‘nan’ values into a new category.

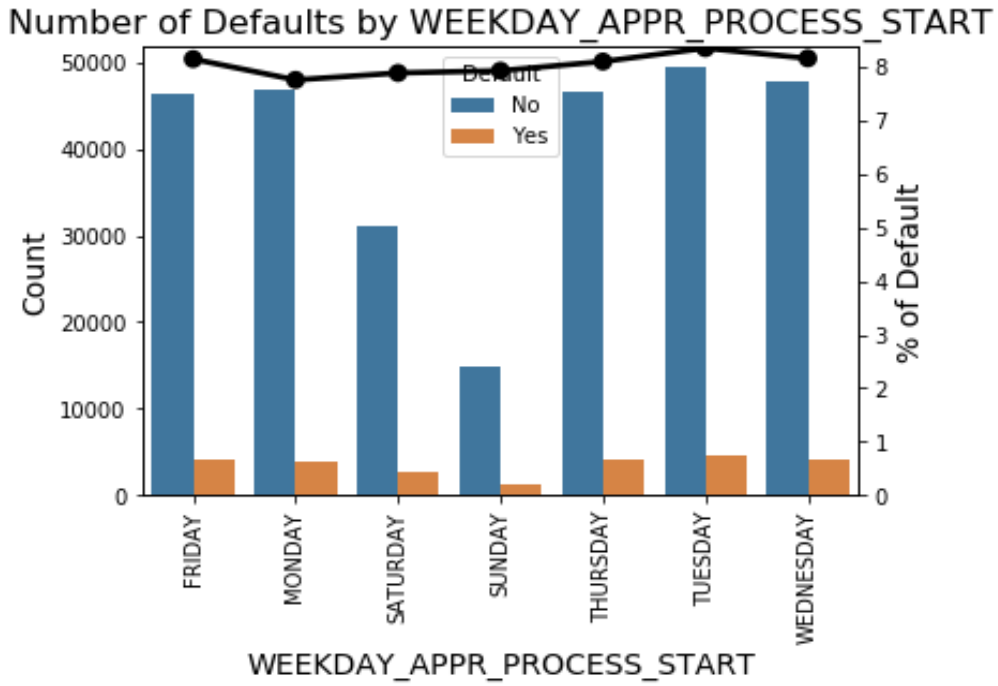




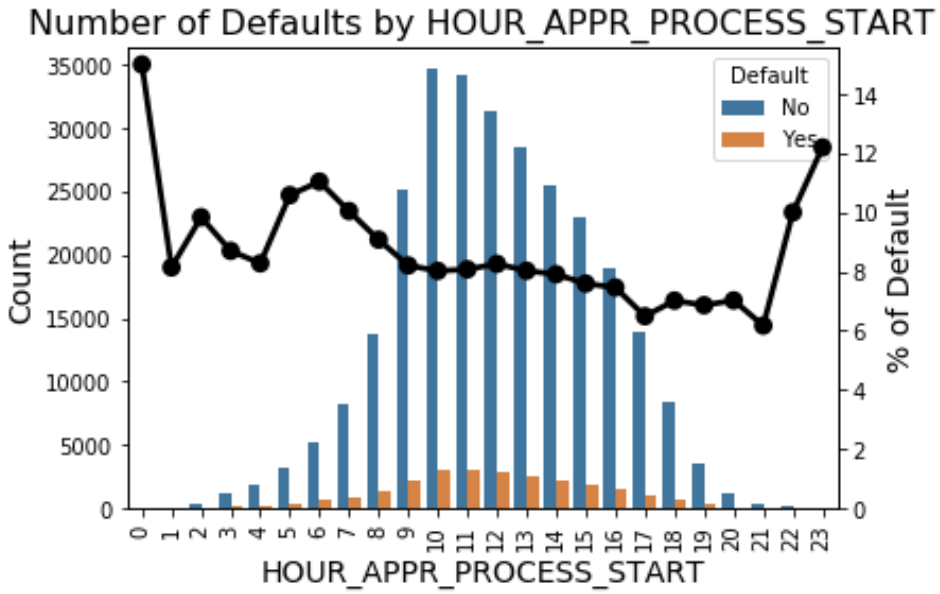
l. Family status.



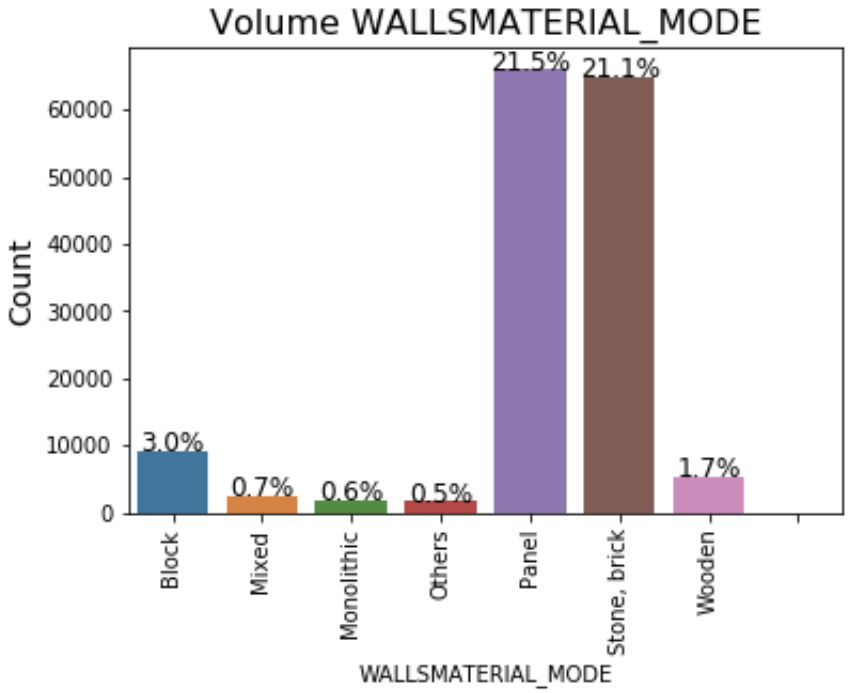
m. Application start time by weekday. It is expected that application volumes are higher during weekdays compared to weekends.



n. Application starting time by hour of the day. Clearly volumes are higher during working hours between 9am and 4pm, and much lower before 9am or after 4pm.

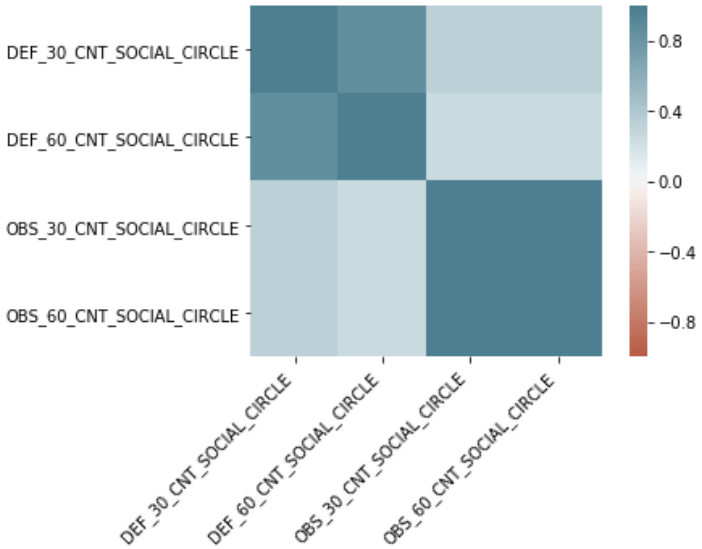


o. Wall materials. This variable tells the wall material of the house that the applicant lives in. Note there are 'nan' values that need to be treated.

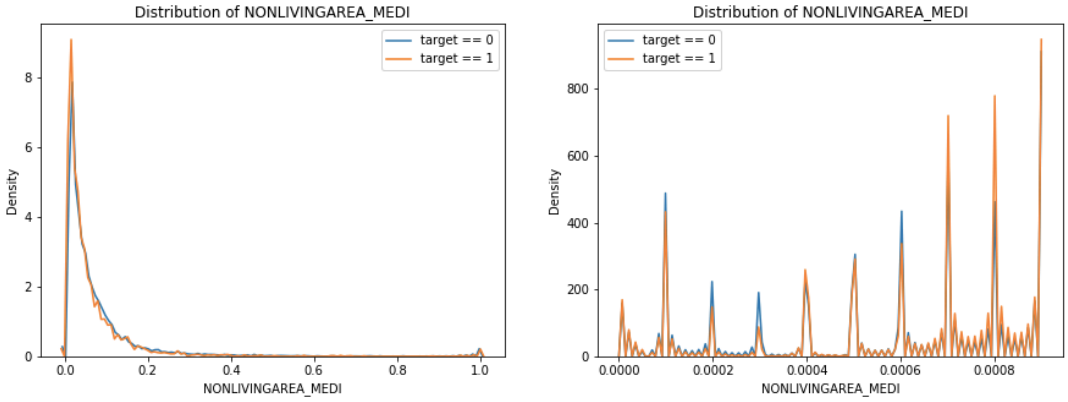


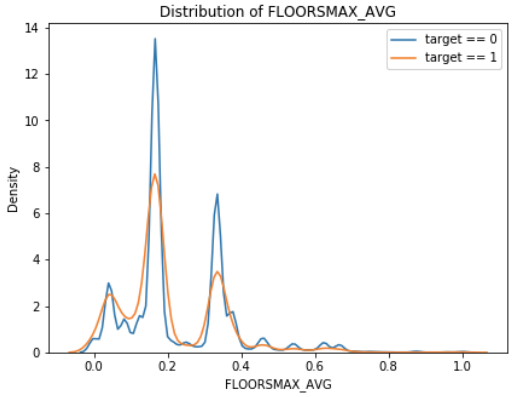
3) Exploration of numeric variables. Here we explore some numeric variables by creating the KDE plots for each variable and split by default vs. non-default.

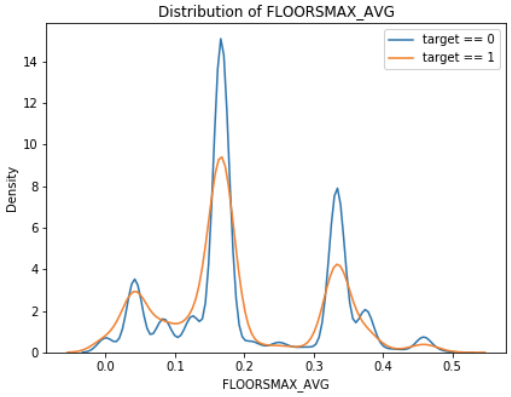
a. There is perfect positive linear correlation between observations of client's social surroundings with observable 30 days past due and 60 days past due. As a result, remove one of them from the dataset. There is also high linear correlation between observation of client's social surroundings defaulted on 30 days past due and defaulted on 60 days past due, so remove one of them.

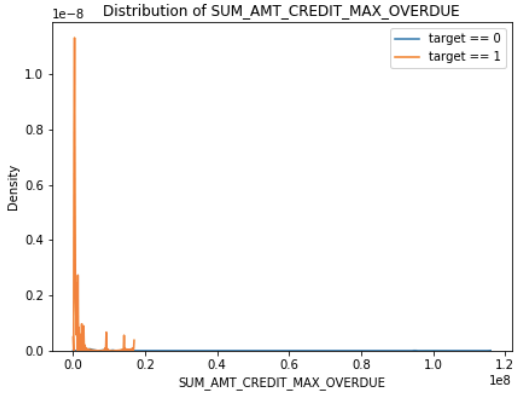


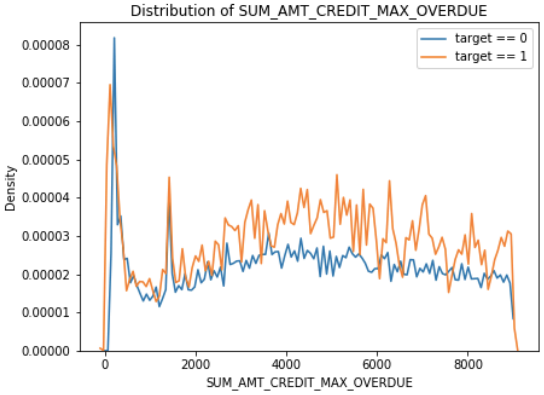
b. Below are KDE plots for some numeric variables. Left panel shows the original range of the variable. Since most of the variables are highly skewed to the right, we zoom in the head of the distribution and plot it on the right panel. A few variables show distinction between the default and non-default groups, while many others are quite similar in these 2 target groups.

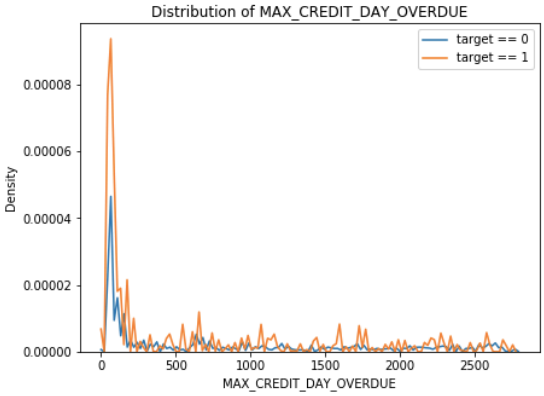


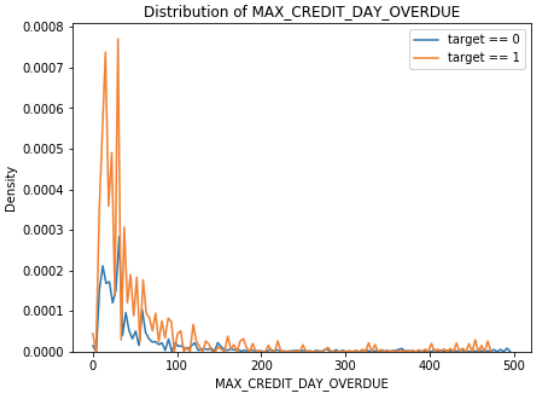

Distribution of FLOORSMAX_AVG
target == 0
target == 1
Density
FLOORSMAX_AVG

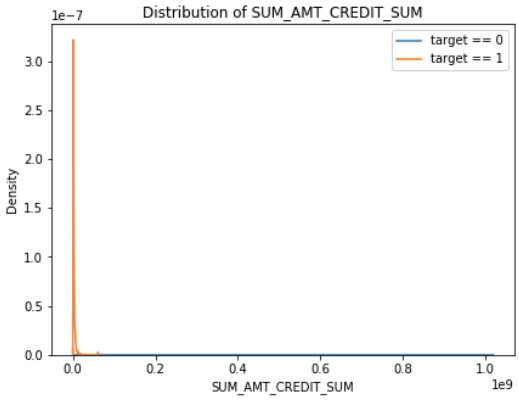

Distribution of FLOORSMAX_AVG
target == 0
target == 1
Density
FLOORSMAX_AVG

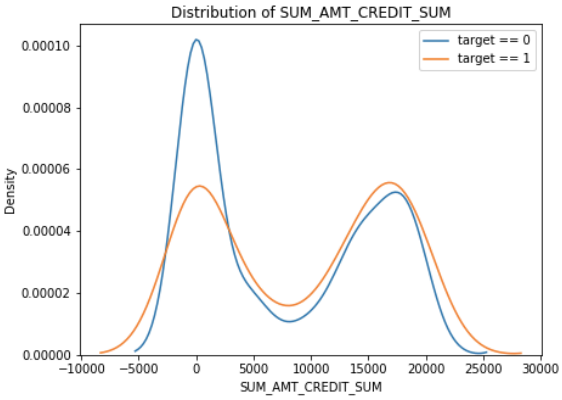

Distribution of SUM_AMT_CREDIT_MAX_OVERDUE
1e−8
target == 0
target == 1
Density
SUM_AMT_CREDIT_MAX_OVERDUE
1e8

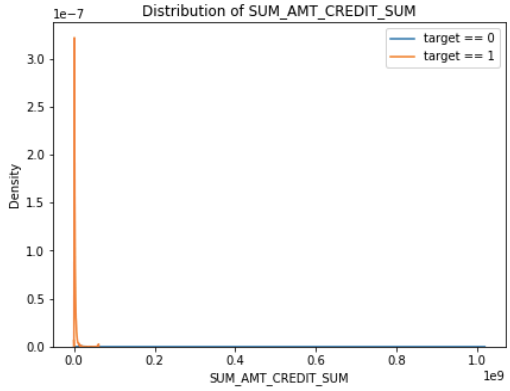

Distribution of SUM_AMT_CREDIT_MAX_OVERDUE
target == 0
target == 1
Density
SUM_AMT_CREDIT_MAX_OVERDUE

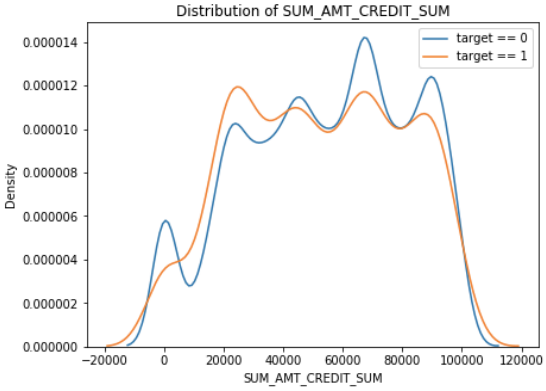

Distribution of MAX_CREDIT_DAY_OVERDUE
target == 0
target == 1
Density
MAX_CREDIT_DAY_OVERDUE

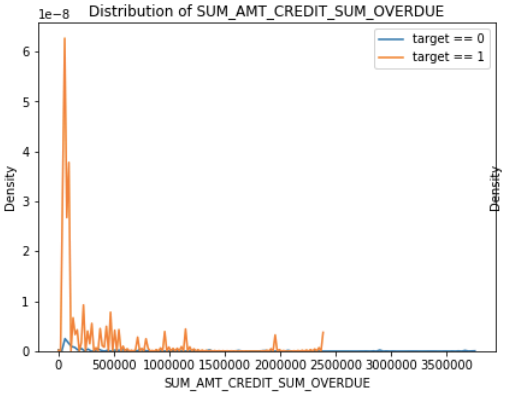

Distribution of MAX_CREDIT_DAY_OVERDUE
target == 0
target == 1
Density
MAX_CREDIT_DAY_OVERDUE

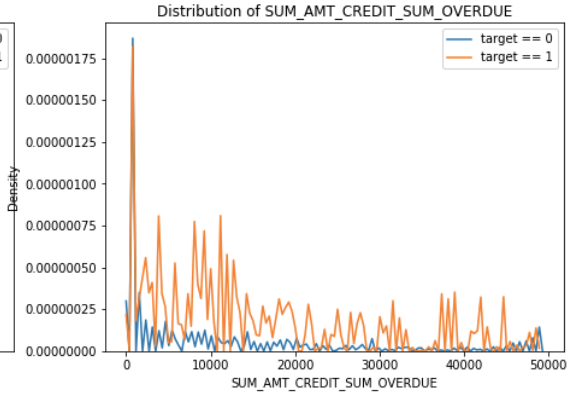

Distribution of SUM_AMT_CREDIT_SUM
1e−7
target == 0
target == 1
Density
SUM_AMT_CREDIT_SUM
1e9

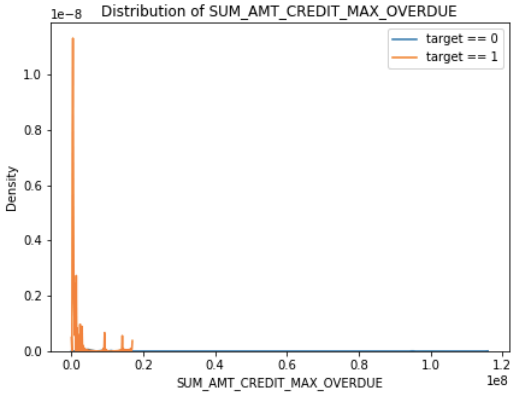

Distribution of SUM_AMT_CREDIT_SUM
target == 0
target == 1
Density
SUM_AMT_CREDIT_SUM

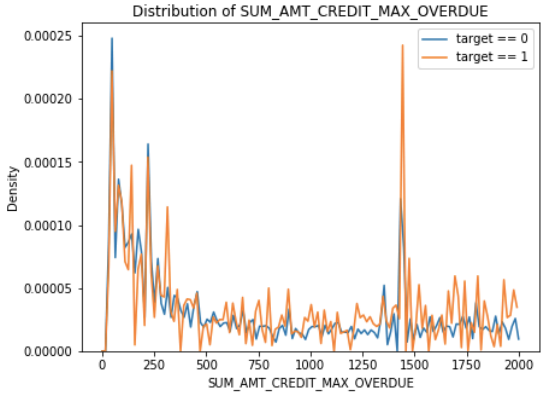

Distribution of SUM_AMT_CREDIT_SUM
1e−7
target == 0
target == 1
Density
SUM_AMT_CREDIT_SUM
1e9

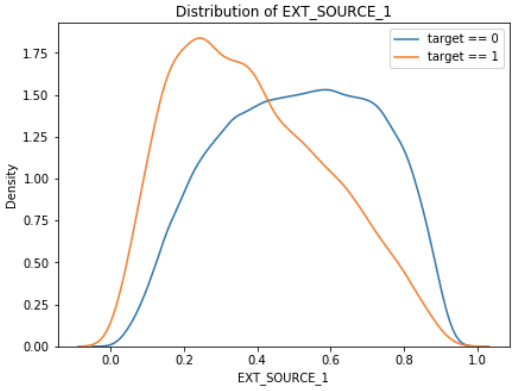

Distribution of SUM_AMT_CREDIT_SUM
target == 0
target == 1
Density
SUM_AMT_CREDIT_SUM

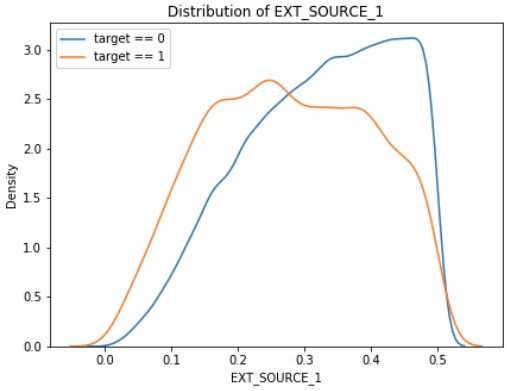

Distribution of SUM_AMT_CREDIT_SUM_OVERDUE
1e−8
target == 0
target == 1
Density
SUM_AMT_CREDIT_SUM_OVERDUE

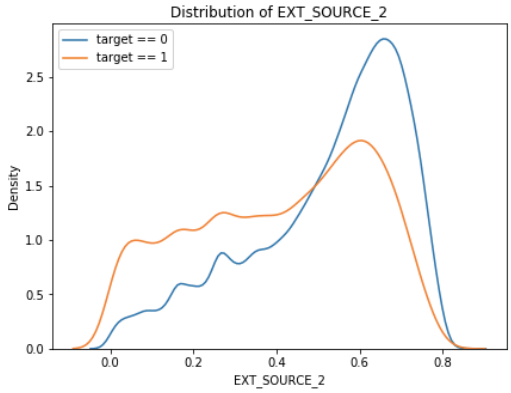

Distribution of SUM_AMT_CREDIT_SUM_OVERDUE
target == 0
target == 1
Density
SUM_AMT_CREDIT_SUM_OVERDUE

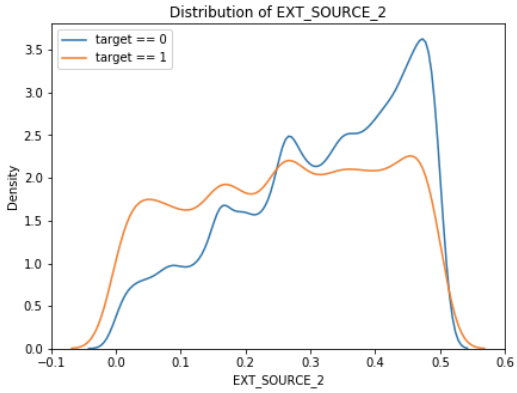

Distribution of SUM_AMT_CREDIT_MAX_OVERDUE
1e−8
target == 0
target == 1
Density
SUM_AMT_CREDIT_MAX_OVERDUE
1e8

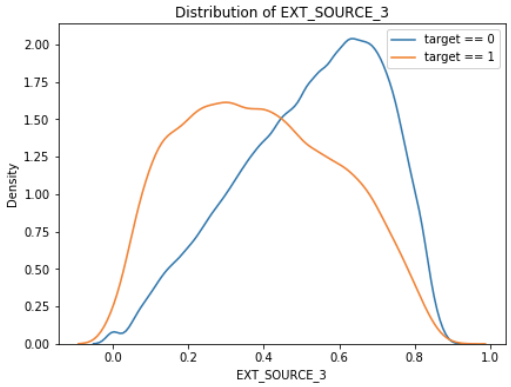

Distribution of SUM_AMT_CREDIT_MAX_OVERDUE
target == 0
target == 1
Density
SUM_AMT_CREDIT_MAX_OVERDUE

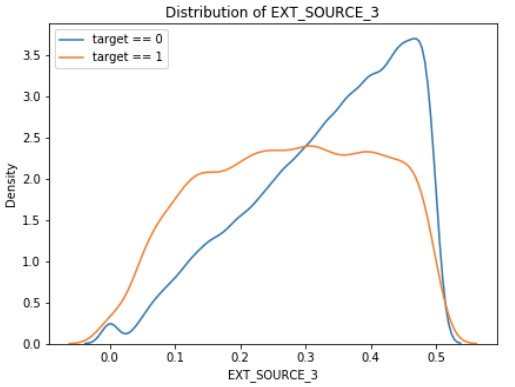

Distribution of EXT_SOURCE_1
target == 0
target == 1
Density
EXT_SOURCE_1

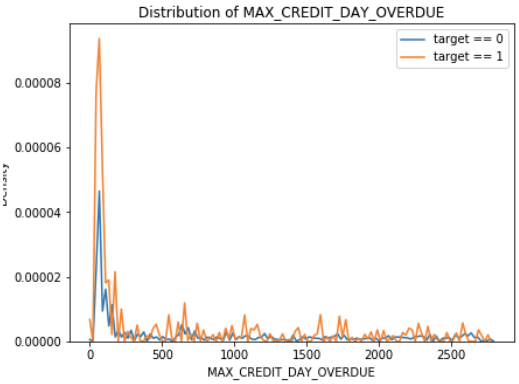

Distribution of EXT_SOURCE_1
target == 0
target == 1
Density
EXT_SOURCE_1

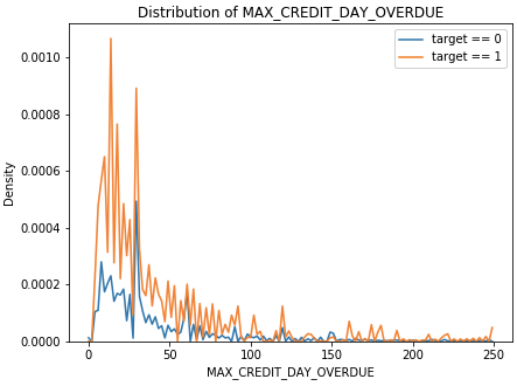

Distribution of EXT_SOURCE_2
target == 0
target == 1
Density
EXT_SOURCE_2

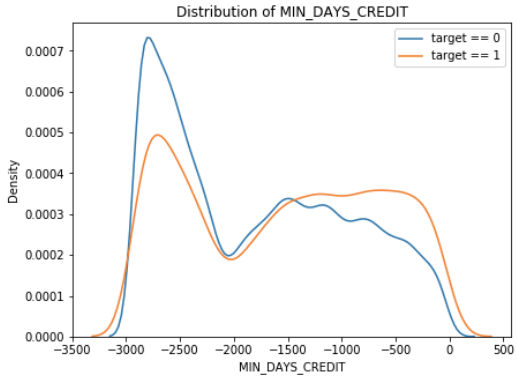

Distribution of EXT_SOURCE_2
target == 0
target == 1
Density
EXT_SOURCE_2

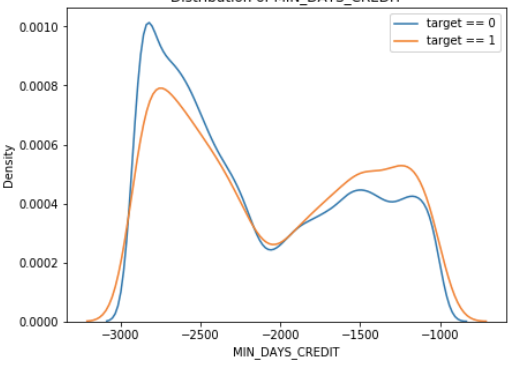

Distribution of EXT_SOURCE_3
target == 0
target == 1
Density
EXT_SOURCE_3


Distribution of EXT_SOURCE_3
target == 0
target == 1
Density
EXT_SOURCE_3

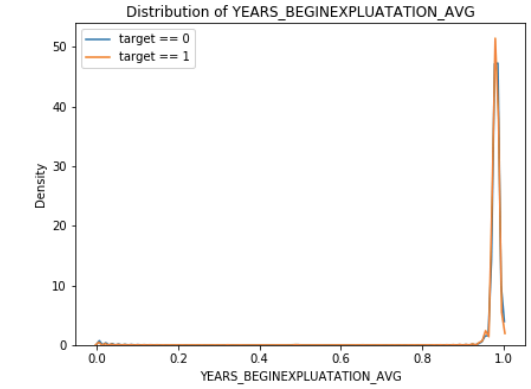

Distribution of MAX_CREDIT_DAY_OVERDUE
target == 0
target == 1
MAX_CREDIT_DAY_OVERDUE

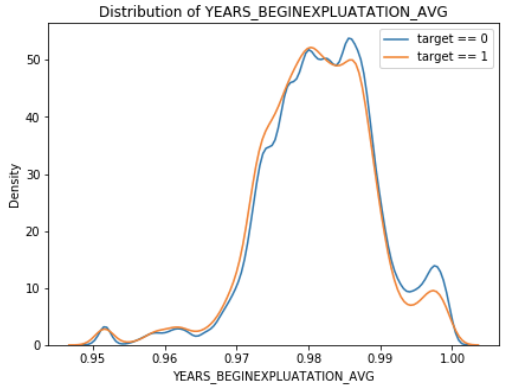

Distribution of MAX_CREDIT_DAY_OVERDUE
target == 0
target == 1
Density
MAX_CREDIT_DAY_OVERDUE

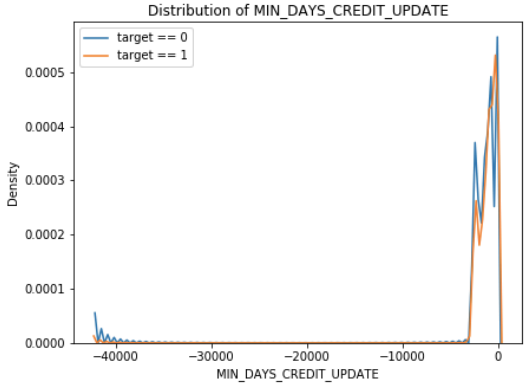

Distribution of MIN_DAYS_CREDIT
target == 0
target == 1
Density
MIN_DAYS_CREDIT

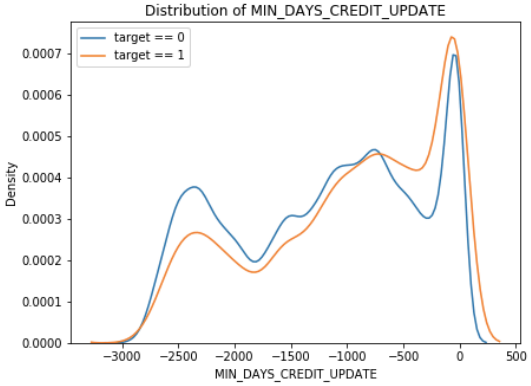

Distribution of MIN_DAYS_CREDIT
target == 0
target == 1
Density
MIN_DAYS_CREDIT

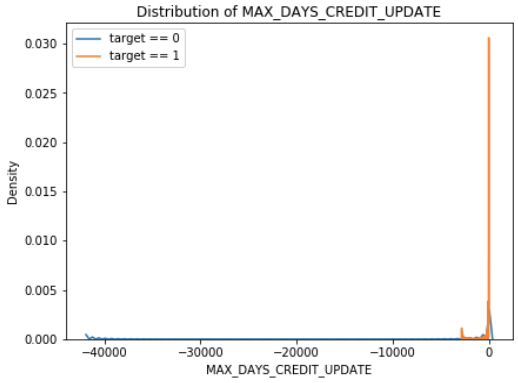
c. Below are KDE plots for some left skewed numeric variables. Left panel shows the original range of the variable, and the right panel shows the zoomed in view of the head in the distribution plot.

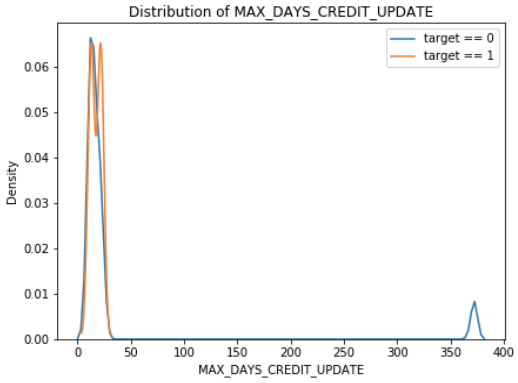


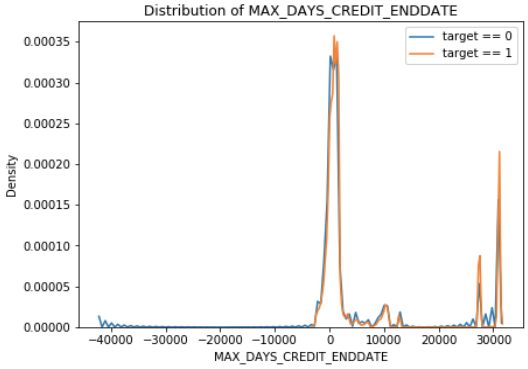


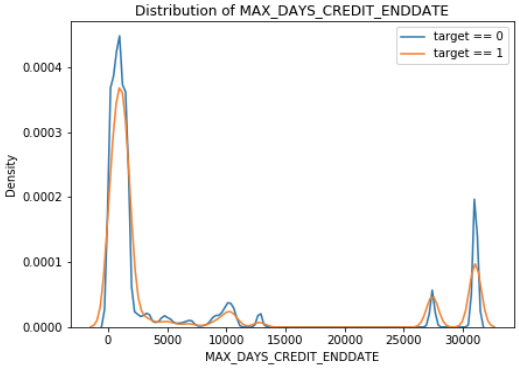




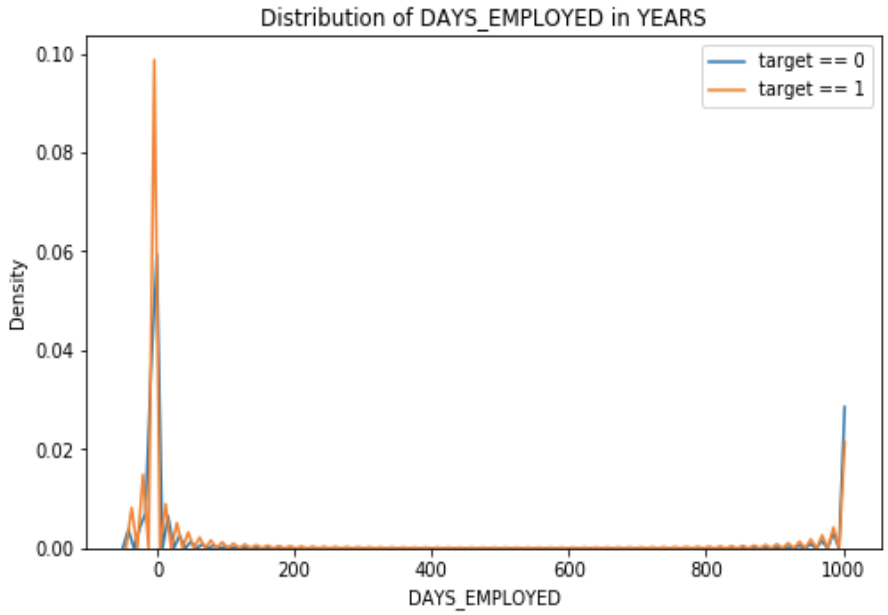


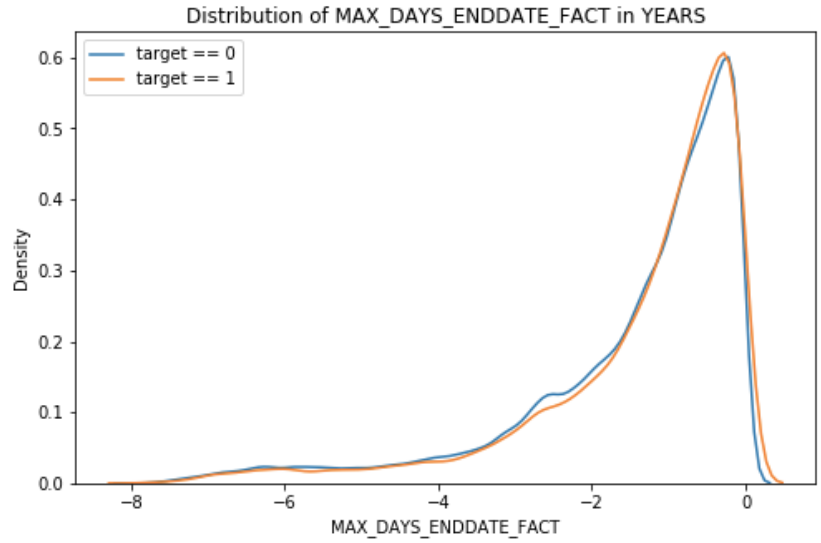


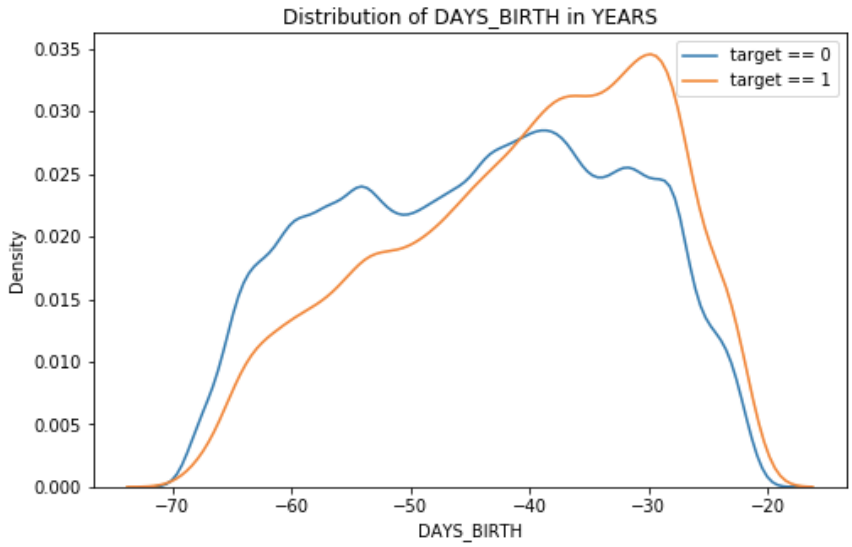


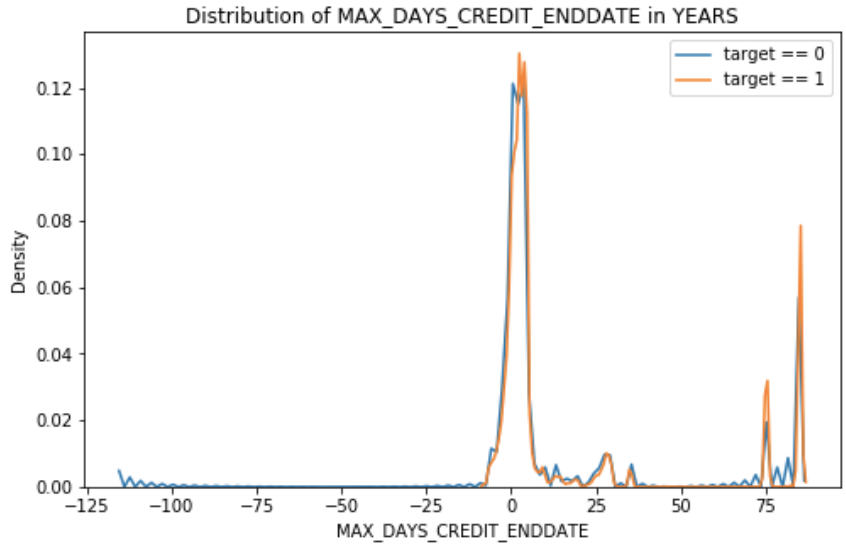


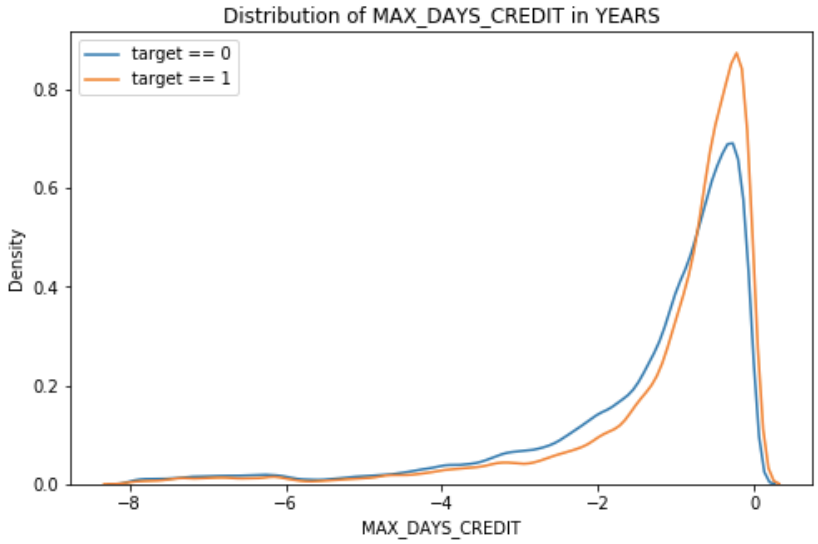
d. Variables below are originally represented in days in the dataset, convert them into years to better visualize the ranges. Here are some observations:
   (i) "DAYS_EMPLOYED" has values over 1000 years, which are apparently errors. Based on other age type of variables, they are all counting backwards, meaning the maximum should be 0 (current application date), so cap the value at 0.
   (ii) 'MAX_DAYS_CREDIT_ENDDATE' is an aggregated field summarizing all previous applications within the same current application ID using the max aggregation function. It means over all the previous applications within the same current application ID, the maximum remaining duration of Bureau credit at the time of application. This variable should be a positive number, but we found negative values in the column. As a result, we floor the variable at 0. For missing values, impute using the median.
   (iii) 'MAX_DAYS_CREDIT_UPDATE' is an aggregated field similar to (ii). It means over all the previous applications within the same current application ID, the maximum days before current application when the last information about the Credit Bureau credit come. This should be a negative number as it's counting backwards from the current application date. For all the positive numbers, define them as 0.
   (iv) 'MIN_DAYS_CREDIT_UPDATE' is treated the same way as MAX_DAYS_CREDIT_UPDATE

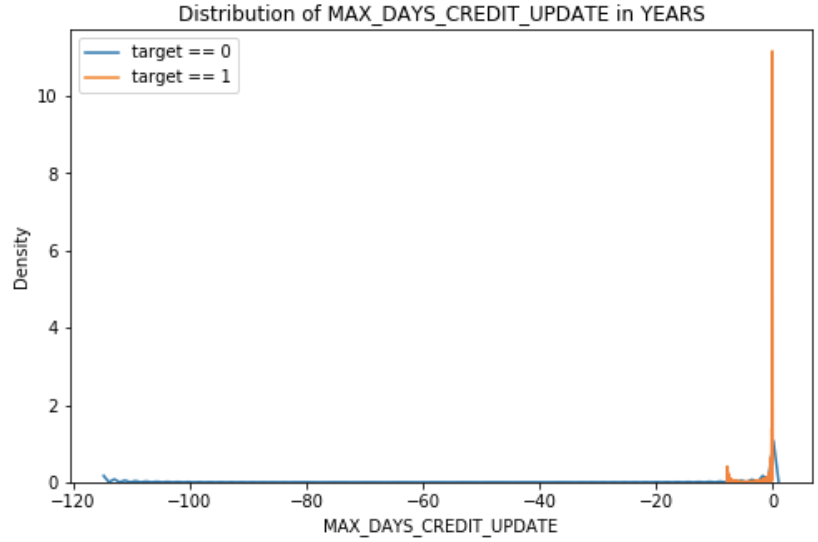


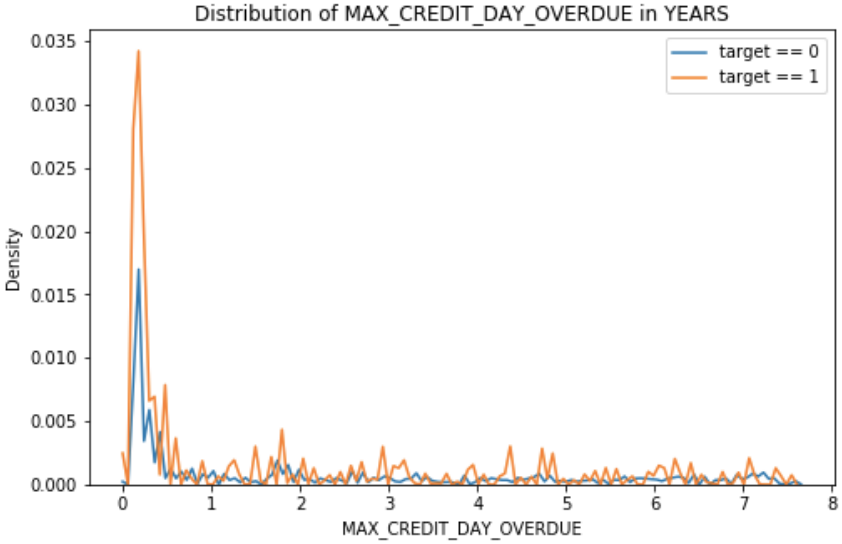

Distribution of MAX_DAYS_ENDDATE_FACT in YEARS
target == 0
target == 1
Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
−8
−6
−4
−2
0
MAX_DAYS_ENDDATE_FACT

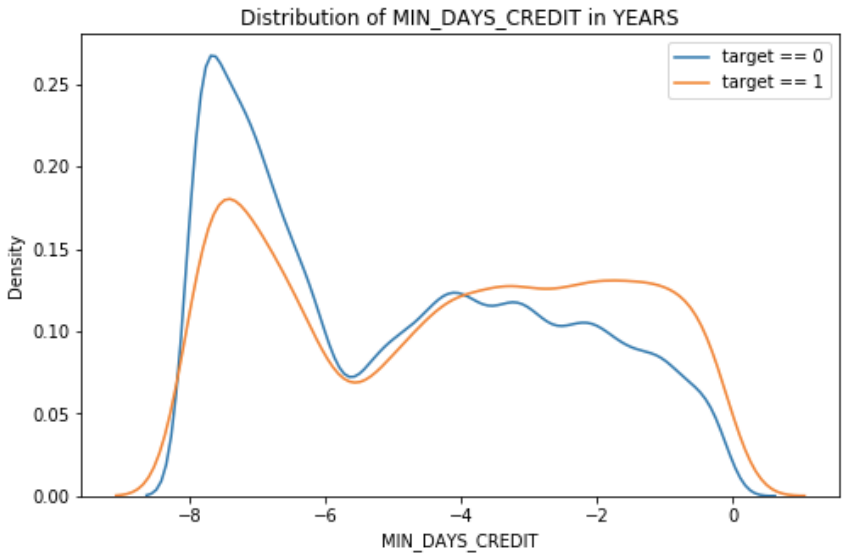

Distribution of DAYS_BIRTH in YEARS
target == 0
target == 1
Density
0.000
0.005
0.010
0.015
0.020
0.025
0.030
0.035
−70
−60
−50
−40
−30
−20
DAYS_BIRTH

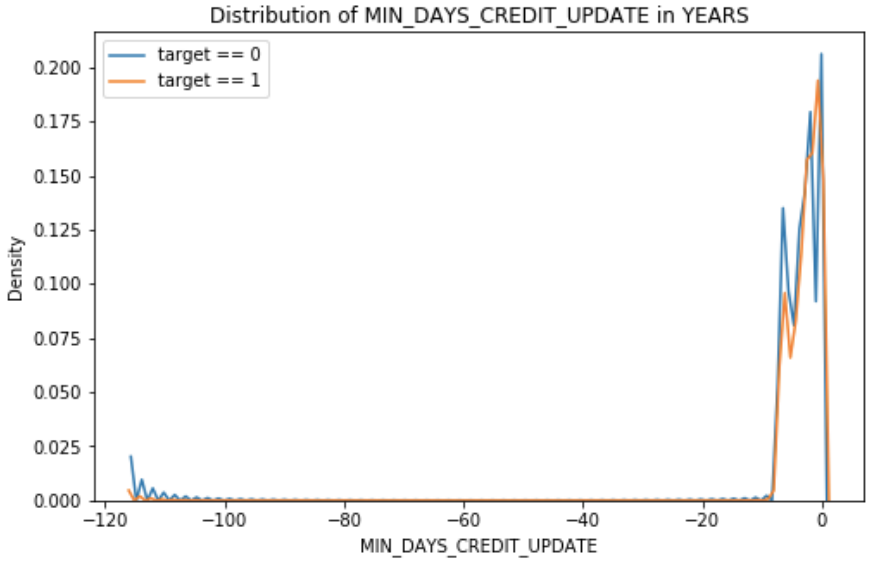

Distribution of MAX_DAYS_CREDIT_ENDDATE in YEARS
target == 0
target == 1
Density
0.00
0.02
0.04
0.06
0.08
0.10
0.12
−125
−100
−75
−50
−25
0
25
50
75
MAX_DAYS_CREDIT_ENDDATE

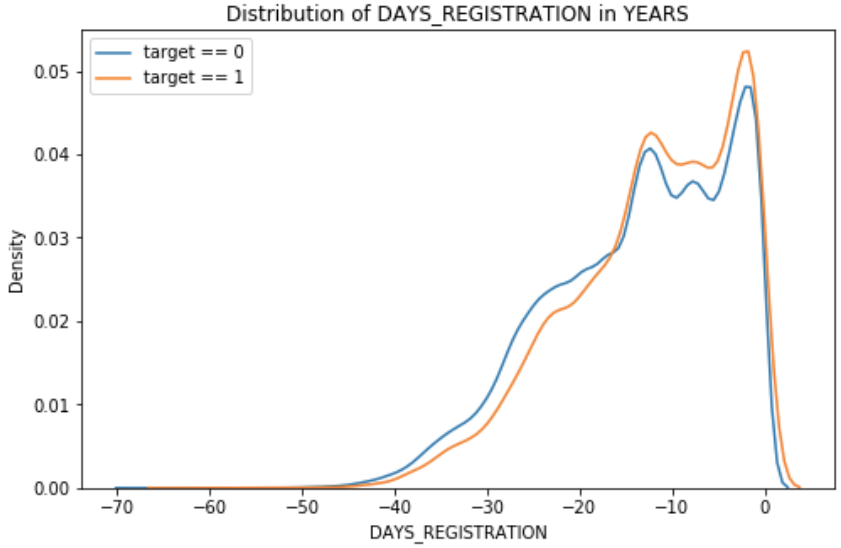

Distribution of MAX_DAYS_CREDIT in YEARS
target == 0
target == 1
Density
MAX_DAYS_CREDIT

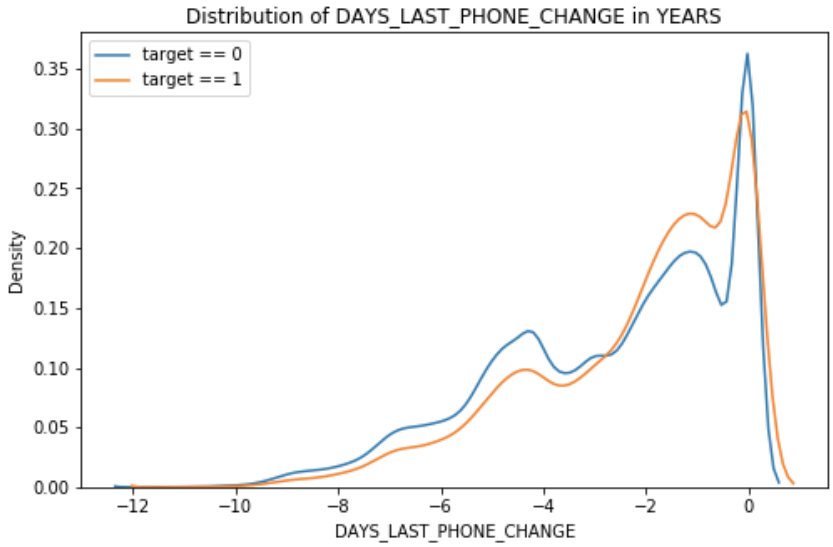

Distribution of MAX_DAYS_CREDIT_UPDATE in YEARS
target == 0
target == 1
Density
MAX_DAYS_CREDIT_UPDATE

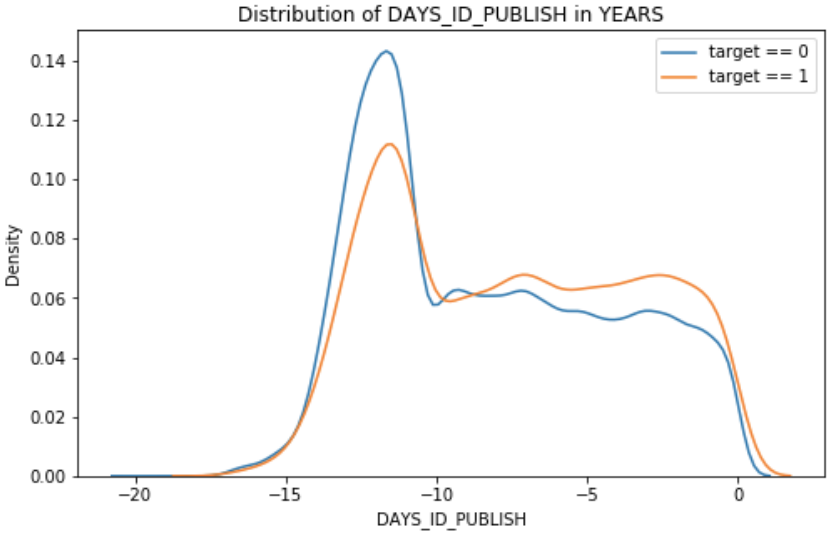

Distribution of MAX_CREDIT_DAY_OVERDUE in YEARS
target == 0
target == 1
Density
MAX_CREDIT_DAY_OVERDUE


Distribution of MIN_DAYS_CREDIT in YEARS
target == 0
target == 1
Density
0.25
0.20
0.15
0.10
0.05
0.00
−8
−6
−4
−2
0
MIN_DAYS_CREDIT


Distribution of MIN_DAYS_CREDIT_UPDATE in YEARS
target == 0
target == 1
Density
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
−120
−100
−80
−60
−40
−20
0
MIN_DAYS_CREDIT_UPDATE

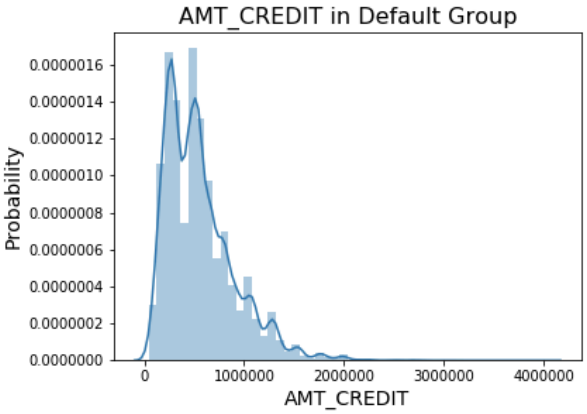

Distribution of DAYS_REGISTRATION in YEARS
target == 0
target == 1
Density
0.05
0.04
0.03
0.02
0.01
0.00
−70
−60
−50
−40
−30
−20
−10
0
DAYS_REGISTRATION

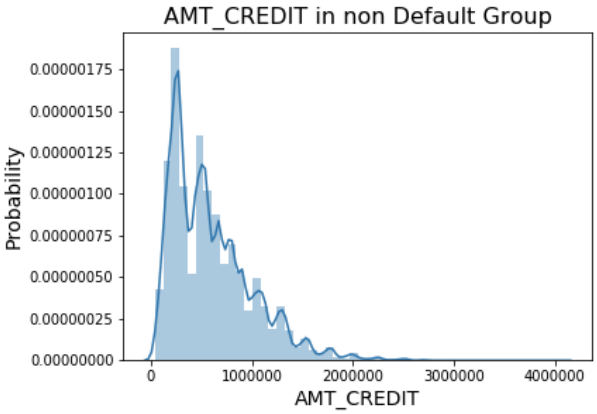


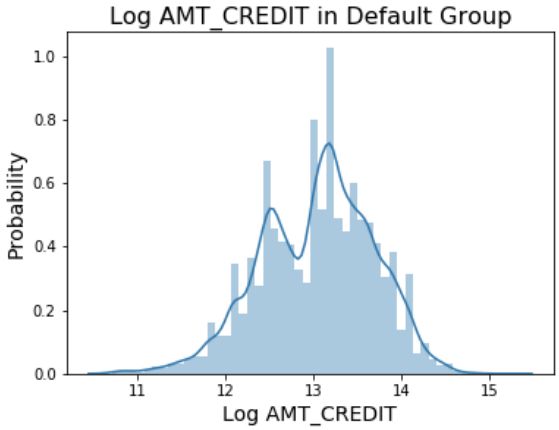


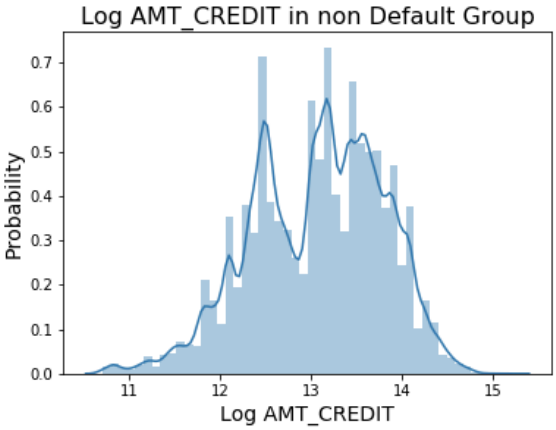
e. Distributions of 'AMT_ANNUITY', 'AMT_CREDIT' and 'AMT_GOODS_PRICE' are skewed to the right, we recommend to take the log transformation to normalize the data.

Amount / Log AMT_CREDIT

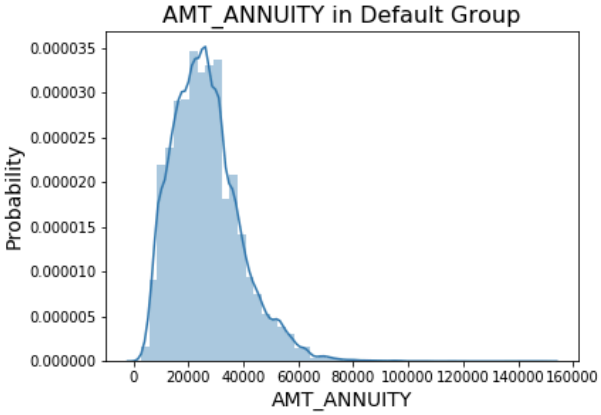


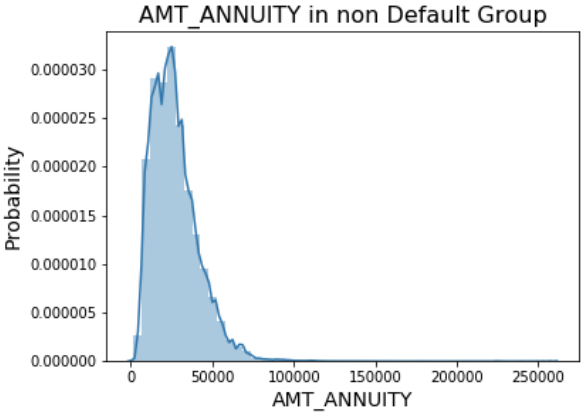


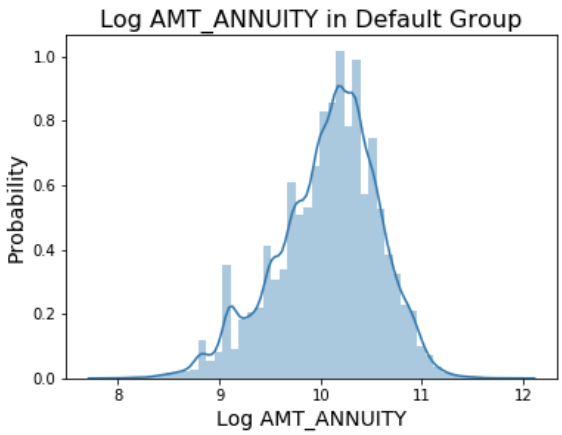


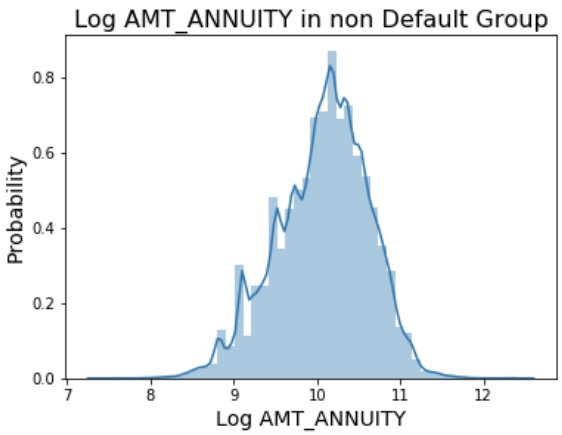


Amount / Log AMT_ANNUITY

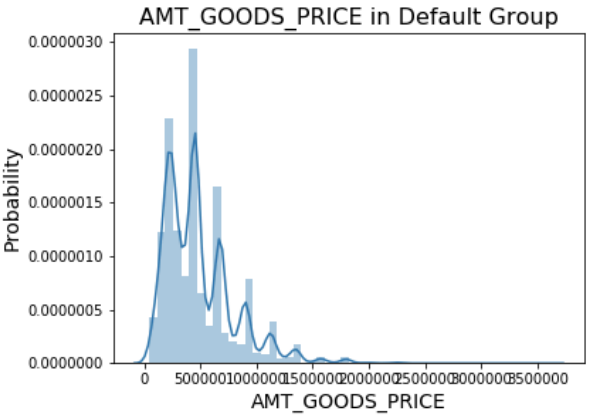


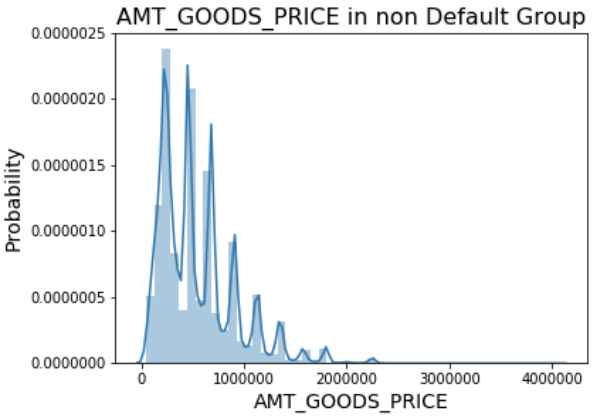


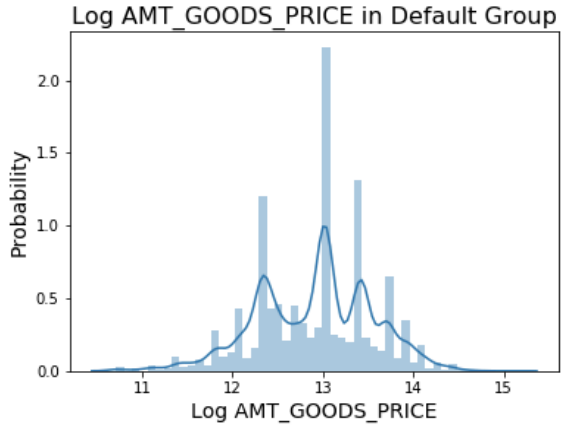


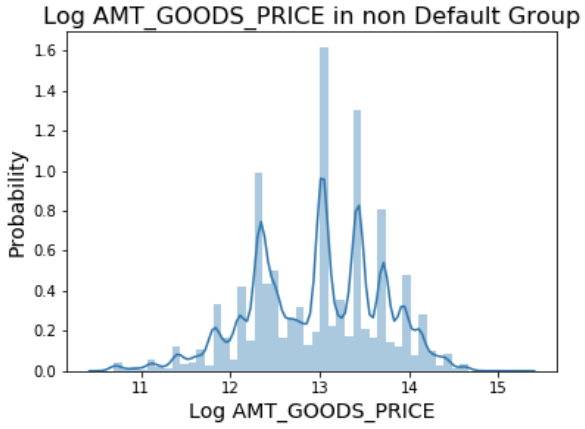


Amount / Log AMT_GOODS_PRICE

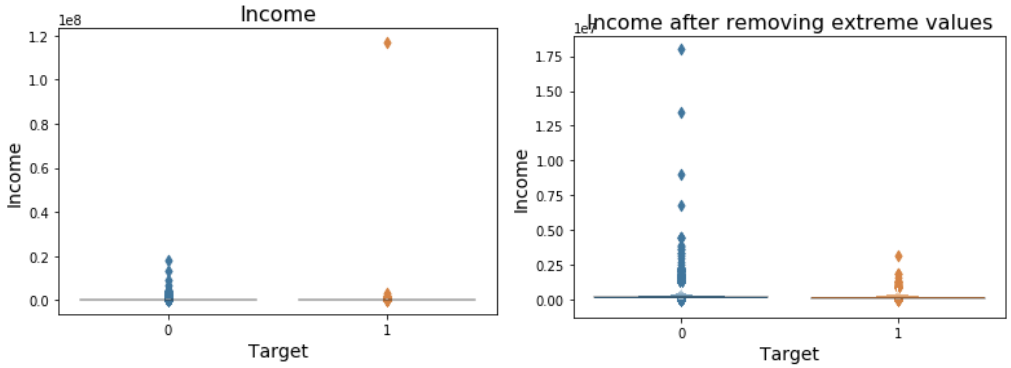


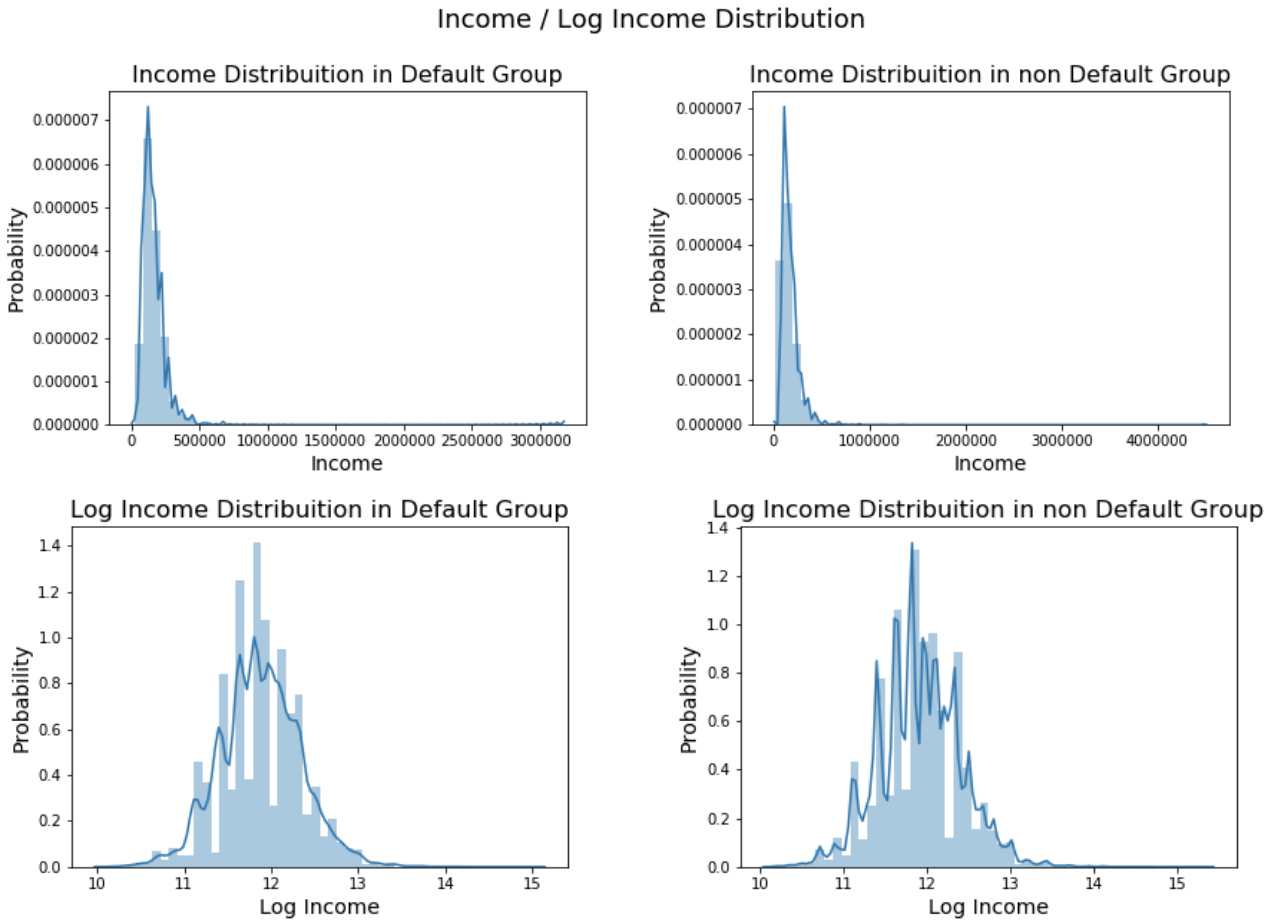




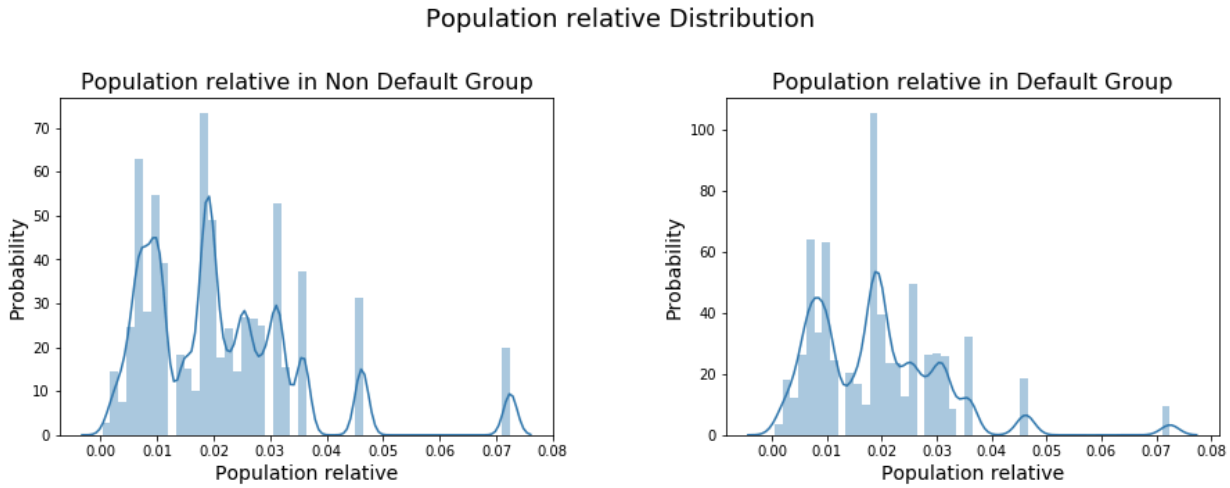


f. There is 1 extremely large income value in the default group ($120 million), and 4 very large values (> $7.5 million) in the non-default group. Recommend to remove the outliers and then do log transformation to normalize the income data.

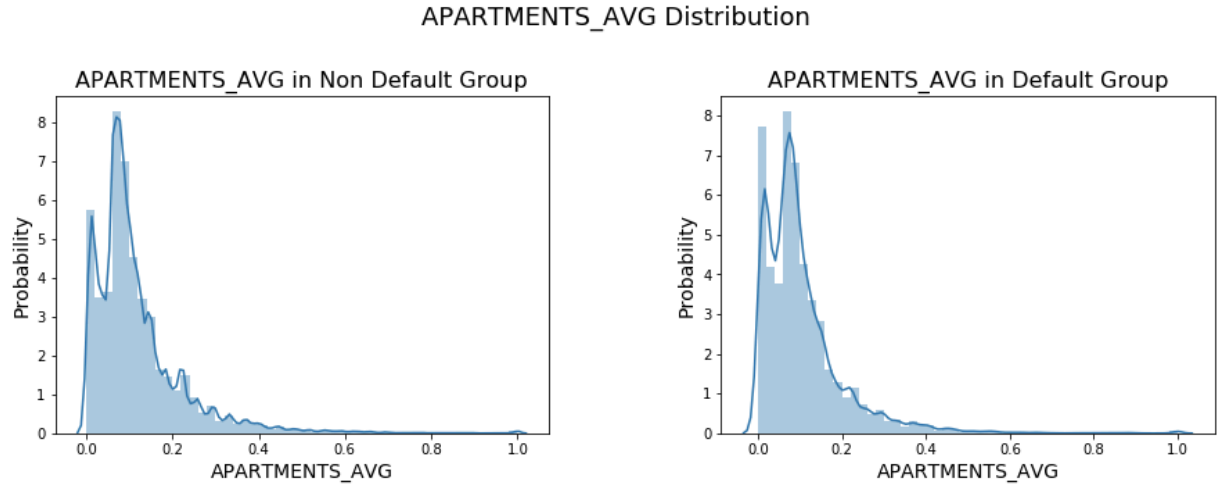




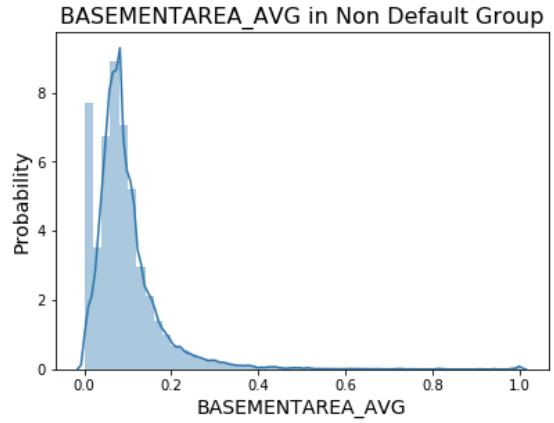
g. 'REGION_POPULATION_RELATIVE' variable has multiple modes, most of the clients live in lower population density places. Visually the distribution for the default and non-default groups are similar.

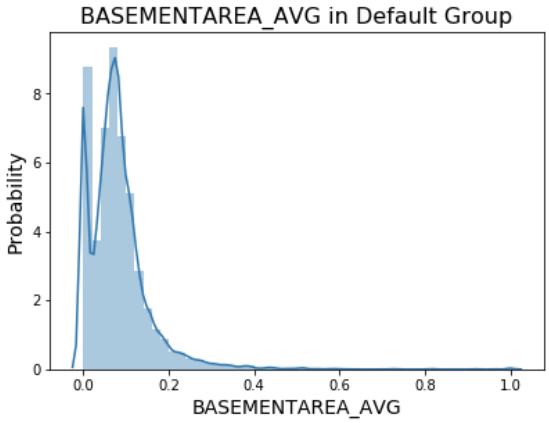


h. The distributions for _AVG, _MODE and _MEDI numeric variables are all kind of skewed, but since they are already normalized to range 0 to 1, no additional transformation is done to these variables. For missing values in these variables, can impute using the median. Here we are only showing the KDE plot for a few variables.

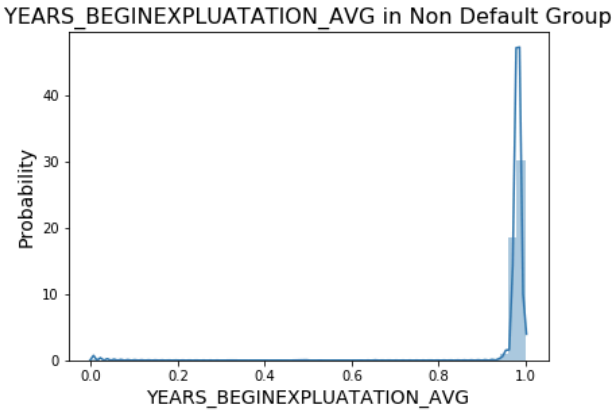


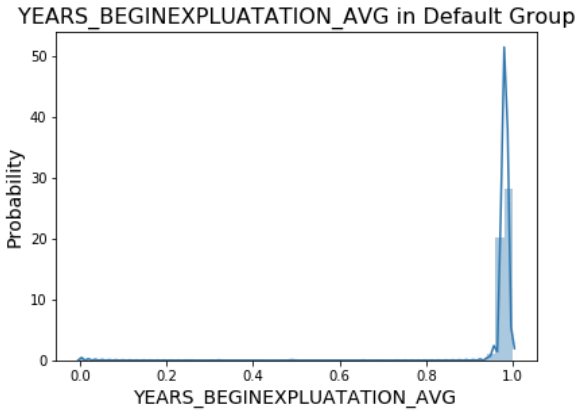
BASEMENTAREA_AVG Distribution



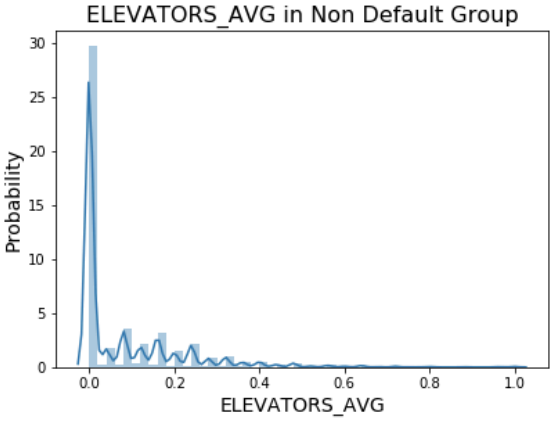


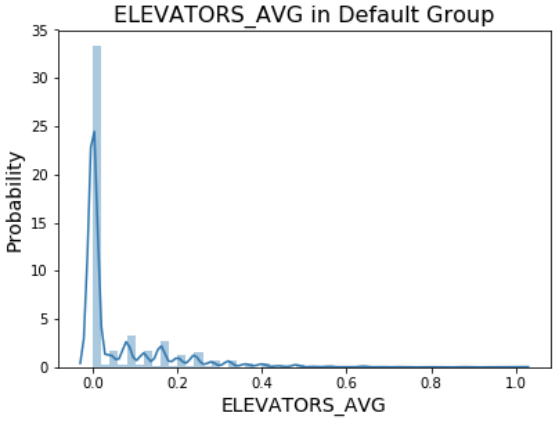
YEARS_BEGINEXPLUATATION_AVG Distribution



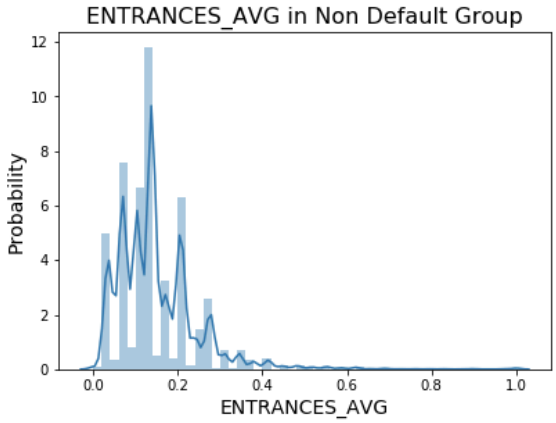


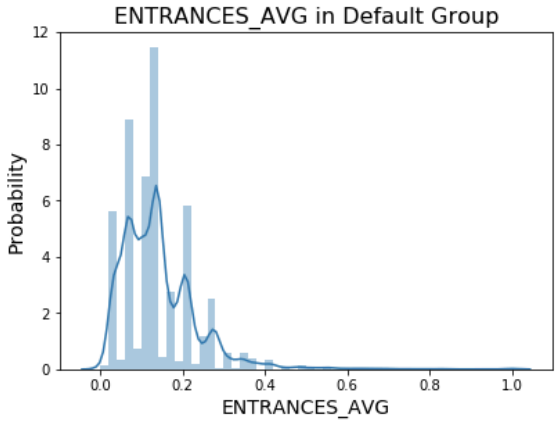
ELEVATORS_AVG Distribution



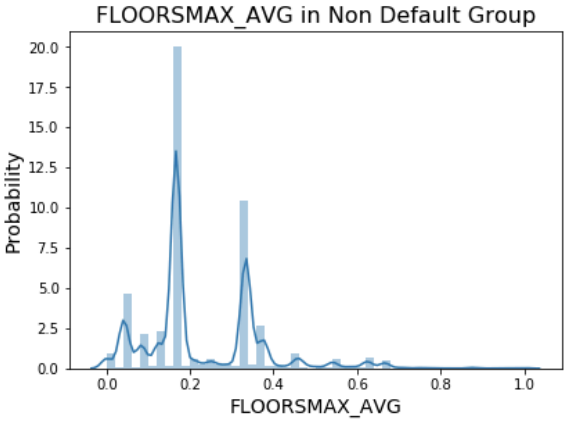


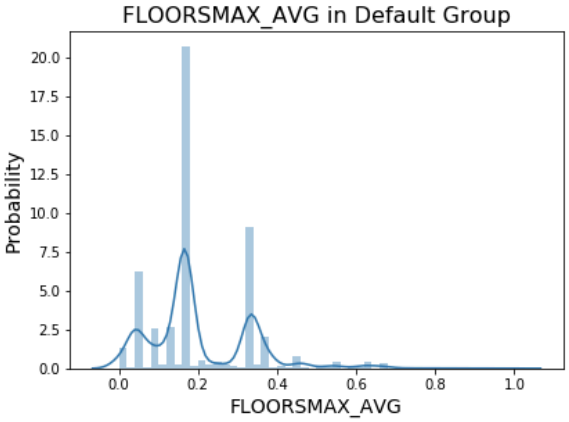
ENTRANCES_AVG Distribution



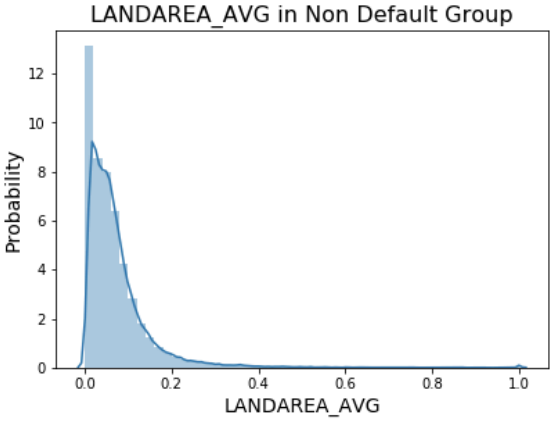


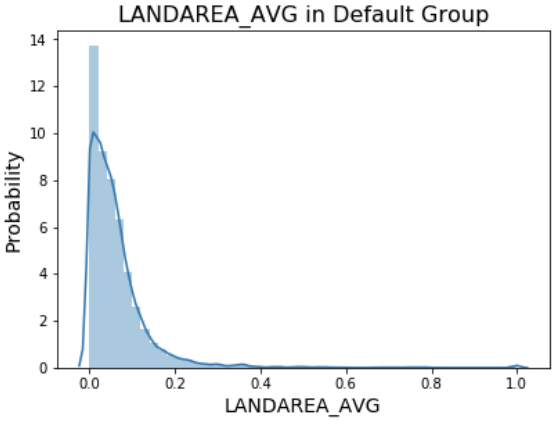
FLOORSMAX_AVG Distribution



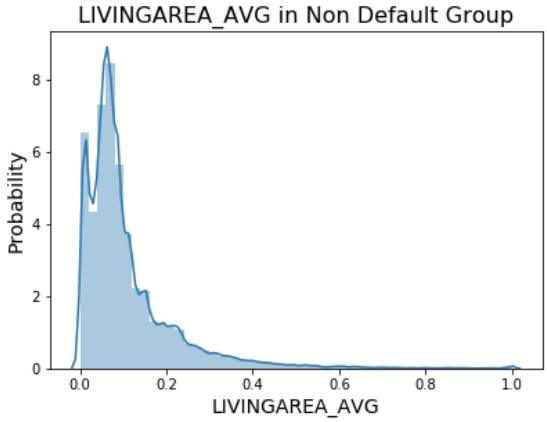


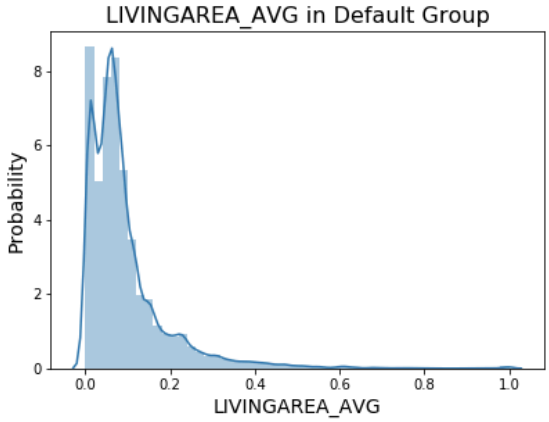
LANDAREA_AVG Distribution



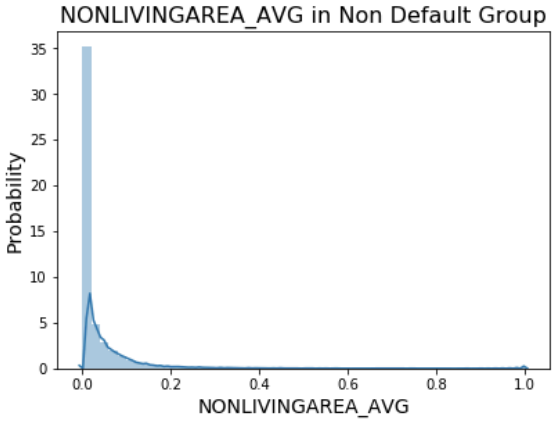


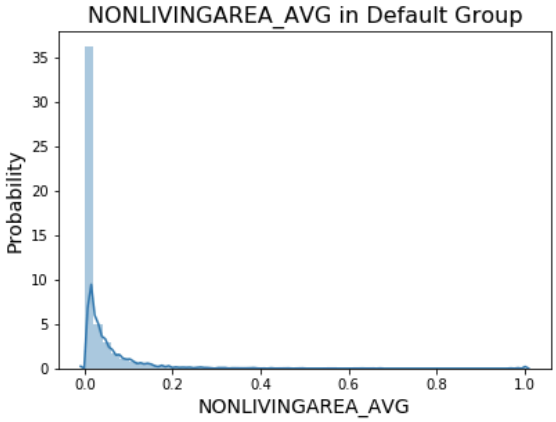
LIVINGAREA_AVG Distribution



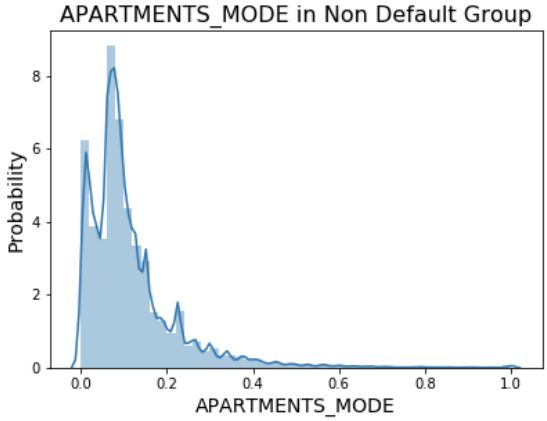


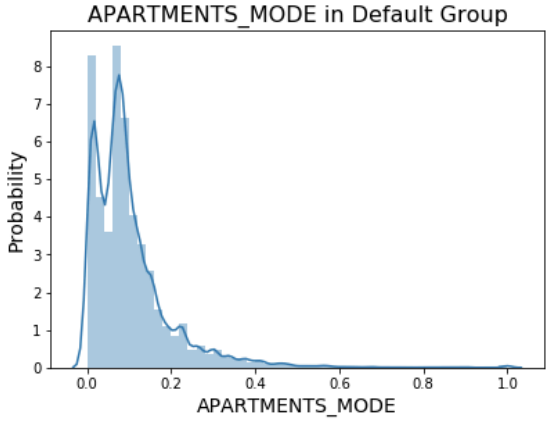
NONLIVINGAREA_AVG Distribution



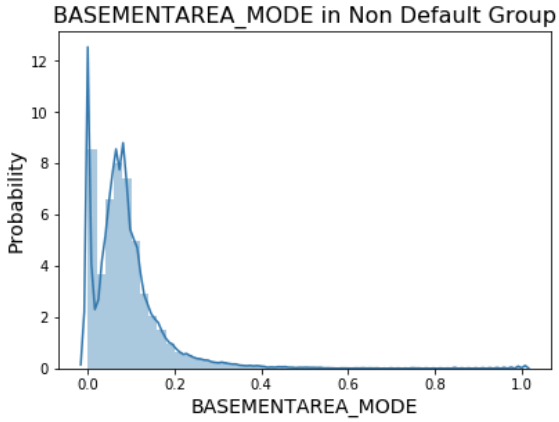


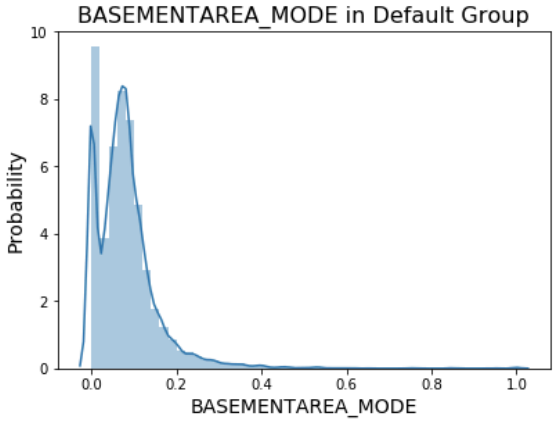
APARTMENTS_MODE Distribution



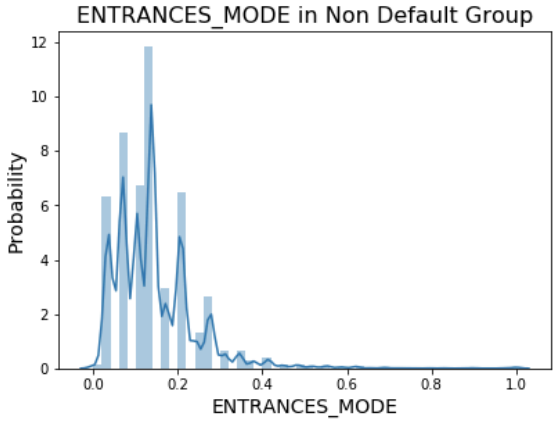


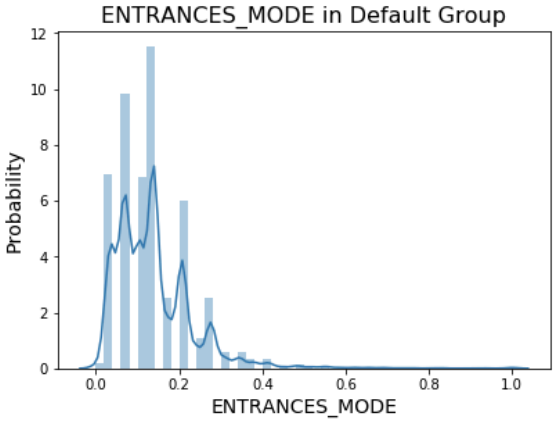
BASEMENTAREA_MODE Distribution



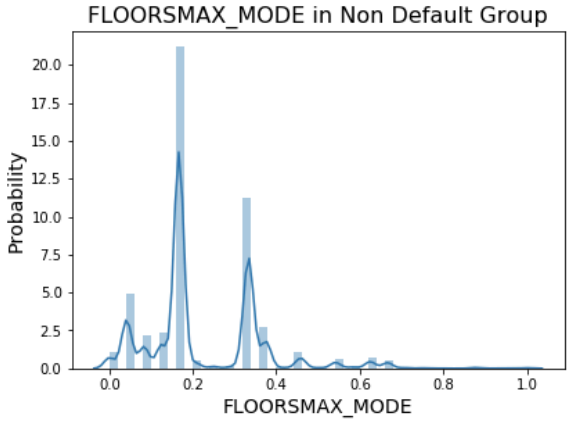


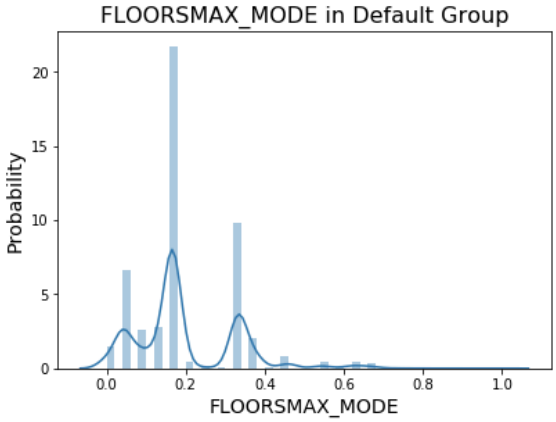
ENTRANCES_MODE Distribution



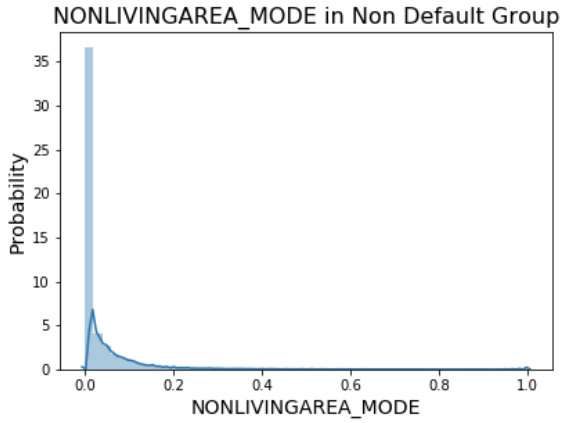


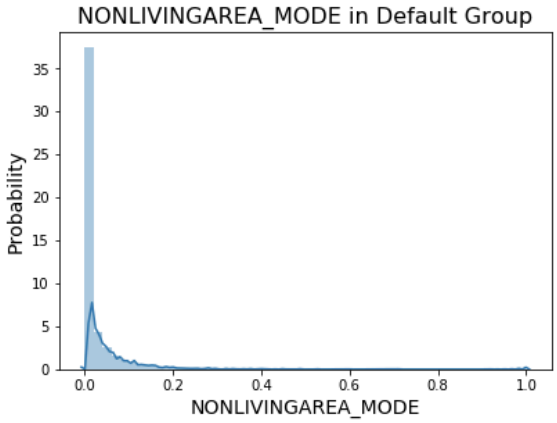
FLOORSMAX_MODE Distribution



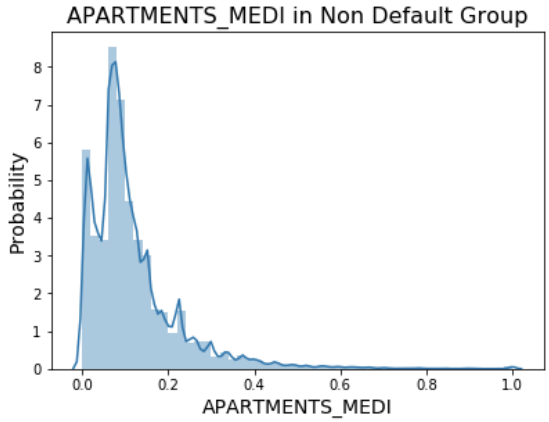


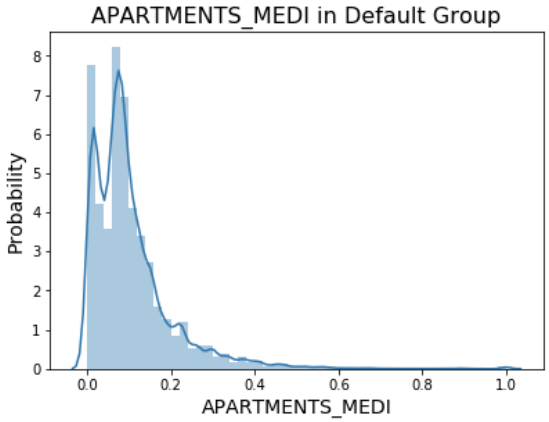
NONLIVINGAREA_MODE Distribution



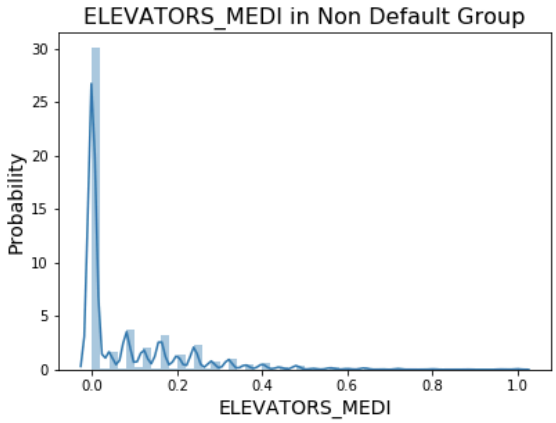


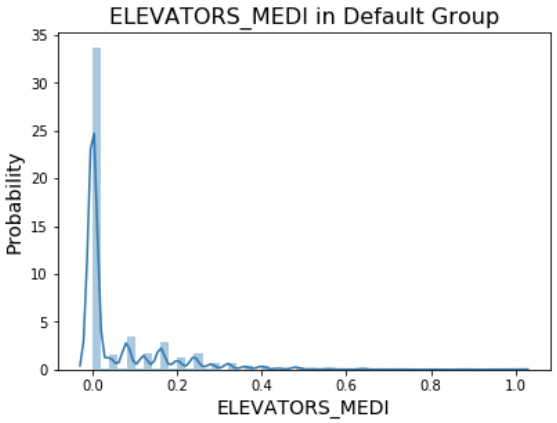
APARTMENTS_MEDI Distribution

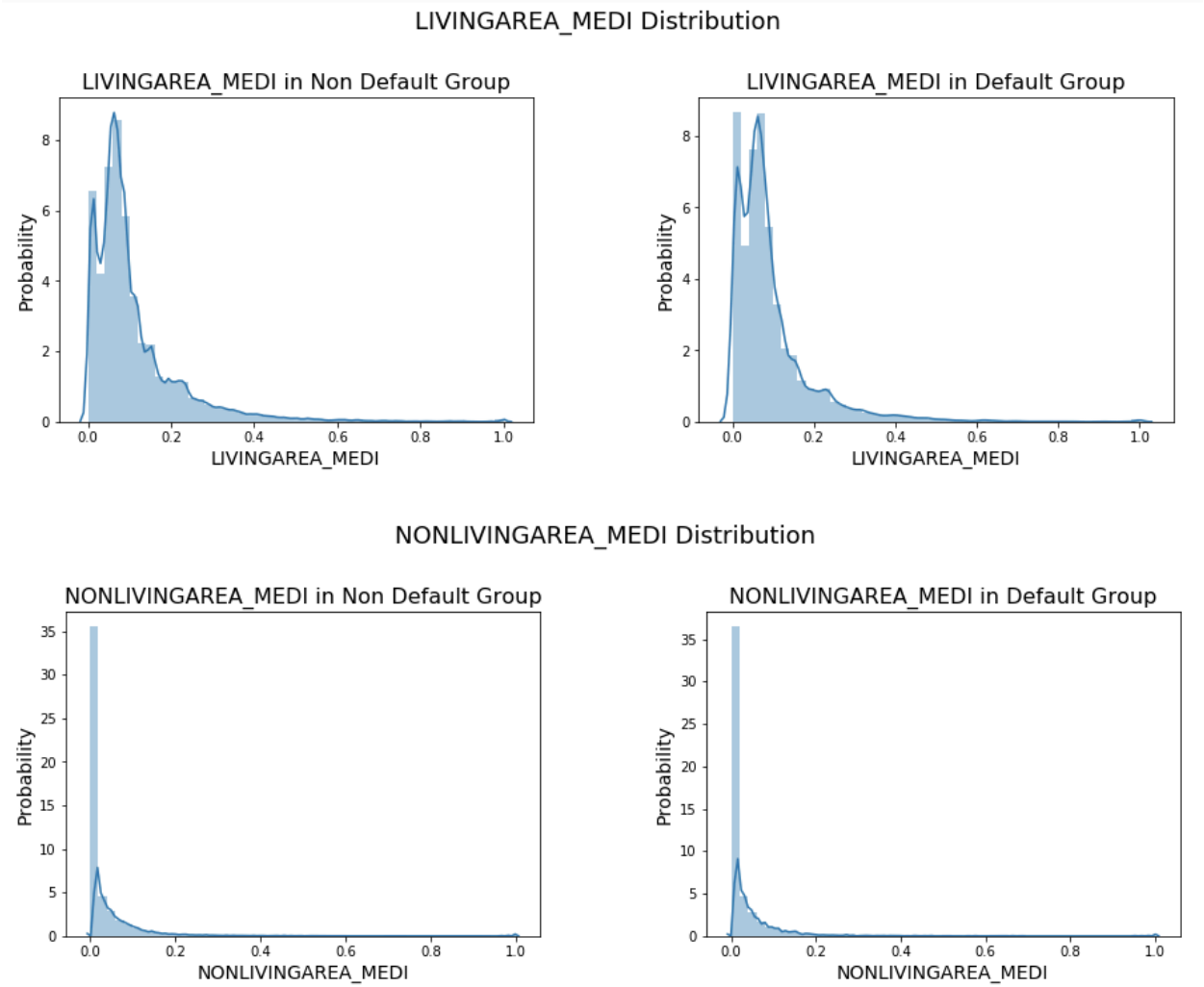




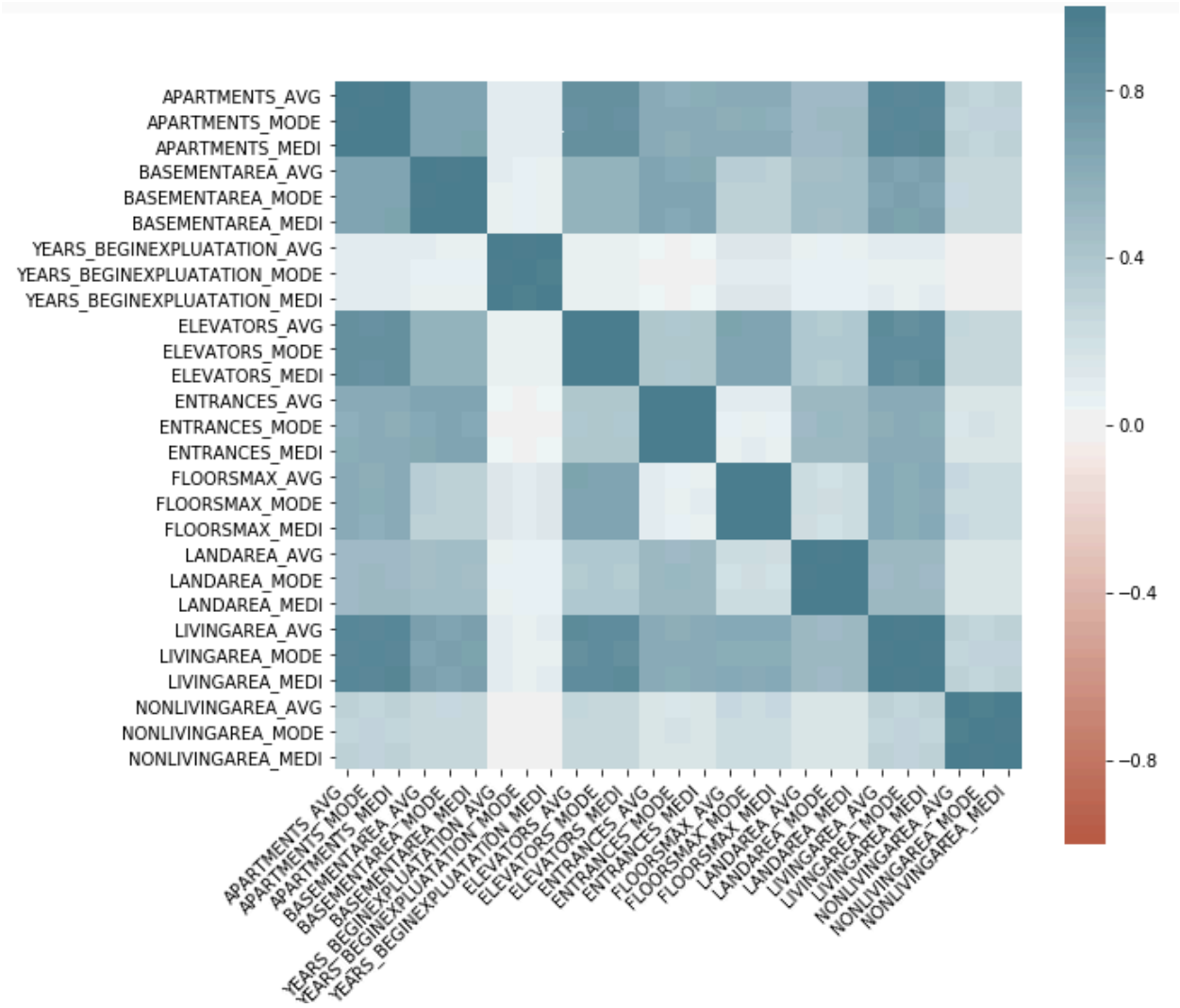
ELEVATORS_MEDI Distribution







i. Correlations between the _AVG, _MODE and _MEDI variables. Some groups of variables are highly correlated with some groups, for example, 'APARTMENTS' variables are highly correlated with 'LIVINGAREA' variables. At this point we don't remove any one of them**.**



4) Data cleaning and transformation, high correlation column removal and outlier removal, missing value imputations.

   a. Remove outliers with extremely large income accounts. Remove gender = XNA rows.
   b. For categorical features: (i) When cardinality is high, combine some low frequency categories. (ii) When missing values ('nan') are present, code it as a new category. (iii) Use one hot encoding to convert all categorical variables into numeric dummy variables and drop the original categorical variables.
   c. For numeric features: (i) Drop variables with high or perfect linear correlations with others. (ii) Do log transformation to normalize some "AMT" variables. (iii) Floor or cap some "DAYS" variables based on exploratory column analysis. (iv) Impute all missing values using the median of the non-missing part.

Section 3. Statistical Tests and Inference

In this section we perform a few hypothesis tests to check if the distributions of a numeric / categorical variable are the same in the default and non-default groups.

1) Two sample t-test of equal mean of numeric variables on default and non-default groups.

   The null hypothesis is that the mean values of the variable are the same in default and non-default groups. The alternative hypothesis is that the mean values are different.

   The 2-sample t-test assumes the distribution of the 2 samples are approximately normal, whereas in our cases, many variables are highly skewed. We first do the test using original scales of the variables, next we chop the middle part of some of the variables where the distribution is less skewed and pass the new ranges to the test again, to check if the results are different.

   For most of the variables, their mean values are statistically different in the default and non-default groups. Among the variables tested, only AMT_ANNUITY and SUM_AMT_CREDIT_MAX_OVERDUE have insignificant p-values, meaning there is no evidence to reject the null hypothesis.

```
t-test p-value for variable AMT_INCOME_TOTAL is 0.0000.
t-test p-value for variable YEARS_BEGINEXPLUATATION_AVG is 0.0167.
t-test p-value for variable MIN_DAYS_CREDIT_UPDATE is 0.0000.
t-test p-value for variable MIN_DAYS_CREDIT_UPDATE is 0.0000.
t-test p-value for variable MAX_DAYS_CREDIT_ENDDATE is 0.0000.
t-test p-value for variable MIN_DAYS_CREDIT is 0.0000.
t-test p-value for variable EXT_SOURCE_1 is 0.0000.
t-test p-value for variable EXT_SOURCE_2 is 0.0000.
t-test p-value for variable EXT_SOURCE_3 is 0.0000.
t-test p-value for variable MAX_DAYS_ENDDATE_FACT is 0.0000.
t-test p-value for variable DAYS_BIRTH is 0.0000.
t-test p-value for variable DAYS_REGISTRATION is 0.0000.
t-test p-value for variable DAYS_LAST_PHONE_CHANGE is 0.0000.
t-test p-value for variable DAYS_ID_PUBLISH is 0.0000.
t-test p-value for variable AMT_CREDIT is 0.0000.
t-test p-value for variable AMT_GOODS_PRICE is 0.0000.
t-test p-value for variable AMT_ANNUITY is 0.6374.
t-test p-value for variable DAYS_EMPLOYED is 0.0000.
t-test p-value for variable SUM_AMT_CREDIT_MAX_OVERDUE is 0.0727.
t-test p-value for variable SUM_AMT_CREDIT_SUM_OVERDUE is 0.0000.
t-test p-value for variable SUM_AMT_CREDIT_SUM is 0.0000.
```

   Next we chop the variables to a range that the variables are less skewed, then do 2 sample t test again. We see some variables' p-values changed from significant to insignificant as we only test on the middle part of the distribution. It seems the differences are mainly caused by the tails.

```
t-test p-value for variable AMT_INCOME_TOTAL is 0.0000.
t-test p-value for variable REGION_POPULATION_RELATIVE is 0.0210.
t-test p-value for variable YEARS_BEGINEXPLUATATION_AVG is 0.0000.
t-test p-value for variable MIN_DAYS_CREDIT_UPDATE is 0.0000.
t-test p-value for variable MAX_DAYS_CREDIT_UPDATE is 0.5390.
t-test p-value for variable MAX_DAYS_CREDIT_ENDDATE is 0.4993.
t-test p-value for variable MIN_DAYS_CREDIT is 0.0000.
t-test p-value for variable EXT_SOURCE_1 is 0.0000.
t-test p-value for variable EXT_SOURCE_2 is 0.0000.
t-test p-value for variable EXT_SOURCE_3 is 0.0000.
t-test p-value for variable MAX_DAYS_ENDDATE_FACT is 0.0638.
t-test p-value for variable DAYS_BIRTH is 0.0000.
t-test p-value for variable DAYS_REGISTRATION is 0.6056.
t-test p-value for variable DAYS_LAST_PHONE_CHANGE is 0.6424.
t-test p-value for variable DAYS_ID_PUBLISH is 0.0000.
t-test p-value for variable AMT_CREDIT is 0.0000.
t-test p-value for variable AMT_GOODS_PRICE is 0.0000.
t-test p-value for variable AMT_ANNUITY is 0.6077.
t-test p-value for variable DAYS_EMPLOYED is 0.0000.
t-test p-value for variable SUM_AMT_CREDIT_MAX_OVERDUE is 0.0016.
t-test p-value for variable SUM_AMT_CREDIT_SUM_OVERDUE is 0.8851.
t-test p-value for variable SUM_AMT_CREDIT_SUM is 0.0000.
```

2) Test of equal distributions of numeric variables on default and non-default groups.

   Use Kolmogorov-Smirnov (KS) test to check if the distributions of one numeric variable are the same on default and non-default groups. Under the null hypothesis the two distributions are identical. The alternative hypothesis is that their distributions are different. The KS test is only valid for continuous distributions.

   Among the numeric features tested, none of them have the same distribution in the default and non-default groups. This is expected as testing the distributions are similar or the same is more restrictive than having the same mean values. Intuitively at least their means should be pretty close, and we already saw in the previous t-test that only 2 variables have similar mean values on the default and non-default groups.

```
ks-test pvalue for variable AMT_INCOME_TOTAL is 0.0000.
ks-test pvalue for variable YEARS_BEGINEXPLUATATION_AVG is 0.0000.
ks-test pvalue for variable MIN_DAYS_CREDIT_UPDATE is 0.0000.
ks-test pvalue for variable MIN_DAYS_CREDIT_UPDATE is 0.0000.
ks-test pvalue for variable MAX_DAYS_CREDIT_ENDDATE is 0.0000.
ks-test pvalue for variable MIN_DAYS_CREDIT is 0.0000.
ks-test pvalue for variable EXT_SOURCE_1 is 0.0000.
ks-test pvalue for variable EXT_SOURCE_2 is 0.0000.
ks-test pvalue for variable EXT_SOURCE_3 is 0.0000.
ks-test pvalue for variable MAX_DAYS_ENDDATE_FACT is 0.0000.
ks-test pvalue for variable DAYS_BIRTH is 0.0000.
ks-test pvalue for variable DAYS_REGISTRATION is 0.0000.
ks-test pvalue for variable DAYS_LAST_PHONE_CHANGE is 0.0000.
ks-test pvalue for variable DAYS_ID_PUBLISH is 0.0000.
ks-test pvalue for variable AMT_CREDIT is 0.0000.
ks-test pvalue for variable AMT_GOODS_PRICE is 0.0000.
ks-test pvalue for variable AMT_ANNUITY is 0.0000.
ks-test pvalue for variable DAYS_EMPLOYED is 0.0000.
ks-test pvalue for variable SUM_AMT_CREDIT_MAX_OVERDUE is 0.0000.
ks-test pvalue for variable SUM_AMT_CREDIT_SUM_OVERDUE is 0.0031.
ks-test pvalue for variable SUM_AMT_CREDIT_SUM is 0.0000.
```

3) Test if the default rates from 2 categories are statistically the same using z-test.

   This is testing the proportion of default in 2 categories are the same. Pick a few categorical variables with 2 levels, and compare the default rate in each level. Test results show a few variables having p-values > 0.05, indicating the default proportions are not significantly different in its 2 categories. Such variables may not have a good separating power of default loans from non-default loans.

```
z-test pvalue for variable NAME_CONTRACT_TYPE is 0.0000.
z-test pvalue for variable FLAG_OWN_CAR is 0.0000.
z-test pvalue for variable FLAG_OWN_REALTY is 0.0007.
z-test pvalue for variable FLAG_DOCUMENT_2 is 0.0027.
z-test pvalue for variable FLAG_DOCUMENT_3 is 0.0000.
z-test pvalue for variable FLAG_DOCUMENT_5 is 0.8610.
z-test pvalue for variable FLAG_DOCUMENT_6 is 0.0000.
z-test pvalue for variable FLAG_DOCUMENT_7 is 0.3994.
z-test pvalue for variable FLAG_DOCUMENT_8 is 0.0000.
z-test pvalue for variable FLAG_DOCUMENT_9 is 0.0158.
z-test pvalue for variable FLAG_DOCUMENT_11 is 0.0190.
z-test pvalue for variable FLAG_DOCUMENT_13 is 0.0000.
z-test pvalue for variable FLAG_DOCUMENT_14 is 0.0000.
z-test pvalue for variable FLAG_DOCUMENT_15 is 0.0003.
z-test pvalue for variable FLAG_DOCUMENT_16 is 0.0000.
z-test pvalue for variable FLAG_DOCUMENT_17 is 0.0611.
z-test pvalue for variable FLAG_DOCUMENT_18 is 0.0000.
z-test pvalue for variable FLAG_DOCUMENT_19 is 0.4516.
z-test pvalue for variable FLAG_DOCUMENT_20 is 0.9049.
z-test pvalue for variable FLAG_DOCUMENT_21 is 0.0397.
```

4) Chi-squared test of independence between categorical variables and the target.

   This is testing the distribution of counts in each category of a categorical variable is independent of the target group, whether default or non-default. Rejecting the null hypothesis would mean the categorical variable differs in the 2 target groups, in which case this categorical variable may be a good feature to separate out default vs. non-default loans; otherwise we would have no evidence to conclude the distribution of counts are different in the 2 target groups.

   Choose a few categorical variables where there are not too many levels, and need to make sure there are > 0 counts in every category in both default and non-default loans.

   It turns out that the p-values obtained from this Chi-squared test are exactly the same as the p-values coming out of the z-test in 3).

```
chisq-test pvalue for variable NAME_CONTRACT_TYPE is 0.0000.
chisq-test pvalue for variable FLAG_OWN_CAR is 0.0000.
chisq-test pvalue for variable FLAG_OWN_REALTY is 0.0007.
chisq-test pvalue for variable FLAG_DOCUMENT_2 is 0.0027.
chisq-test pvalue for variable FLAG_DOCUMENT_3 is 0.0000.
chisq-test pvalue for variable FLAG_DOCUMENT_5 is 0.8610.
chisq-test pvalue for variable FLAG_DOCUMENT_6 is 0.0000.
chisq-test pvalue for variable FLAG_DOCUMENT_7 is 0.3994.
chisq-test pvalue for variable FLAG_DOCUMENT_19 is 0.4516.
chisq-test pvalue for variable FLAG_DOCUMENT_20 is 0.9049.
chisq-test pvalue for variable FLAG_DOCUMENT_21 is 0.0397.
chisq-test pvalue for variable WEEKDAY_APPR_PROCESS_START is 0.0174.
```