Credit EDA Assignment

Should you get the loan? Will you pay the loan on time?

Deekshashree KM

Problem Statement:

When the company receives a loan application, the company has to decide for loan approval based on the applicant’s profile. Risks associated with the bank’s decision:

- Not approving loan to applicant who can repay the amount.
- Approving loan to applicant who can't repay or likely to default.

Both result in loss of business to the company. So, The loan providing companies find it hard to give loans to the people due to their insufficient or non-existent credit history. Because of that, some consumers use it as their advantage by becoming a defaulter.

Data given in ‘**applicant_data**' file contains the information about the loan application at the time of applying for the loan. It contains two types of scenarios:

- The client with payment difficulties: he/she had late payment more than X days on at least one of the first Y instalments of the loan in our sample (**Defaulters**)
- All other cases: All other cases when the payment is paid on time.(**Non-Defaulters**)

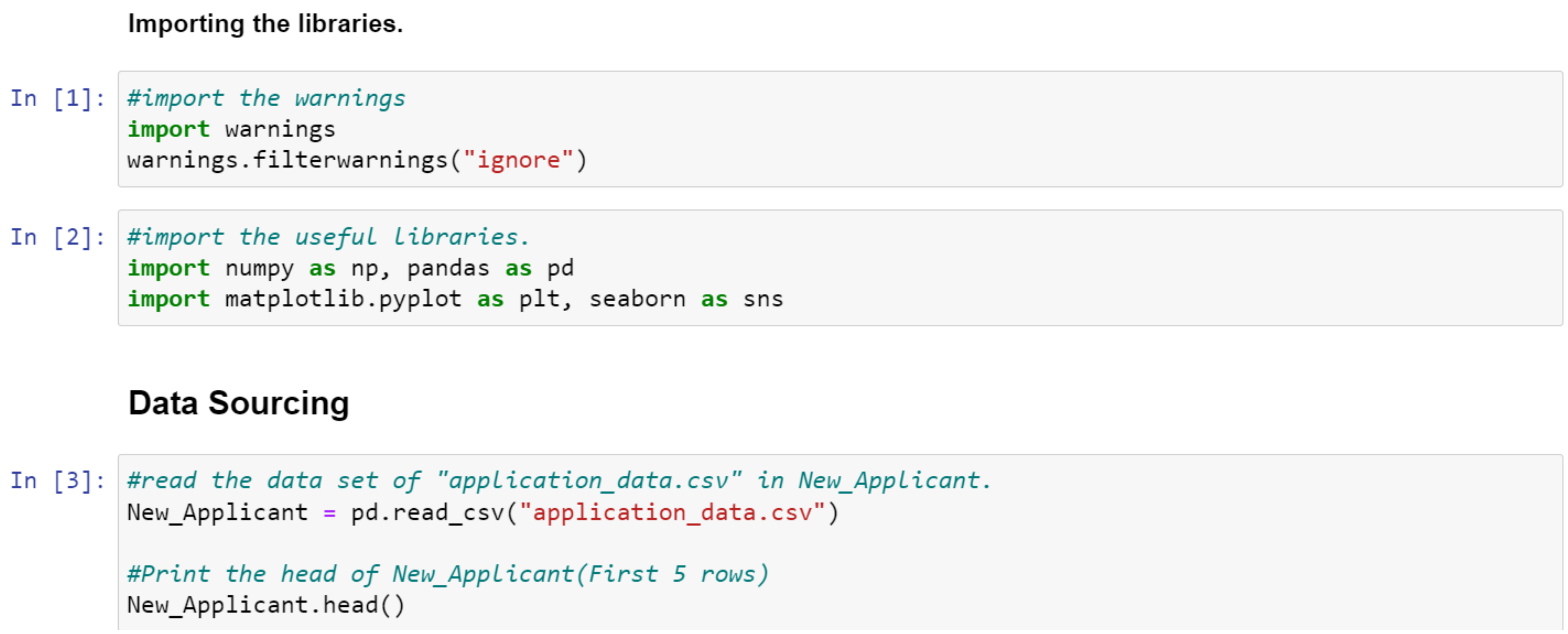
Data given in '**previous_application**' contains information about the client’s previous loan data. When a client applies for a loan, there are four types of decisions that could be taken by the client/company):

- **Approved**: The Company has approved loan Application
- **Cancelled**: The client cancelled the application sometime during approval. Either the client changed her/his mind about the loan or in some cases due to a higher risk of the client he received worse pricing which he did not want.
- **Refused**: The company had rejected the loan (because the client does not meet their requirements etc.).
- **Unused offer**: Loan has been cancelled by the client but on different stages of the process.

Objective:

- To identify patterns which indicate if a client has difficulty paying their installments (i.e., company wants to understand the driving factors/variables or strong indicators behind loan default) Based on the trend analysis/ result, company can: Deny loan, Reduce amount of loan or lend loan at high interest to risky applicant. Also, applicants who are capable of repaying the loan are not rejected.

In this case study, will use EDA to understand how consumer attributes and loan attributes influence the tendency of default.



Importing the libraries.

```
In [1]: #import the warnings
        import warnings
        warnings.filterwarnings("ignore")
```

```
In [2]: #import the useful libraries.
        import numpy as np, pandas as pd
        import matplotlib.pyplot as plt, seaborn as sns
```

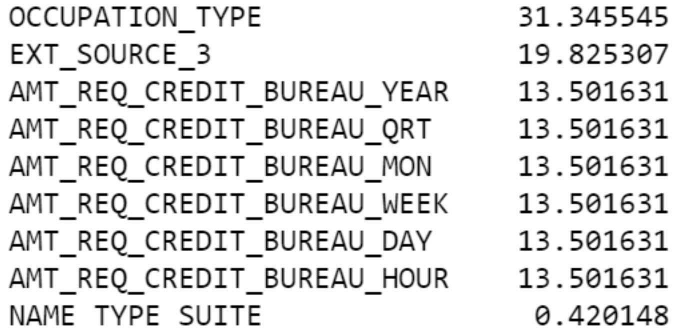
Data Sourcing

```
In [3]: #read the data set of "application_data.csv" in New_Applicant.
        New_Applicant = pd.read_csv("application_data.csv")

        #Print the head of New_Applicant(First 5 rows)
        New_Applicant.head()
```

Required libraries are imported and loaded data set.

Data cleaning activities:

- Application_data was initially checking for missing values.
- Columns having more than 40% of null values were dropped, as it’s insufficient for analysis.
- Columns having less than 40% of null values were checked if those missing value can be imputed or ‘dropna’ related rows

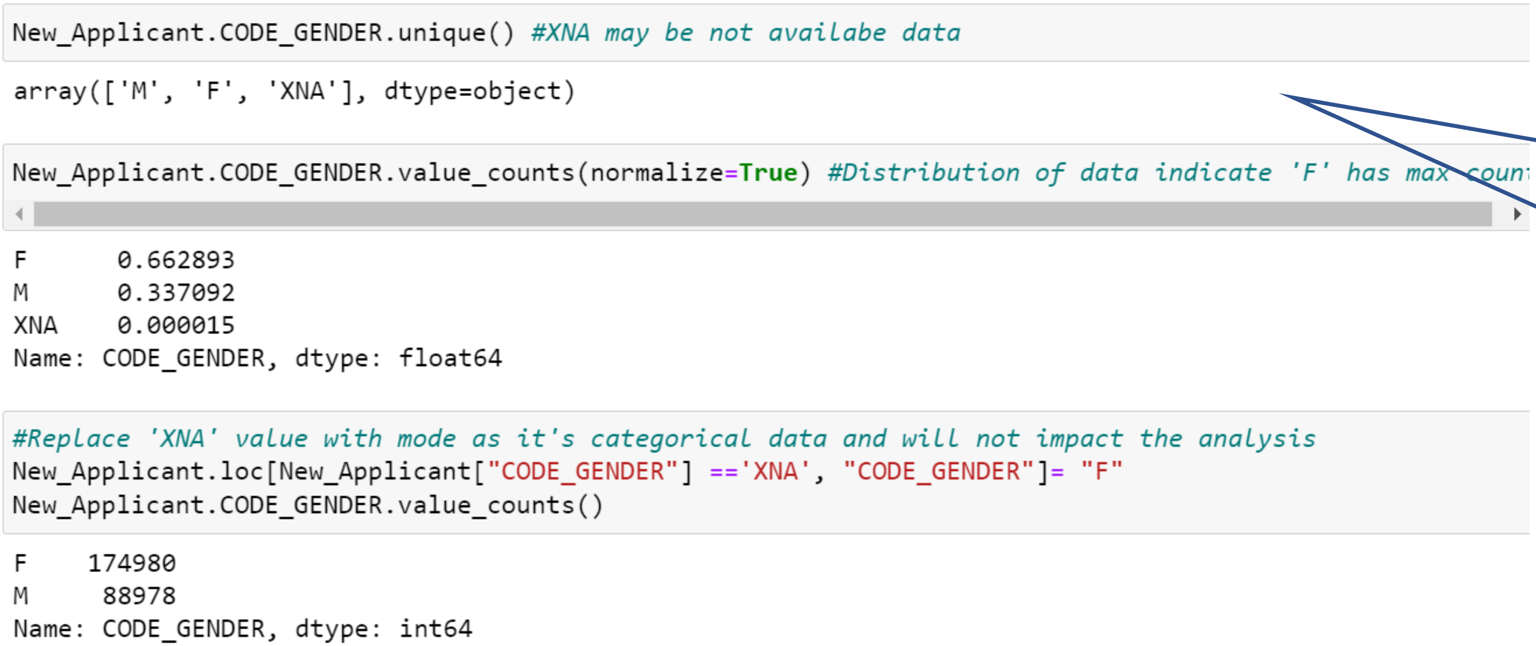


Column name	Null %
OCCUPATION_TYPE	31.345545
EXT_SOURCE_3	19.825307
AMT_REQ_CREDIT_BUREAU_YEAR	13.501631
AMT_REQ_CREDIT_BUREAU_QRT	13.501631
AMT_REQ_CREDIT_BUREAU_MON	13.501631
AMT_REQ_CREDIT_BUREAU_WEEK	13.501631
AMT_REQ_CREDIT_BUREAU_DAY	13.501631
AMT_REQ_CREDIT_BUREAU_HOUR	13.501631
NAME TYPE SUITE	0.420148

Impute Missing values

- ‘**OCCUPATION_TYPE’** column had 31% null values. Occupation type had multiple categories and required column data. So, to impute the missing data checked for ‘mode’ (most frequent type), since, it’s categorical column and distribution mode has ‘Laborers’ contributing 26% data. But can't replace 31% missing data with laborer as this would impact the analysis. Hence, missing values are retained as is.
- ‘**AMT_REQ_CREDIT_BUREAU**’ columns had 13% missing data. Maximum distribution of data was seen at one value. Hence, dropped the rows with missing value here as this missing data will not have major impact to column (the column already has constant/max value for analysis, if required)
- Similarly for ‘**NAME_TYPE_SUITE**’ column, Maximum distribution of data is seen at one value. Hence, dropped the rows with missing value.
- **AMT_ANNUITY** column had *very less % of null value*. Since it’s a numerical column-checked for outliers if missing value can be replaced with Mean or median. And there were Outlier in the annuity, as the difference between max value and 75% values was high. So, the missing value was replaced with Median value.
- Columns with no significant data for analysis were dropped- 'EXT_SOURCE_2', 'EXT_SOURCE_3','FLAG_DOCUMENT_2', 'FLAG_DOCUMENT_3','FLAG_DOCUMENT_4', 'FLAG_DOCUMENT_5', 'FLAG_DOCUMENT_6','FLAG_DOCUMENT_7', 'FLAG_DOCUMENT_8', 'FLAG_DOCUMENT_9','FLAG_DOCUMENT_10', 'FLAG_DOCUMENT_11', 'FLAG_DOCUMENT_12’, 'FLAG_DOCUMENT_13', 'FLAG_DOCUMENT_14', 'FLAG_DOCUMENT_15', 'FLAG_DOCUMENT_16', 'FLAG_DOCUMENT_17’, 'FLAG_DOCUMENT_18','FLAG_DOCUMENT_19', 'FLAG_DOCUMENT_20', 'FLAG_DOCUMENT_21’.

Further, data in each columns were checked to identify errors in data and data type.



```
New_Applicant.CODE_GENDER.unique() #XNA may be not availabe data
```

```
array(['M', 'F', 'XNA'], dtype=object)
```

```
New_Applicant.CODE_GENDER.value_counts(normalize=True) #Distribution of data indicate 'F' has max coun
```

```
F      0.662893
M      0.337092
XNA    0.000015
Name: CODE_GENDER, dtype: float64
```

```
#Replace 'XNA' value with mode as it's categorical data and will not impact the analysis
New_Applicant.loc[New_Applicant["CODE_GENDER"] =='XNA', "CODE_GENDER"]= "F"
New_Applicant.CODE_GENDER.value_counts()
```

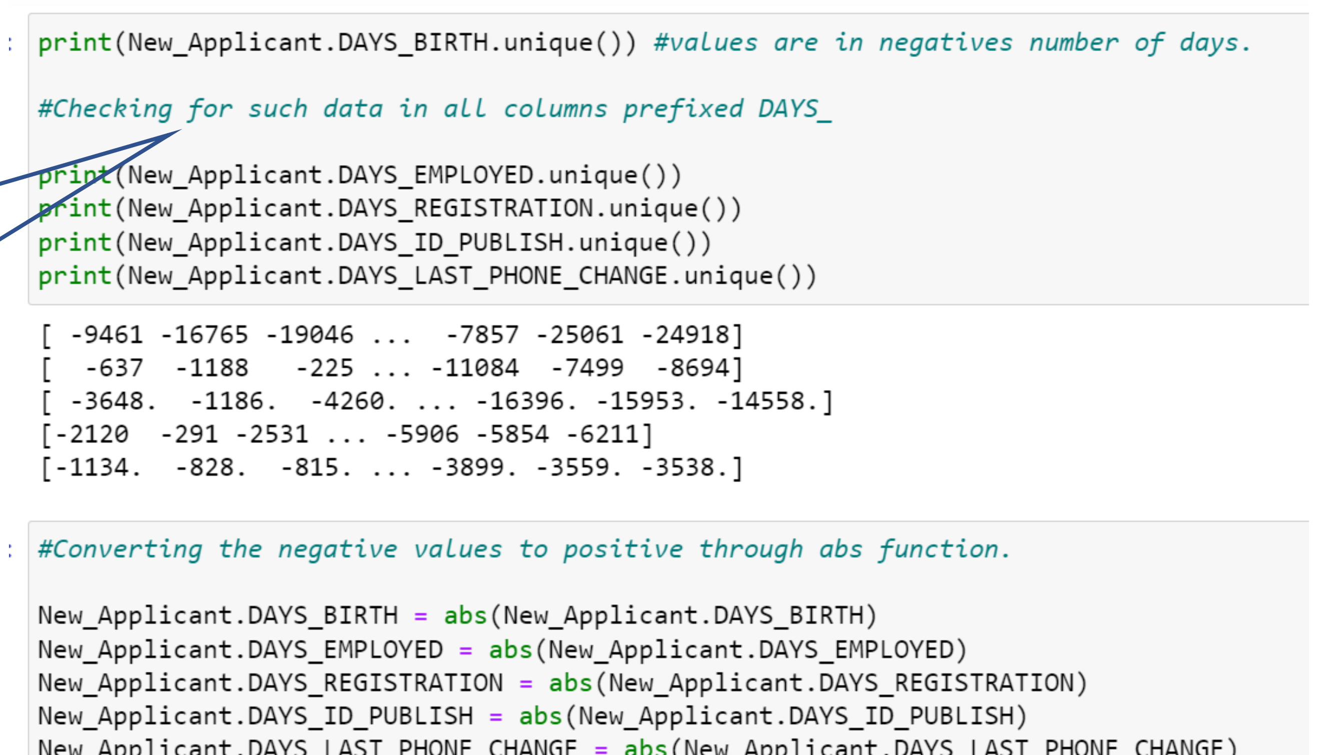
```
F    174980
M     88978
Name: CODE_GENDER, dtype: int64
```

Errors in data

Gender had ‘XNA’ which may be not available data. Since, it’s categorical data, replaced it with distribution mode- Female ‘F’.

Errors in data

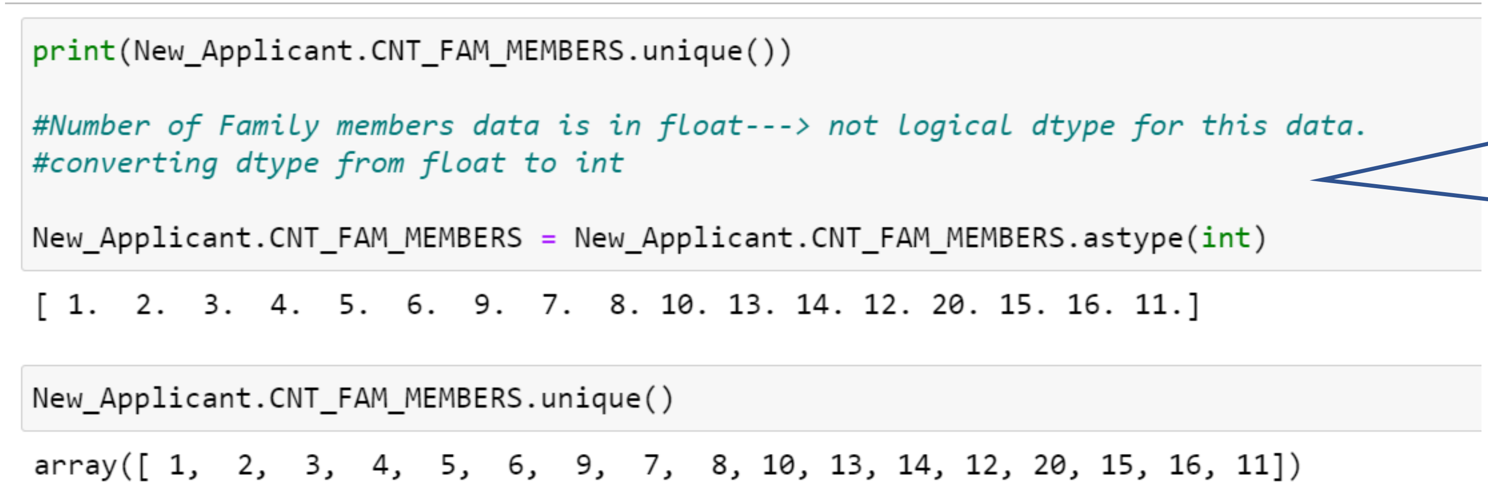
Columns prefixed DAYS_ has negative values for no. of days. So, values are converted to positive number using absolute function ‘abs()’



```
print(New_Applicant.DAYS_BIRTH.unique()) #values are in negatives number of days.

#Checking for such data in all columns prefixed DAYS_

print(New_Applicant.DAYS_EMPLOYED.unique())
print(New_Applicant.DAYS_REGISTRATION.unique())
print(New_Applicant.DAYS_ID_PUBLISH.unique())
print(New_Applicant.DAYS_LAST_PHONE_CHANGE.unique())
```

```
[ -9461 -16765 -19046 ...  -7857 -25061 -24918]
[  -637  -1188   -225 ... -11084  -7499  -8694]
[ -3648.  -1186.  -4260. ... -16396. -15953. -14558.]
[-2120  -291 -2531 ... -5906 -5854 -6211]
[-1134.  -828.  -815. ... -3899. -3559. -3538.]
```

```
#Converting the negative values to positive through abs function.

New_Applicant.DAYS_BIRTH = abs(New_Applicant.DAYS_BIRTH)
New_Applicant.DAYS_EMPLOYED = abs(New_Applicant.DAYS_EMPLOYED)
New_Applicant.DAYS_REGISTRATION = abs(New_Applicant.DAYS_REGISTRATION)
New_Applicant.DAYS_ID_PUBLISH = abs(New_Applicant.DAYS_ID_PUBLISH)
New_Applicant.DAYS_LAST_PHONE_CHANGE = abs(New_Applicant.DAYS_LAST_PHONE_CHANGE)
```



```
print(New_Applicant.CNT_FAM_MEMBERS.unique())

#Number of Family members data is in float---> not logical dtype for this data.
#converting dtype from float to int

New_Applicant.CNT_FAM_MEMBERS = New_Applicant.CNT_FAM_MEMBERS.astype(int)
```

```
[ 1.  2.  3.  4.  5.  6.  9.  7.  8. 10. 13. 14. 12. 20. 15. 16. 11.]
```

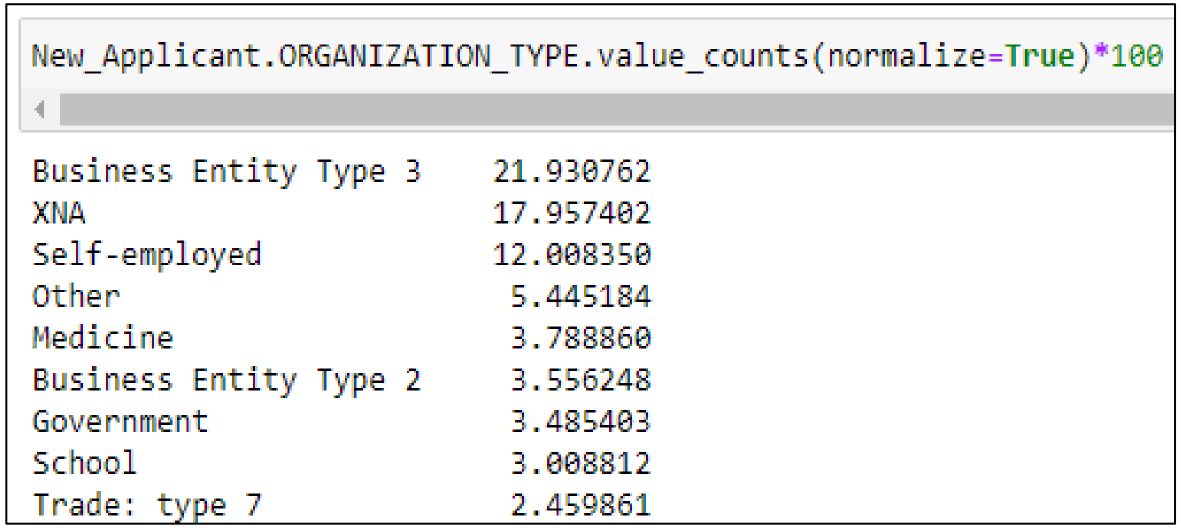
```
New_Applicant.CNT_FAM_MEMBERS.unique()
```

```
array([ 1,  2,  3,  4,  5,  6,  9,  7,  8, 10, 13, 14, 12, 20, 15, 16, 11])
```

Number people can’t be counted in decimals. Family member column datatype is changed to int from float

Similarly, for ORGANIZATION_TYPE column had ‘XNA’. Again, its categorical column, so, checked for Mode. Distribution of data is across and has mode as 'Business Entity Type 3’ of 22%.

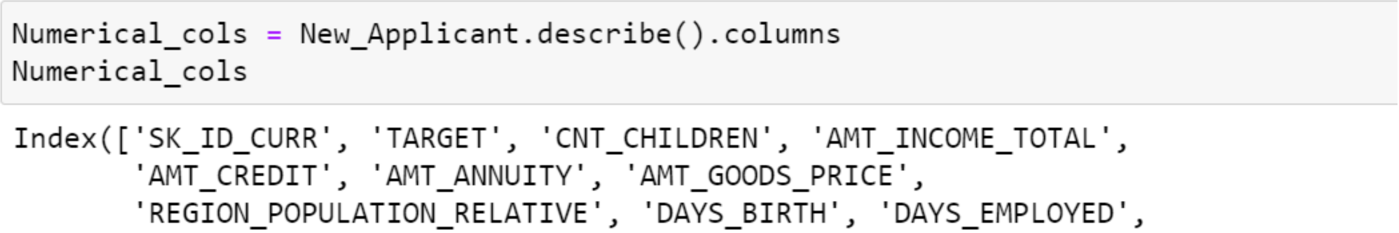
Replacing ‘XNA’ data with mode 'Business Entity Type 3' will impact the analysis as data is distributed across 58 unique categories. So, XNA data in ORGANIZATION_TYPE were dropped.



```
New_Applicant.ORGANIZATION_TYPE.value_counts(normalize=True)*100
```

```
Business Entity Type 3    21.930762
XNA                       17.957402
Self-employed             12.008350
Other                      5.445184
Medicine                   3.788860
Business Entity Type 2     3.556248
Government                 3.485403
School                     3.008812
Trade: type 7              2.459861
```

Identify Numerical columns and check for outliers in it.



```
Numerical_cols = New_Applicant.describe().columns
Numerical_cols
```

```
Index(['SK_ID_CURR', 'TARGET', 'CNT_CHILDREN', 'AMT_INCOME_TOTAL',
       'AMT_CREDIT', 'AMT_ANNUITY', 'AMT_GOODS_PRICE',
       'REGION_POPULATION_RELATIVE', 'DAYS_BIRTH', 'DAYS_EMPLOYED',
```

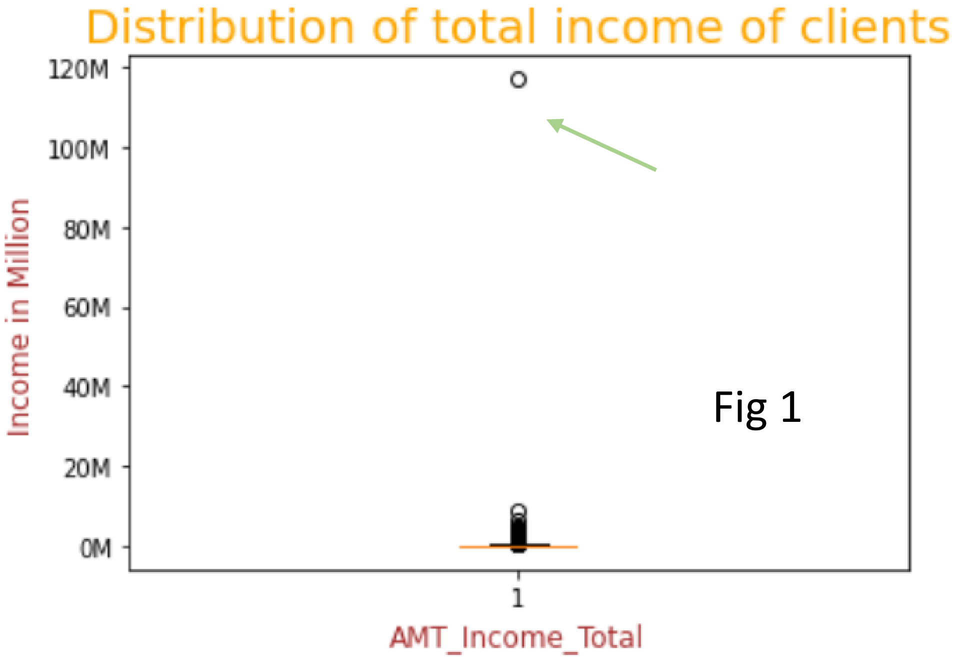


Fig 1: In the above boxplot of AMT_Income_Total, we can observe outliers- Max income 117000000 (near to 120M) is surely an outlier here.

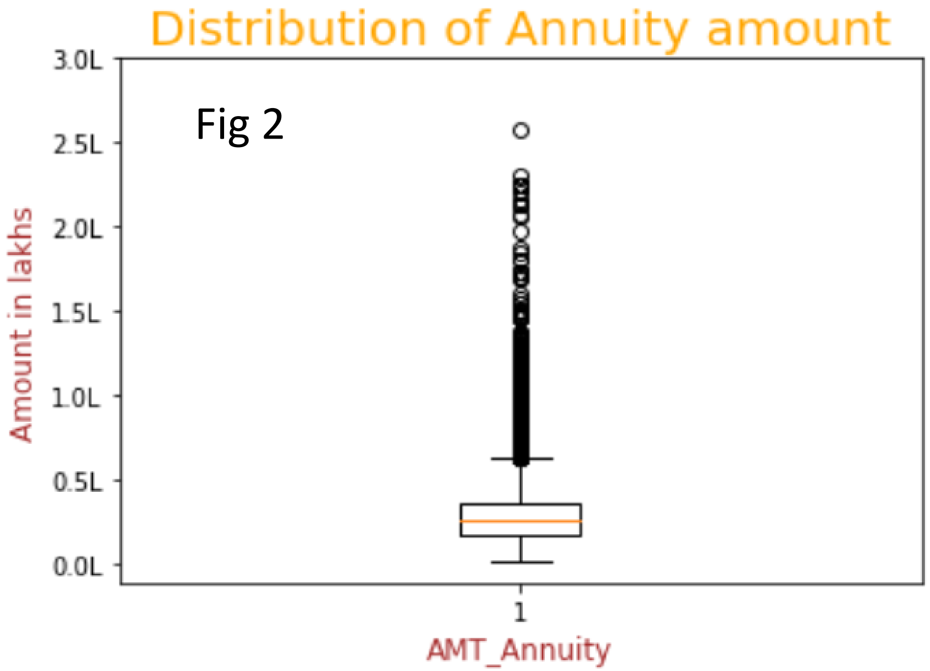
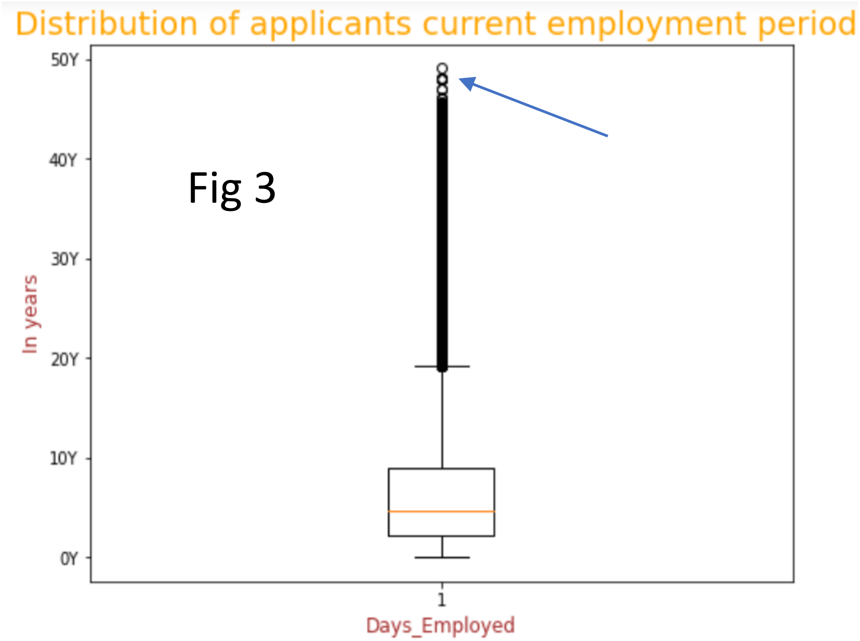


Fig 2: In the above boxplot of AMT_Annuity, we can observe some outliers. Max loan annuity at 2.5L appears to be an outlier here.



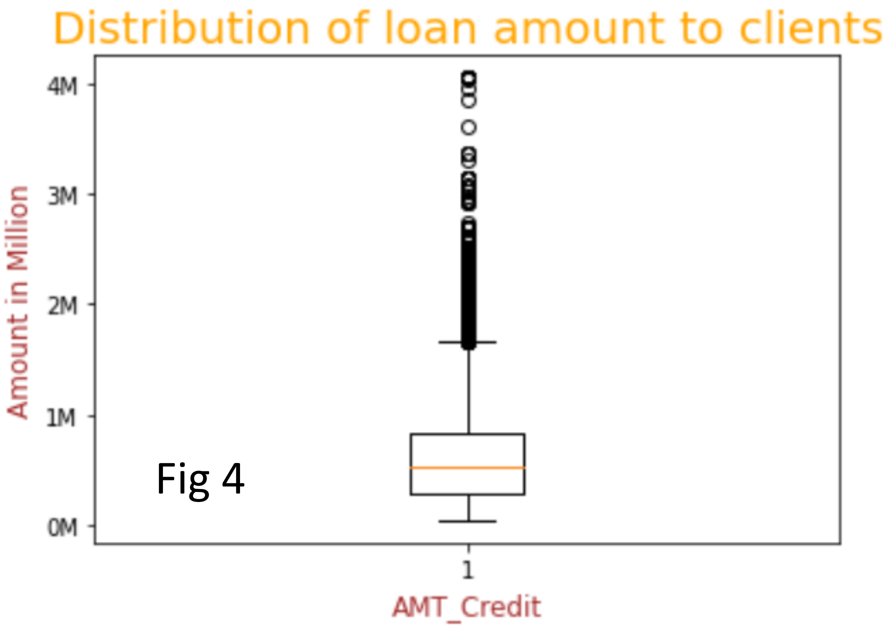


Fig 3: In the above boxplot of DAYS_EMPLOYED, there is values near to 50 years is surely outlier. As per DAYS_BIRTH data, applicants Max age here is 69 years. Considering that, lets say applicant started working early in 21 years to which if we add the outlier values (ex: 21+49) that gives applicant age as 70-- serving in the same office where he/she started without retirement?!..

Fig 4: In the above boxplot of AMT_Credit, we can observe some outliers. Considering the income of the applicant loan credit can vary and can't deem Max loan credited (4 Million) as outlier.

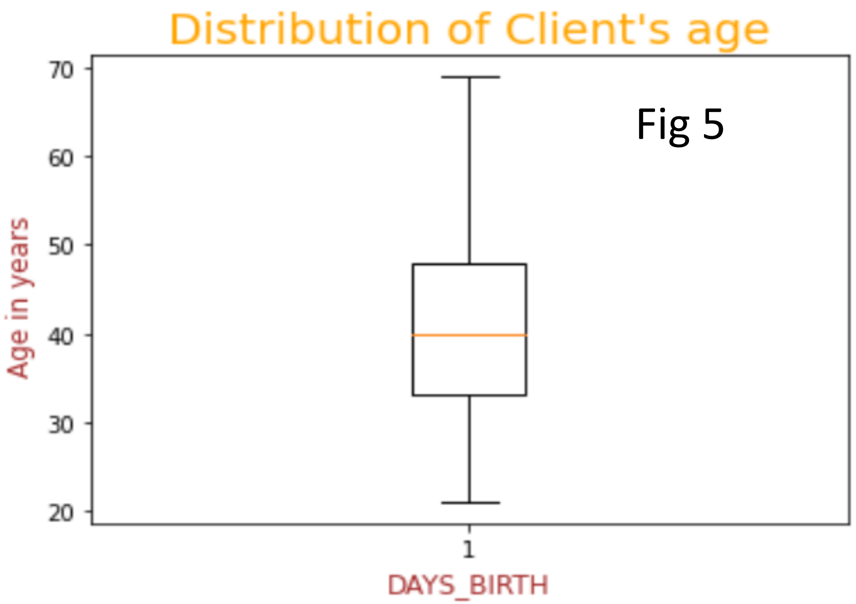
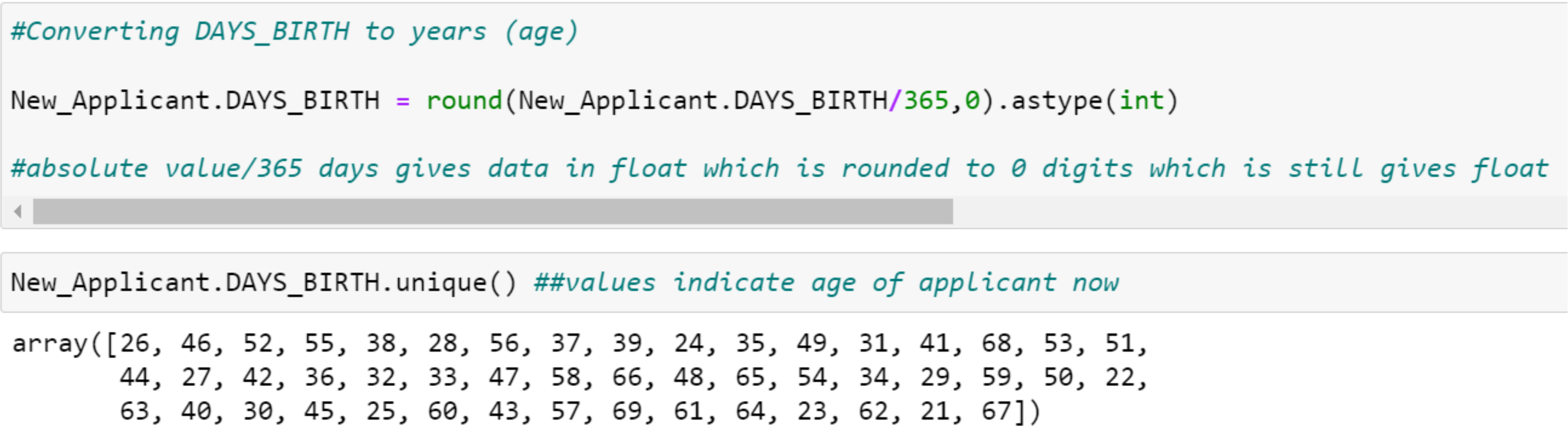


Fig 5

Fig 5: In the above boxplot of DAY_BIRTH, there are no outliers w.r.t age.

Next, creating bins for Continuous variables/columns: AMT_INCOME_TOTAL, AMT_CREDIT and DAYS_BIRTH for further analysis.

‘DAYS_BIRTH’ column had data in number of days, which were converted to years (age of the applicants) here

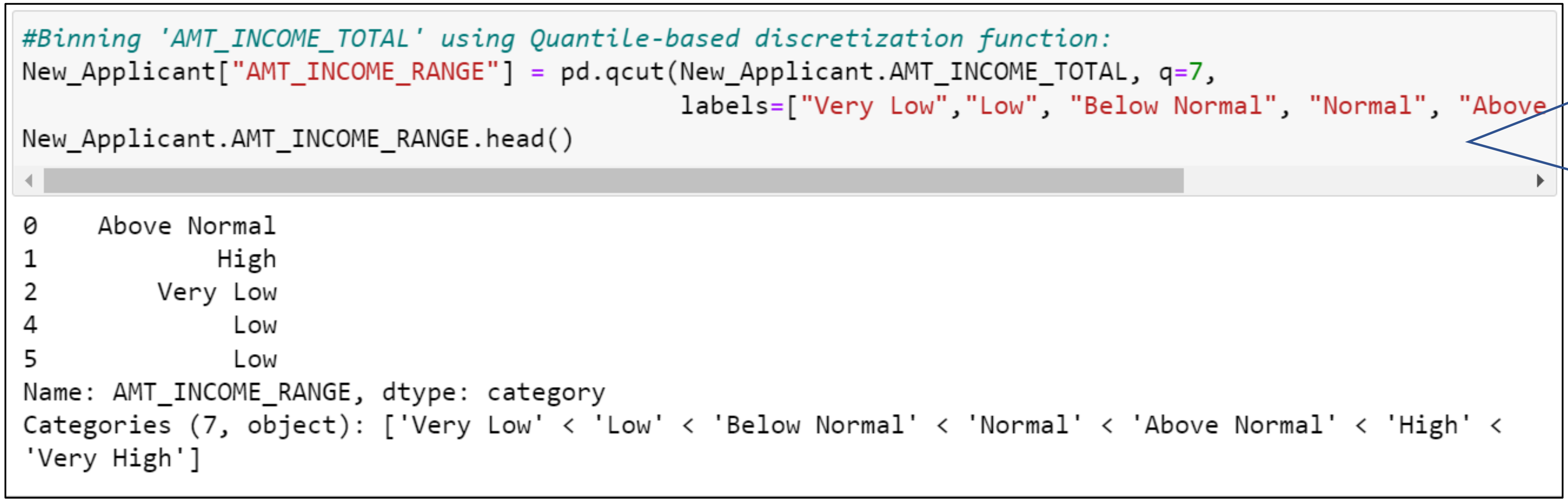


```
#Converting DAYS_BIRTH to years (age)

New_Applicant.DAYS_BIRTH = round(New_Applicant.DAYS_BIRTH/365,0).astype(int)

#absolute value/365 days gives data in float which is rounded to 0 digits which is still gives float
```

```
New_Applicant.DAYS_BIRTH.unique() ##values indicate age of applicant now
```

```
array([26, 46, 52, 55, 38, 28, 56, 37, 39, 24, 35, 49, 31, 41, 68, 53, 51,
       44, 27, 42, 36, 32, 33, 47, 58, 66, 48, 65, 54, 34, 29, 59, 50, 22,
       63, 40, 30, 45, 25, 60, 43, 57, 69, 61, 64, 23, 62, 21, 67])
```

Binning of Continuous variables

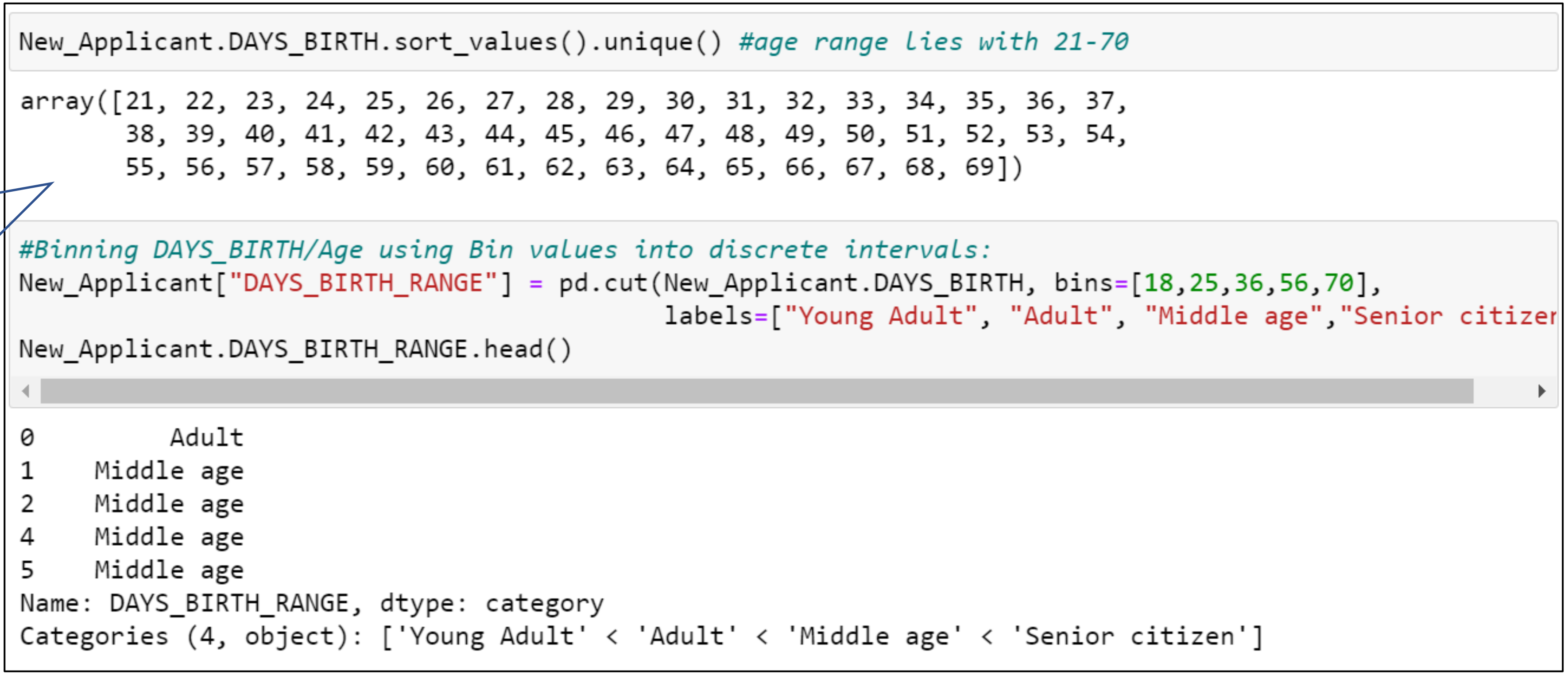


```
#Binning 'AMT_INCOME_TOTAL' using Quantile-based discretization function:
New_Applicant["AMT_INCOME_RANGE"] = pd.qcut(New_Applicant.AMT_INCOME_TOTAL, q=7,
                                    labels=["Very Low","Low", "Below Normal", "Normal", "Above
New_Applicant.AMT_INCOME_RANGE.head()
```

```
0    Above Normal
1            High
2        Very Low
4             Low
5             Low
Name: AMT_INCOME_RANGE, dtype: category
Categories (7, object): ['Very Low' < 'Low' < 'Below Normal' < 'Normal' < 'Above Normal' < 'High' <
'Very High']
```

'AMT_INCOME_TOTAL' and 'AMT_CREDIT' numerical data were converted to categorical data based on range buckets by creating new col 'AMT_INCOME_RANGE' and 'AMT_CREDIT_RANGE'.

Similarly, applicant 'Age' numerical data were converted to categorical data based on range buckets by creating new column 'DAYS_BIRTH_RANGE'



```
New_Applicant.DAYS_BIRTH.sort_values().unique() #age range lies with 21-70
```

```
array([21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37,
       38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54,
       55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69])
```

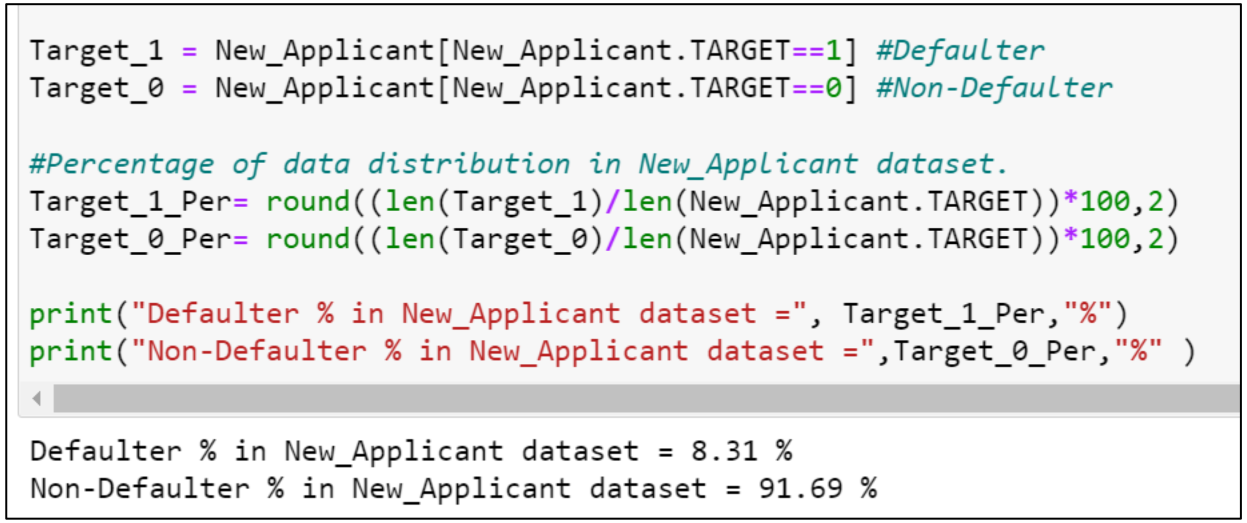
```
#Binning DAYS_BIRTH/Age using Bin values into discrete intervals:
New_Applicant["DAYS_BIRTH_RANGE"] = pd.cut(New_Applicant.DAYS_BIRTH, bins=[18,25,36,56,70],
                                           labels=["Young Adult", "Adult", "Middle age","Senior citizer
New_Applicant.DAYS_BIRTH_RANGE.head()
```

```
0          Adult
1     Middle age
2     Middle age
4     Middle age
5     Middle age
Name: DAYS_BIRTH_RANGE, dtype: category
Categories (4, object): ['Young Adult' < 'Adult' < 'Middle age' < 'Senior citizen']
```

Imbalance Percentage between 1(Defaulter) and 0 (Non-Defaulter), in Target variable:

‘New_Applicant’ dataset is divided into 2 set based on values in Target column:

- Target_1(Defaulter)
- Target_0(Non-Defaulter)



```
Target_1 = New_Applicant[New_Applicant.TARGET==1] #Defaulter
Target_0 = New_Applicant[New_Applicant.TARGET==0] #Non-Defaulter

#Percentage of data distribution in New_Applicant dataset.
Target_1_Per= round((len(Target_1)/len(New_Applicant.TARGET))*100,2)
Target_0_Per= round((len(Target_0)/len(New_Applicant.TARGET))*100,2)

print("Defaulter % in New_Applicant dataset =", Target_1_Per,"%")
print("Non-Defaulter % in New_Applicant dataset =",Target_0_Per,"%" )
```

```
Defaulter % in New_Applicant dataset = 8.31 %
Non-Defaulter % in New_Applicant dataset = 91.69 %
```

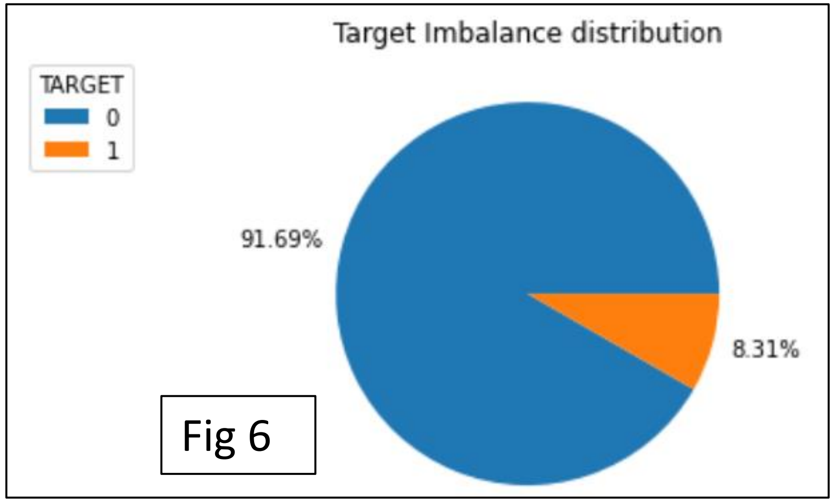
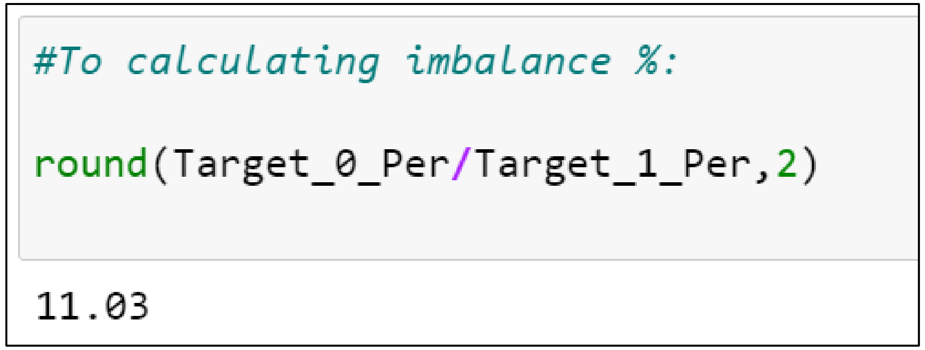


Fig 6



```
#To calculating imbalance %:

round(Target_0_Per/Target_1_Per,2)
```

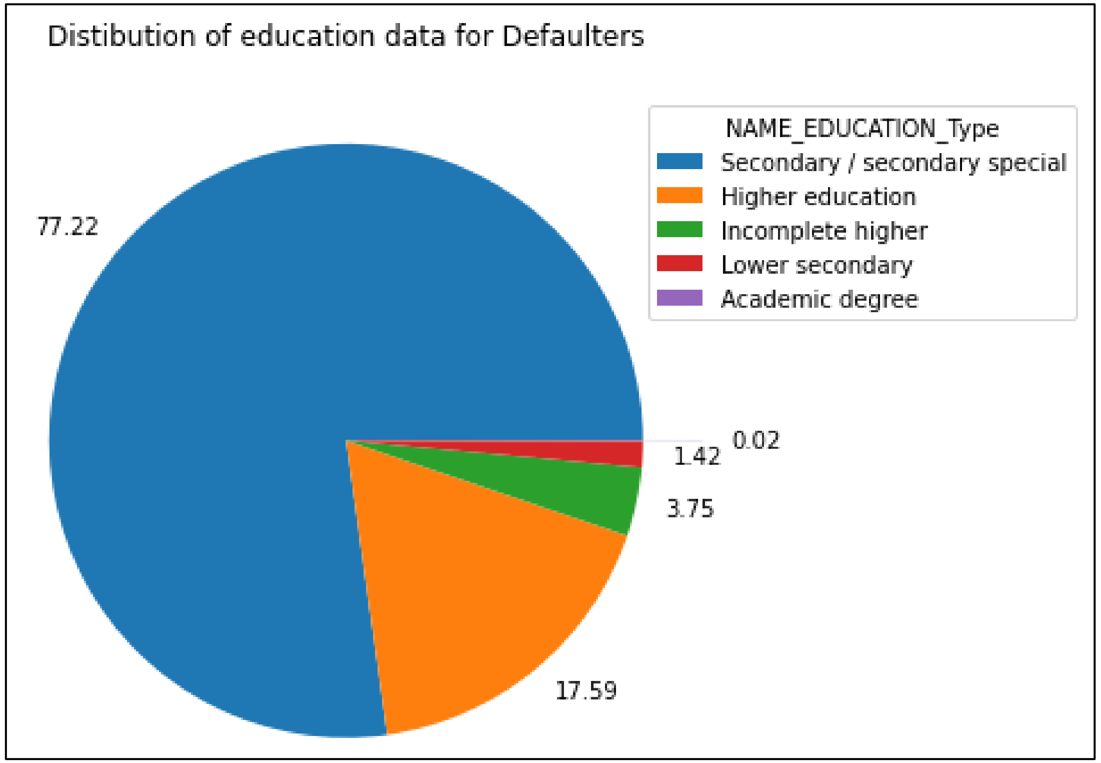
```
11.03
```

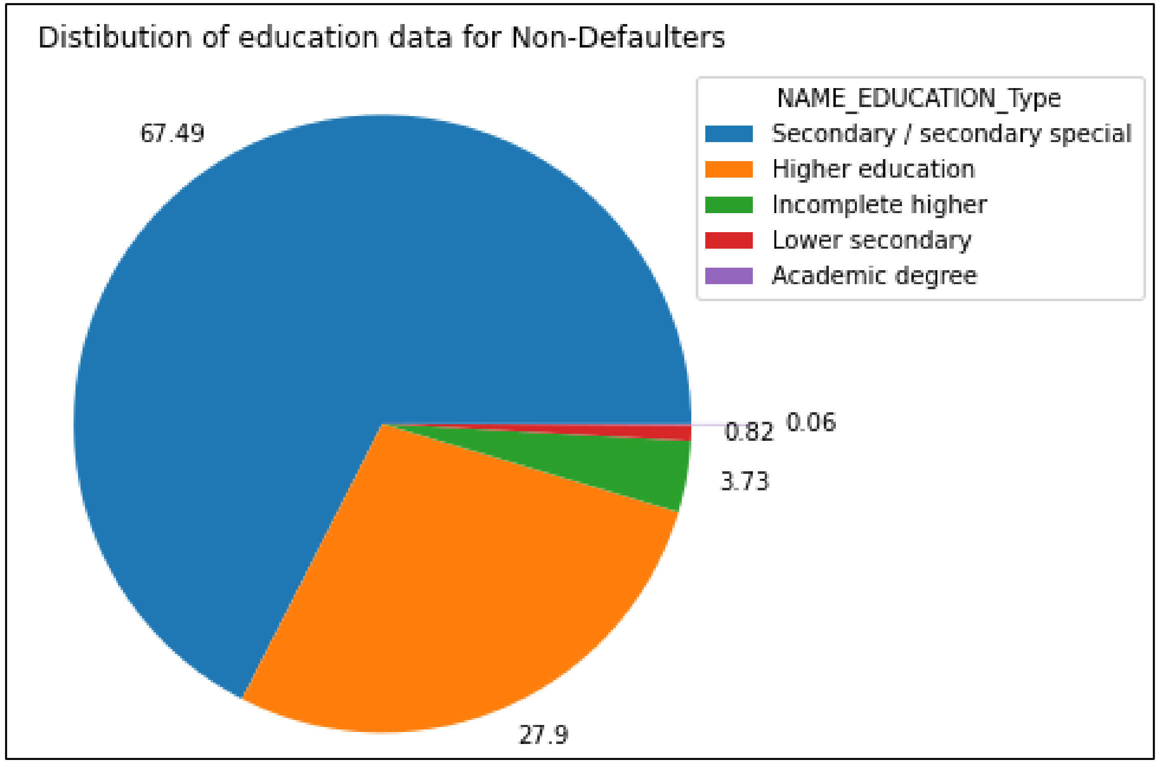
Fig 6: We can observe majority of data is seen in Target_0 (Non-Defaulter) dataset. Overall, 8.31% clients are defaulters in the given dataset.

Imbalance ratio is 11.03. There is an imbalance in the target variable where 8.31% of the clients are found to be defaulter and 91.69% are found to be non-defaulter.

Univariant analysis for different categorial data

EDUCATION TYPE:

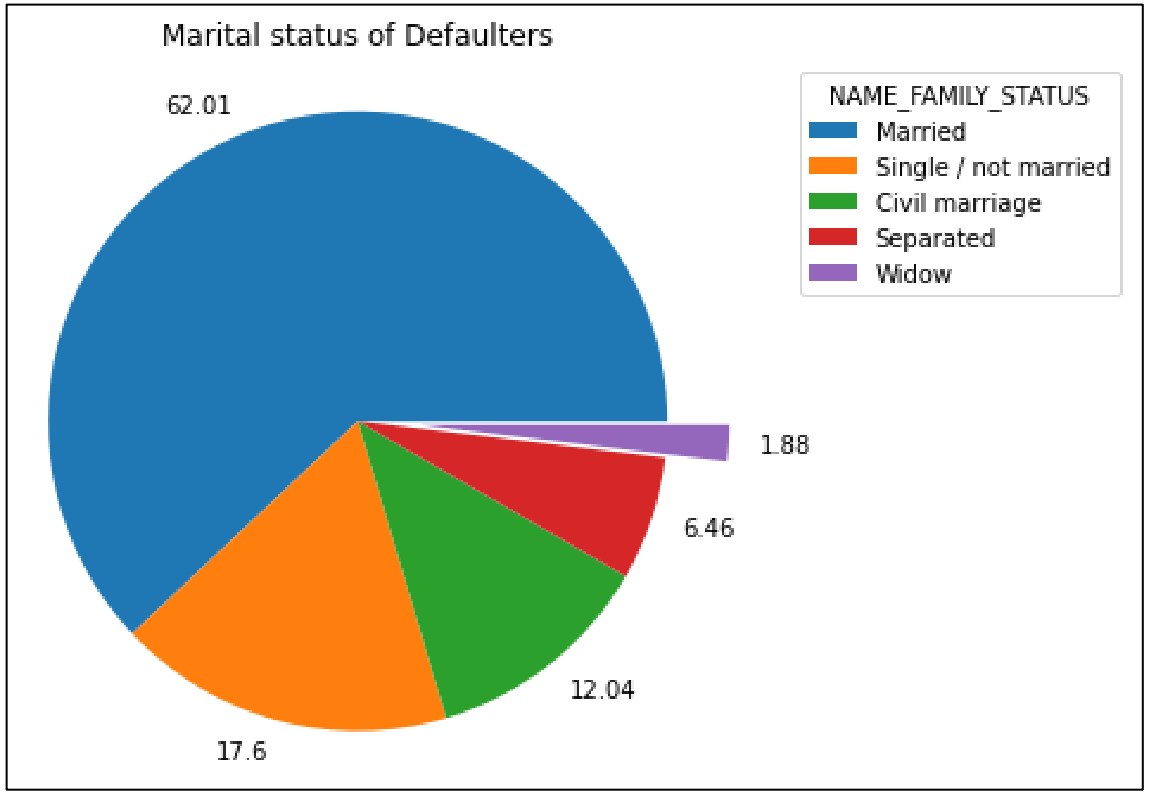


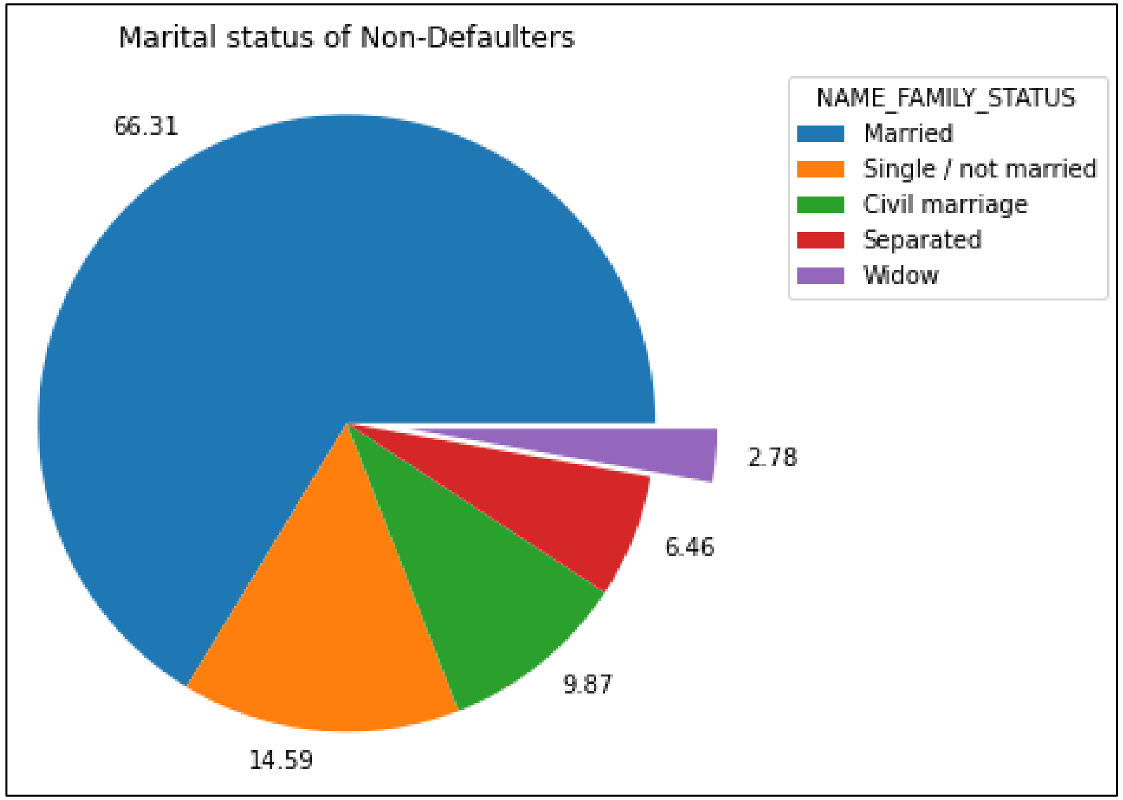


By comparing % of Defaulter and Non-Defaulter education qualification, We can observe :

- •Increase in percentage of Defaulters who has "Secondary/Secondary special" education.
- •Decrease in percentage of Defaulters who has "Higher Education".

NAME_FAMILY_STATUS (Marital Status)

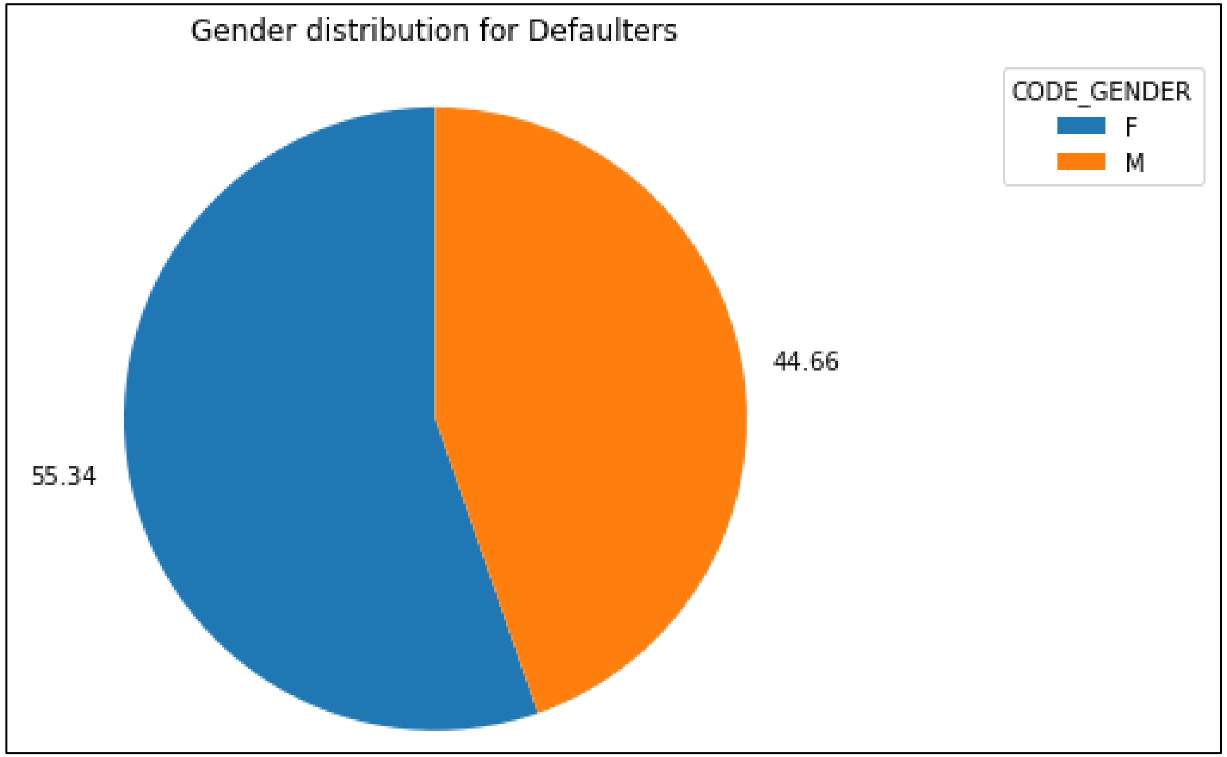


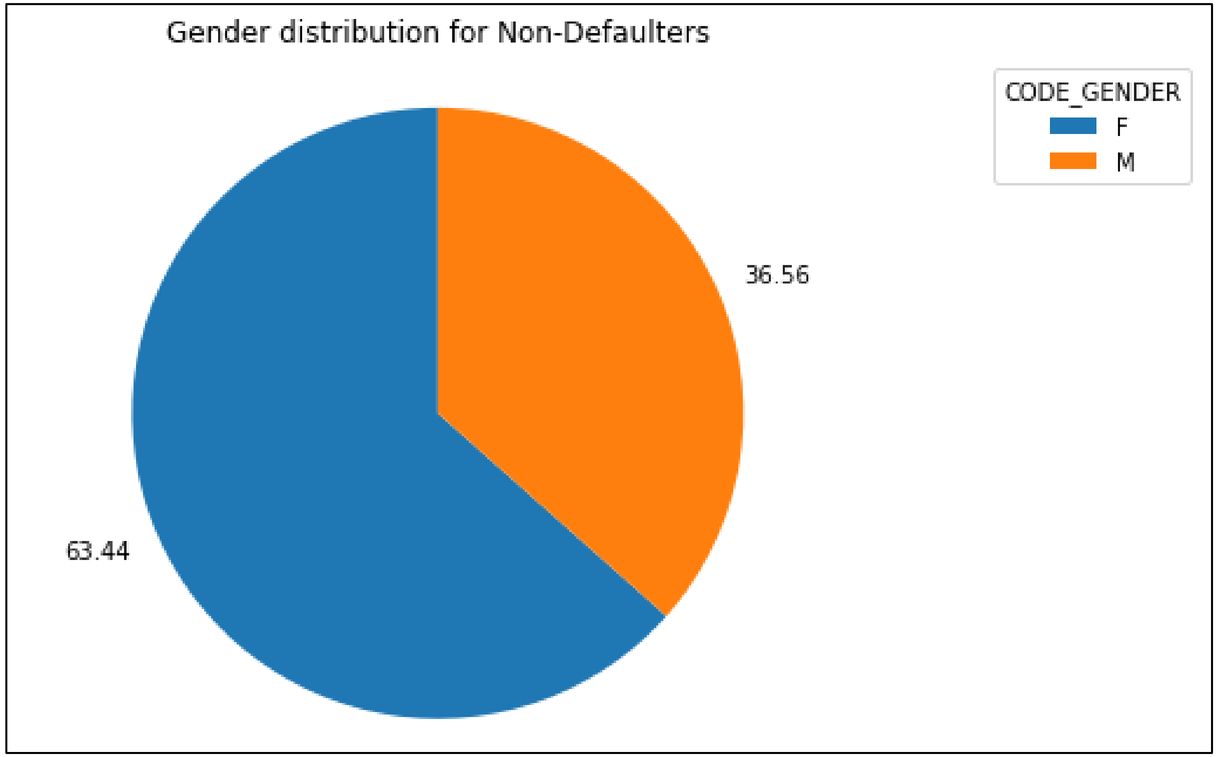


By comparing % of Defaulter and Non-Defaulter marital status, We can observe :

- Increase in percentage of Defaulters who are "Single" and 'Civil Married’.
- Decrease in percentage of Defaulters who are "Married" and 'Widow’.

GENDER:

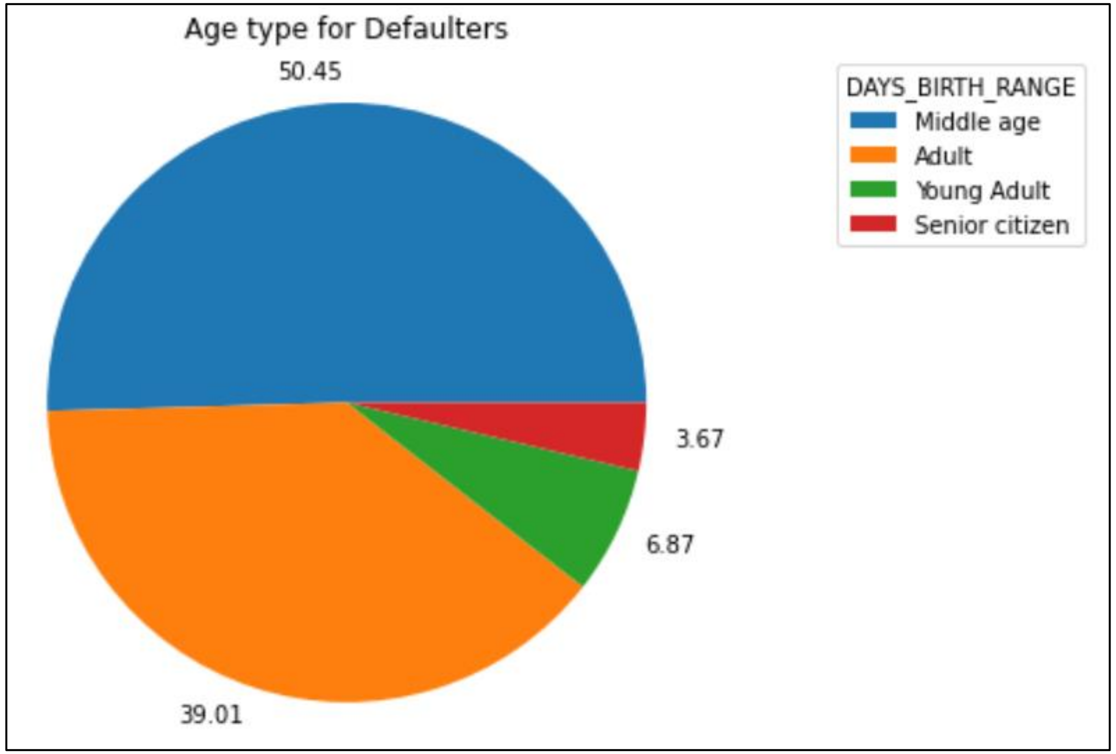


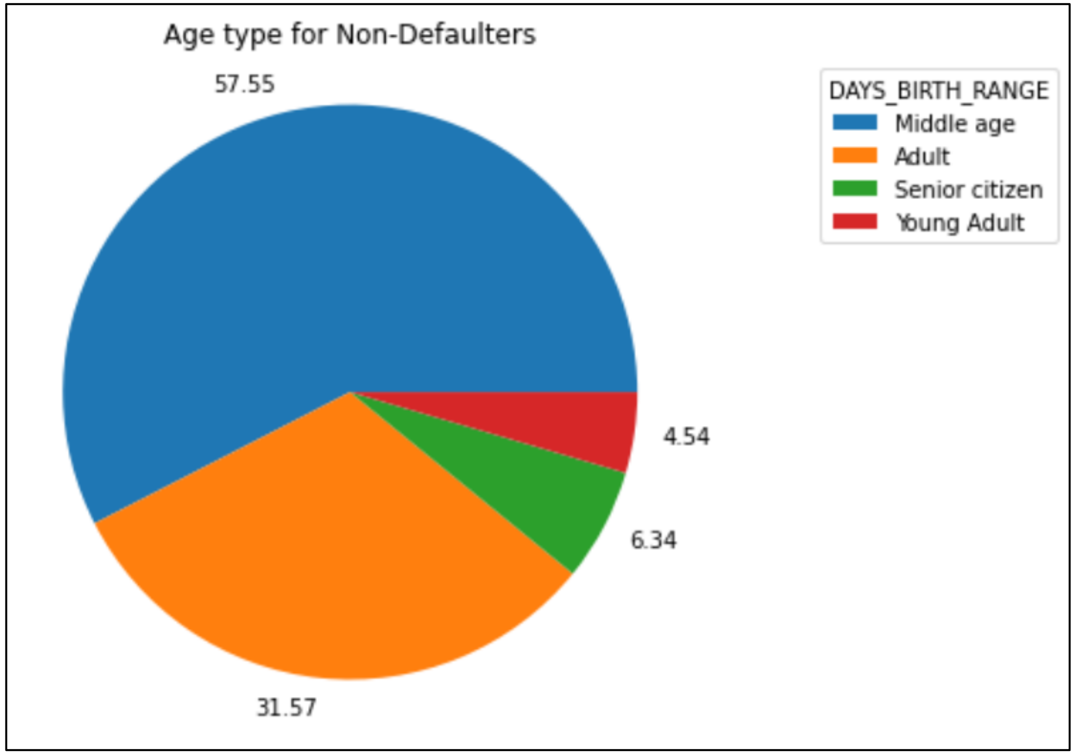


In both cases we can observe female are majority in both cases.
By comparing % of Defaulter and Non-Defaulter gender, we can observe:

- Increase in percentage of Male Defaulters.
- Decrease in percentage of Female Defaulters.

AGE:

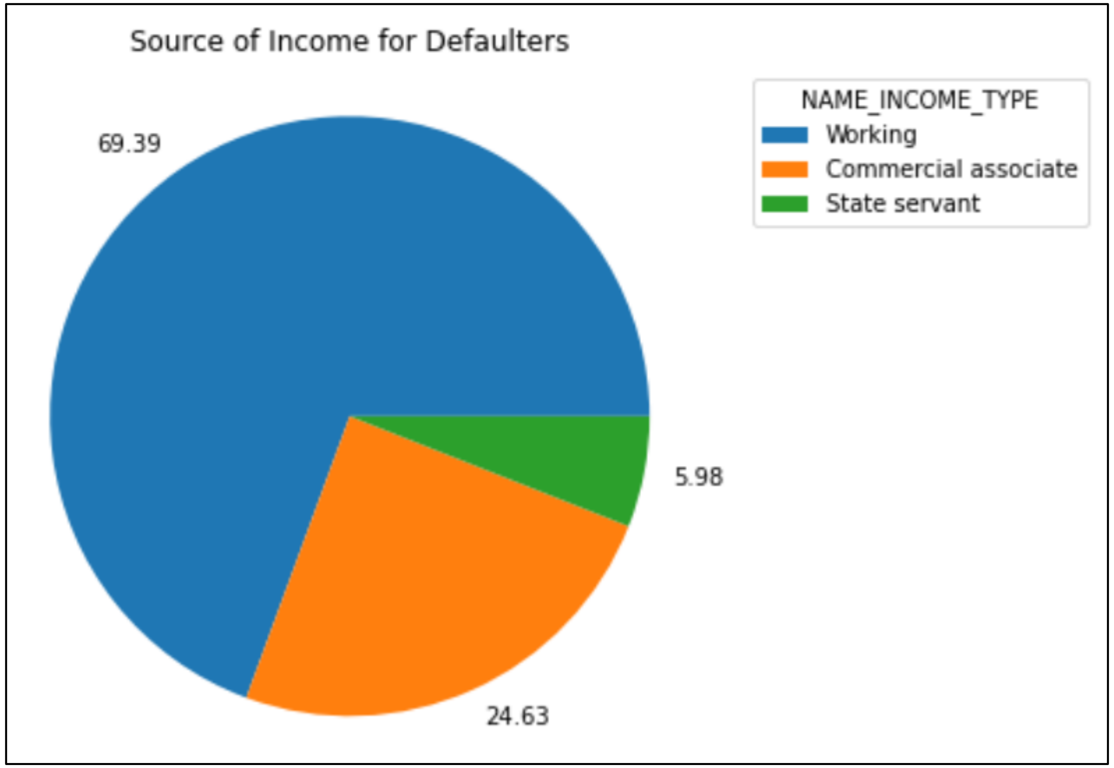


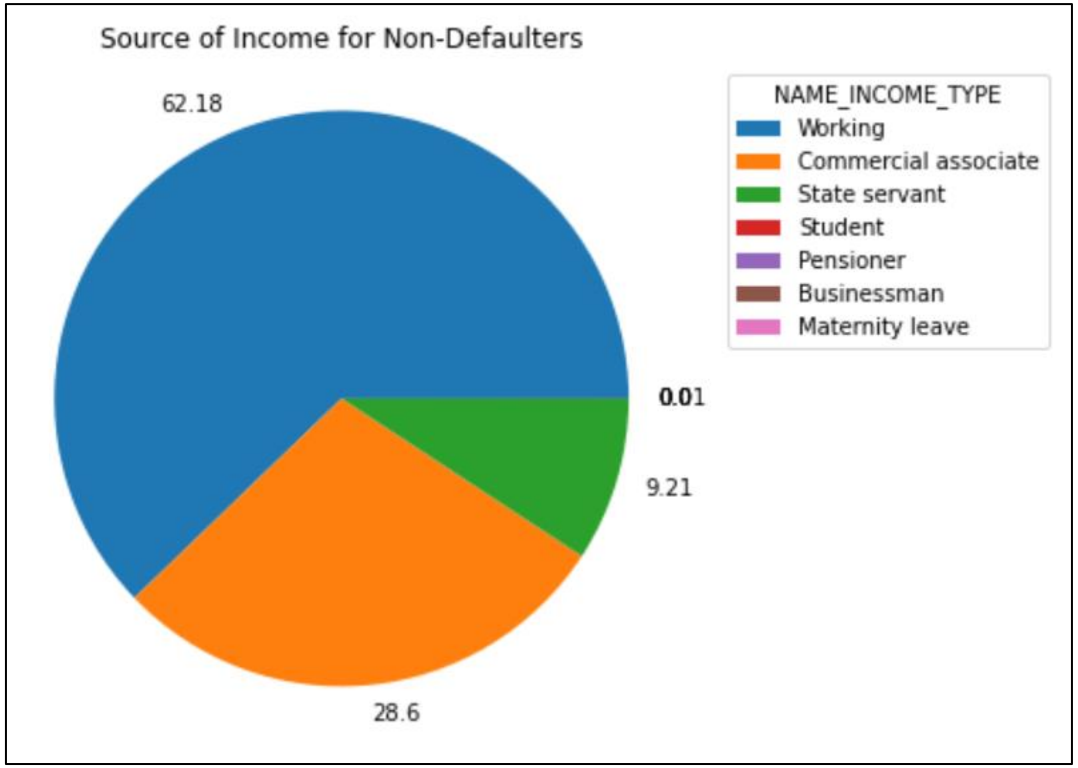


By comparing % of Defaulter and Non-Defaulter age group, We can observe :

- Increase % of Defaulters who are Young adults and Adults.
- Decrease % of Defaulters who are Senior citizen and Middle Aged.

Source of Income:

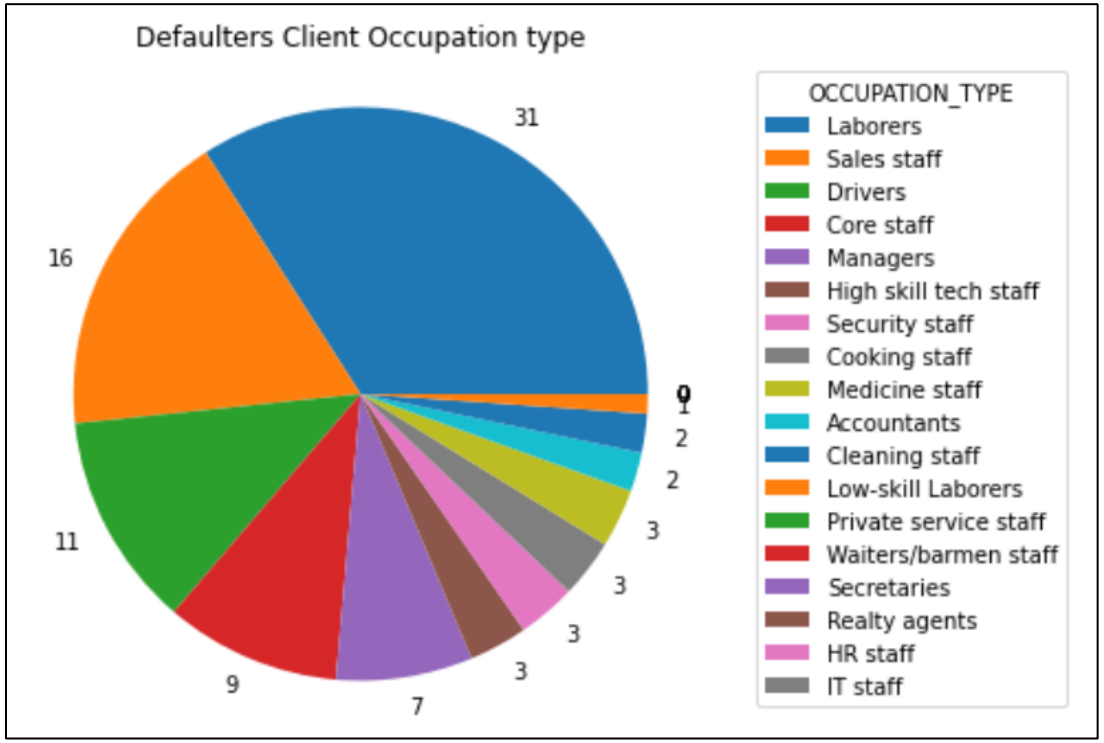


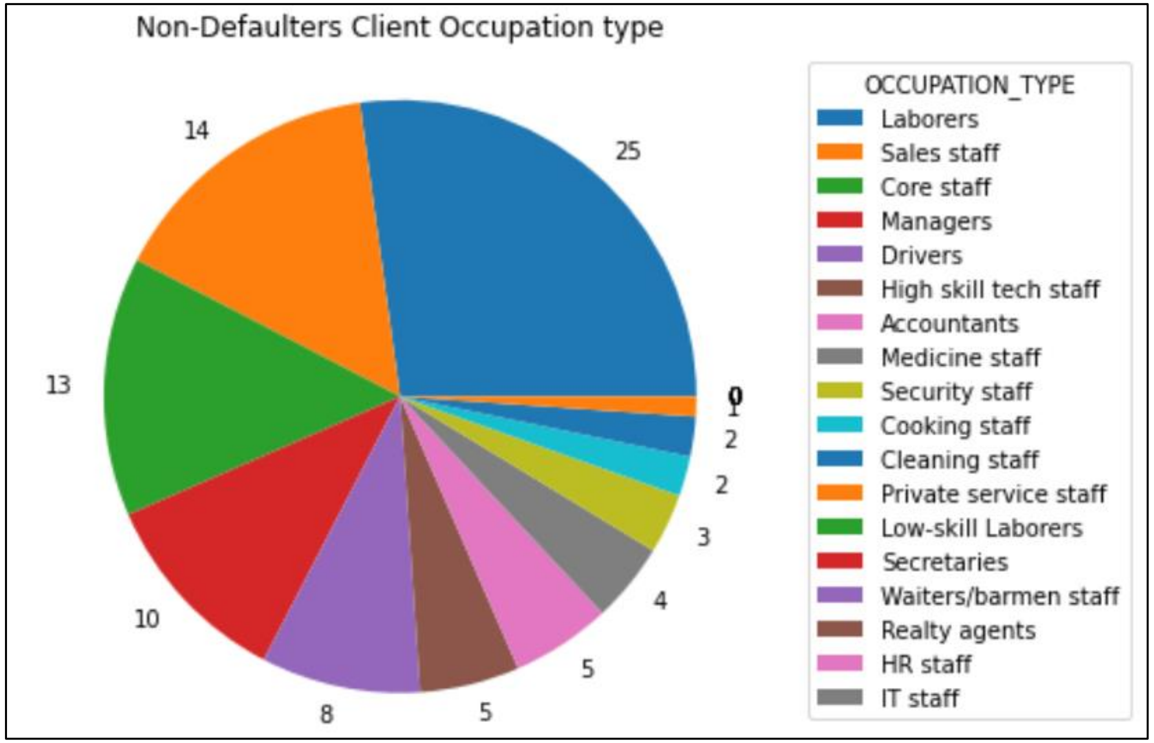


We can observe defaulters Income type mainly falls under Working, Commercial associate & State servants category. In comparison with defaulter and non-defaulter % graph, we can say that:

- Increase in percentage of defaulters who are 'working'.
- Decrease in percentage of defaulters who are 'Commercial associate' & 'State servants'.

Occupation type:





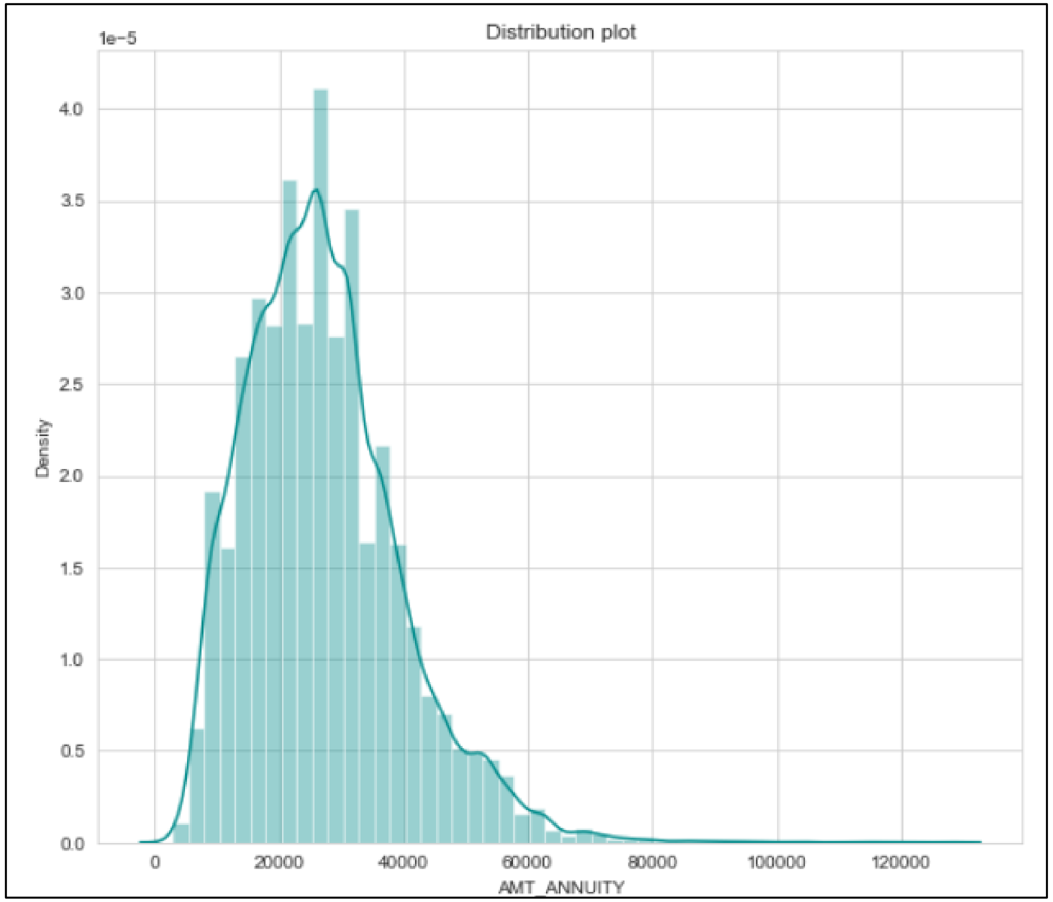
In comparison with defaulter and non-defaulter % graph, We can observe:

- Increase in defaulter % who are 'Laborers', 'Sales staff', 'Drivers’
- Decrease in defaulter % who are 'Core staff’.

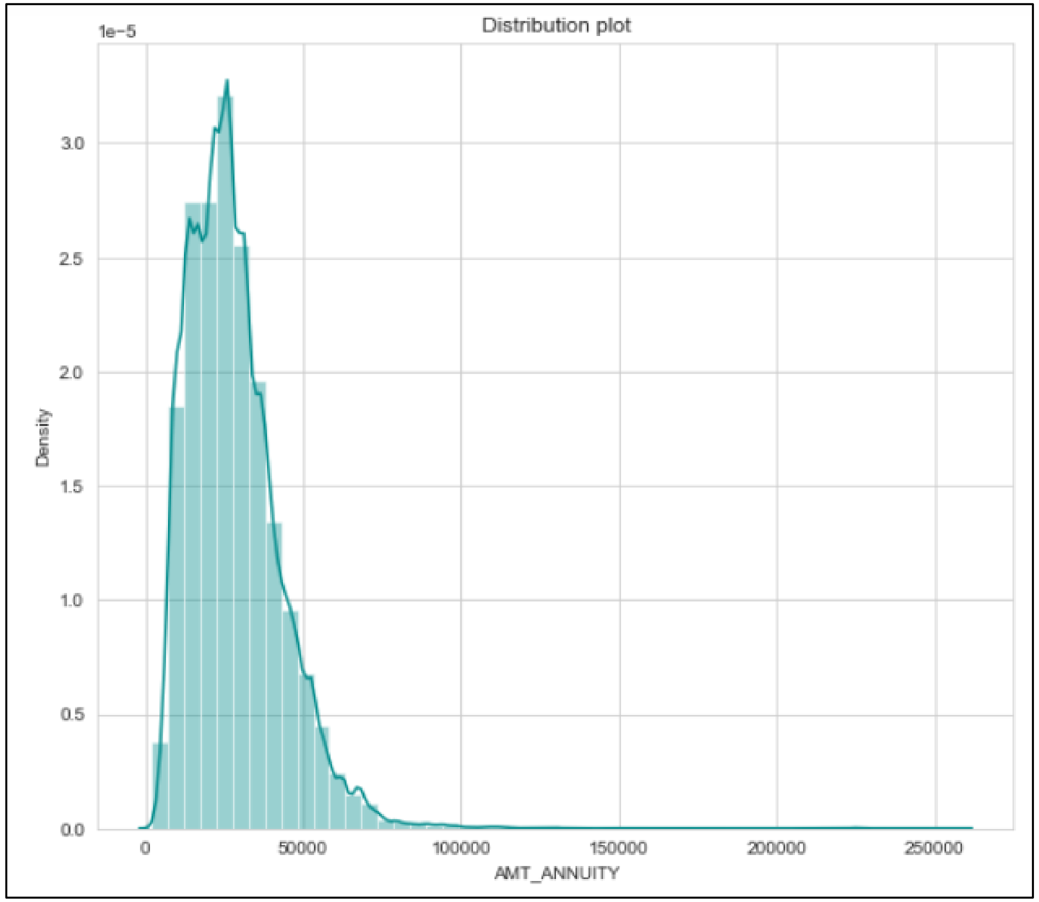
Univariant analysis for different Numerical data

Loan annuity

Defaulters

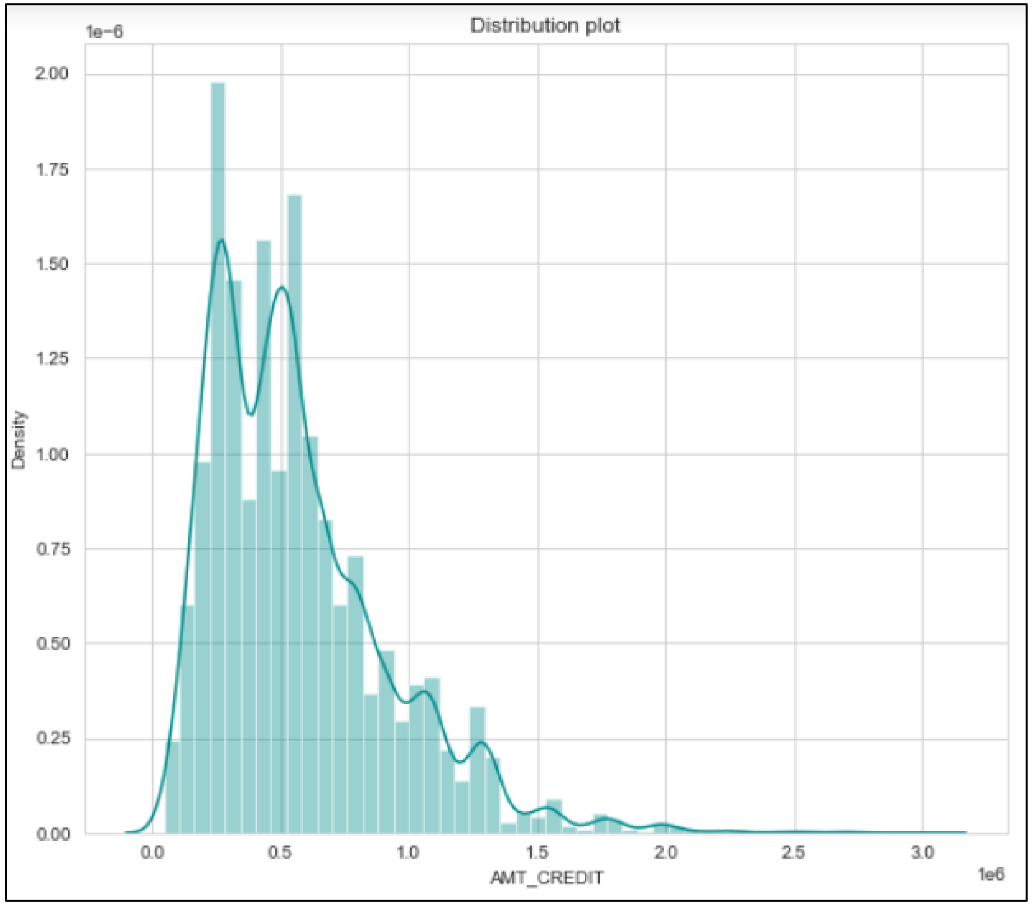


Non-Defaulters

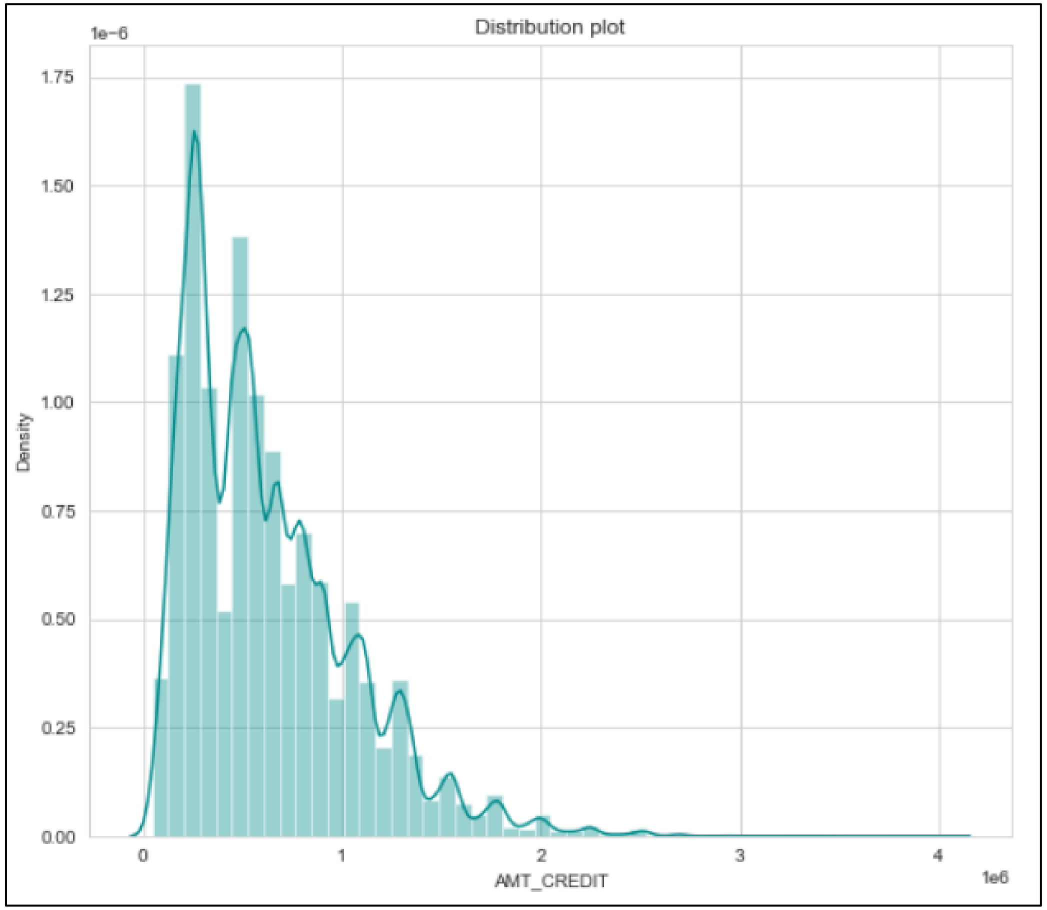


Loan Credit:

Defaulters

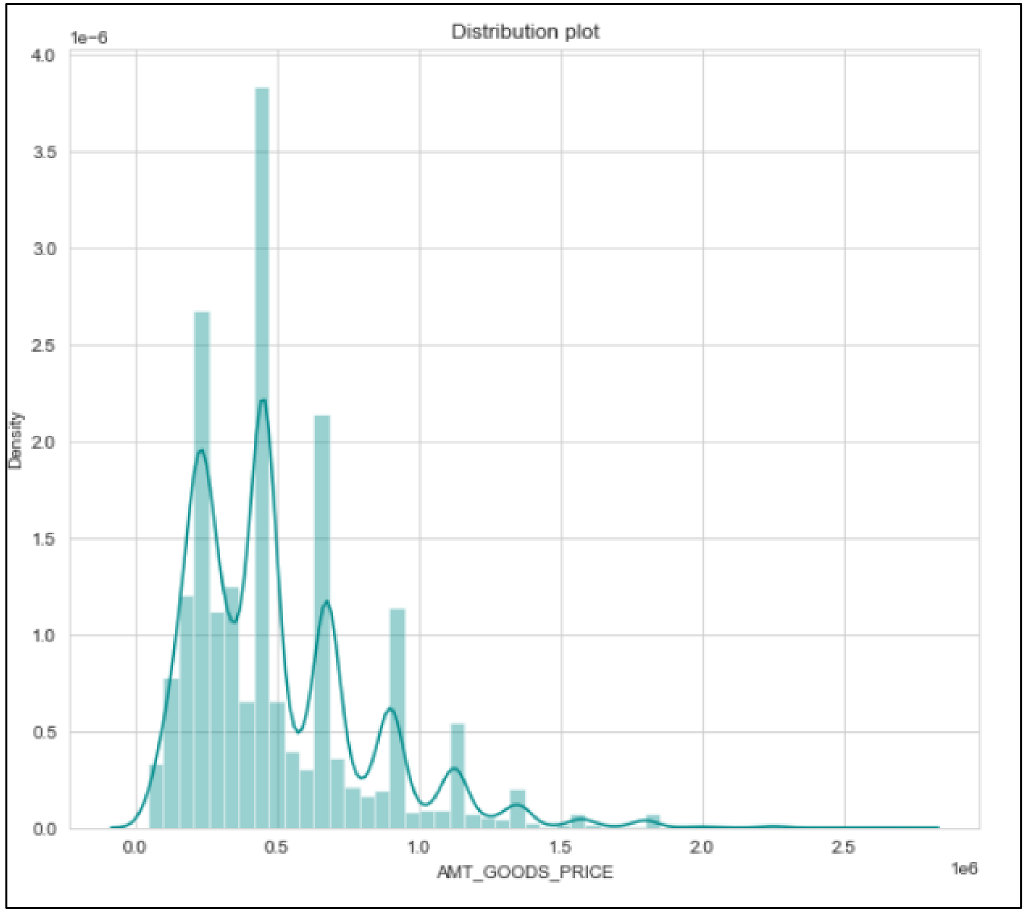


Non-Defaulters

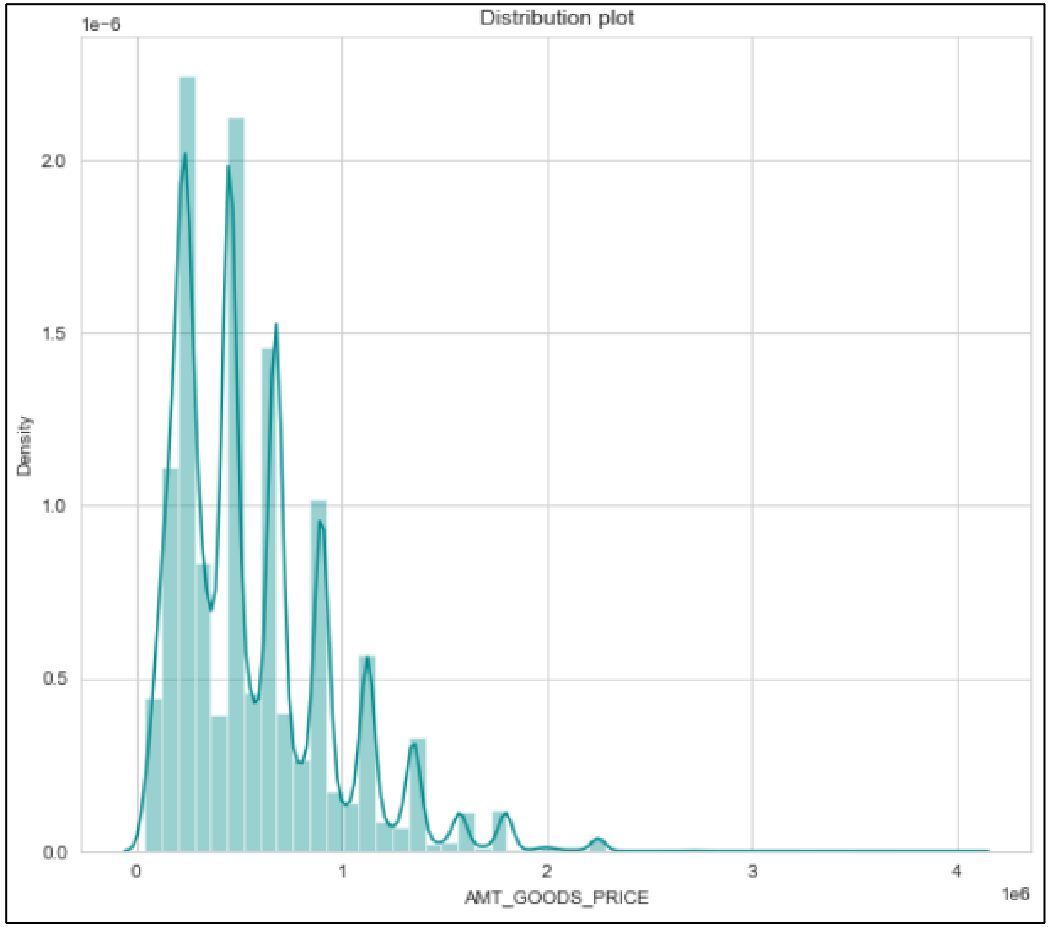


Goods Price

Defaulters

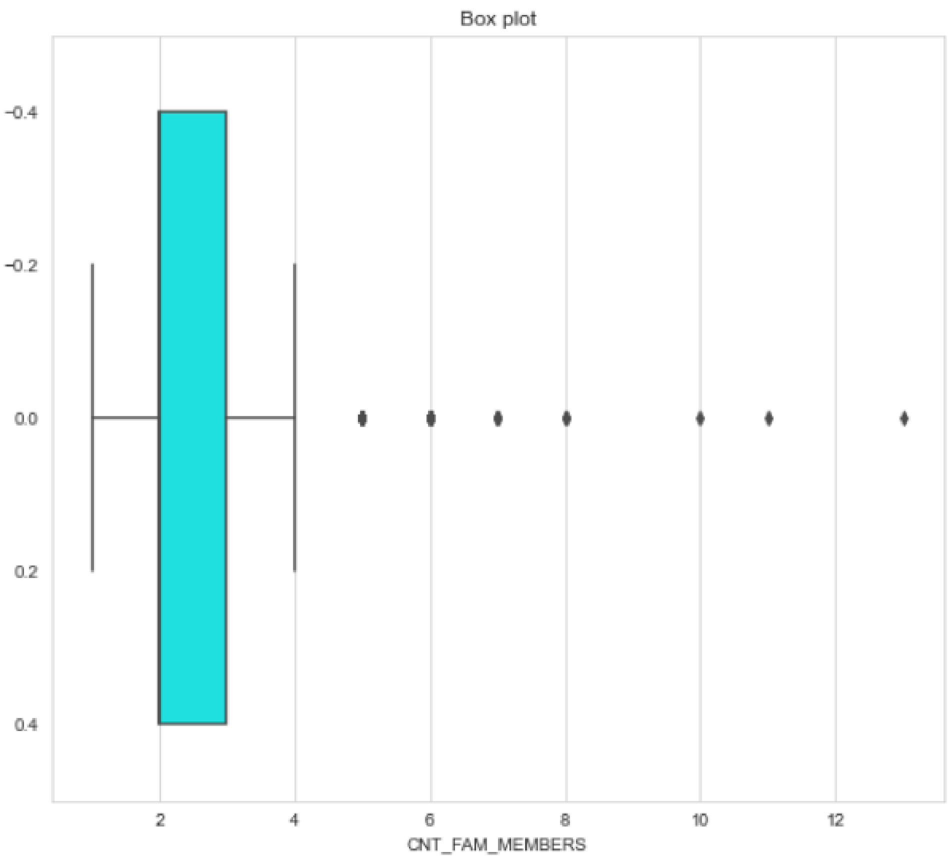


Non-Defaulters

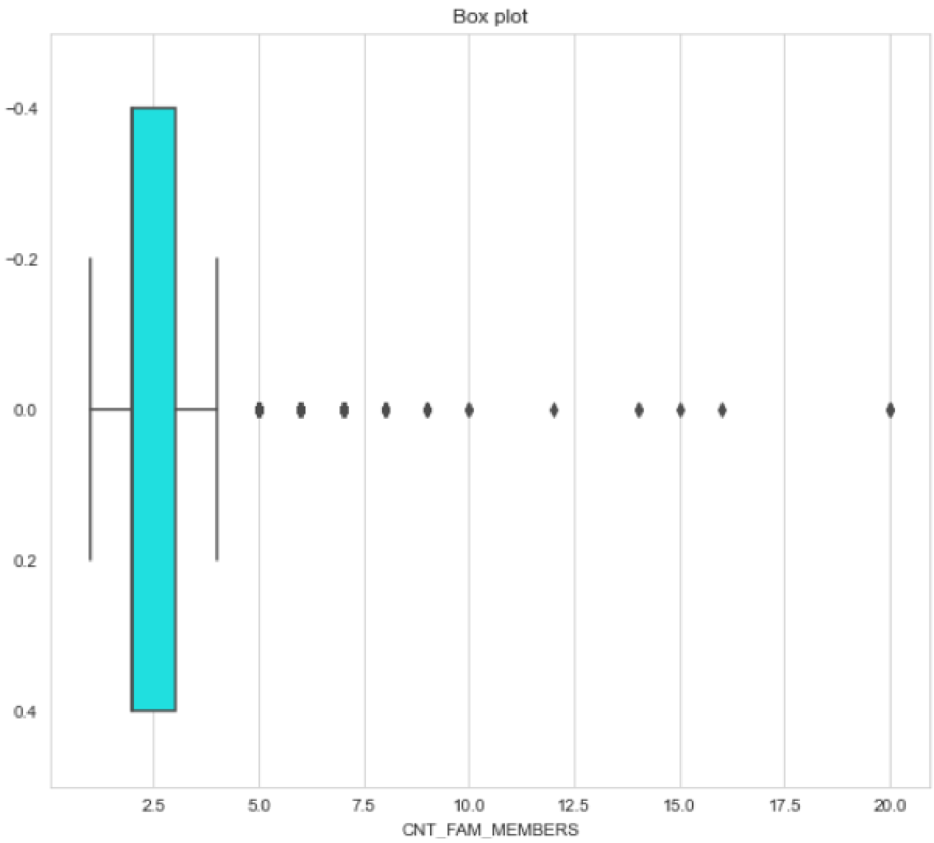


Family members

Defaulters

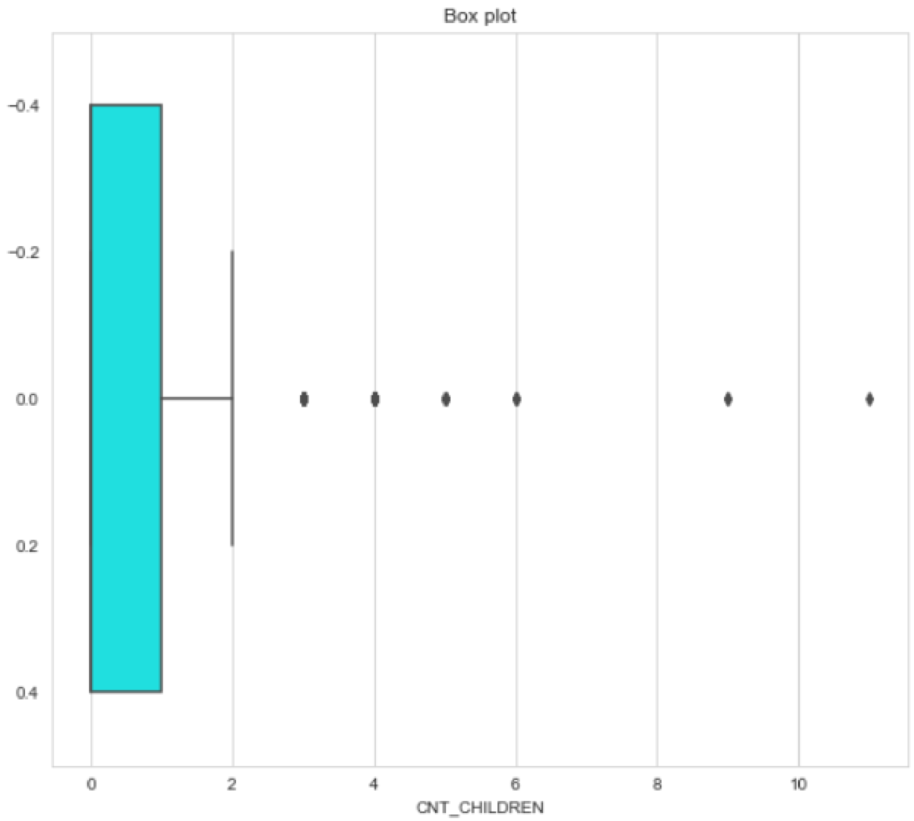


Non-Defaulters

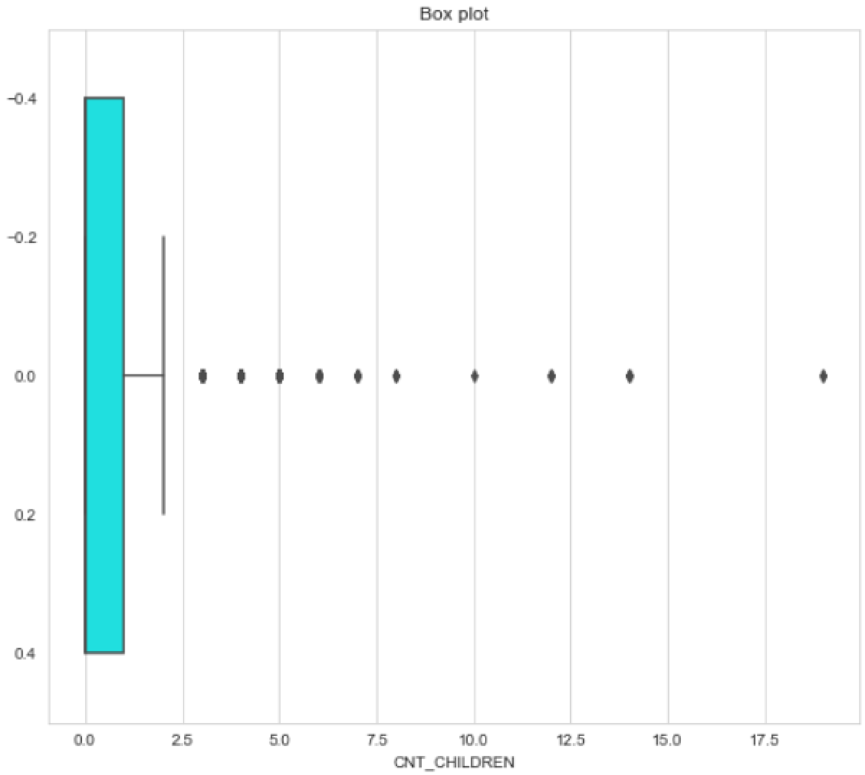


Number of children

Defaulters

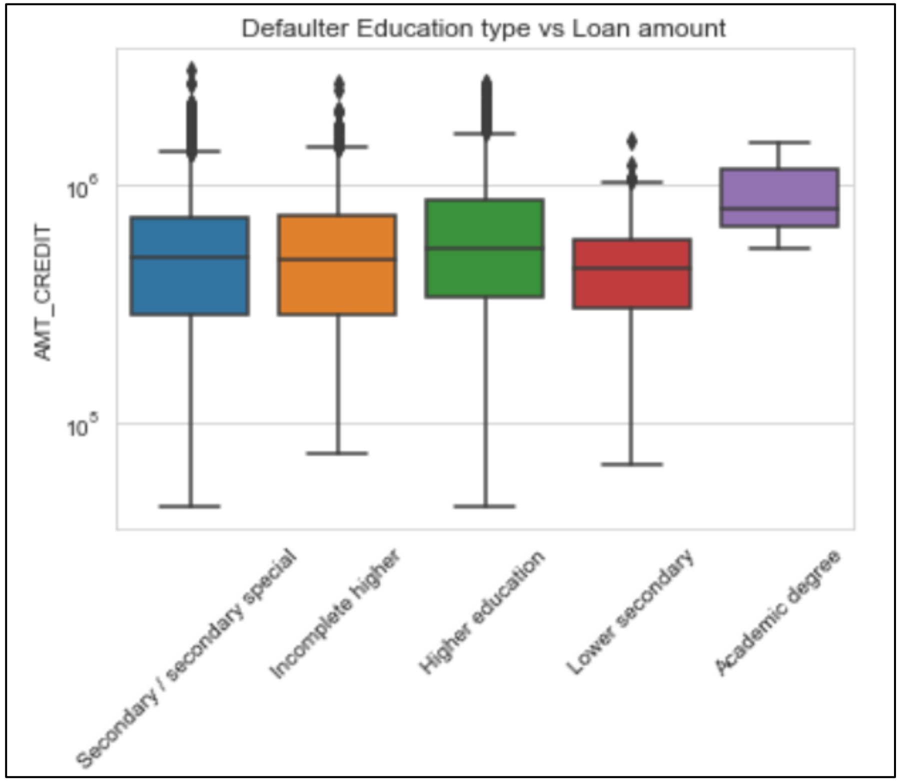


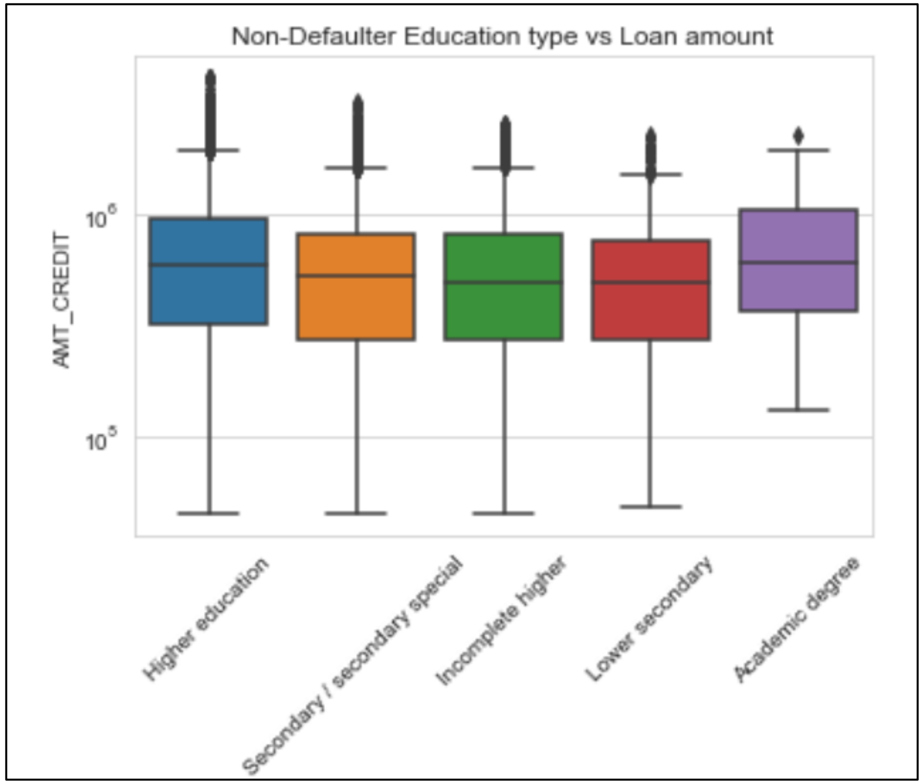
Non-Defaulters



Bivariate analysis
for categorical variables vs numerical variables

Education type vs Loan amount

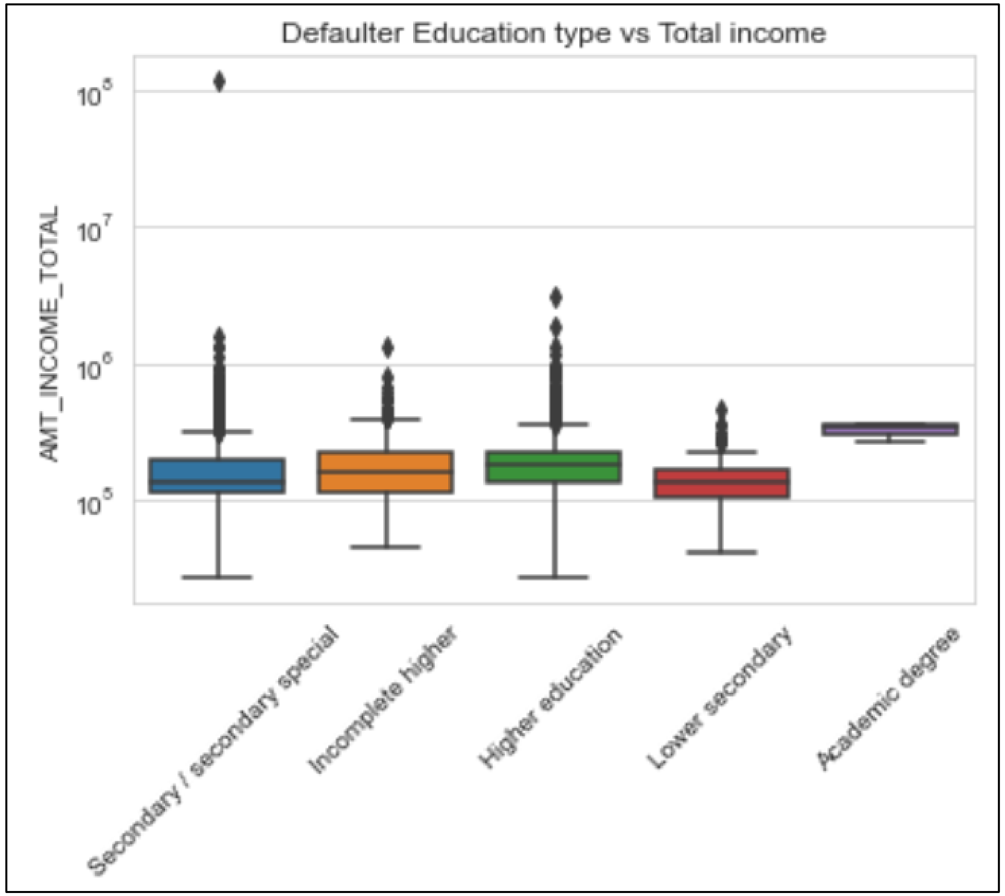


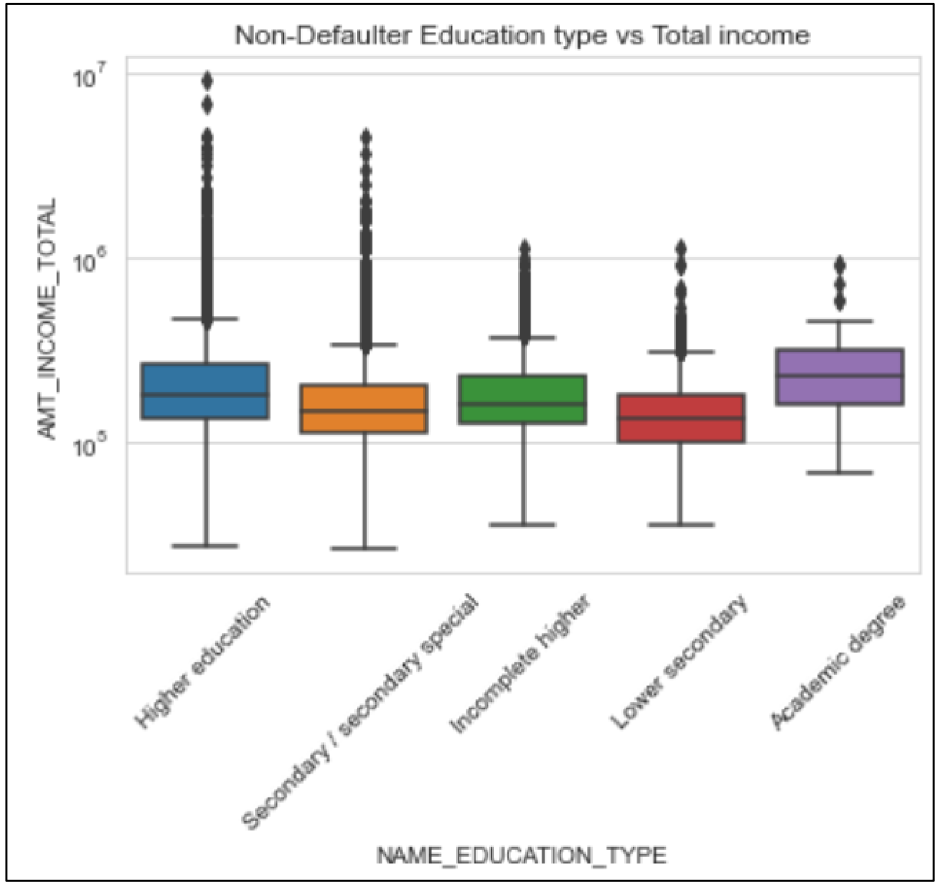


In the above graphs of Education type vs loan amount for Defaulter and Non-Defaulter, we can observe

- Secondary and incomplete higher education type applicant are on higher side of defaulter
- Defaulter applicant with Academic degree type of education are having higher credit/loan amount than others.

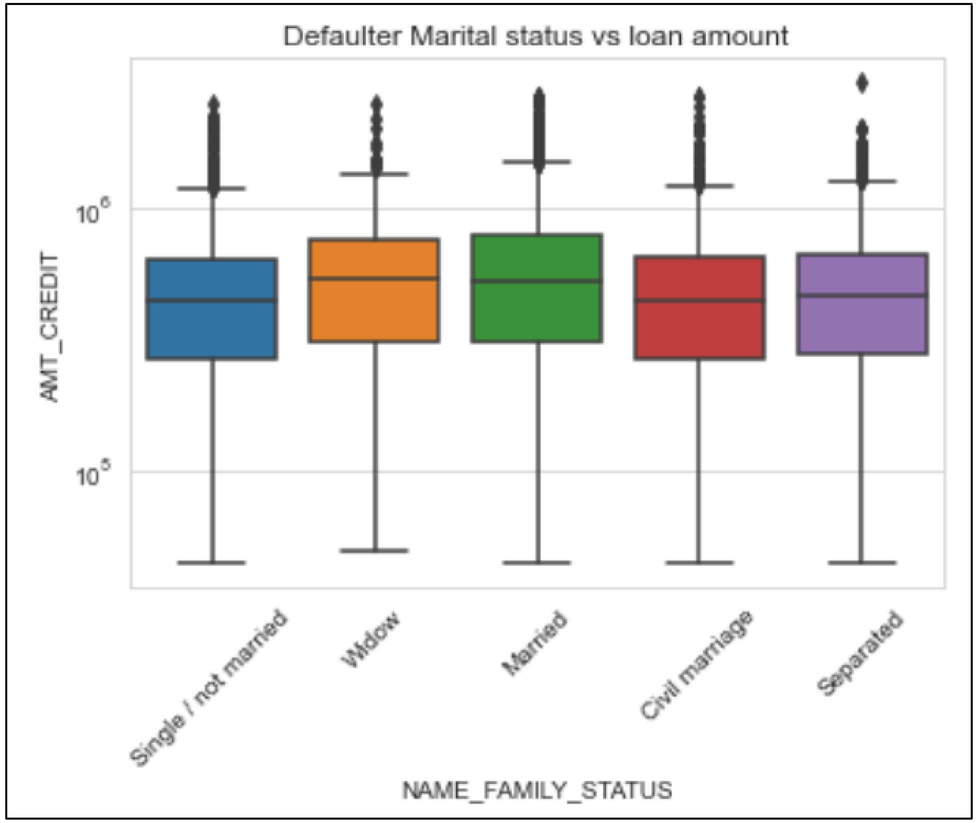
Education type vs Income

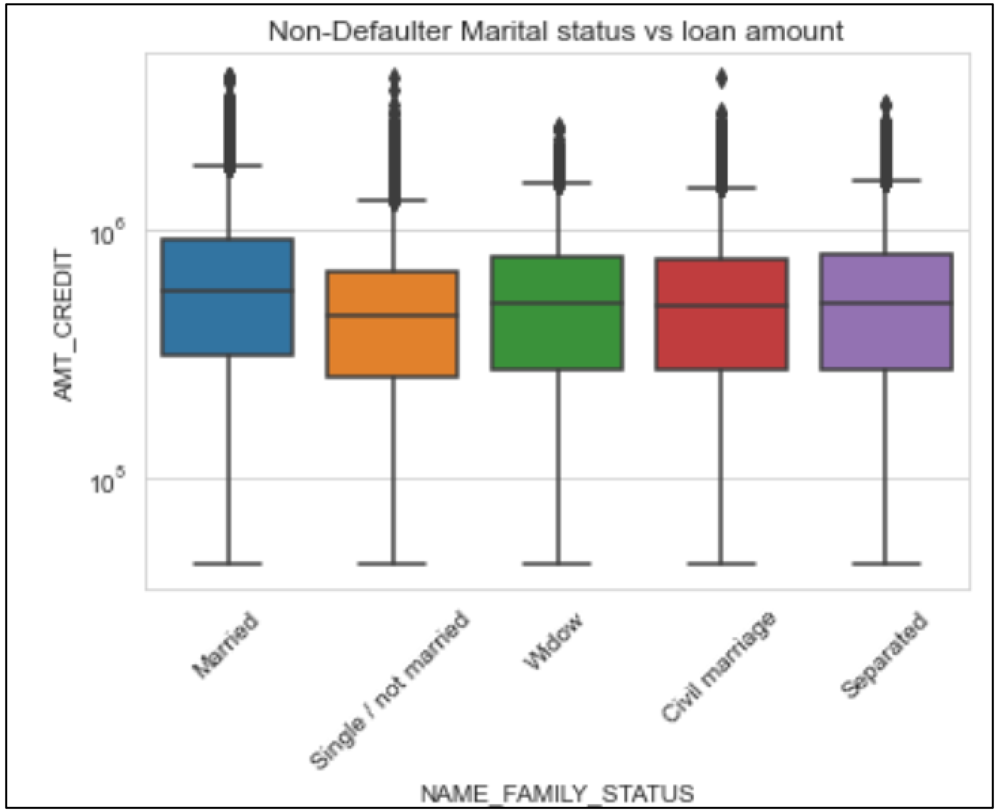




In the above graphs of Education type vs Total income for Defaulter and Non-Defaulter, can observe Defaulter applicant with Academic degree are having higher income than others at low extreme.

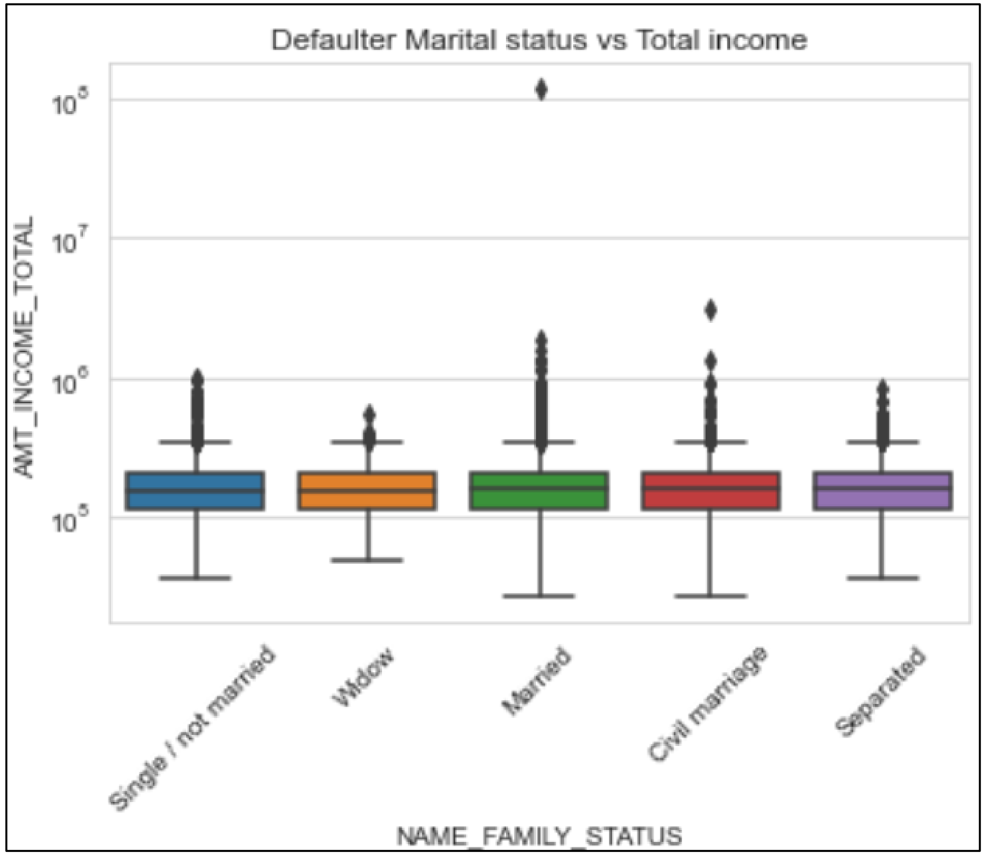
Marital status vs Loan amount

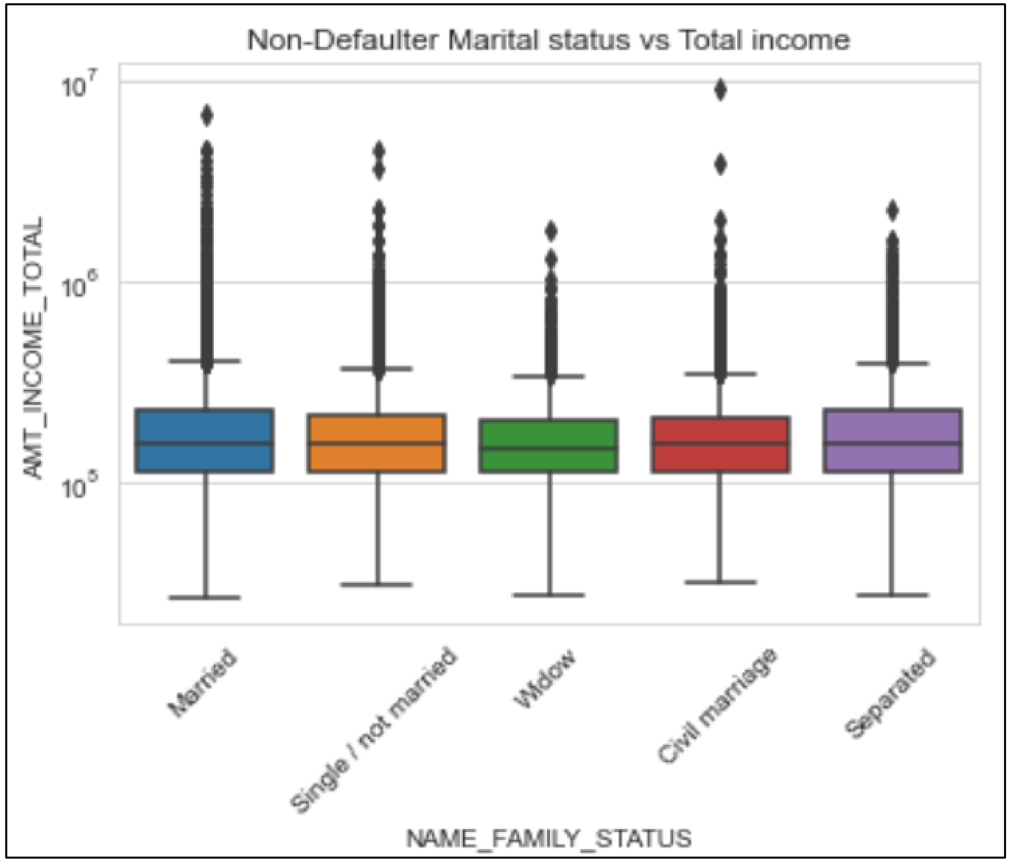




Above graphs of Marital status vs loan amount for Defaulter and Non-Defaulter, appears to be similar. Single/ Not married appears be on the higher side to be a defaulter. Married Non-Defaulter has higher loan amount than other.

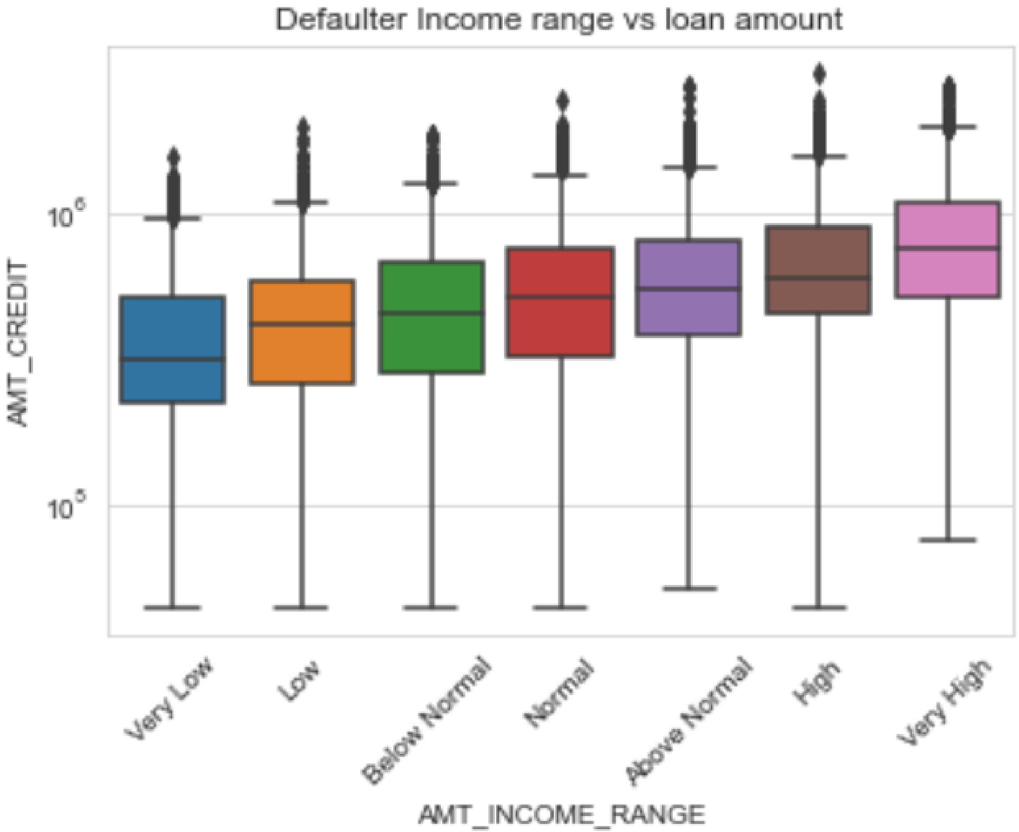
Marital status vs Income

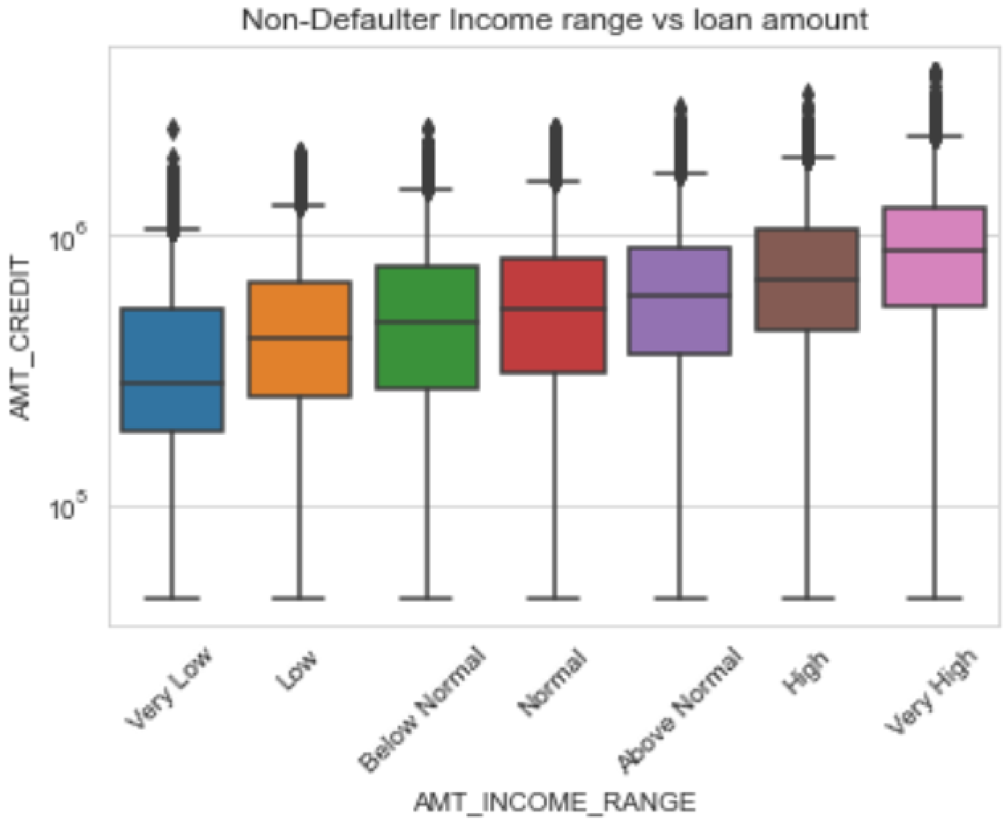




Above graphs of Marital status vs Total income for Defaulter and Non-Defaulter, can observe: 'Single/not married', ‘Separated', 'civil married' and 'Married' are having higher income

Income range vs Loan credit

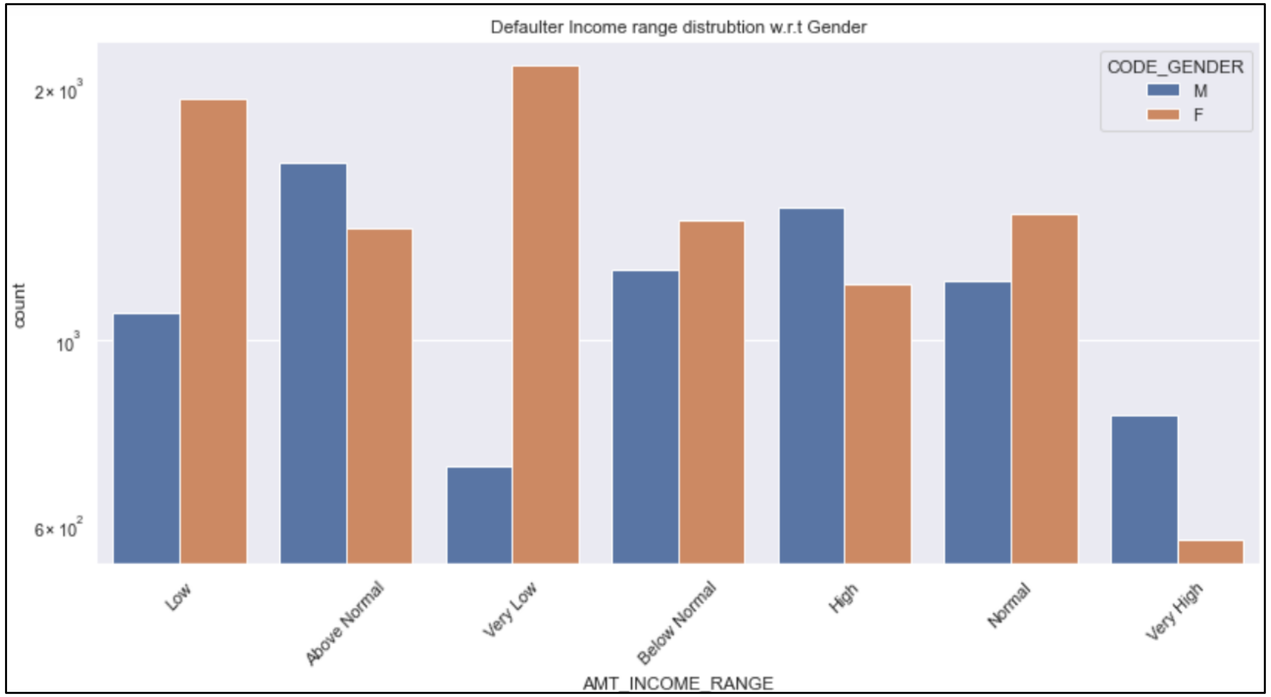


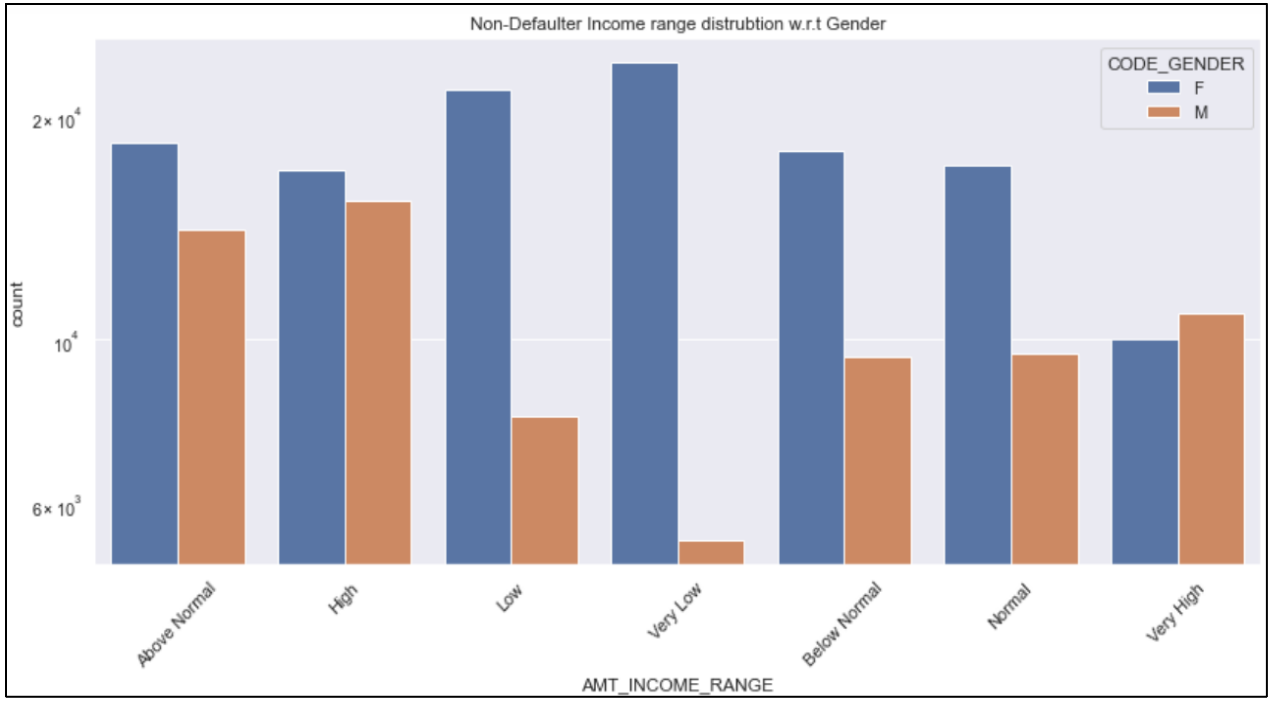


Above graphs of Marital status vs loan amount for Defaulter and Non-Defaulter, appears to be similar. Lower income range has low credit/loan amount. Also, there are few outliers : in non-defaulter with very low income range and in defaulters with high income range.

Bivariate analysis
for categorical variables vs categorical variables

Income range vs Gender

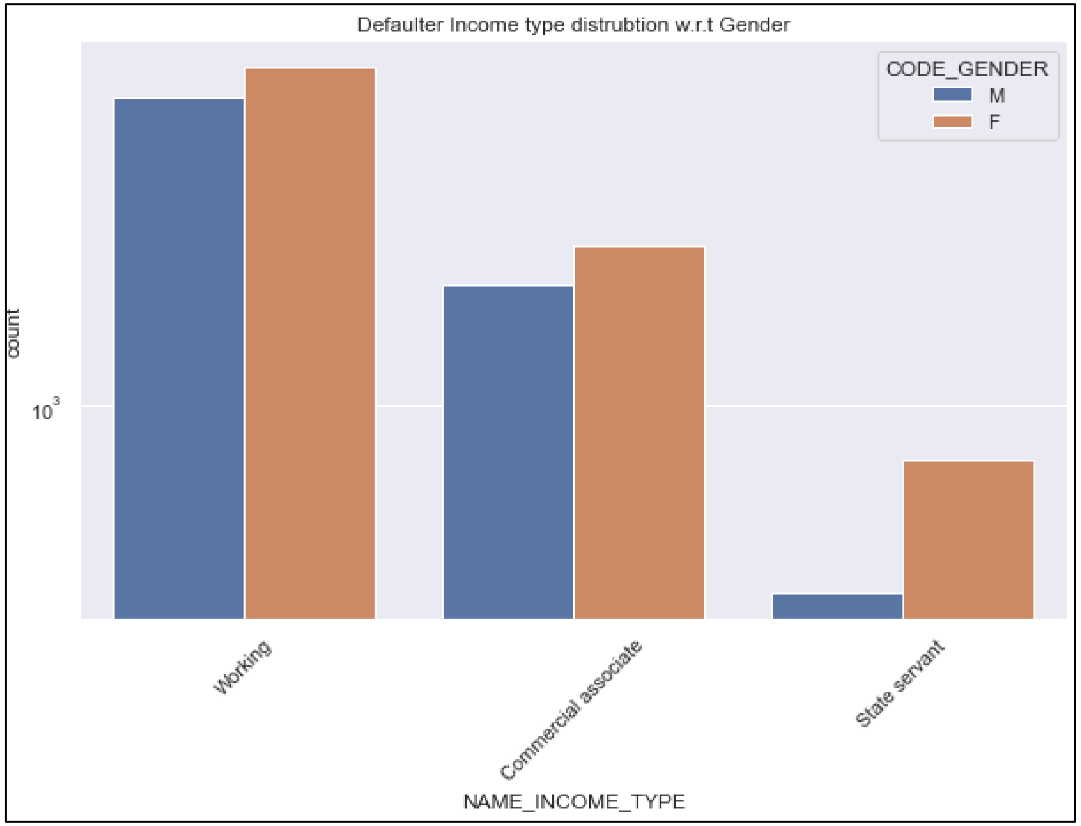


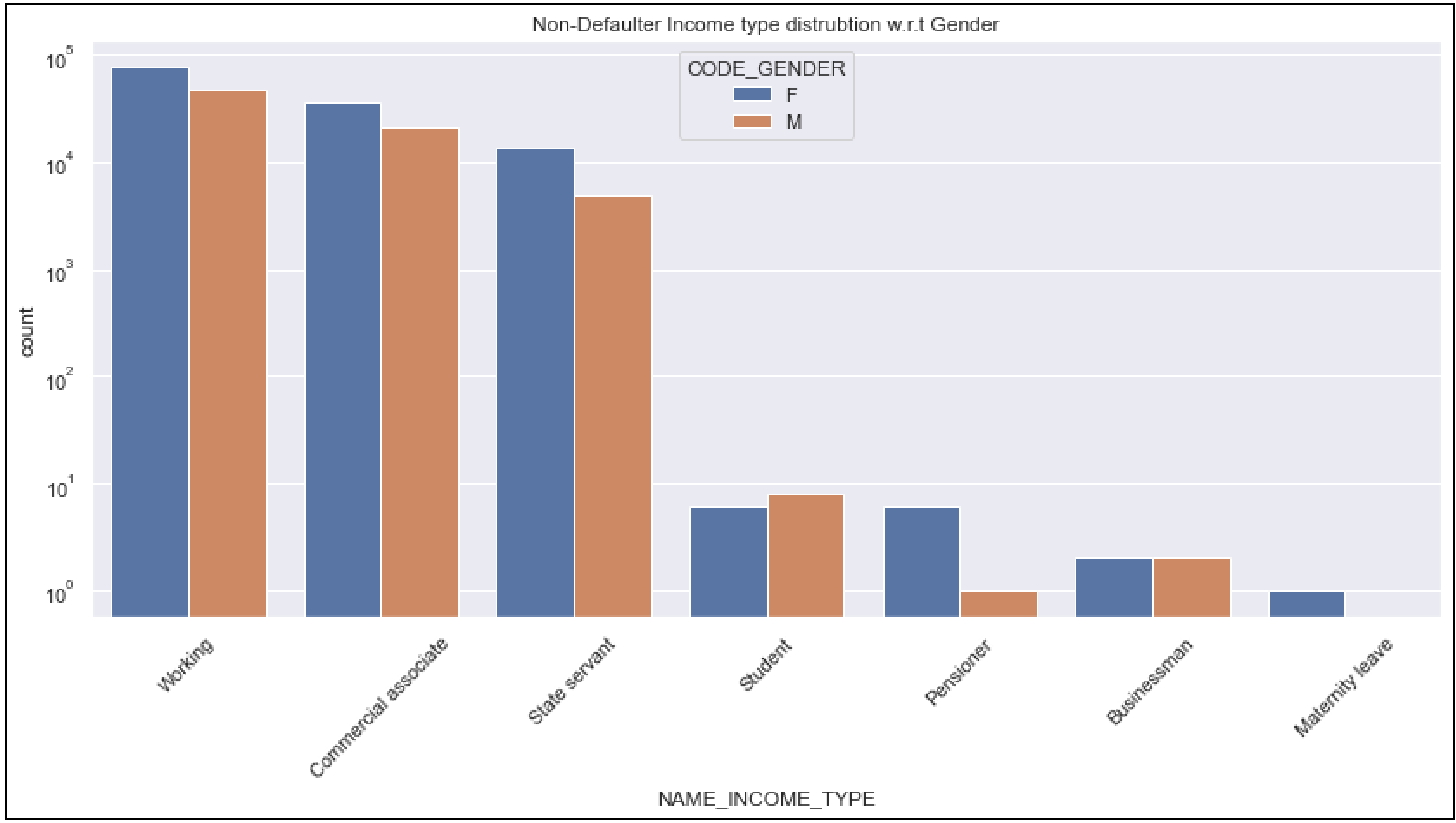


From the above graphs, We can observe:

- Male count is higher than Female in defaulter list. Female defaulter are more in 'Very low' and 'Low' income.
- Female count is higher than Male in Non-defaulter list, under all range of income.

Income type vs Gender

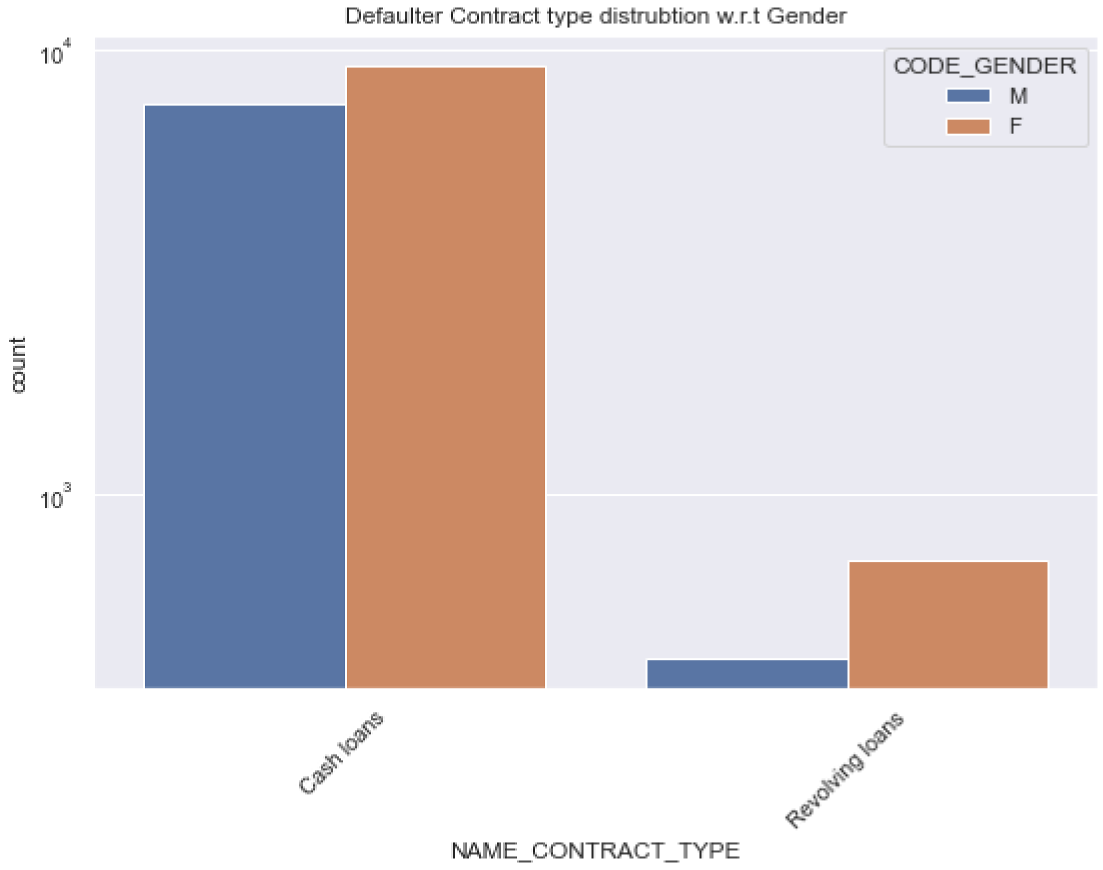




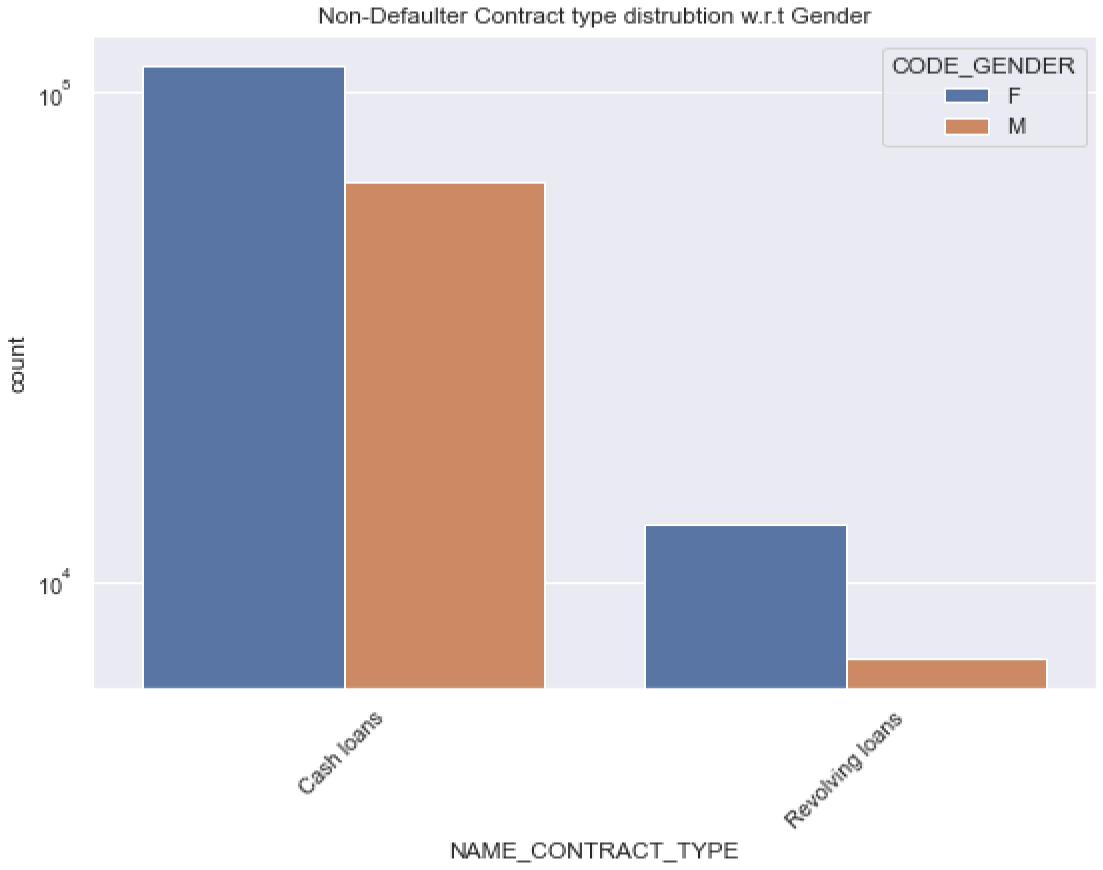
From the above graphs, We can observe:

- Male count is more than female in defaulter list. Majority of defaulters are working type.
- Females count is more than male in non-defaulter list. Majority of loans credited for 'Working', 'Commercial associate' and 'State servant'. There is less no. of loan credited to 'Students', 'Pensioner', Businessman' and lesser number credited during maternity leave.

Contract type vs Gender



Defaulter Contract type distrubtion w.r.t Gender

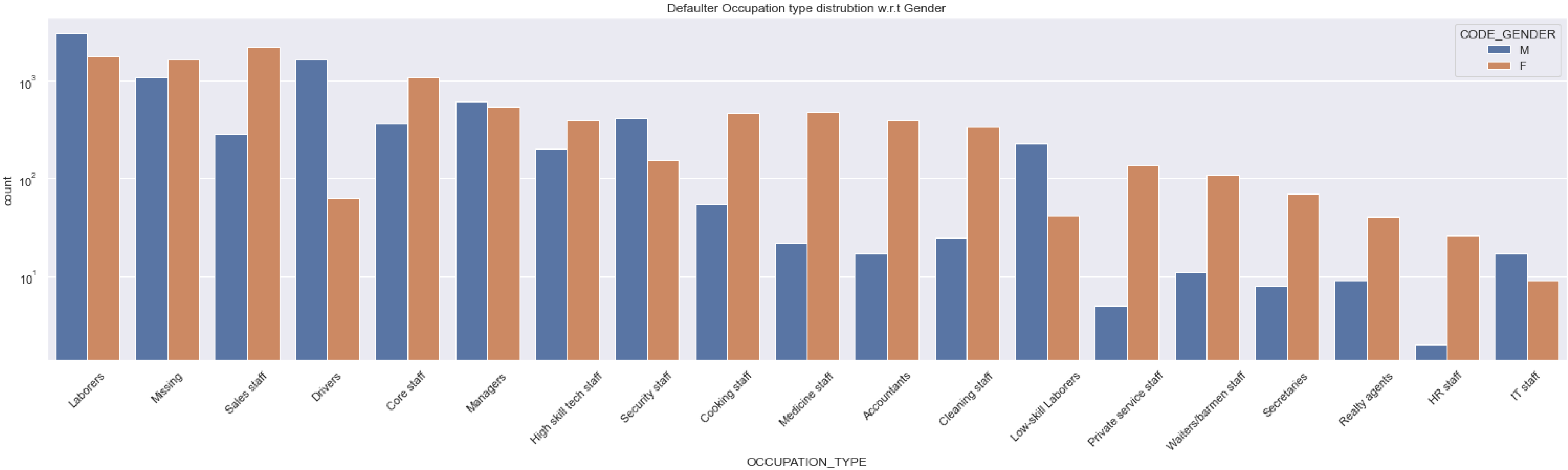


Non-Defaulter Contract type distrubtion w.r.t Gender

From the above graphs, We can observe:

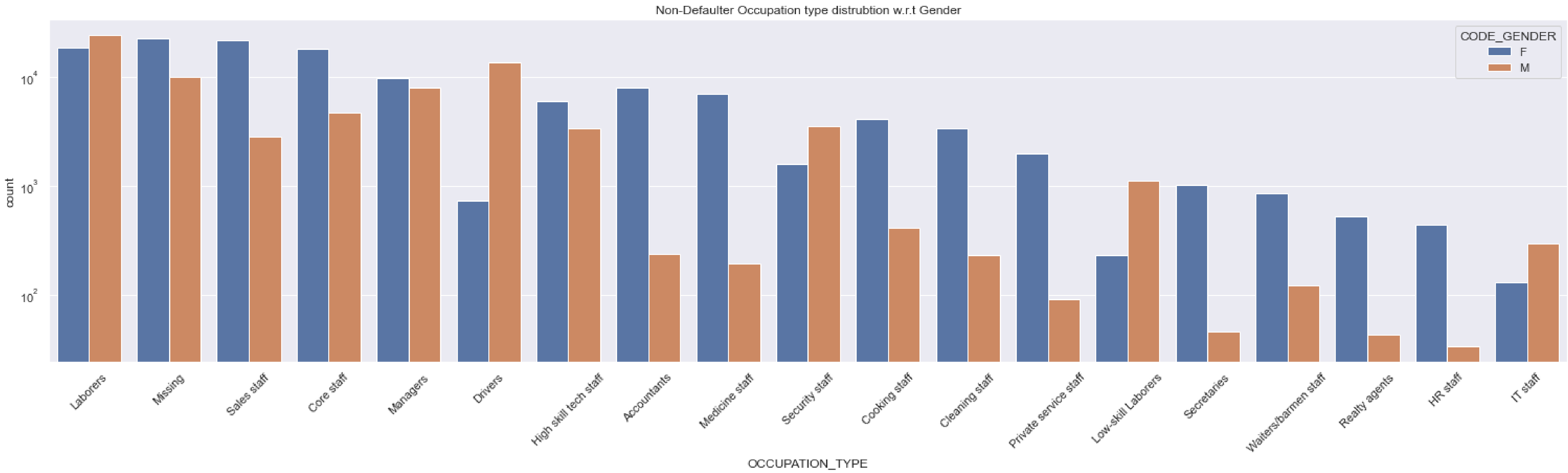
- Male count is more than female in defaulter list. Majority of defaulters are from Cash loans as this type of loan is preferred most by the clients.
- Female count is more than Male in Non-defaulter list. Majority of the loans are of Cash loan which is preferred more than Revolving loans.

Occupation type vs Gender



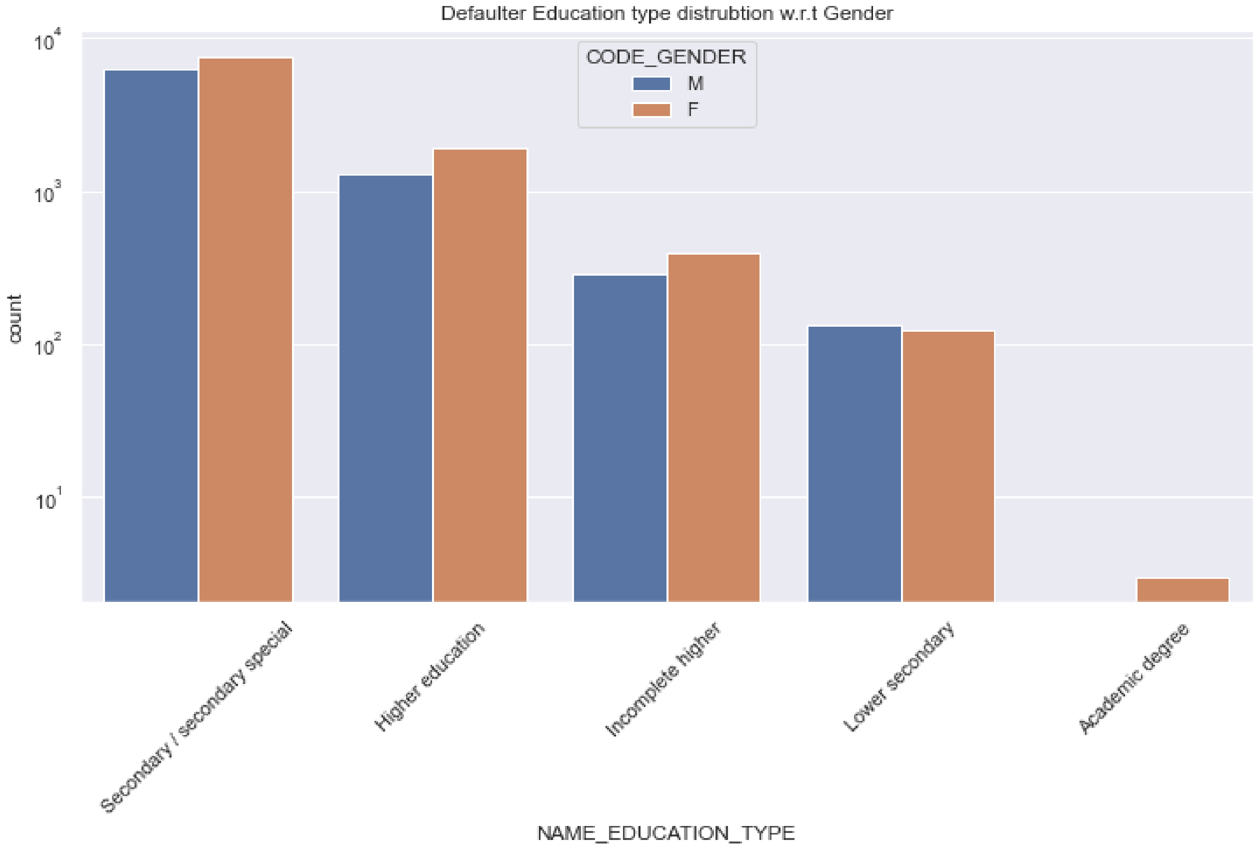
From the above graph, we can observe Male count is more than female in defaulter list. We can observe increase in defaulter % who are 'Laborers', 'Sales staff', 'Drivers'.

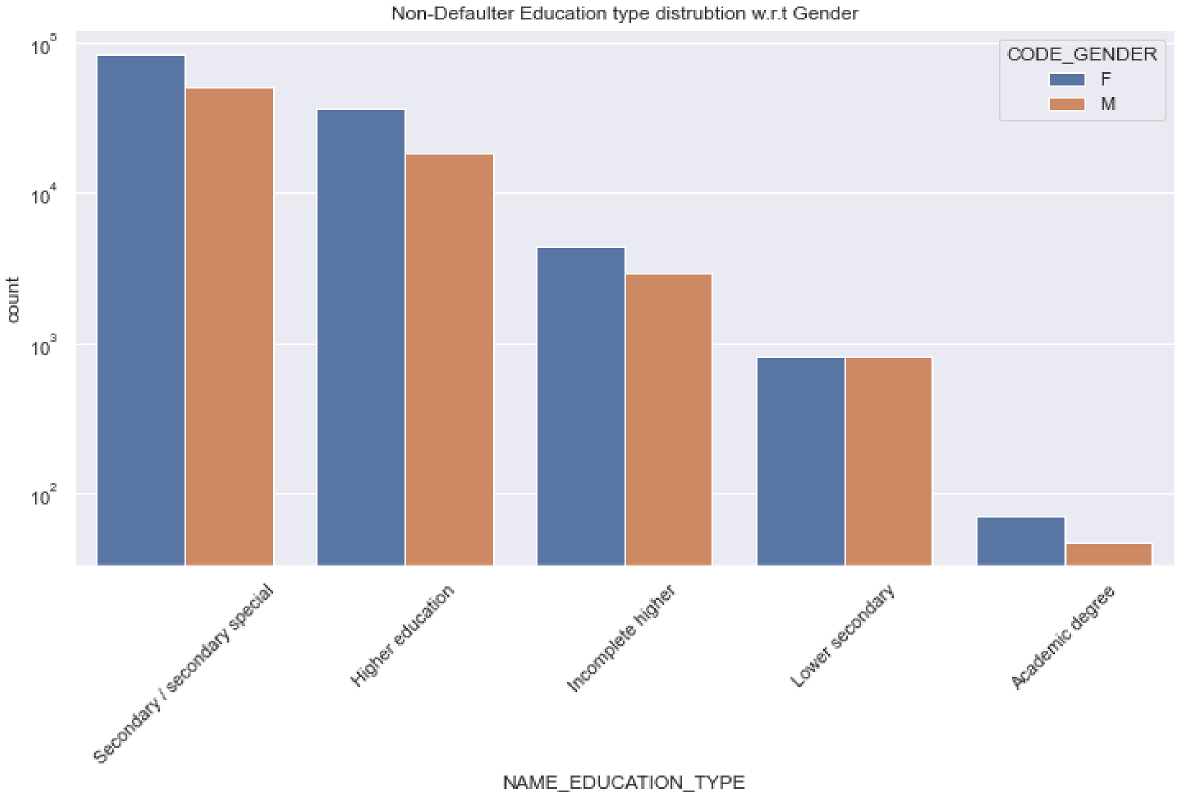
Occupation type vs Gender



From the above graph, we can observe Female count is more than Male in Non-defaulter list.We can observe decrease in defaulter % who are 'Core staff'.

Education type vs Gender

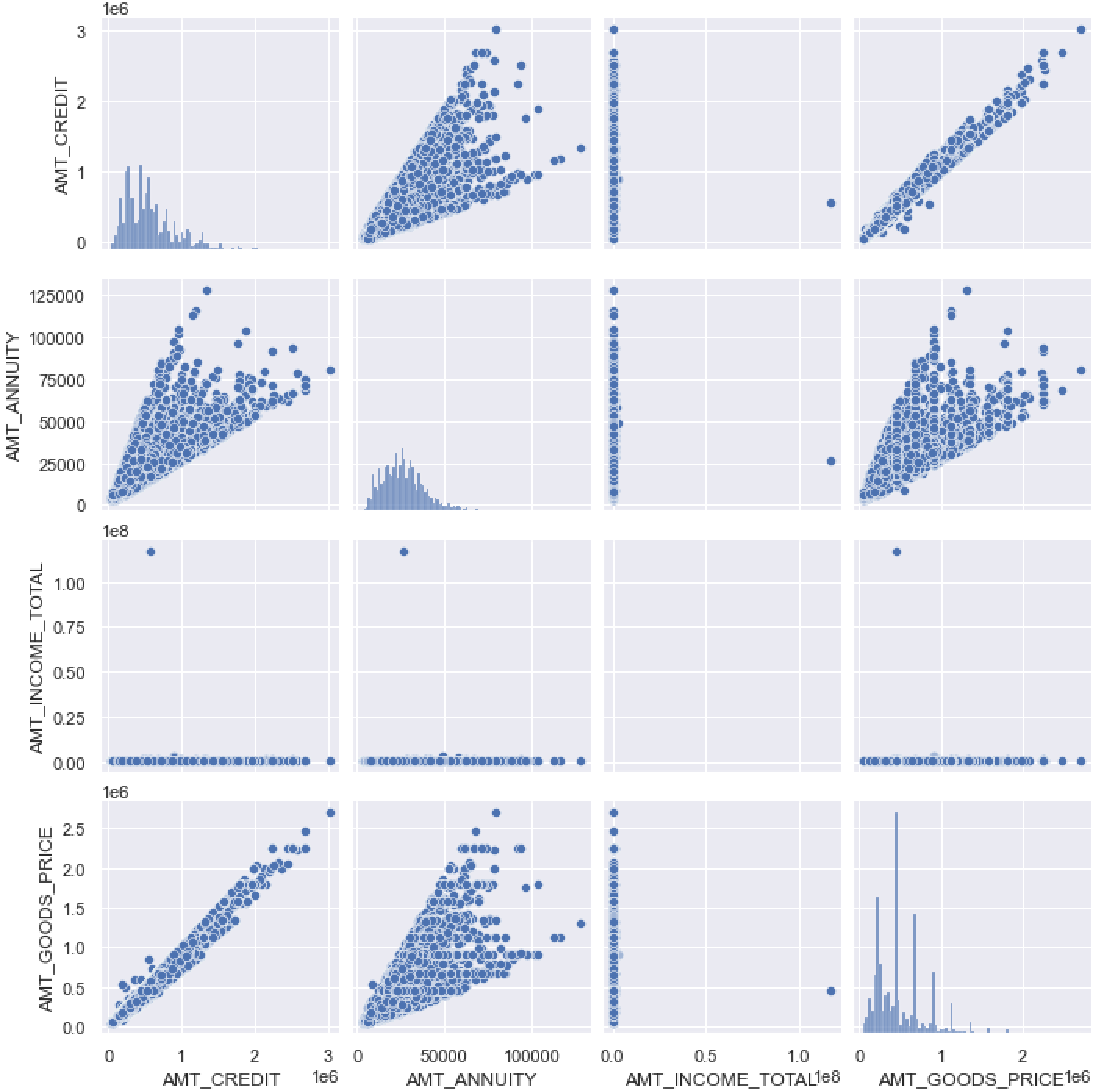




In the above graph, we can observe:

- Female non-defaulters is more than Male. Less number of loan provided for clients with academic degree and Majority of loan given/applied by the clients with 'Secondary' / 'Higher education’
- Male is leading in the defaulter list. Interesting observation is Male with academic degree education is not seen as defaulter here. Since, Majority of loans given for education type 'Secondary' and 'Higher education' defaulter under this type is also high.

Bivariate analysis
for Numerical variables vs Numerical variables

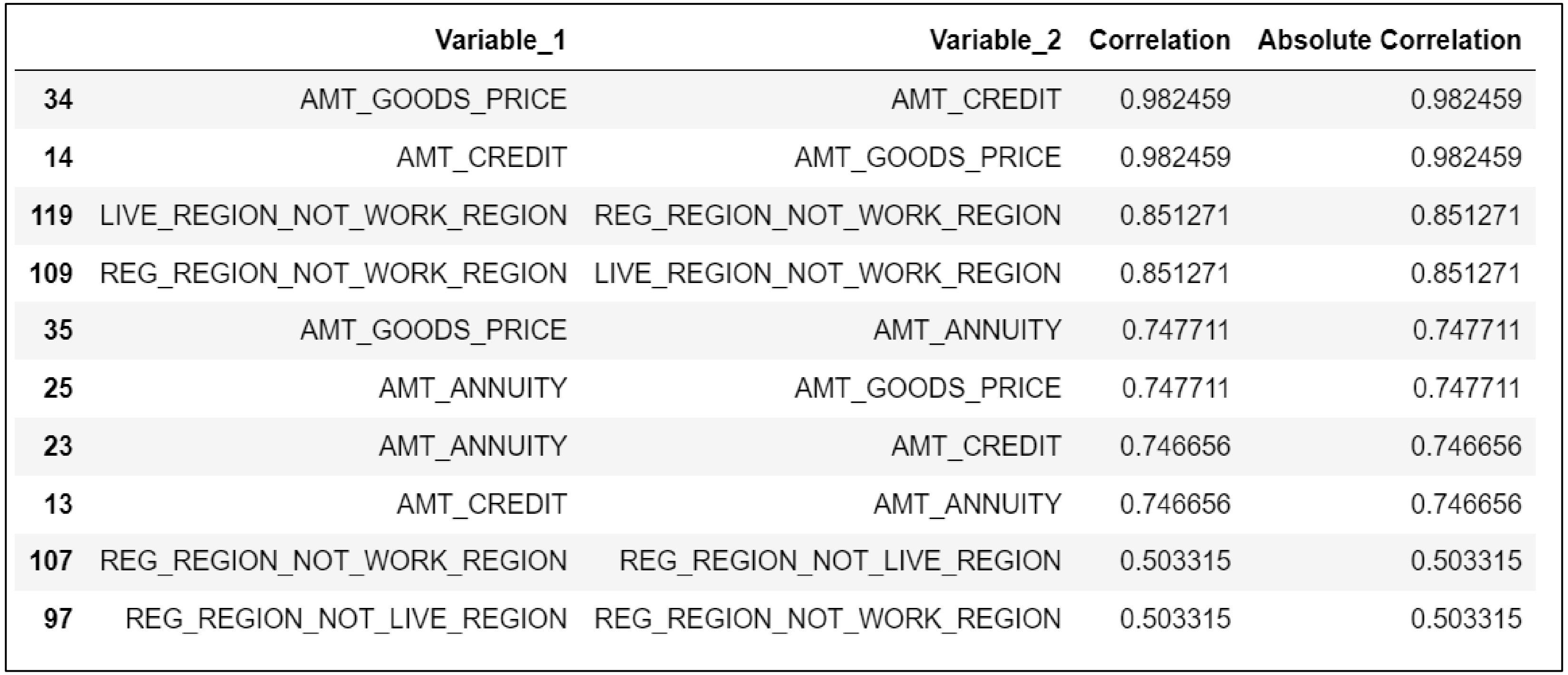


Analysis of different numerical variables for Defaulters

Here, 4 numerical columns compared with each other– Loan amount , Annuity Income and Goods price.

In the plots, at ax[0][3] and ax[3][0], loan amount vs goods price, we can observe positive linear correlation.

Top 10 Correlation for client with payment difficulties



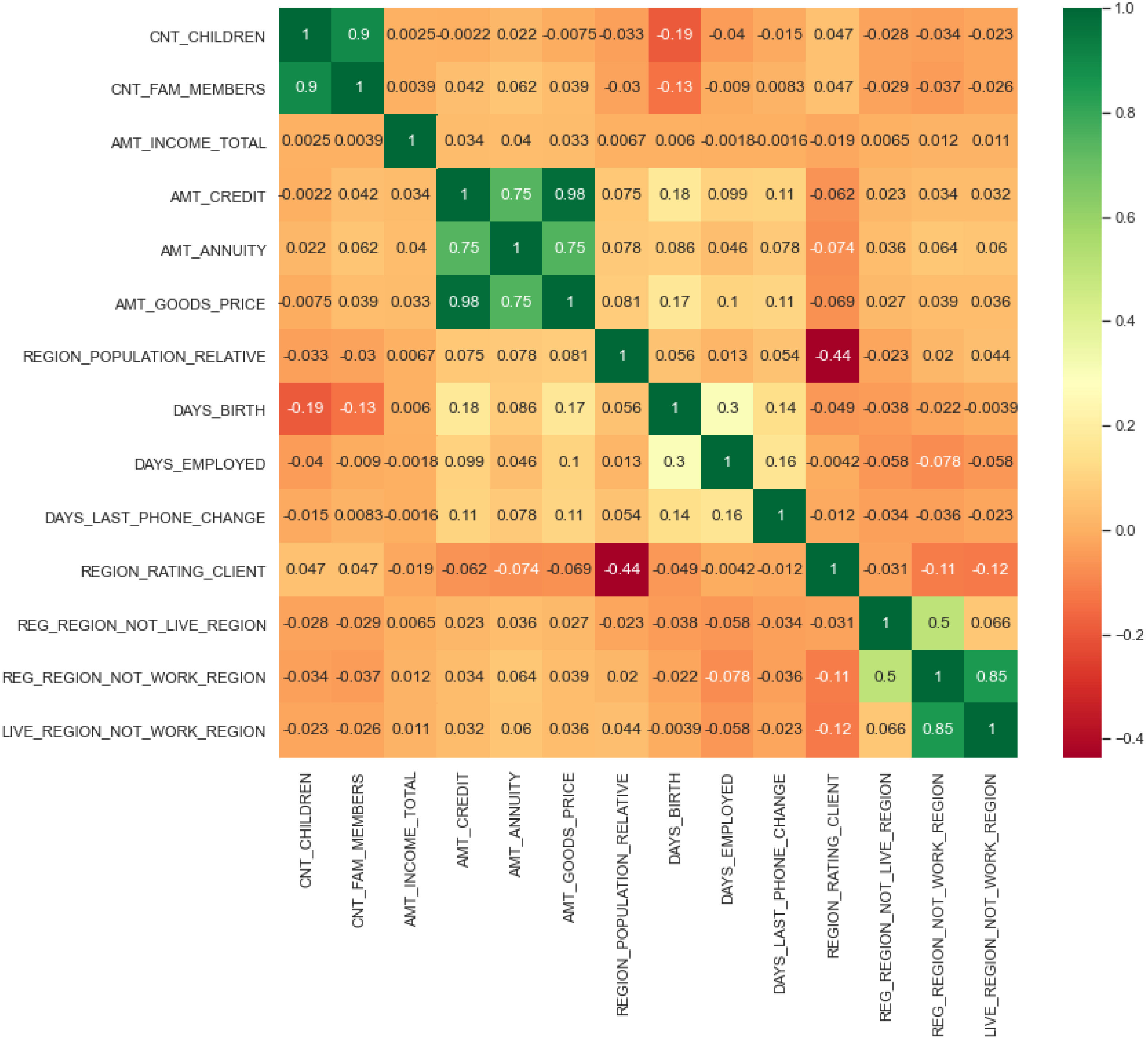
	Variable_1	Variable_2	Correlation	Absolute Correlation
34	AMT_GOODS_PRICE	AMT_CREDIT	0.982459	0.982459
14	AMT_CREDIT	AMT_GOODS_PRICE	0.982459	0.982459
119	LIVE_REGION_NOT_WORK_REGION	REG_REGION_NOT_WORK_REGION	0.851271	0.851271
109	REG_REGION_NOT_WORK_REGION	LIVE_REGION_NOT_WORK_REGION	0.851271	0.851271
35	AMT_GOODS_PRICE	AMT_ANNUITY	0.747711	0.747711
25	AMT_ANNUITY	AMT_GOODS_PRICE	0.747711	0.747711
23	AMT_ANNUITY	AMT_CREDIT	0.746656	0.746656
13	AMT_CREDIT	AMT_ANNUITY	0.746656	0.746656
107	REG_REGION_NOT_WORK_REGION	REG_REGION_NOT_LIVE_REGION	0.503315	0.503315
97	REG_REGION_NOT_LIVE_REGION	REG_REGION_NOT_WORK_REGION	0.503315	0.503315

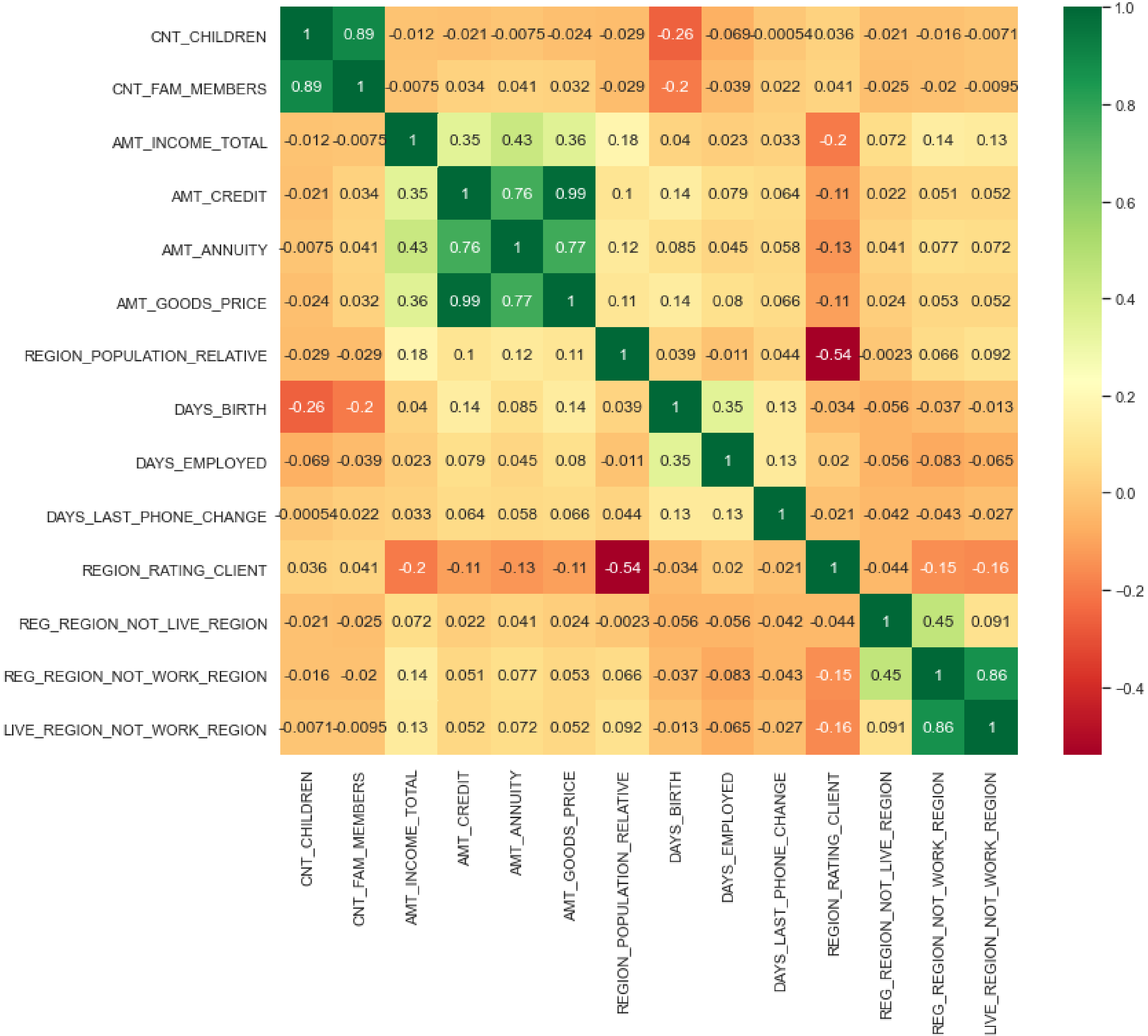
Analysis of different numerical variables for Defaulters

Here, many numerical columns are compared with each other. Higher positive number indicate more direct relatedness and lesser or negative number indicate inverse proportionality.

Observation from the graphs are:

- Loan amount vs number children are inversely proportional
- Loan amount, Goods price, Annuity are directly proportional.
- Loan amount, Annuity vs companies region rating are inversely proportional
- Population of region where client lives vs companies region rating are inversely proportional





	CNT_CHILDREN	CNT_FAM_MEMBERS	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_PRICE	REGION_POPULATION_RELATIVE	DAYS_BIRTH	DAYS_EMPLOYED	DAYS_LAST_PHONE_CHANGE	REGION_RATING_CLIENT	REG_REGION_NOT_LIVE_REGION	REG_REGION_NOT_WORK_REGION	LIVE_REGION_NOT_WORK_REGION
CNT_CHILDREN	1	0.89	-0.012	-0.021	-0.0075	-0.024	-0.029	-0.26	-0.069	-0.00054	0.036	-0.021	-0.016	-0.0071
CNT_FAM_MEMBERS	0.89	1	-0.0075	0.034	0.041	0.032	-0.029	-0.2	-0.039	0.022	0.041	-0.025	-0.02	-0.0095
AMT_INCOME_TOTAL	-0.012	-0.0075	1	0.35	0.43	0.36	0.18	0.04	0.023	0.033	-0.2	0.072	0.14	0.13
AMT_CREDIT	-0.021	0.034	0.35	1	0.76	0.99	0.1	0.14	0.079	0.064	-0.11	0.022	0.051	0.052
AMT_ANNUITY	-0.0075	0.041	0.43	0.76	1	0.77	0.12	0.085	0.045	0.058	-0.13	0.041	0.077	0.072
AMT_GOODS_PRICE	-0.024	0.032	0.36	0.99	0.77	1	0.11	0.14	0.08	0.066	-0.11	0.024	0.053	0.052
REGION_POPULATION_RELATIVE	-0.029	-0.029	0.18	0.1	0.12	0.11	1	0.039	-0.011	0.044	-0.54	-0.0023	0.066	0.092
DAYS_BIRTH	-0.26	-0.2	0.04	0.14	0.085	0.14	0.039	1	0.35	0.13	-0.034	-0.056	-0.037	-0.013
DAYS_EMPLOYED	-0.069	-0.039	0.023	0.079	0.045	0.08	-0.011	0.35	1	0.13	0.02	-0.056	-0.083	-0.065
DAYS_LAST_PHONE_CHANGE	-0.00054	0.022	0.033	0.064	0.058	0.066	0.044	0.13	0.13	1	-0.021	-0.042	-0.043	-0.027
REGION_RATING_CLIENT	0.036	0.041	-0.2	-0.11	-0.13	-0.11	-0.54	-0.034	0.02	-0.021	1	-0.044	-0.15	-0.16
REG_REGION_NOT_LIVE_REGION	-0.021	-0.025	0.072	0.022	0.041	0.024	-0.0023	-0.056	-0.056	-0.042	-0.044	1	0.45	0.091
REG_REGION_NOT_WORK_REGION	-0.016	-0.02	0.14	0.051	0.077	0.053	0.066	-0.037	-0.083	-0.043	-0.15	0.45	1	0.86
LIVE_REGION_NOT_WORK_REGION	-0.0071	-0.0095	0.13	0.052	0.072	0.052	0.092	-0.013	-0.065	-0.027	-0.16	0.091	0.86	1

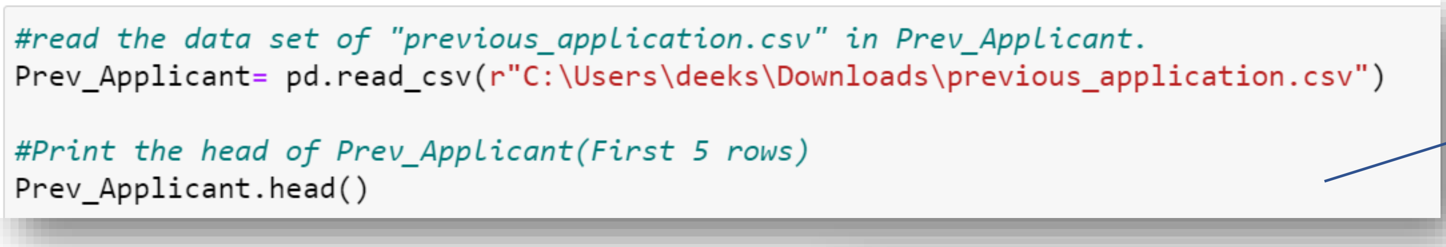
Analysis of different numerical variables for Non-Defaulters

Here, many numerical columns are compared with each other. Higher positive number indicate more direct relatedness and lesser or negative number indicate inverse proportionality.

Observation from the graphs are:

- Positive correlation between Credit vs Annuity, Goods price, Income and vice-versa.
- Negative correlation between Population of region where client lives vs rating of the region where client lives, Region Rating vs Credit, Annuity, Goods price, Income; Similar, observation seen for family size and number of children. Also, can observe that if client's permanent address does not match contact address--then it doesn't match with work address either

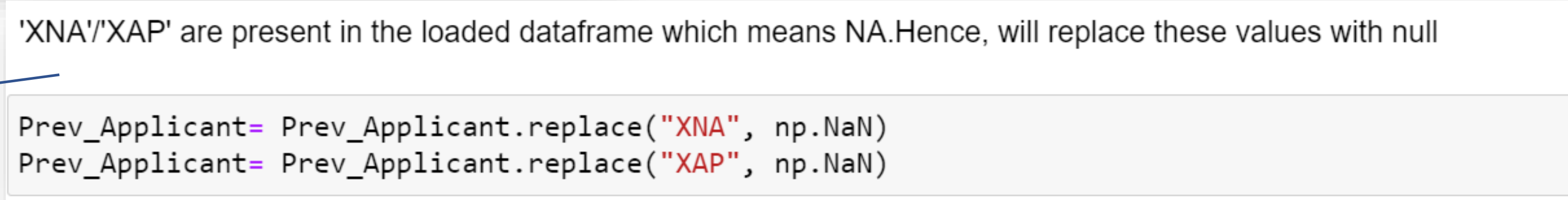
Analysis of previous application data:



```
#read the data set of "previous_application.csv" in Prev_Applicant.
Prev_Applicant= pd.read_csv(r"C:\Users\deeks\Downloads\previous_application.csv")

#Print the head of Prev_Applicant(First 5 rows)
Prev_Applicant.head()
```

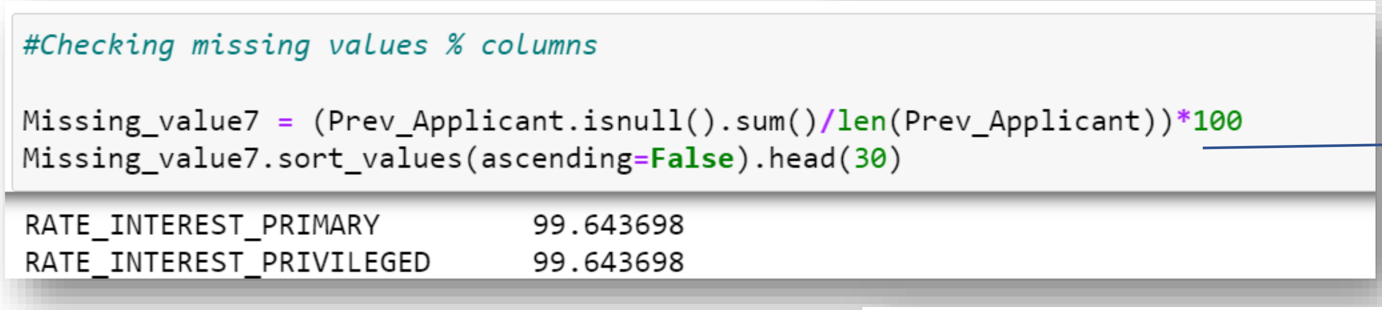
Read previous_data.csv file

Fixing errors in data



'XNA'/'XAP' are present in the loaded dataframe which means NA.Hence, will replace these values with null

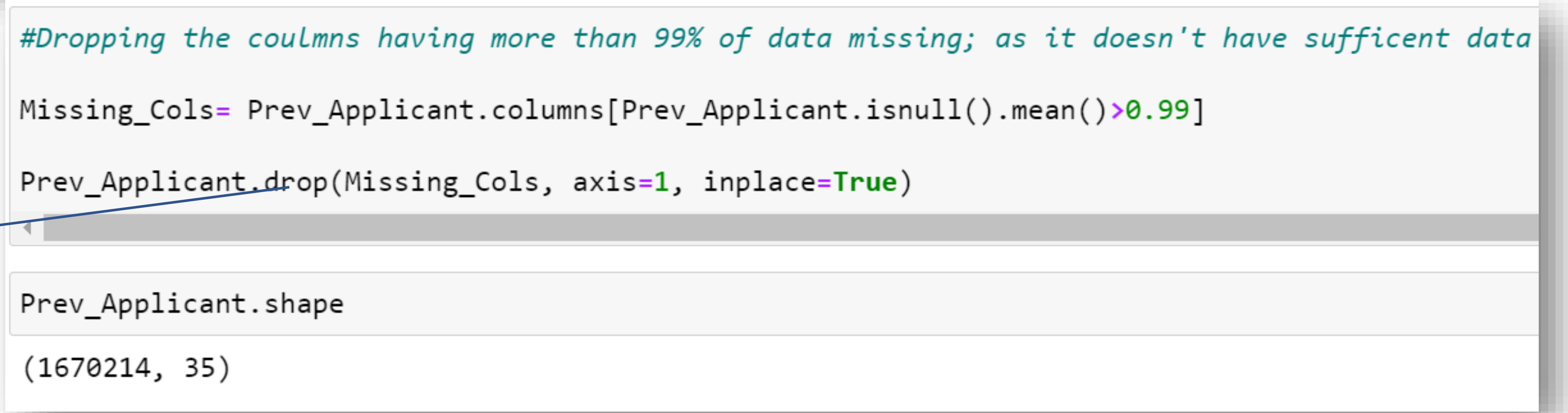
```
Prev_Applicant= Prev_Applicant.replace("XNA", np.NaN)
Prev_Applicant= Prev_Applicant.replace("XAP", np.NaN)
```



```
#Checking missing values % columns

Missing_value7 = (Prev_Applicant.isnull().sum()/len(Prev_Applicant))*100
Missing_value7.sort_values(ascending=False).head(30)
```

```
RATE_INTEREST_PRIMARY         99.643698
RATE_INTEREST_PRIVILEGED      99.643698
```

Checking for Missing values count %



```
#Dropping the coulmns having more than 99% of data missing; as it doesn't have sufficent data

Missing_Cols= Prev_Applicant.columns[Prev_Applicant.isnull().mean()>0.99]

Prev_Applicant.drop(Missing_Cols, axis=1, inplace=True)
```

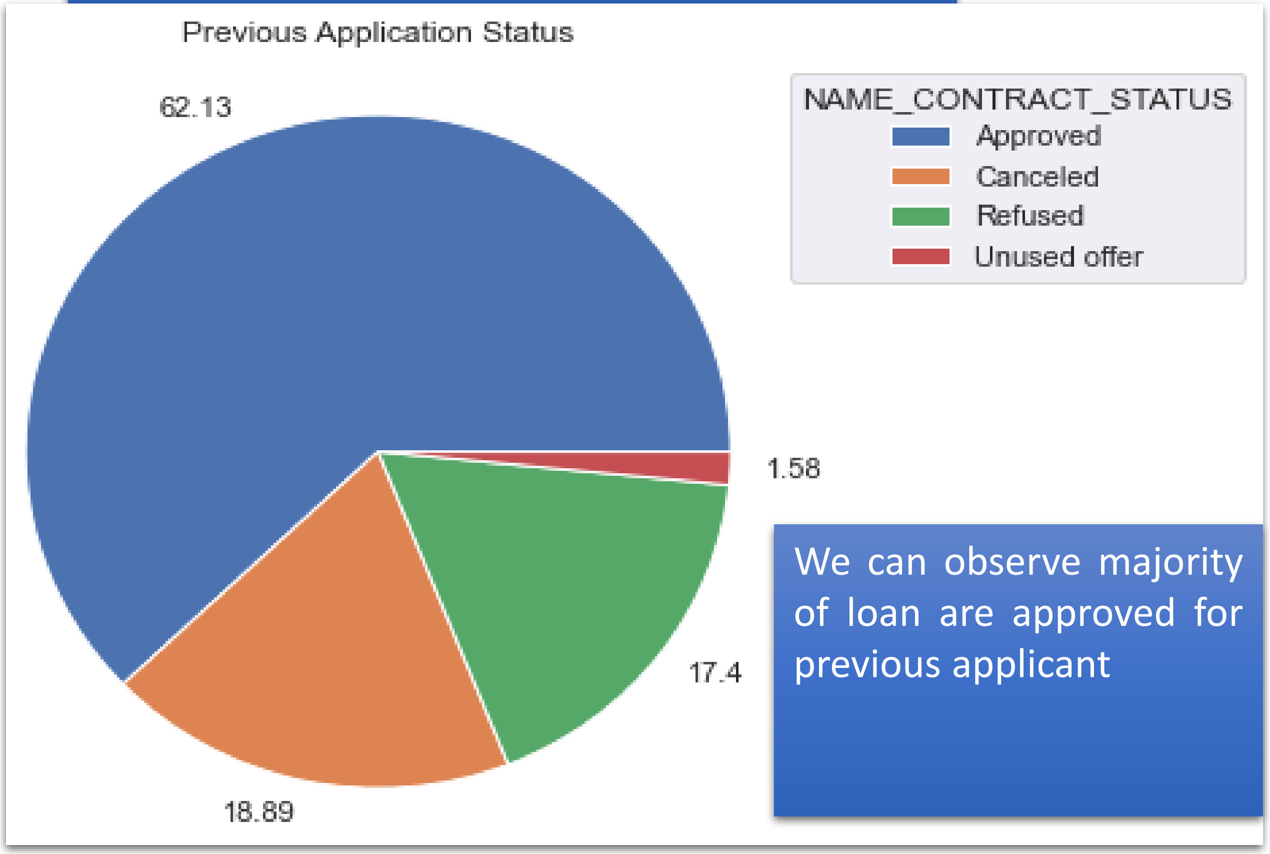
```
Prev_Applicant.shape
```

```
(1670214, 35)
```

Dropping columns where >99% of data is missing

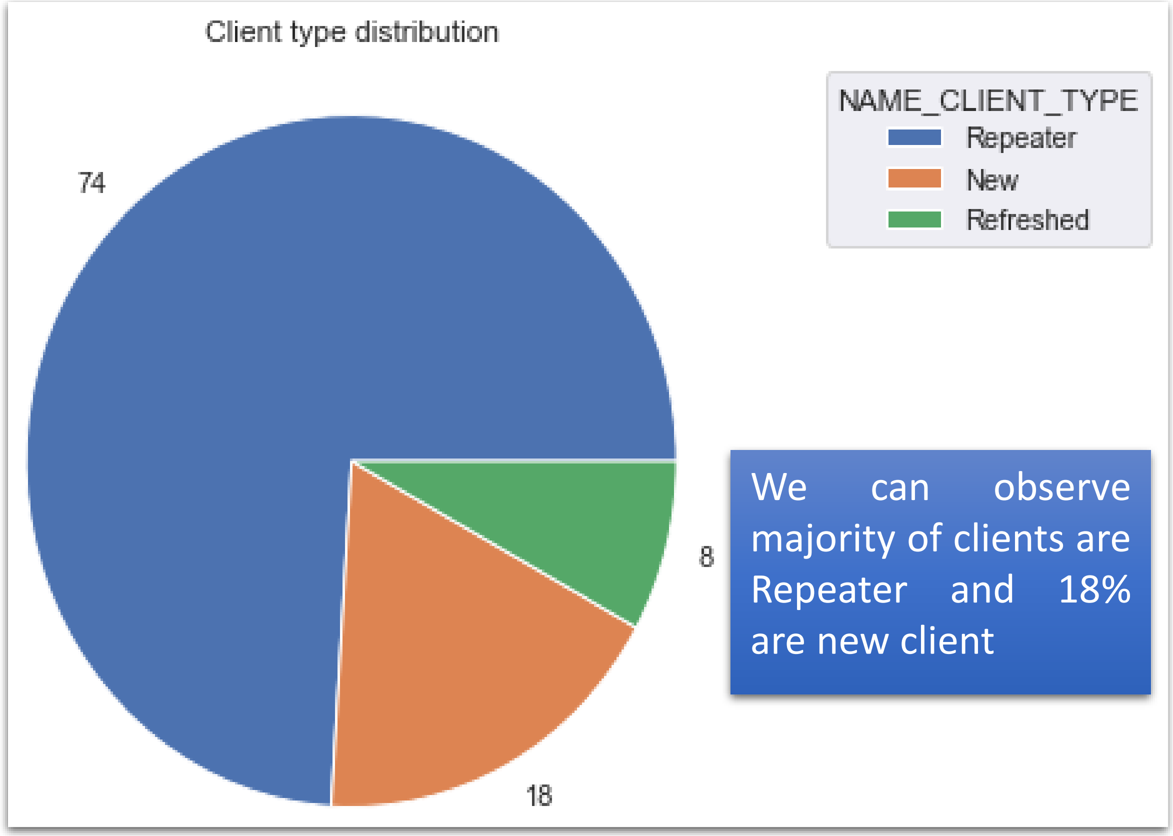
Univariant analysis on previous application for Categorical data

NAME_CONTRACT_STATUS



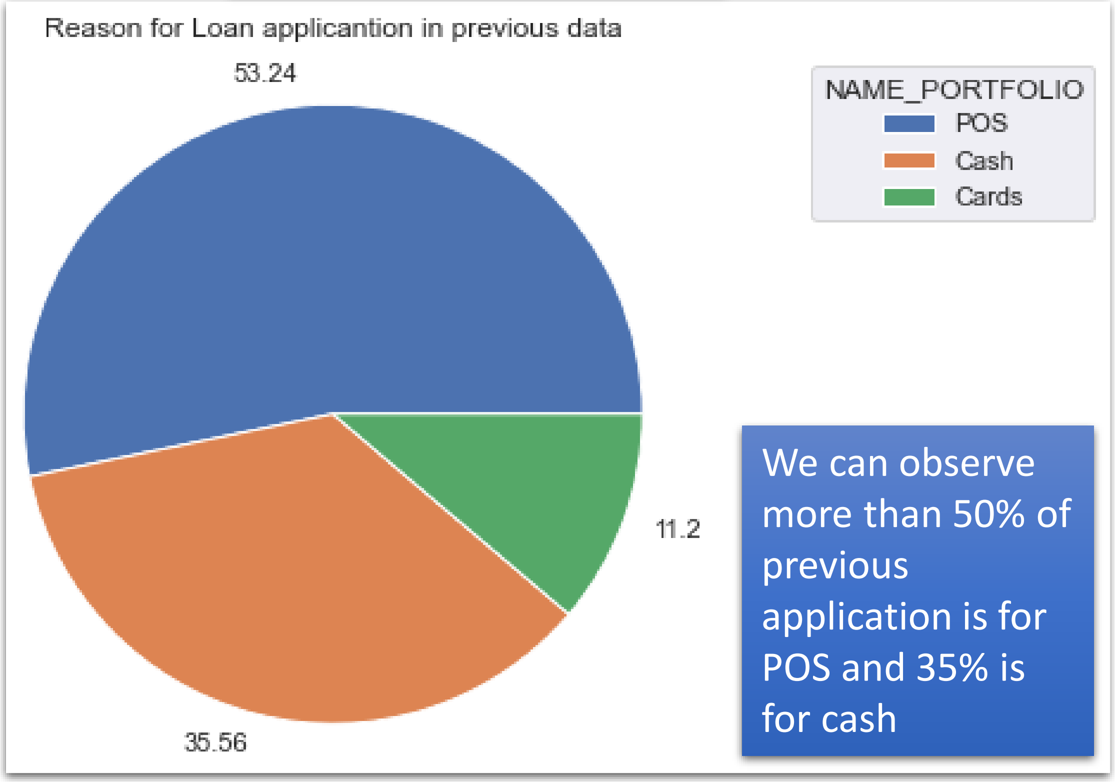
We can observe majority of loan are approved for previous applicant

CLIENT TYPE



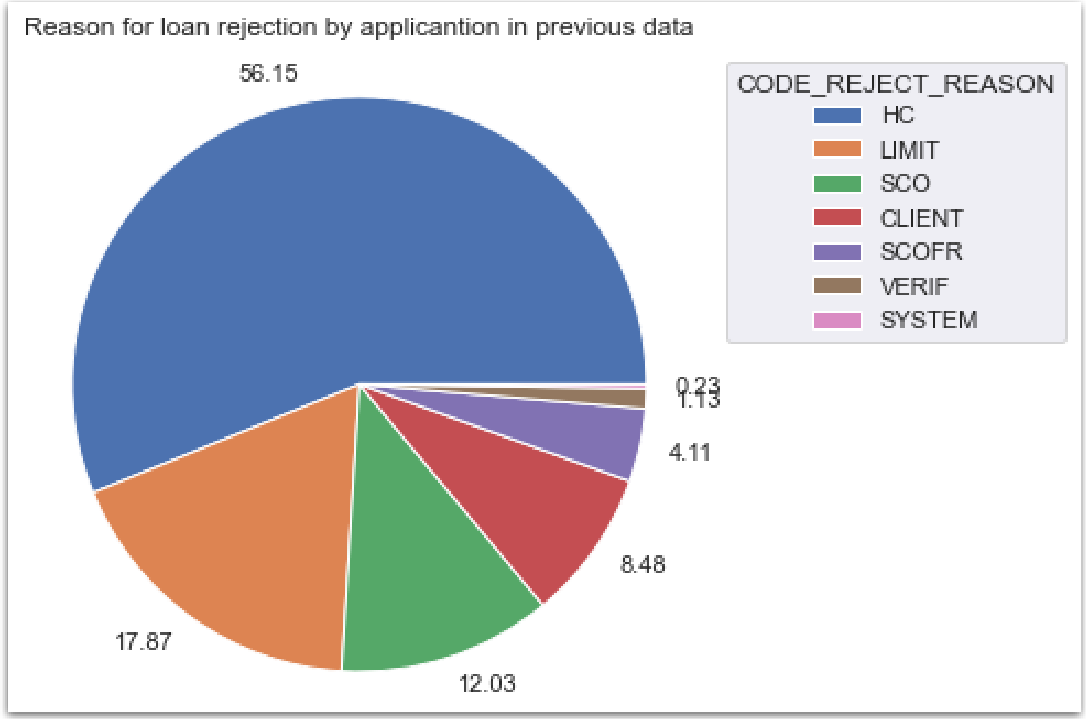
We can observe majority of clients are Repeater and 18% are new client

NAME_PORTFOLIO



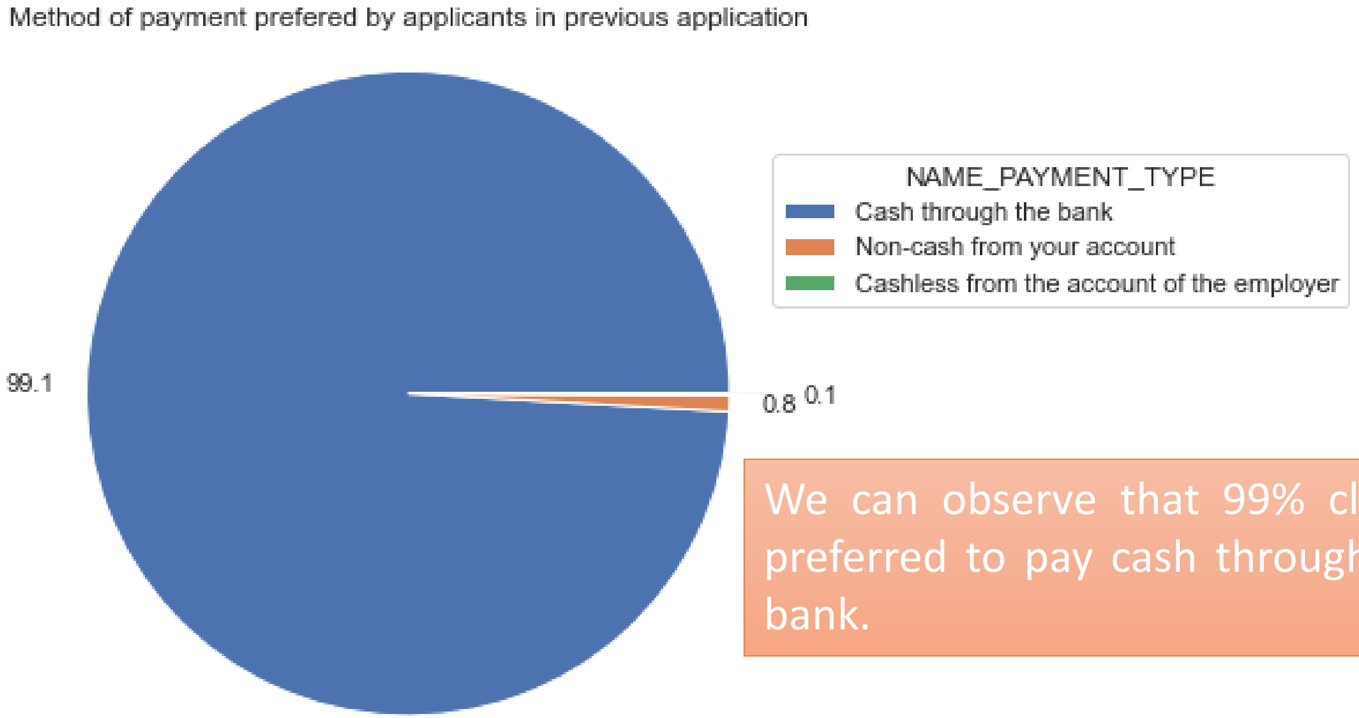
We can observe more than 50% of previous application is for POS and 35% is for cash

LOAN REJECTION REASON



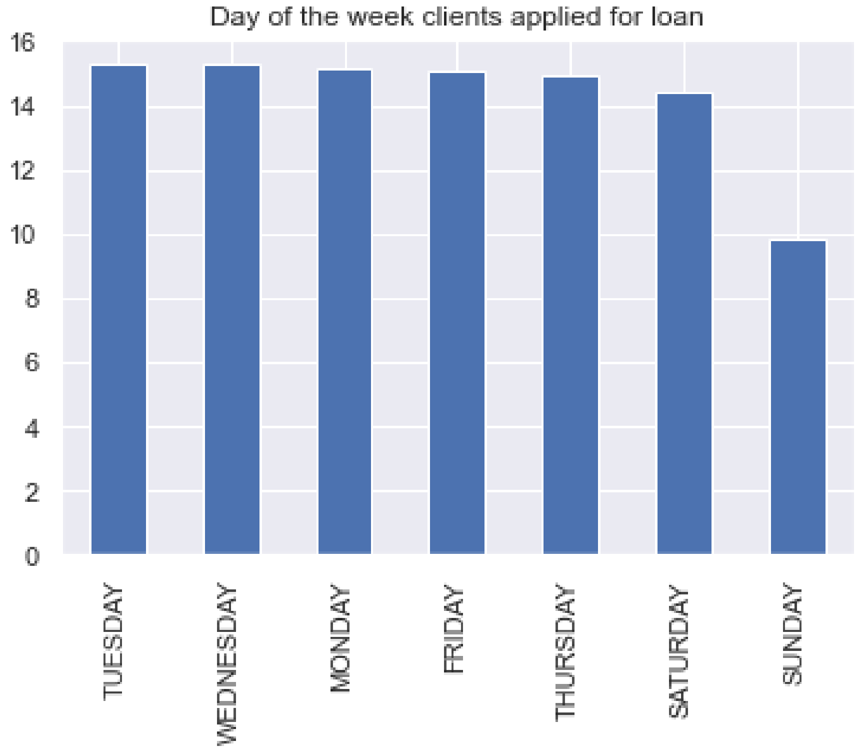
We can observe that more than 50% of applicants have rejected the loan due to HC and 18% due to limit.

LOAN PAYMENT TYPE



We can observe that 99% clients preferred to pay cash through the bank.

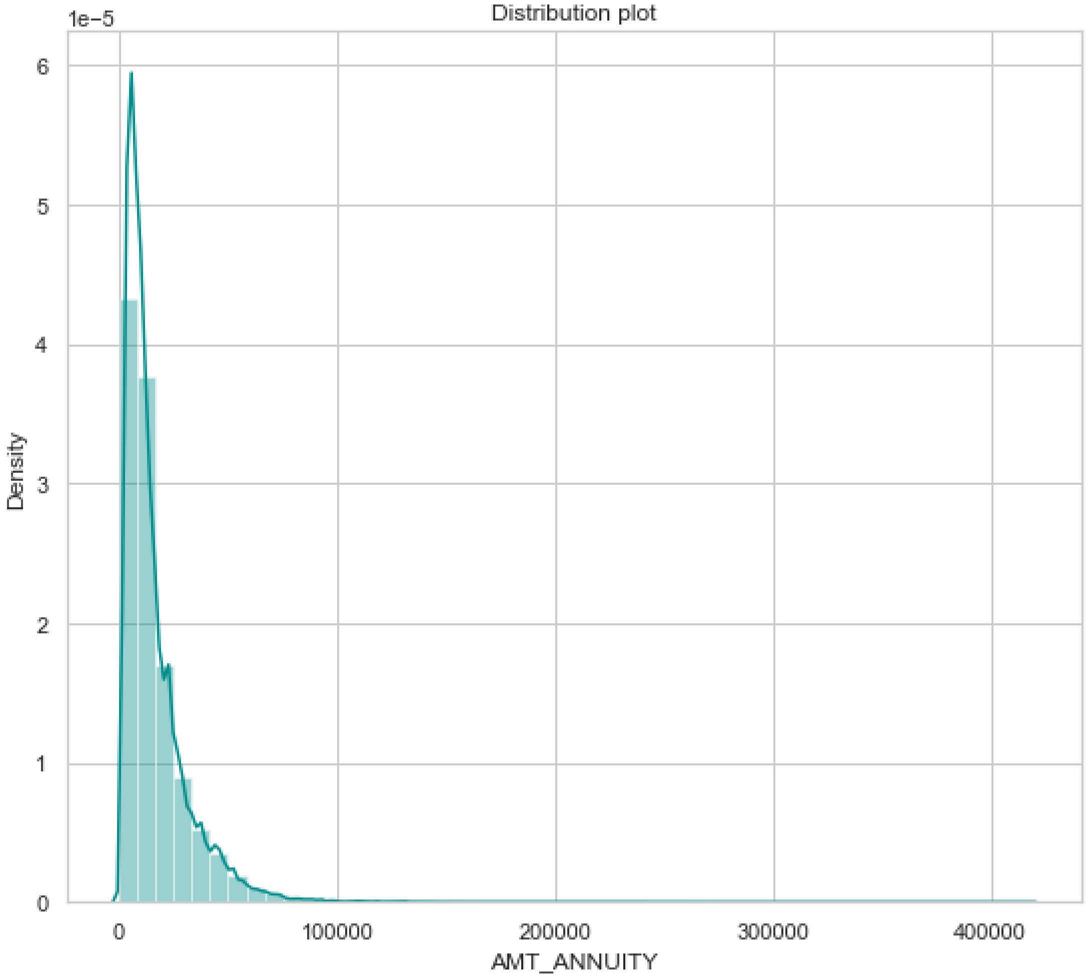
WEEKDAY_APPR_PROCESS_START

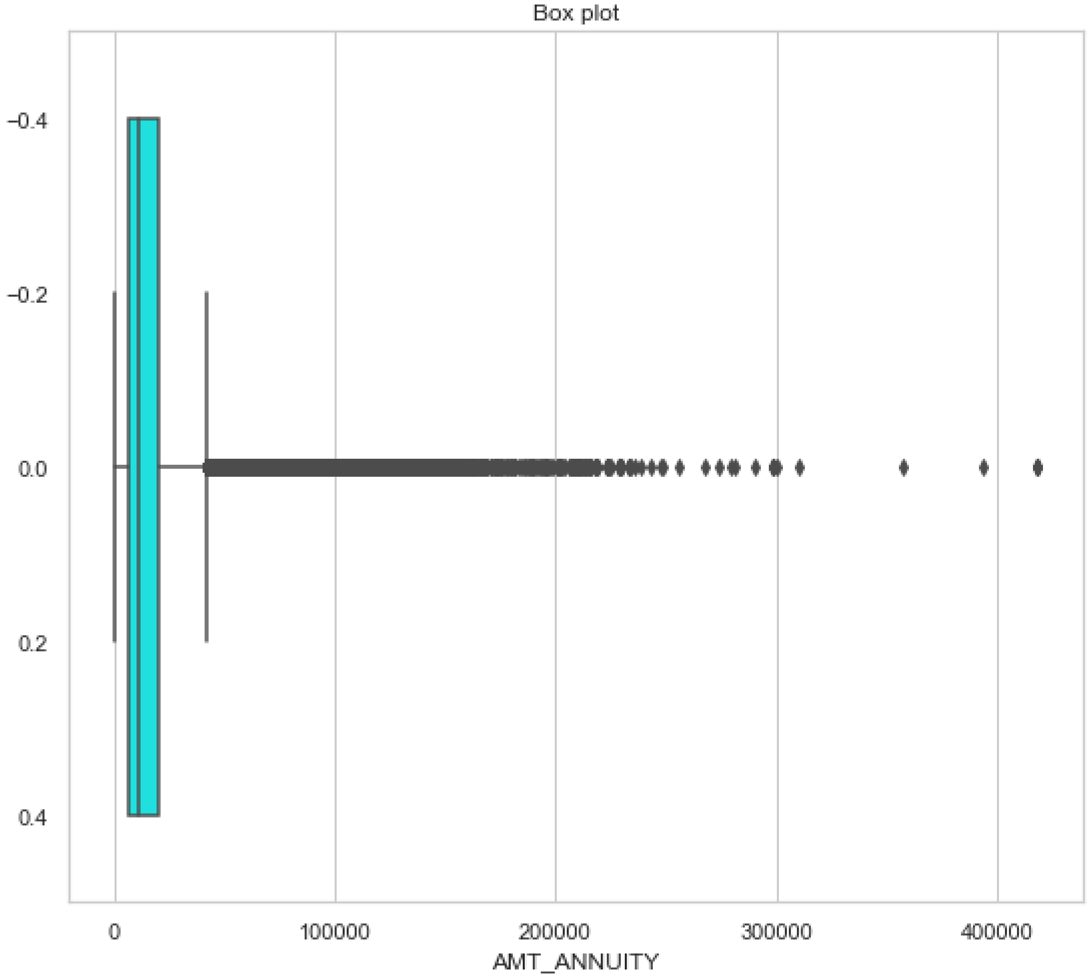


We can observe lesser number of applicant on weekend/Sunday.

Univariant analysis on previous application for Numerical data

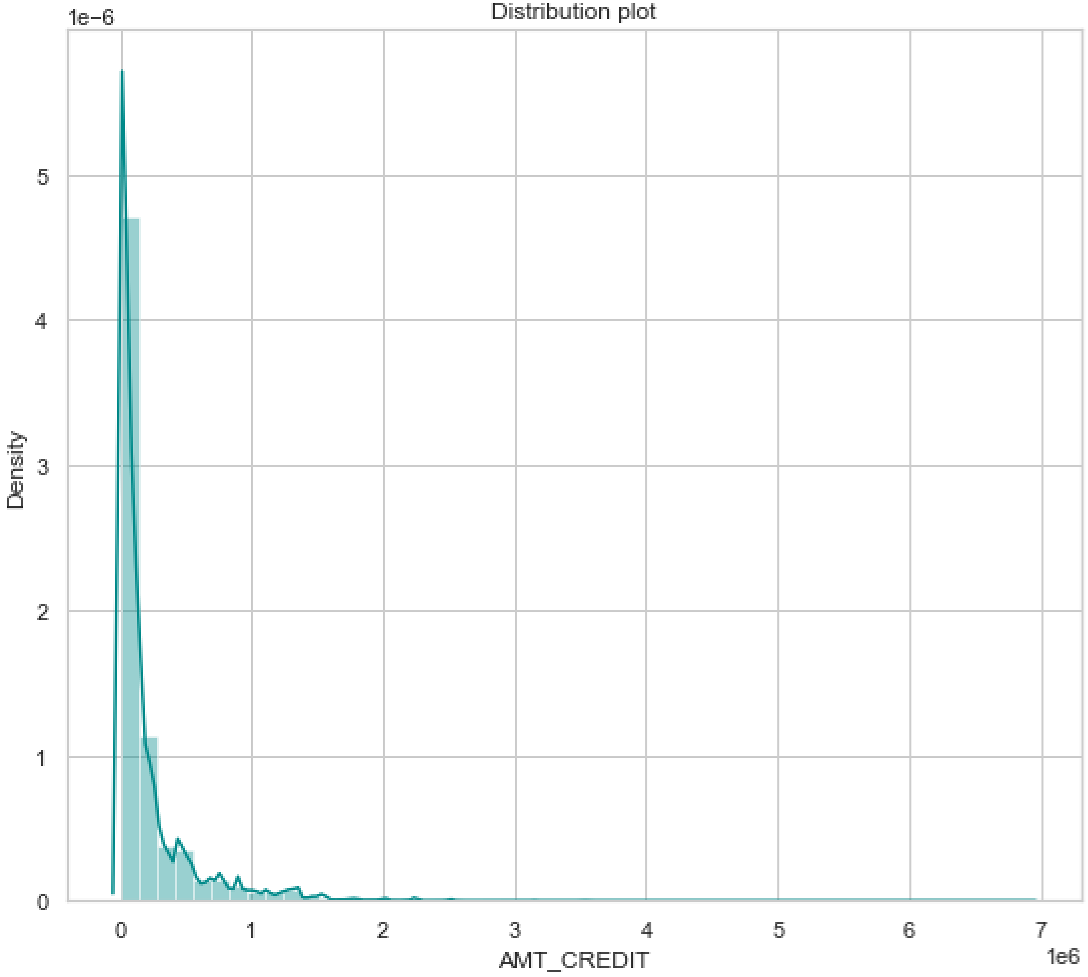
LOAN ANNUITY

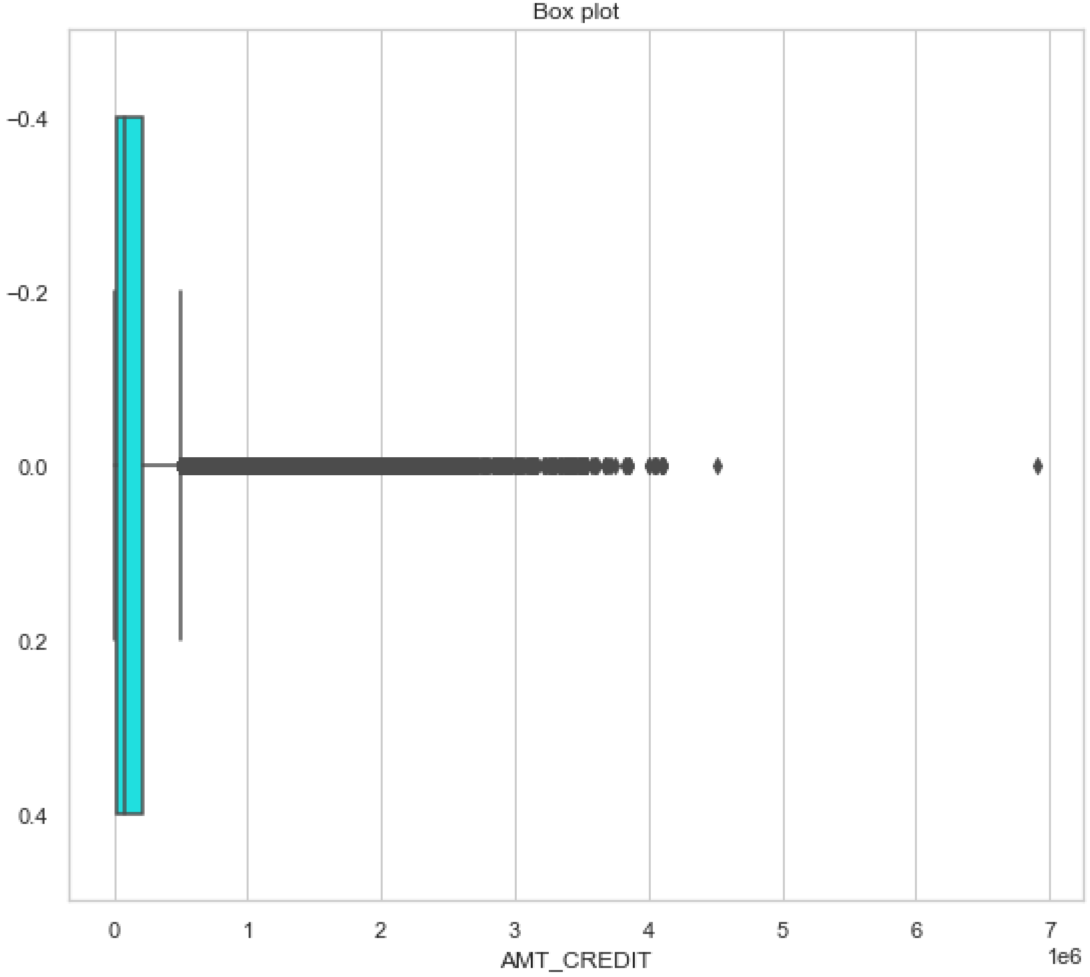




We can observe some outliers and non-normal distribution for AMT_ANNUITY data in previous application. Outliers are present in clients annuity and Q1 is smaller than Q3 for annuity distribution.

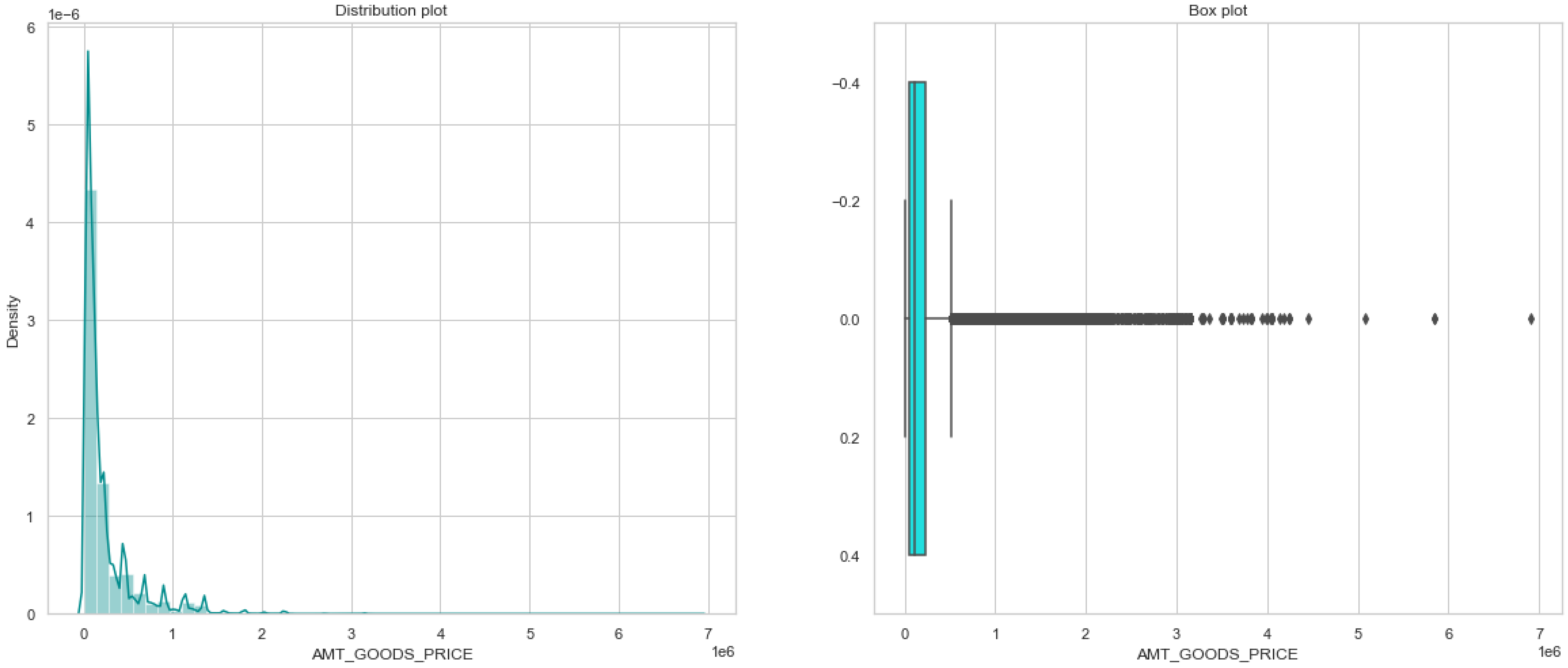
LOAN CREDIT





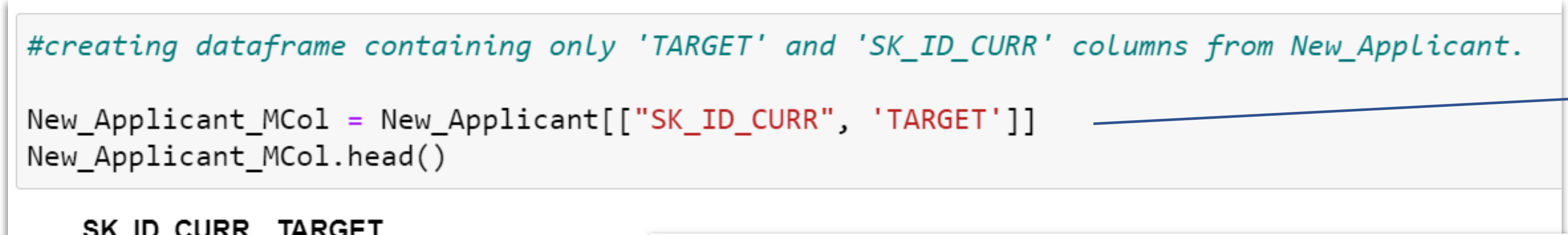
We can observe some outliers and non-normal distribution for AMT_CREDIT data in previous application. There are some outliers and Q1 is almost overlapping with lower fence and it's smaller than Q3 for credit distribution.

GOODS_PRICE



We can observe some outliers and non-normal distribution for AMT_GOODS_PRICE data in previous application. There are some outliers and Q1 is almost overlapping with lower fence and it's smaller than Q3 for Goods price distribution.

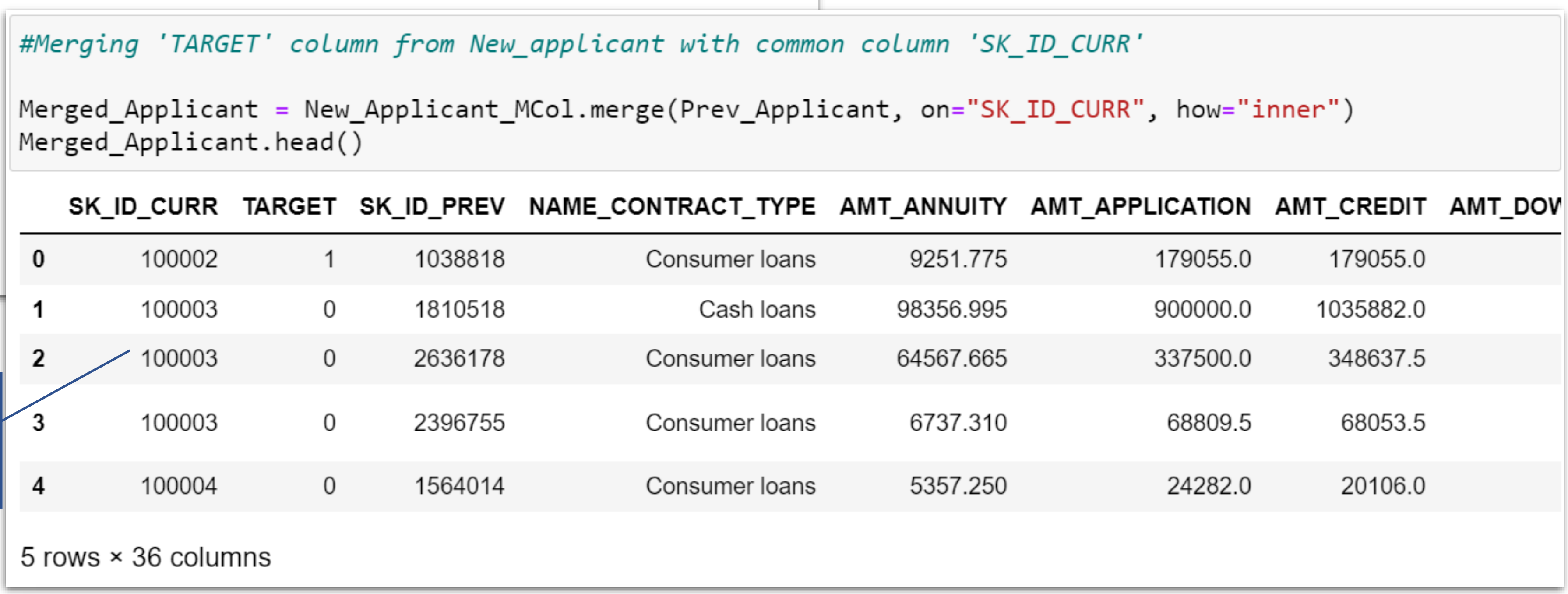
Merging Target data from New_applicant with Prev_applicant for Analysis



```
#creating dataframe containing only 'TARGET' and 'SK_ID_CURR' columns from New_Applicant.

New_Applicant_MCol = New_Applicant[["SK_ID_CURR", 'TARGET']]
New_Applicant_MCol.head()
```

	SK_ID_CURR	TARGET
0	100002	1
1	100003	0
2	100004	0
4	100007	0
5	100008	0

Created new DF for TARGET and Current applicant ID

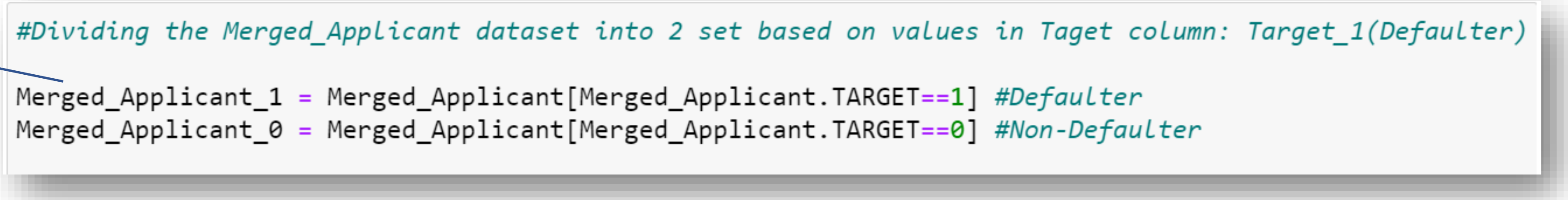


```
#Merging 'TARGET' column from New_applicant with common column 'SK_ID_CURR'

Merged_Applicant = New_Applicant_MCol.merge(Prev_Applicant, on="SK_ID_CURR", how="inner")
Merged_Applicant.head()
```

	SK_ID_CURR	TARGET	SK_ID_PREV	NAME_CONTRACT_TYPE	AMT_ANNUITY	AMT_APPLICATION	AMT_CREDIT	AMT_DOV
0	100002	1	1038818	Consumer loans	9251.775	179055.0	179055.0	
1	100003	0	1810518	Cash loans	98356.995	900000.0	1035882.0	
2	100003	0	2636178	Consumer loans	64567.665	337500.0	348637.5	
3	100003	0	2396755	Consumer loans	6737.310	68809.5	68053.5	
4	100004	0	1564014	Consumer loans	5357.250	24282.0	20106.0	

5 rows × 36 columns

New DF is merged with previous applicant data

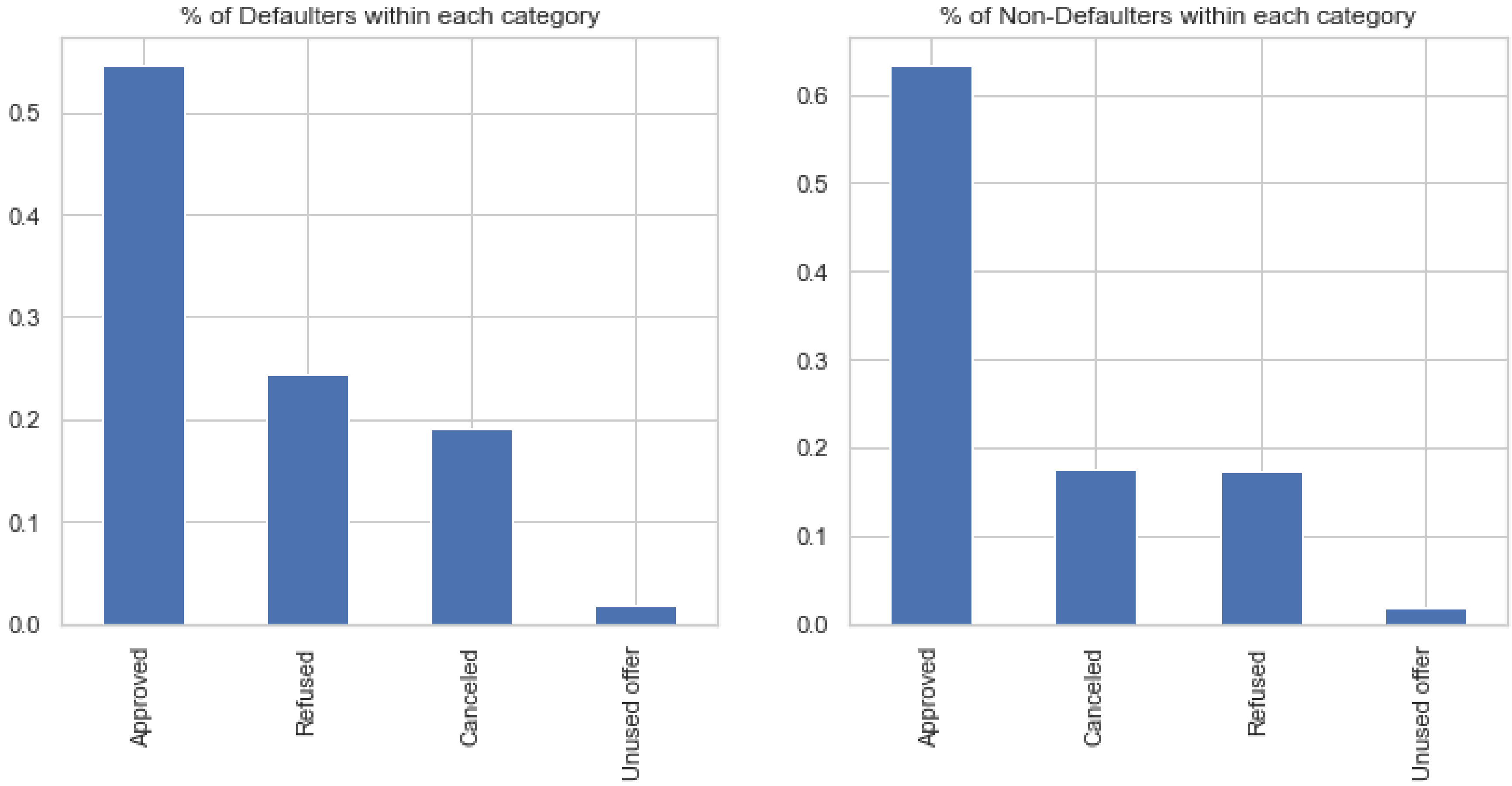


```
#Dividing the Merged_Applicant dataset into 2 set based on values in Taget column: Target_1(Defaulter)

Merged_Applicant_1 = Merged_Applicant[Merged_Applicant.TARGET==1] #Defaulter
Merged_Applicant_0 = Merged_Applicant[Merged_Applicant.TARGET==0] #Non-Defaulter
```

Merged Dataset split into 2 set based on Target 0/1

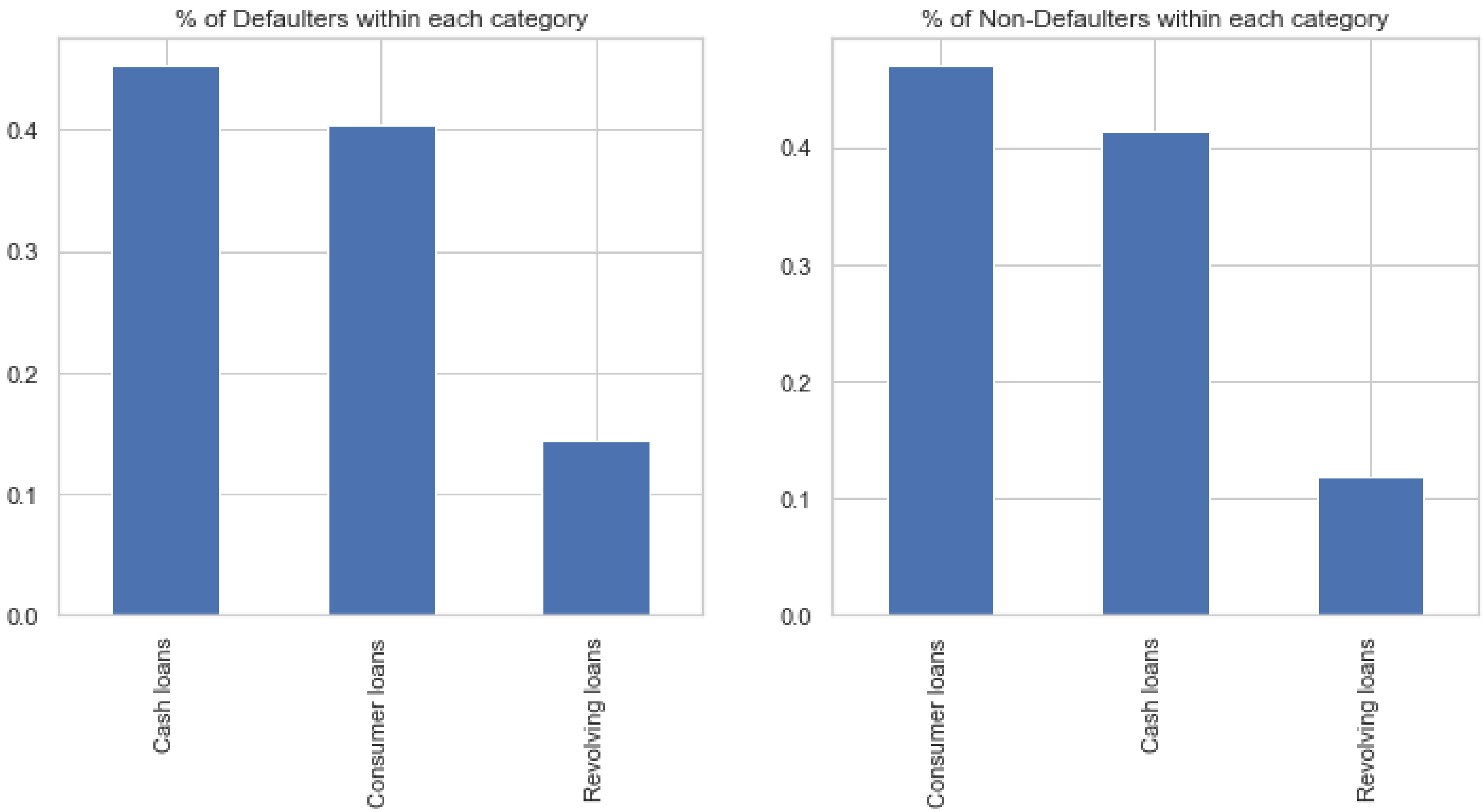
Univariant Analysis of Merged applicant data

CONTRACT STATUS



From the above graph we can observe that percentage of defaulter are more with clients who had previously refused the loan

CONTRACT TYPE



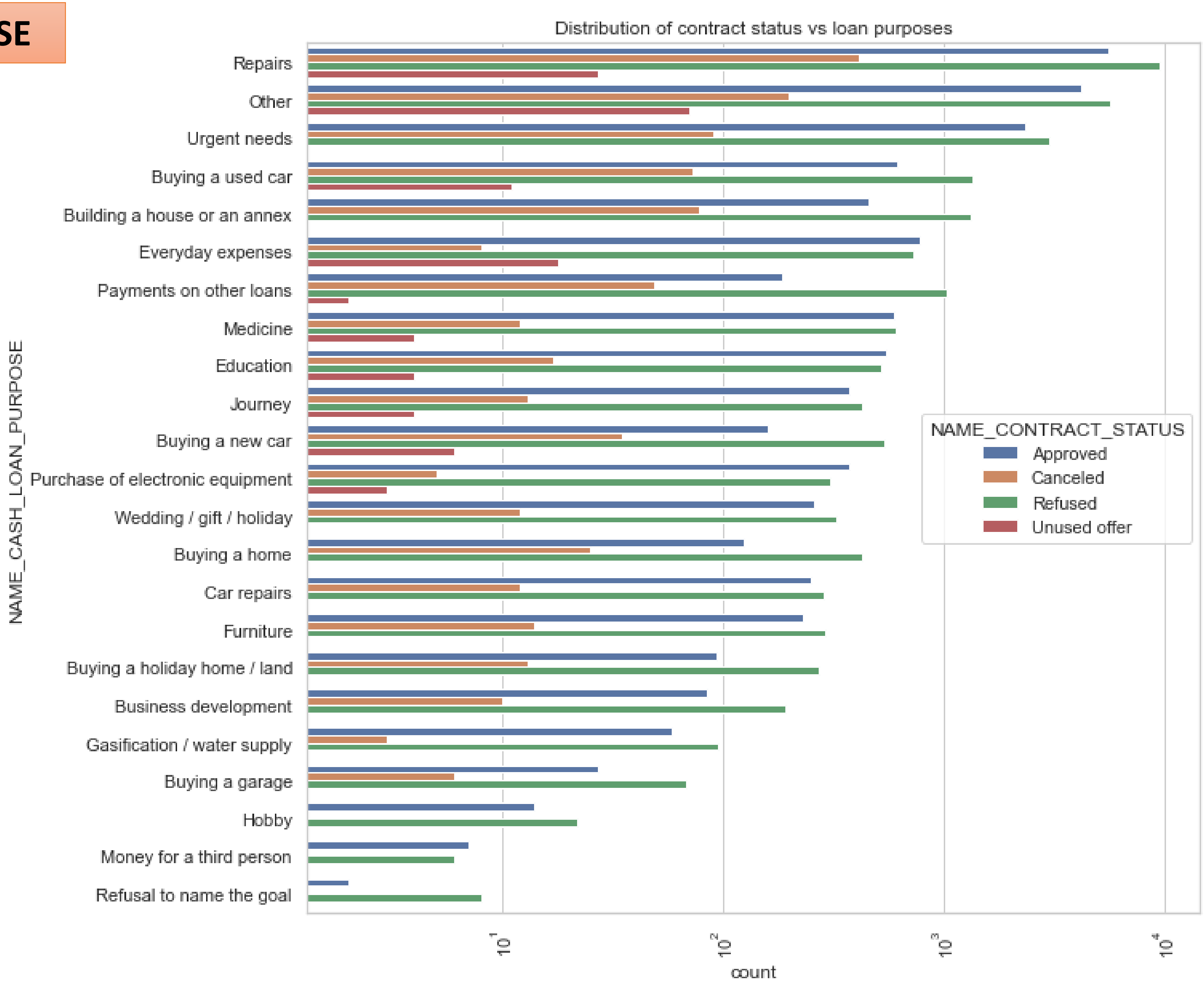
From the above graph we can observe that percentage of defaulter are more with Cash loan. Also, there is decrease % in defaulters who had consumer loan type and increased % of defaulters who had revolving loan.

Bivariant Analysis of Merged applicant data

CONTRACT STATUS vs LOAN PURPOSE

From the graph, we can observe :

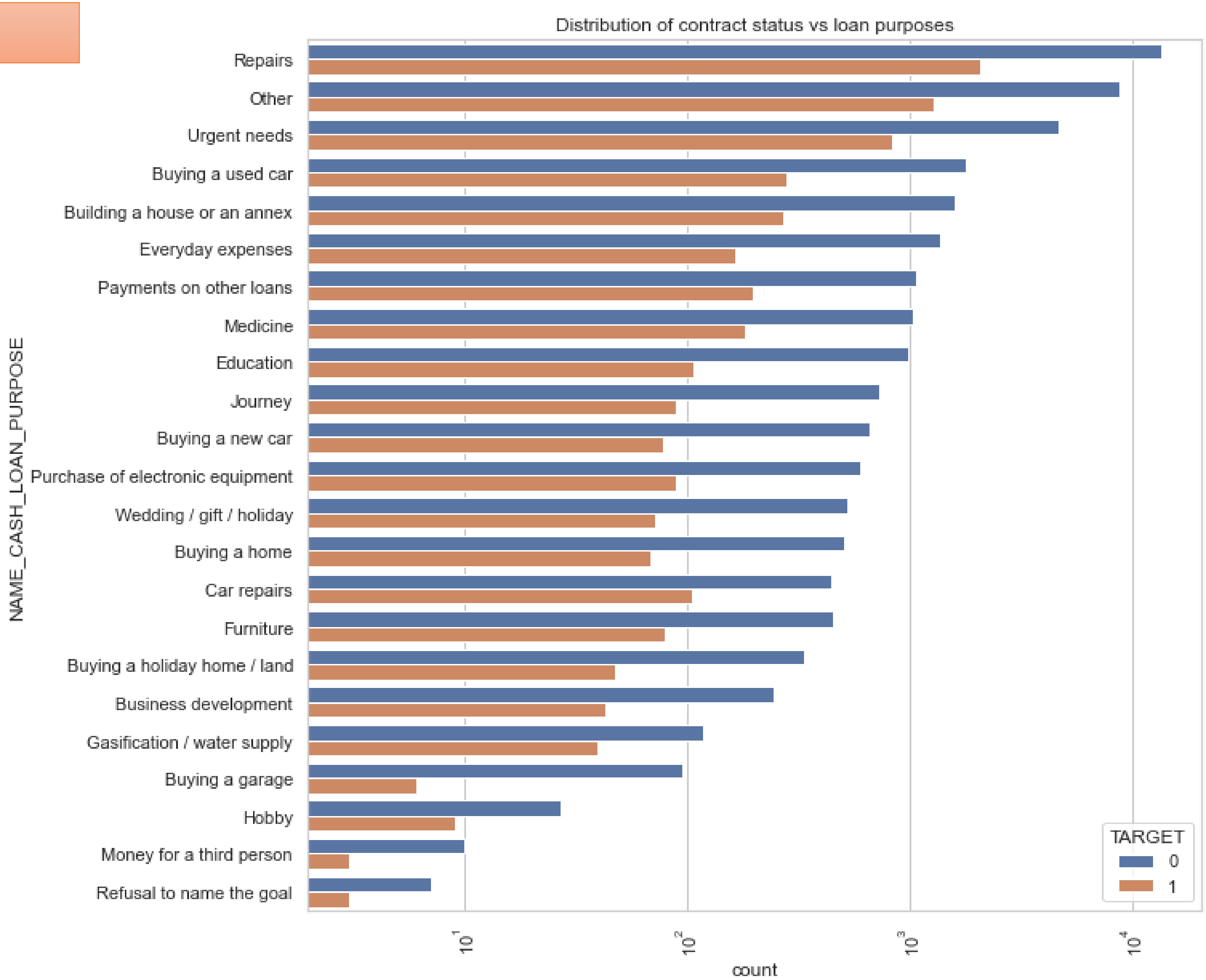
- Most of loans are rejected for repair purpose than approved.
- For 'Buying house/home', Urgent needs', 'weddings', 'Car repair'- there are no unused offer for these category but there is higher rate of refusal than approval.

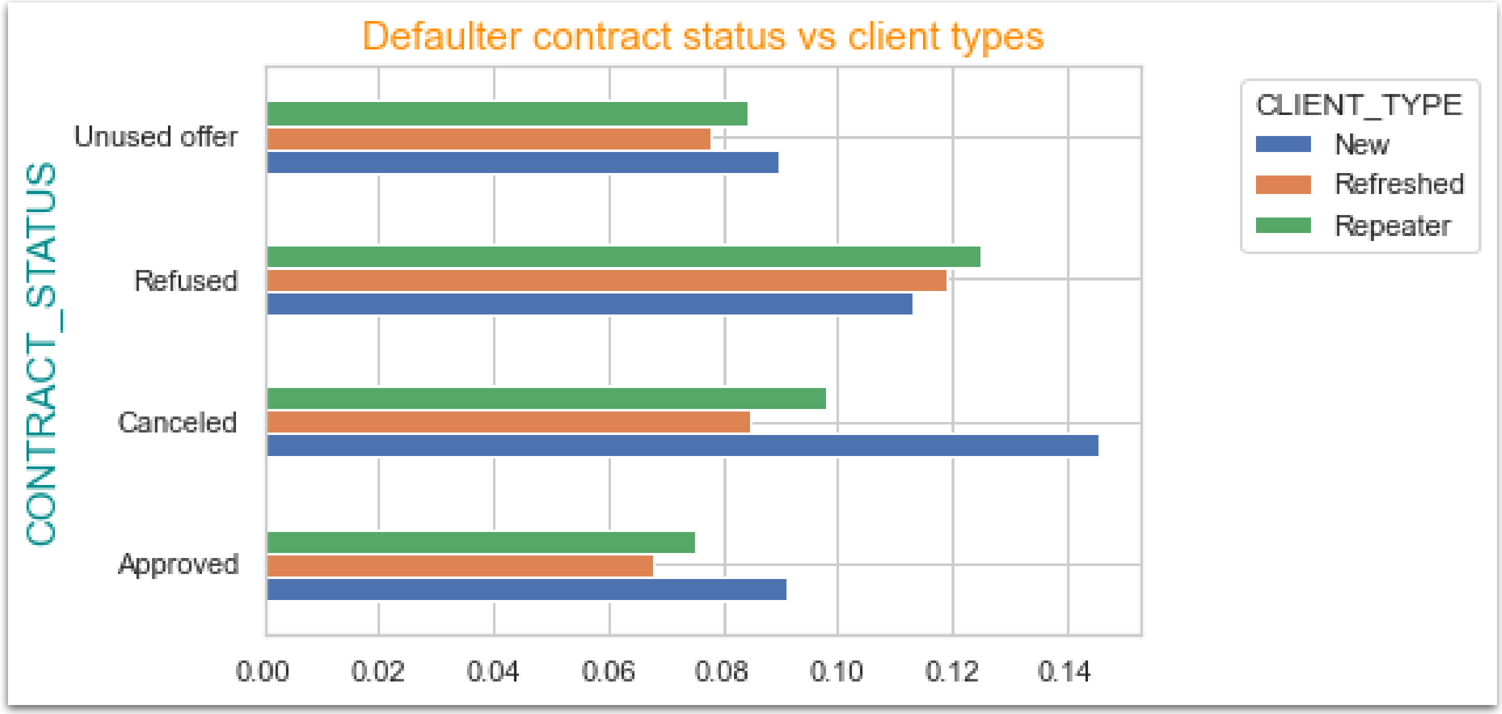


TARGET vs LOAN PURPOSE

From the graph, we can observe that:

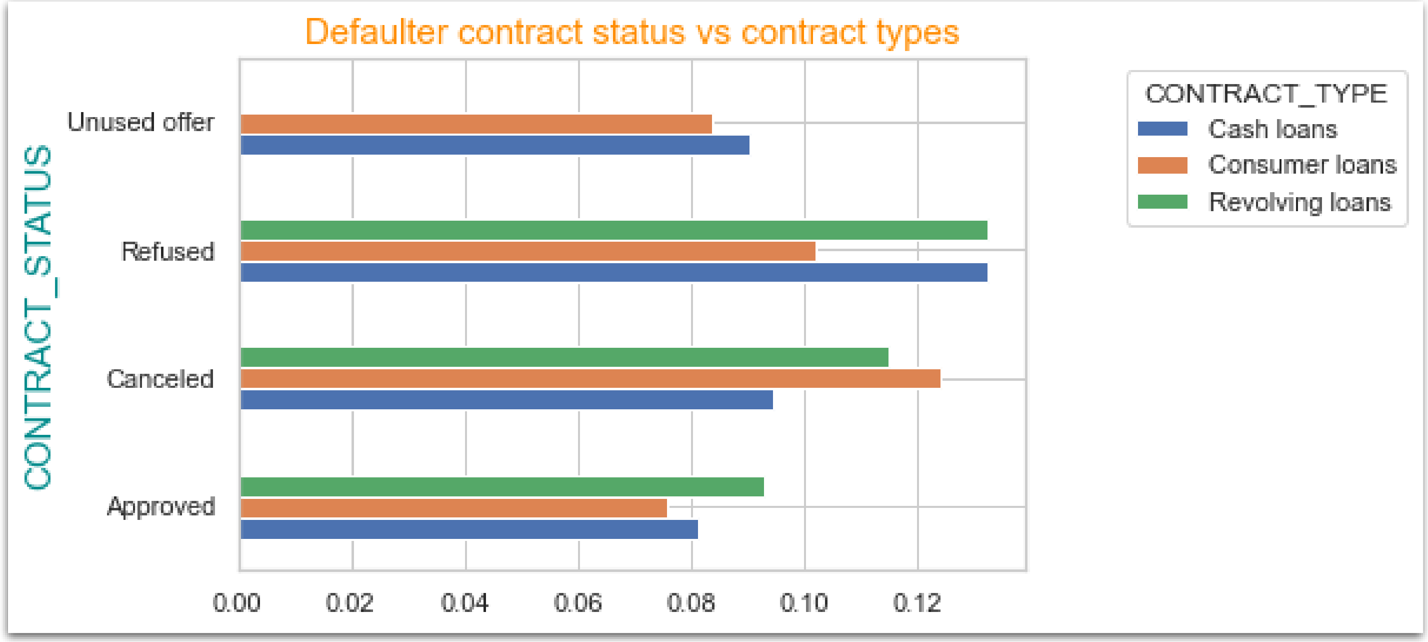
- Loans taken for 'Repair' purpose has more defaulters.
- Loans taken for 'Buying garage' has less defaulters

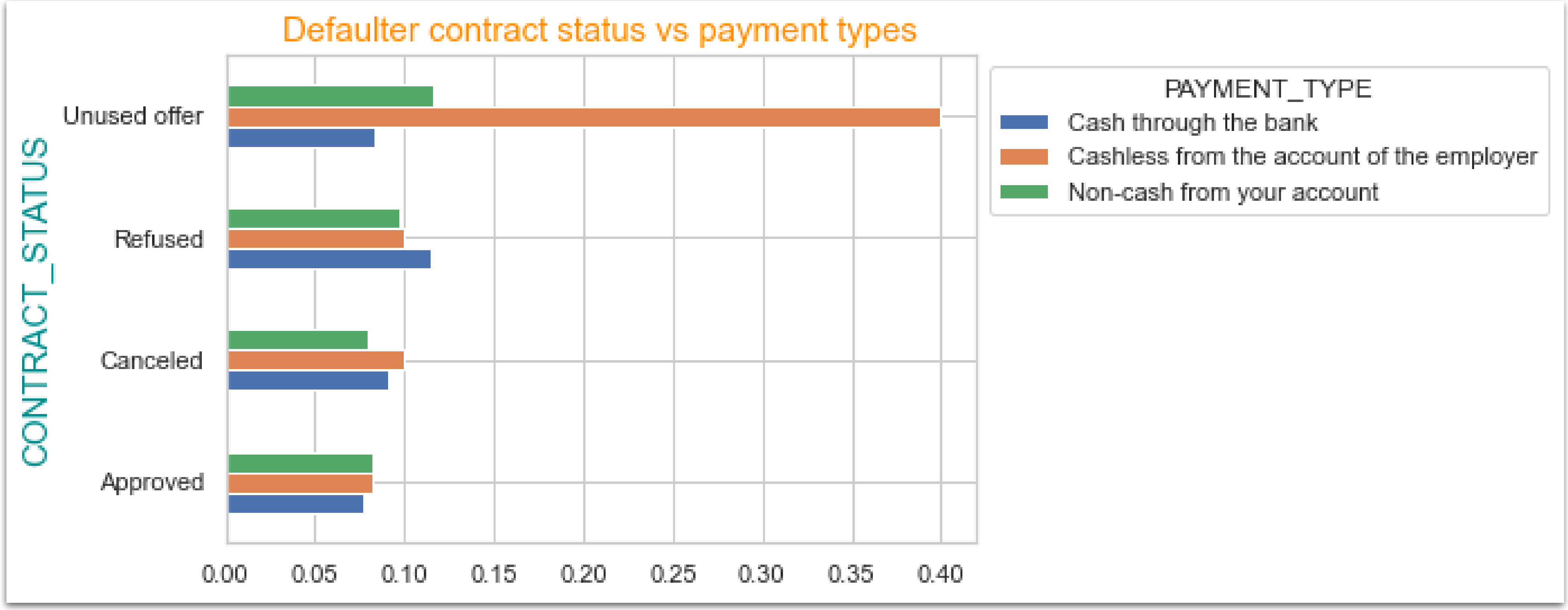




From the graph, we can observe that more defaulters are from new clients who had previously canceled the loan.

From the graph, we can observe that clients with 'Cash loan' / 'Revolving loans' who has previously refused loan tend be on the higher defaulter side.





From the above graph, we can observe that clients who chose 'Cashless from the account of employer' who has previously unused loan offer appears to be the defaulter on higher side. Less defaulters is seen with clients who chose 'Non-cash payment from account'.

Insights from analysis are:

1. Prefer Female having normal and above type of income range, as there is less percentage of difficulties observed in loan payment.
2. Senior citizens and Middle aged adults are having lesser difficulty than young adults and adults in loan payments.
3. Married applicant with lesser family members are on the lower side of being defaulter, though their loan amounts are higher than others. Applicant who are Single and those who lives with their parents have higher chance of being defaulter.
4. Prefer applicant with Higher and above education type as applicant with 'Secondary' or incomplete higher' find it difficult with loan payments. Also, majority of loans are given to 'Secondary' / 'Higher '.
5. There are no defaulter Male applicant with academic degree education. However, less number of loans provided to them. Prefer applicant with academic degree education but at less loan credit.
6. Increase loan credits to applicants - 'Students', 'Pensioner', Businessman' and females during maternity leave.
7. Working applicant get majority of loan, hence forth, more defaulters are seen here. So, lessen the loan credit amount or provide with lesser interest rates.
8. Applicants working as 'Laborers', 'Sales staff', 'Drivers' are on higher side of being defaulter. As mentioned above, prefer them over less loan credit.
9. Majority of defaulters are seen for applicants with Cash loans. So, prefer digital method of payment. Avoid 'Cashless from the account of employer', prefer 'Non-cash payment from account' type of loan.
10. Applicant who previously refused/cancelled loan find it difficulty with loan payments.
11. Majority of loans is taken for 'Repair' purpose and defaulters rate is also for the same. However, loans credited for 'Buying garage' has less defaulters.
12. 50% of applicants have rejected the loan due to HC and 18% due to limit:
    - Most of loans are rejected for repair purpose than approved.
    - No unused offer 'Buying house/home', Urgent needs', 'weddings', 'Car repair' purposes.