Credit EDA Case Study

SUBMITTED BY : ARPITA MISHRA
DARSHANA AWALGAONKAR

Table Of Content

Introduction

- Loan providing companies often find it difficult to access the loan repayment likelihood of a customer. This is primarily due to either unavailability of credit history or due to insufficient data
- The Company decides on the loan application based on applicant's profile, considering few risks such as :
  - Not approving loan of a potential customer with high likelihood to repay the loan
  - Approving the loan of a customer who is more likely to default
- This case study aims to solve such problems using the method of Exploratory Data Analysis

Business Objective

- Aim of this case study is to identify patterns indicating likelihood of a customer category defaulting on the loan based on various categories and demographics
- To identify driving factors behind loan default

Data Understanding

The analysis has been done using two datasets :

- ✓ **Application Data -** Containing information about the customer at the time of application for loan
- ✓ **Previous Application Data –** Containing information about customer's previous loan application, and decision of company on that application
- ✓ A metadata was also provided for deriving meaning of various variables used in the datasets

Importing the required libraries for EDA

- For reading data and performing EDA operations, we have primarily use the NumPy and pandas Python packages.
- For Data Visualization Seaborn package is used which is a Python-based data visualization library built on Matplotlib

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

#import the warnings.
import warnings
warnings.filterwarnings('ignore')
```

Data Cleaning And Manipulation

- Approach used :
  - Identify the data types
  - Identify the missing values by calculating percentage of null values in each variable

```
application_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Columns: 122 entries, SK_ID_CURR to AMT_REQ_CREDIT_BUREAU_YEAR
dtypes: float64(65), int64(41), object(16)
memory usage: 286.2+ MB
```

```
application_data.dtypes
```

```
SK_ID_CURR                      int64
TARGET                          int64
NAME_CONTRACT_TYPE             object
CODE_GENDER                    object
FLAG_OWN_CAR                   object
FLAG_OWN_REALTY                object
CNT_CHILDREN                    int64
AMT_INCOME_TOTAL              float64
AMT_CREDIT                    float64
AMT_ANNUITY                   float64
AMT_GOODS_PRICE               float64
NAME_TYPE_SUITE                object
NAME_INCOME_TYPE               object
NAME_EDUCATION_TYPE            object
NAME_FAMILY_STATUS             object
NAME_HOUSING_TYPE              object
REGION_POPULATION_RELATIVE    float64
DAYS_BIRTH                      int64
DAYS_EMPLOYED                   int64
```

```
application_data.isnull().mean() * 100
```

```
SK_ID_CURR                    0.000000
TARGET                        0.000000
NAME_CONTRACT_TYPE            0.000000
CODE_GENDER                   0.000000
FLAG_OWN_CAR                  0.000000
FLAG_OWN_REALTY               0.000000
CNT_CHILDREN                  0.000000
AMT_INCOME_TOTAL              0.000000
AMT_CREDIT                    0.000000
AMT_ANNUITY                   0.003902
AMT_GOODS_PRICE               0.090403
NAME_TYPE_SUITE               0.420148
NAME_INCOME_TYPE              0.000000
NAME_EDUCATION_TYPE           0.000000
NAME_FAMILY_STATUS            0.000000
NAME_HOUSING_TYPE             0.000000
REGION_POPULATION_RELATIVE    0.000000
DAYS_BIRTH                    0.000000
DAYS_EMPLOYED                 0.000000
```

Data Cleaning And Manipulation

- Dropped all columns which had more than 50% missing value
- Analyzed the dataset by using variable definition in data dictionary to identify columns which will not add value to the analysis process for the given business objective, and removing all such columns

We have observed there were many missing values in this data frame hence dropped all columns from data frame for which missing values % is more than 50%. ¶

```
columns = application_data.columns
percent_missing = application_data.isnull().mean() * 100
missing_value_df = pd.DataFrame({'column_name': columns,
                                 'percent_missing': percent_missing})
missing_drop = list(missing_value_df[missing_value_df.percent_missing>50].column_name)
application_data.drop(missing_drop, axis=1,inplace=True)
```

```
application_data_not_req_columns=['FLAG_MOBIL','FLAG_EMP_PHONE','FLAG_WORK_PHONE','FLAG_CONT_MOBILE','FLAG_PHONE','FLAG_EMAIL',
                                  'REGION_RATING_CLIENT','REGION_RATING_CLIENT_W_CITY','FLAG_EMAIL','CNT_FAM_MEMBERS',
                                  'REGION_RATING_CLIENT', 'REGION_RATING_CLIENT_W_CITY','DAYS_LAST_PHONE_CHANGE',
                                  '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','NAME_TYPE_SUITE','EXT_SOURCE_2',
                                  'EXT_SOURCE_3','REG_REGION_NOT_LIVE_REGION','REG_REGION_NOT_WORK_REGION',
                                  'LIVE_REGION_NOT_WORK_REGION','REG_CITY_NOT_LIVE_CITY','REG_CITY_NOT_WORK_CITY',
                                  'LIVE_CITY_NOT_WORK_CITY','WEEKDAY_APPR_PROCESS_START','HOUR_APPR_PROCESS_START',
                                  'YEARS_BEGINEXPLUATATION_AVG','FLOORSMAX_AVG','YEARS_BEGINEXPLUATATION_MODE',
                                  'FLOORSMAX_MODE','YEARS_BEGINEXPLUATATION_MEDI','FLOORSMAX_MEDI','TOTALAREA_MODE',
                                  'EMERGENCYSTATE_MODE','OBS_30_CNT_SOCIAL_CIRCLE','DEF_30_CNT_SOCIAL_CIRCLE',
                                  'OBS_60_CNT_SOCIAL_CIRCLE','DEF_60_CNT_SOCIAL_CIRCLE','AMT_REQ_CREDIT_BUREAU_HOUR',
                                  'AMT_REQ_CREDIT_BUREAU_DAY', 'AMT_REQ_CREDIT_BUREAU_WEEK','AMT_REQ_CREDIT_BUREAU_MON',
                                  'AMT_REQ_CREDIT_BUREAU_QRT','AMT_REQ_CREDIT_BUREAU_YEAR','OCCUPATION_TYPE']

application_data.drop(labels=application_data_not_req_columns,axis=1,inplace=True)
```

Data Cleaning And Manipulation

- Analyzed few categorical columns where not available was represented as XNA. So, manipulated those values to best aid the analysis

There are some categorical columns where values are mentioned as 'XNA' corresponding to Not Available. Lets analyse those columns!!

```
application_data[application_data["CODE_GENDER"]=="XNA"].shape
```

```
(4, 21)
```

```
application_data[application_data["ORGANIZATION_TYPE"]=="XNA"].shape
```

```
(55374, 21)
```

So CODE_GENDER column has 4 such rows, and ORGANIZATION_TYPE column has 55374 such rows. Lets fix this !!

```
application_data['CODE_GENDER'].value_counts()
```

```
F      202448
M      105059
XNA         4
Name: CODE_GENDER, dtype: int64
```

Since majority of the column values are females, it is safe to assign XNA as F

```
application_data.loc[application_data['CODE_GENDER']=='XNA','CODE_GENDER']='F'
application_data['CODE_GENDER'].value_counts()
```

```
F    202452
M    105059
Name: CODE_GENDER, dtype: int64
```

```
# Describing data for ORGANIZATION_TYPE column
application_data['ORGANIZATION_TYPE'].describe()
```

```
count                     307511
unique                        58
top       Business Entity Type 3
freq                       67992
Name: ORGANIZATION_TYPE, dtype: object
```

As out of 307511 values in ORGANIZATION_TYPE column 55374 values are XNA which is ~18%, dropping these records will not have much impact on dataset.So dropping all such records

```
application_data=application_data.drop(application_data.loc[application_data['ORGANIZATION_TYPE']=='XNA'].index)
application_data[application_data['ORGANIZATION_TYPE']=='XNA'].shape
```

```
(0, 21)
```

Data Cleaning And Manipulation

- Applied bucketing technique to few continuous variables such as AMT_INCOME_TOTAL and AMT_CREDIT for better analysis

Bucketing Continuous variables AMT_INCOME_TOTAL and AMT_CREDIT for better analysis

```python
# Creating buckets for income amount

buckets = [0,50000,100000,150000,200000,250000,300000,350000,400000,450000,500000,10000000000]
INCOME_RANGE = ['0-50000','50000-100000','100000-150000', '150000-200000','200000-250000','250000-300000',
                '300000-350000','350000-400000','400000-450000','450000-500000','500000 and above']

application_data['INCOME_RANGE']=pd.cut(application_data['AMT_INCOME_TOTAL'],buckets,labels=INCOME_RANGE)
```

```python
application_data.head(10)
```

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT
0	100002	1	Cash loans	M	N	Y	0	202500.0	
1	100003	0	Cash loans	F	N	N	0	270000.0	
2	100004	0	Revolving loans	M	Y	Y	0	67500.0	
3	100006	0	Cash loans	F	N	Y	0	135000.0	
4	100007	0	Cash loans	M	N	Y	0	121500.0	
5	100008	0	Cash loans	M	N	Y	0	99000.0	
6	100009	0	Cash loans	F	Y	Y	1	171000.0	
7	100010	0	Cash loans	M	Y	Y	0	360000.0	

```python
# Creating buckets for Credit amount

buckets = [0,150000,300000,400000,500000,600000,700000,800000,900000,1000000000]
CREDIT_RANGE = ['0-150000', '150000-300000','300000-400000','400000-500000','500000-600000','600000-700000','700000-800000',
                '800000-900000','900000 and above']

application_data['CREDIT_RANGE']=pd.cut(application_data['AMT_CREDIT'],buckets,labels=CREDIT_RANGE)
```

```python
application_data.head(10)
```

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT
0	100002	1	Cash loans	M	N	Y	0	202500.0	
1	100003	0	Cash loans	F	N	N	0	270000.0	
2	100004	0	Revolving loans	M	Y	Y	0	67500.0	
3	100006	0	Cash loans	F	N	Y	0	135000.0	
4	100007	0	Cash loans	M	N	Y	0	121500.0	
5	100008	0	Cash loans	M	N	Y	0	99000.0	
6	100009	0	Cash loans	F	Y	Y	1	171000.0	

Data Cleaning And Manipulation

- Converted "DAYS_BIRTH" column into age by defining a custom function and applied bucketing on the same

- Analyzed variable "NAME_FAMILY_STATU" and replaced Unknown with the most common value "Married"

Converting DAYS_BIRTH column to age in years and bucketing the same

```
def age_in_years(x):
    age  = int((x * -1) / 365)
    return age

application_data['AGE'] = application_data['DAYS_BIRTH'].apply(lambda x : age_in_years(x))
#facts['pop2050'] = facts.apply(lambda x: final_pop(x), axis=1)
```

```
# Creating buckets for Age
#Assuming no applicant is > 100 yrs of age
buckets = [0,20,30,40,50,60,100]
AGE_RANGE = ['0-20', '20-30','30-40', '40-50', '50-60', '60 and above']

application_data['AGE_RANGE']=pd.cut(application_data['AGE'],buckets,labels=AGE_RANGE)
```

```
application_data[["INCOME_RANGE","CREDIT_RANGE","AGE","AGE_RANGE"]]
```

	INCOME_RANGE	CREDIT_RANGE	AGE	AGE_RANGE
0	200000-250000	400000-500000	25	20-30
1	250000-300000	900000 and above	45	40-50
2	50000-100000	0-150000	52	50-60
3	100000-150000	300000-400000	52	50-60
4	100000-150000	500000-600000	54	50-60
...	...	...	...	...
307504	150000-200000	300000-400000	45	40-50
307506	150000-200000	150000-300000	25	20-30
307508	150000-200000	600000-700000	41	40-50
307509	150000-200000	300000-400000	32	30-40
307510	150000-200000	600000-700000	46	40-50

Analysing column "NAME_FAMILY_STATUS"

```
application_data["NAME_FAMILY_STATUS"].unique()
```

```
array(['Single / not married', 'Married', 'Civil marriage', 'Widow',
       'Separated'], dtype=object)
```

As there are Unknown values in NAME_FAMILY_STATUS getting their count

```
application_data['NAME_FAMILY_STATUS'].value_counts()
```

```
Married                 163916
Single / not married     39316
Civil marriage           26197
Separated                16000
Widow                     6708
Name: NAME_FAMILY_STATUS, dtype: int64
```

As there are only two rows with NAME_FAMILY_STATUS = Unknown, replacing it with most common value of Married

```
application_data.loc[application_data['NAME_FAMILY_STATUS']=='Unknown','NAME_FAMILY_STATUS']='Married'
application_data['NAME_FAMILY_STATUS'].value_counts()
```

```
Married                 163916
Single / not married     39316
Civil marriage           26197
Separated                16000
Widow                     6708
Name: NAME_FAMILY_STATUS, dtype: int64
```

Data Cleaning And Manipulation

- Analyzed column "CNT_CHILDREN" and bucketed the same
- Analyzed variable "Organization Type" and merged common organization type labels as one

Analysing and bucketing column CNT_CHILDREN

```
application_data["CNT_CHILDREN"].value_counts()
```

```
0     161911
1      59698
2      26365
3       3629
4        414
5         81
6         19
7          7
19         2
14         2
12         2
10         2
9          2
8          2
11         1
Name: CNT_CHILDREN, dtype: int64
```

As there are few records where CNT_CHILDREN is >=10 , bucketing this column further

```
# Creating buckets for CNT_CHILDREN
#Assuming CNT_CHILDREN > 7 is extreme values
buckets = [0,1,3,5,7,20]
COUNT_RANGE = ['0-3','3-5','5-7','7-10','10 and above']

application_data['CHILDREN_COUNT_RANGE']=pd.cut(application_data['CNT_CHILDREN'],5,labels=COUNT_RANGE)
```

```
application_data['CHILDREN_COUNT_RANGE'].value_counts()
```

```
0-3             251603
3-5                521
5-7                  7
7-10                 4
10 and above         2
Name: CHILDREN_COUNT_RANGE, dtype: int64
```

Analysing column "ORGANIZATION_TYPE" and merging common Organization Type labels ¶

```
org_type = {'Business Entity Type 1':'Business Entity','Business Entity Type 2':'Business Entity',
            'Business Entity Type 3':'Business Entity','Industry: type 1':'Industry','Industry: type 2':'Industry',
            'Industry: type 3':'Industry','Industry: type 4':'Industry','Industry: type 5':'Industry',
            'Industry: type 6':'Industry','Industry: type 7':'Industry','Industry: type 8':'Industry',
            'Industry: type 9':'Industry','Industry: type 10':'Industry','Industry: type 11':'Industry',
            'Industry: type 12':'Industry','Industry: type 13':'Industry','Trade: type 1':'Trade','Trade: type 2':'Trade',
            'Trade: type 3':'Trade','Trade: type 4':'Trade','Trade: type 5':'Trade','Trade: type 6':'Trade',
            'Trade: type 7':'Trade','Transport: type 1':'Transport','Transport: type 2':'Transport',
            'Transport: type 3':'Transport','Transport: type 4':'Transport'
           }

application_data["ORGANIZATION_TYPE"]=application_data["ORGANIZATION_TYPE"].replace(org_type)
```

Data Analysis

- Divide the dataset into two datasets one for each Target=1 and Target=0 and calculated the imbalance ratio

Divide the dataframes into two datasets for Target=1 and Target=0 and calculate Imbalance ratio between Target=1 and Target=0

```
application_data_target0=application_data.loc[application_data["TARGET"]==0]
application_data_target1=application_data.loc[application_data["TARGET"]==1]
```

```
application_data_target0.shape
```

```
(230302, 26)
```

```
application_data_target1.shape
```

```
(21835, 26)
```

```
Imbalance_ratio = round(len(application_data_target0)/len(application_data_target1),2)
Imbalance_ratio
```

```
10.55
```

Data Analysis

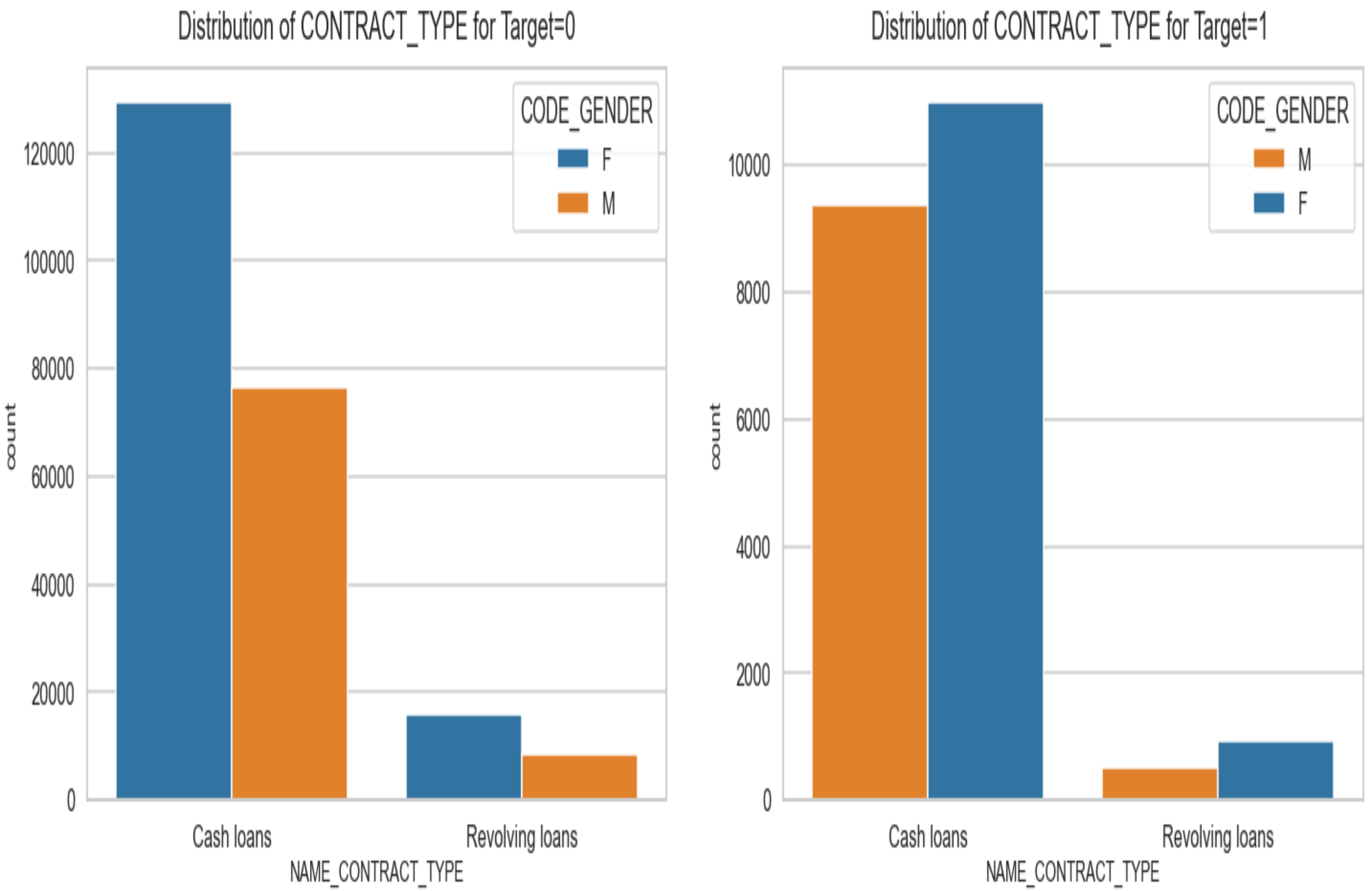
Categorical ordered Univariate Analysis

Distribution of CONTRACT TYPE For Male and Female Applicants

Inferences:

(True for both Target=0 and Target=1)

- Cash Loans are more in popular than Revolving Loans for both Male and Female Applicants
- There are more applications from Female applicants as compared to Males

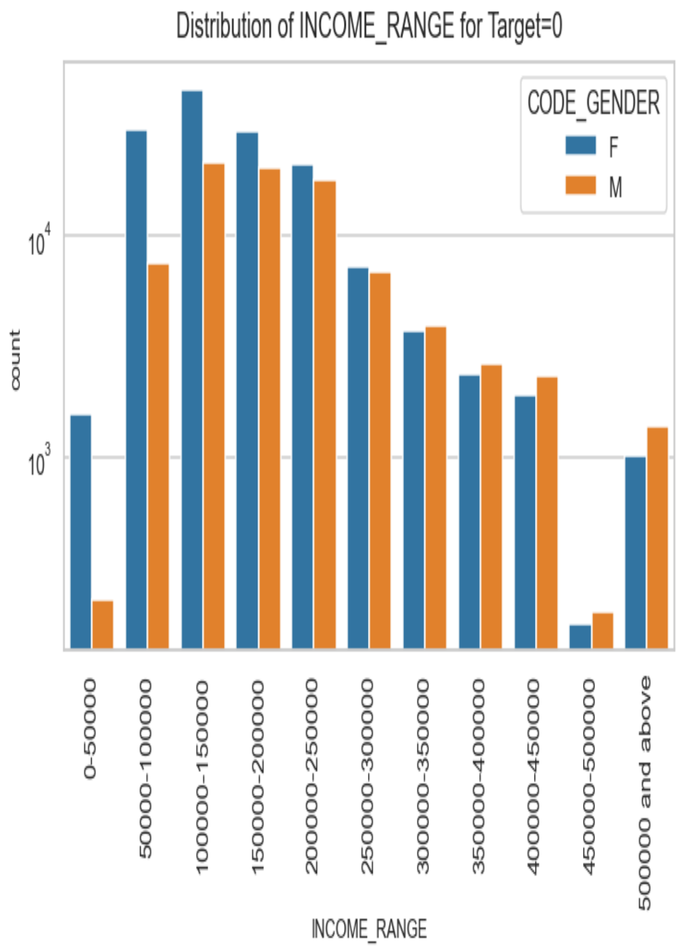


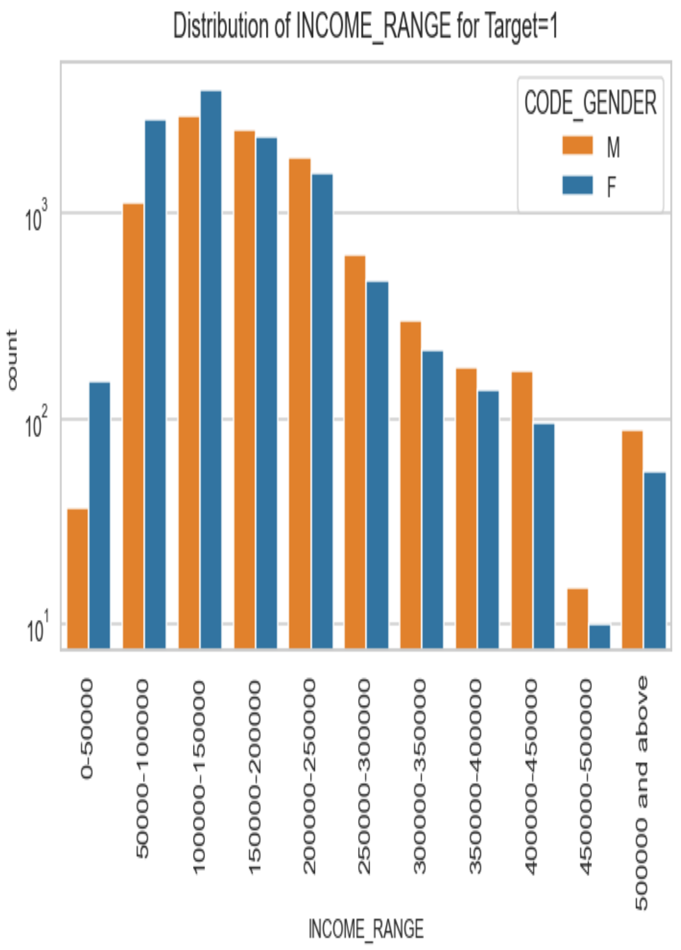
Data Analysis
Categorical ordered Univariate Analysis

Distribution of Income Range For Male and Female Applicants

Inferences:

- There are more Females as compared to males in income range of 0-300000 for Target=0, and for income > 300000 count of males in that income range is more
- For Target=1, there are more Females as compared to males in income range of 0-150000, and for income > 150000 count of males is more
- Most of the applicant's annual income is in range of 100000 to 150000 for both Target=0 and Target=1
- There are very few people who have income in range of 450000-500000 for both Target=0 and Target=1



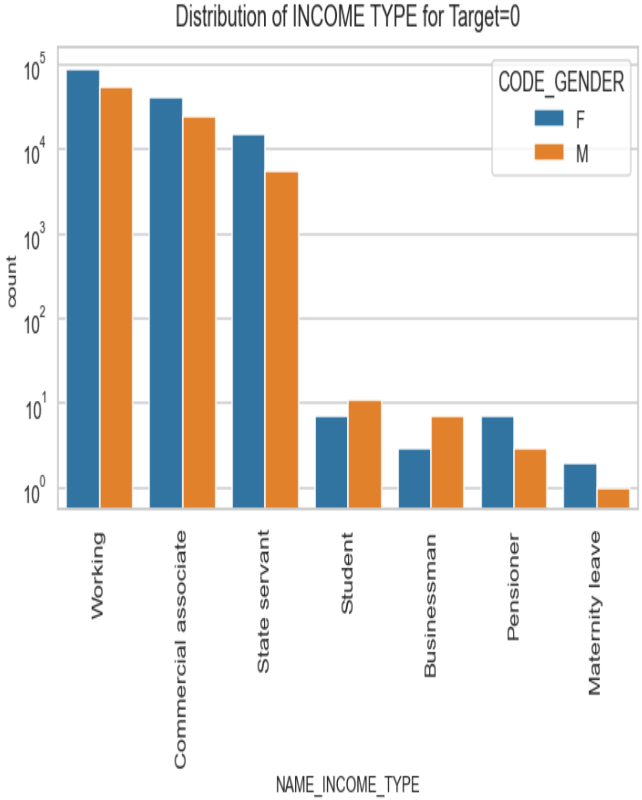


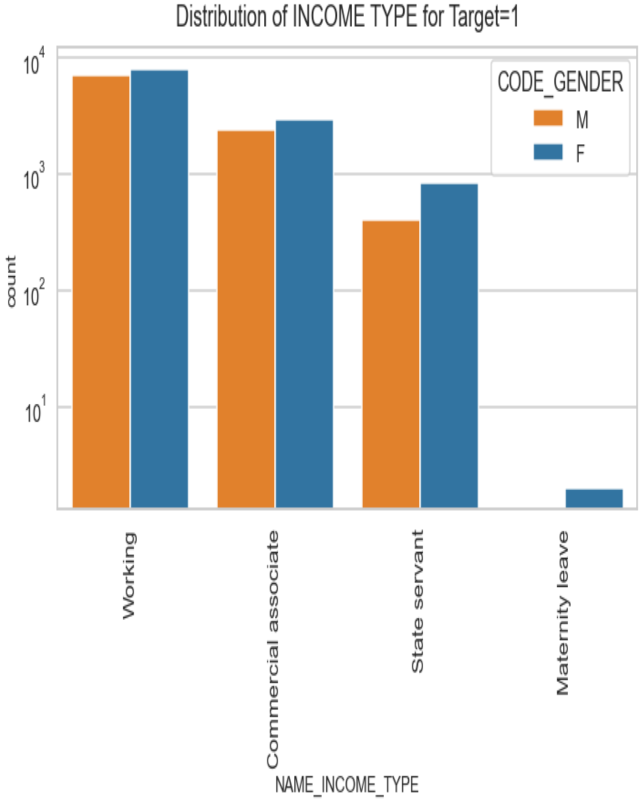
Data Analysis
Categorical ordered Univariate Analysis

Distribution of Income Type For Male and Female Applicants

Inferences:

- Top 3 credit categories for Target=0 and Target=1 are Working, Commercial Associate and State Servant
- None of the Student, Businessman or Pensioner have payment difficulties
- People having Income Type as Student, Businessman, Pensioners or people on Maternity Leave have not applied for more number of loans



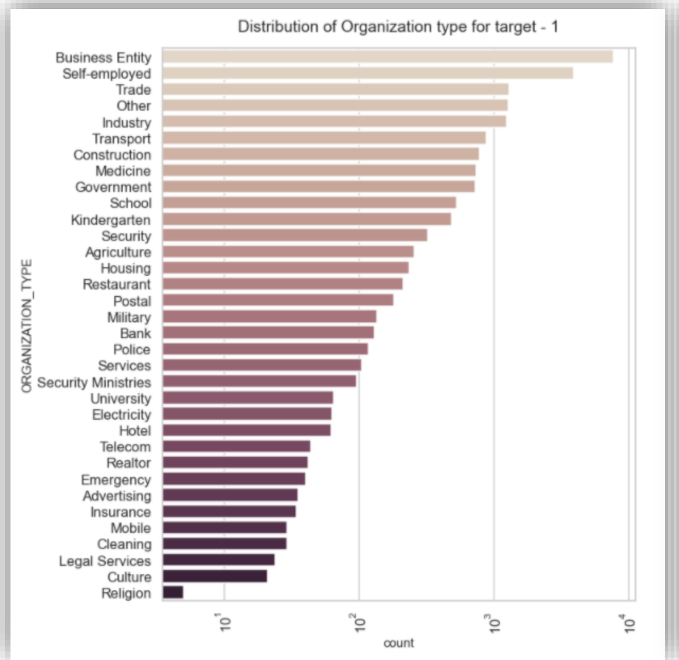


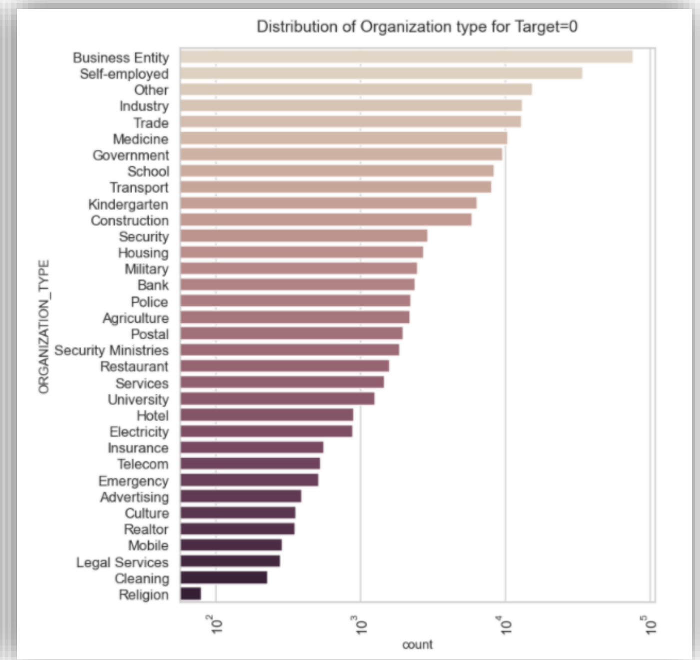
Data Analysis
Categorical unordered Univariate Analysis

Distribution of Organization Type

Inferences:

- **Businessman**, **Self-Employed** people, and people involved in **Trade** have highest count among applicants who are likely to default (Target=1)
- Number of applications for people with organization type Business-Entity, Trade, Industry, people who are Self-Employed is highest



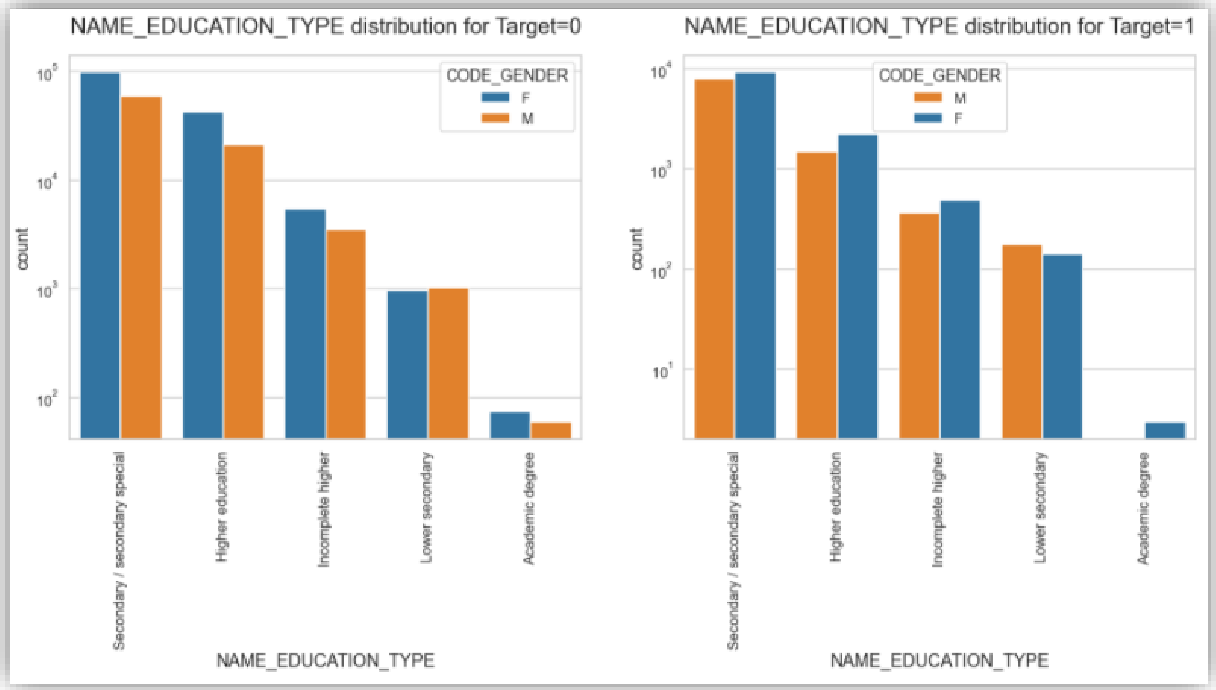


Data Analysis
Categorical ordered Univariate Analysis

Distribution of Education Type

Inferences:

- Maximum number of loan applications are being made by people having Secondary/Secondary Special education
- People having only an academic degree have applied for least number of loans

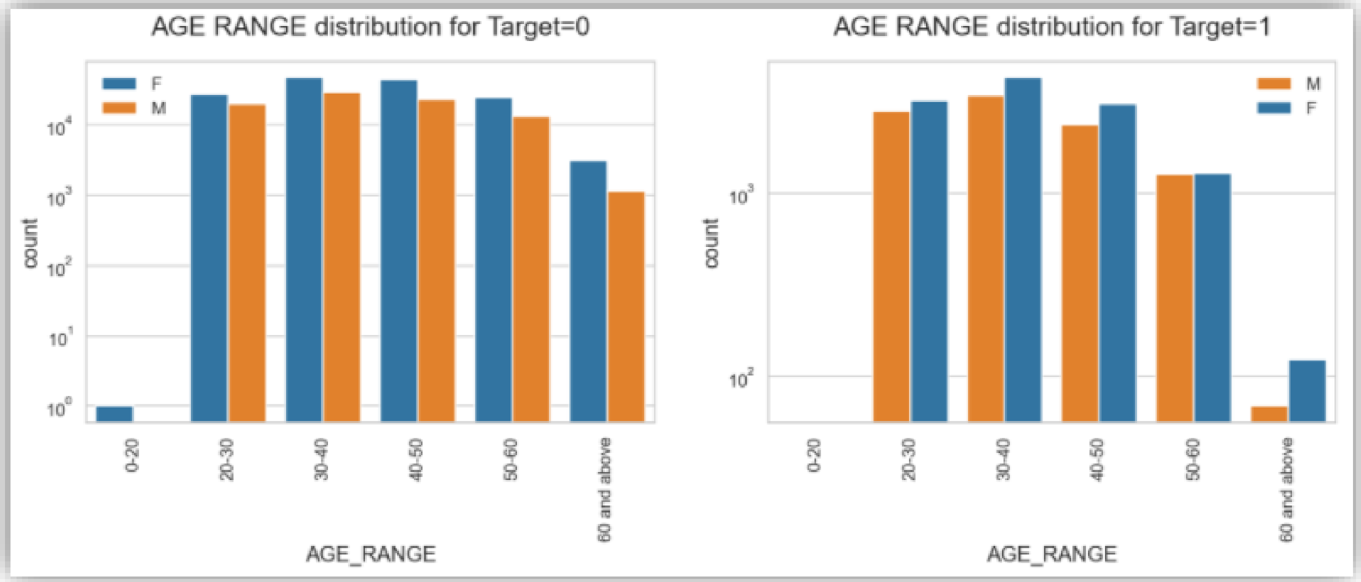


Data Analysis
Categorical ordered Univariate Analysis

Distribution of Age Range

Inferences:

- Most of the people who are likely to default are in age group 30-40
- People in age group 0-20 have least number of loan applications and zero likelihood of default
- People above 60 years of age are very unlikely to default

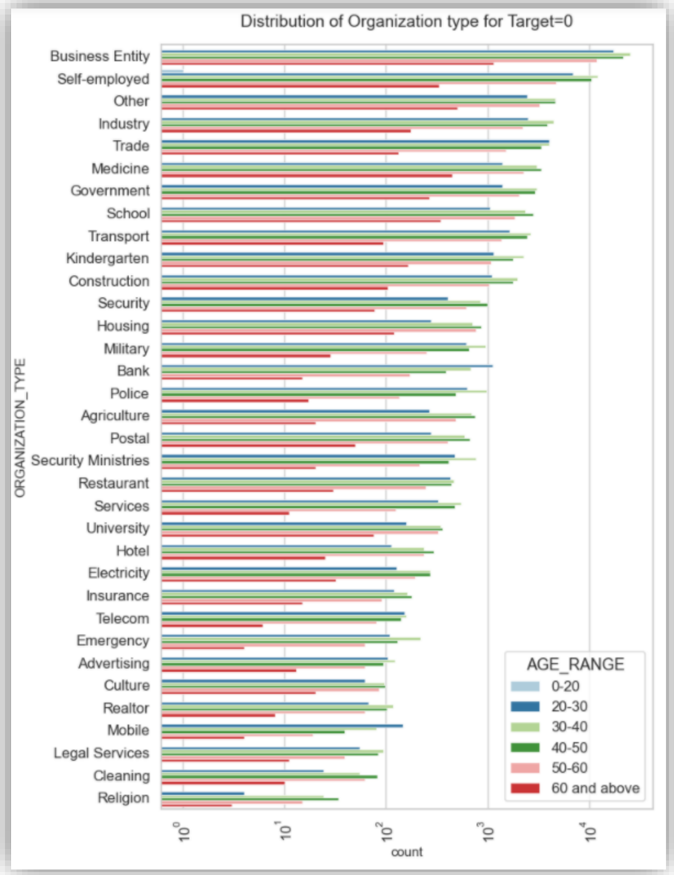


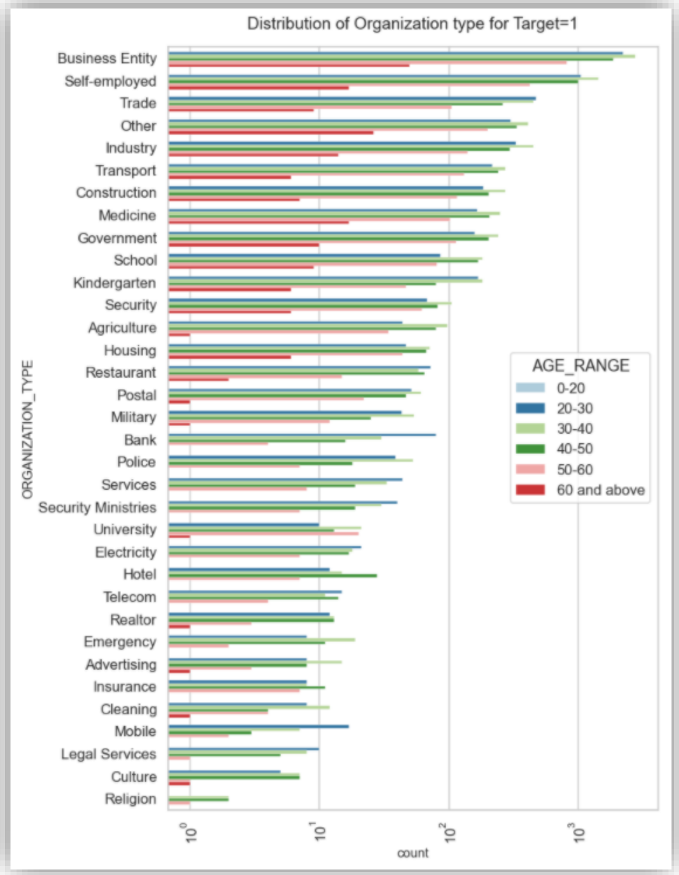
Data Analysis
Bi-variate Analysis

Distribution of Organization Type By Age-Range

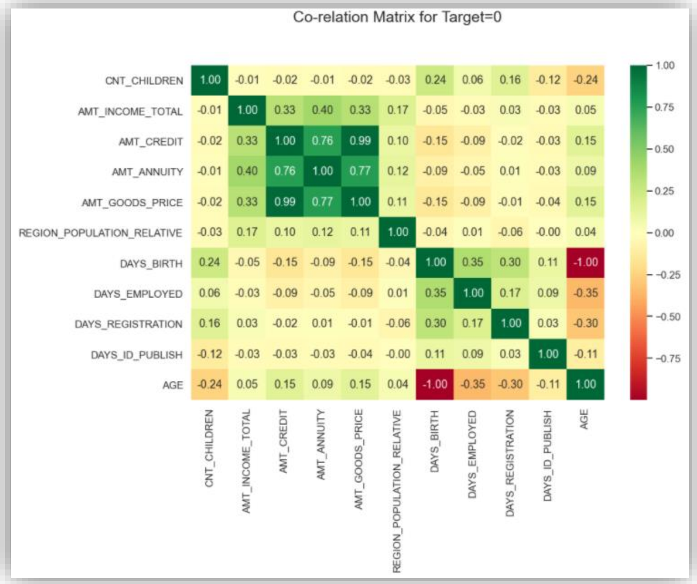
Inferences:

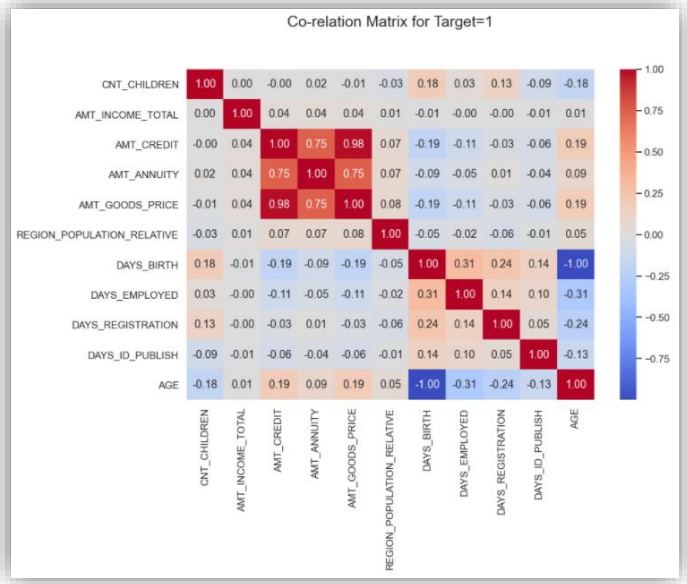
- People in age group of 30-40 with organization type as Business Entity are most likely to default
- People in age group 30-40 are most likely to default across all organization type
- Religion is least popular organization type
- Count of people in age group of 30-40 is highest across all organization type categories except trade, school, bank, agriculture, mobile, hotel





<u>Data Analysis</u>
Co-relation Analysis





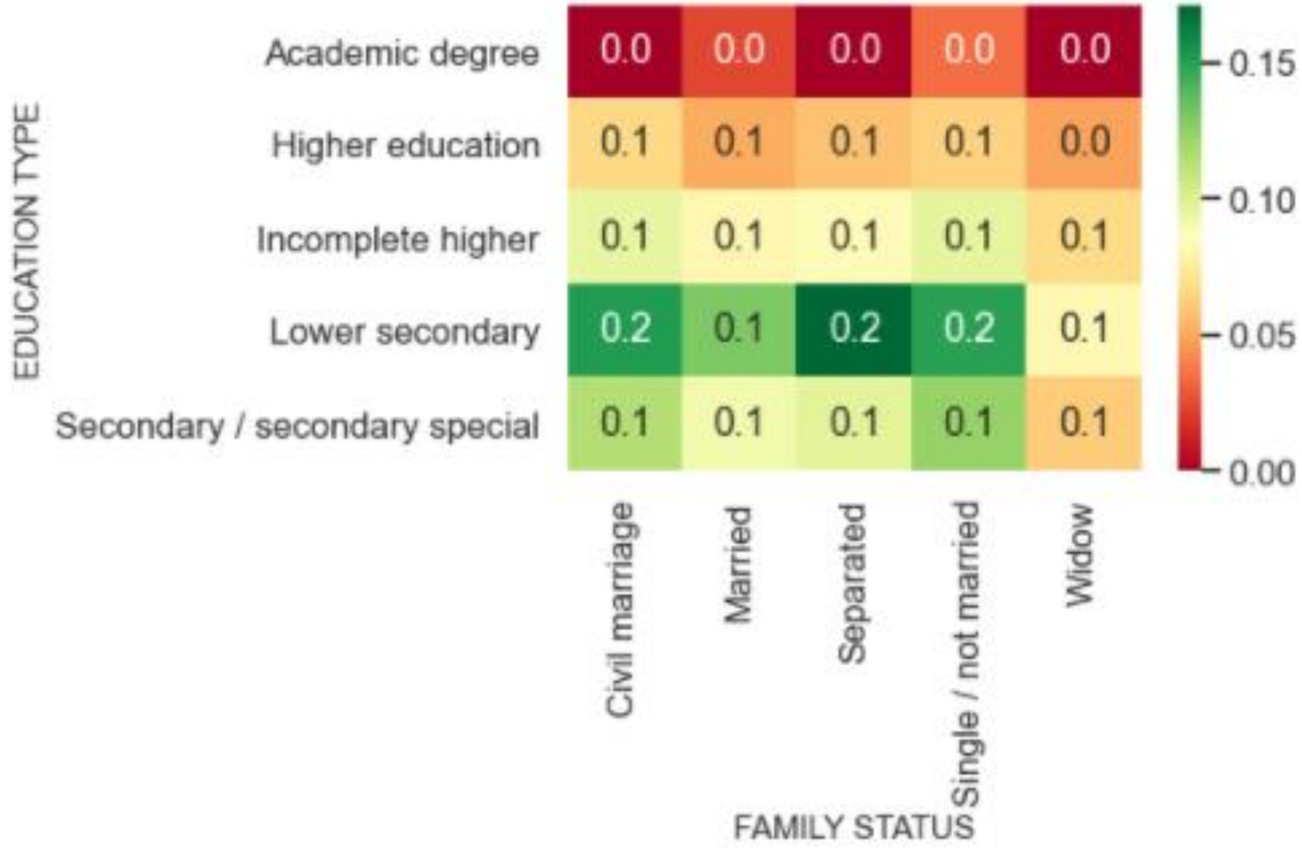
- AMT_CREDIT and AMT_ANNUITY have high co-relation
- AMT_ANNUITY and AMT_GOODS_PRICE have highly co-relation
- AMT_CREDIT and AMT_GOOD_PRICE have high co-relation

Data Analysis
Co-relation Analysis

Family Status Vs Education Type Vs Target Variable

Inference

- Applicants having lower secondary education are most likely to default
- Applicants having Academic Degree are least likely to default

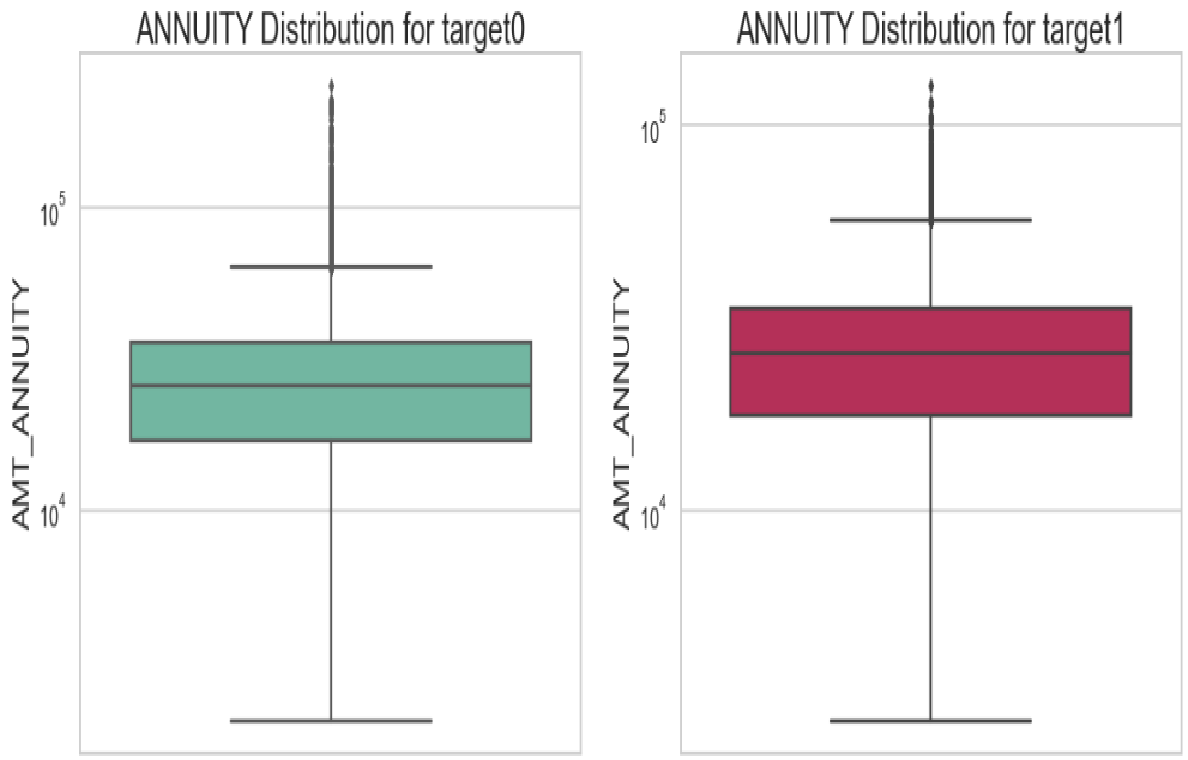


Data Analysis
Numerical variable Univariate Analysis

Distribution of **Annuity Amount**

Inferences:

- Amount of annuity is higher in target 1
- The first quartile is bigger than third quartile for annuity amount which means most of the annuity clients are from first quartile for both the cases.

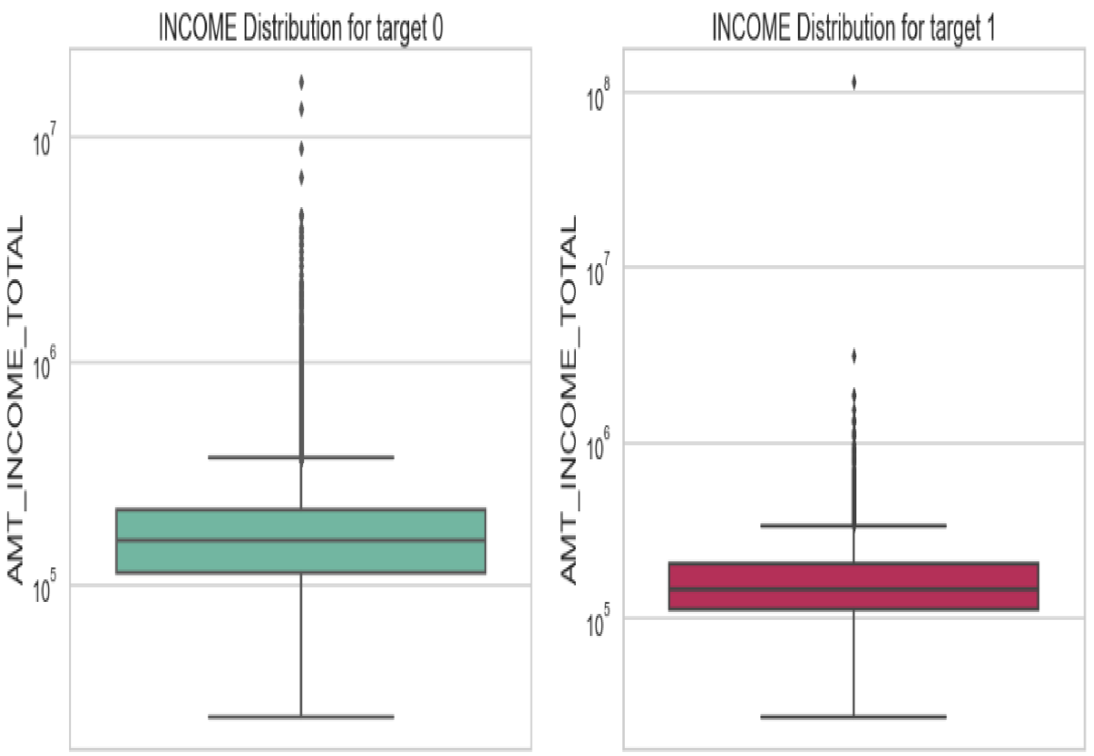


Data Analysis
Numerical variable Univariate Analysis

Distribution of **Income Amount**

Inferences:

- Amount of income is less for the clients who are likely to default than clients who are unlikely to default
- Outliers are present for both target 1 and target 2.

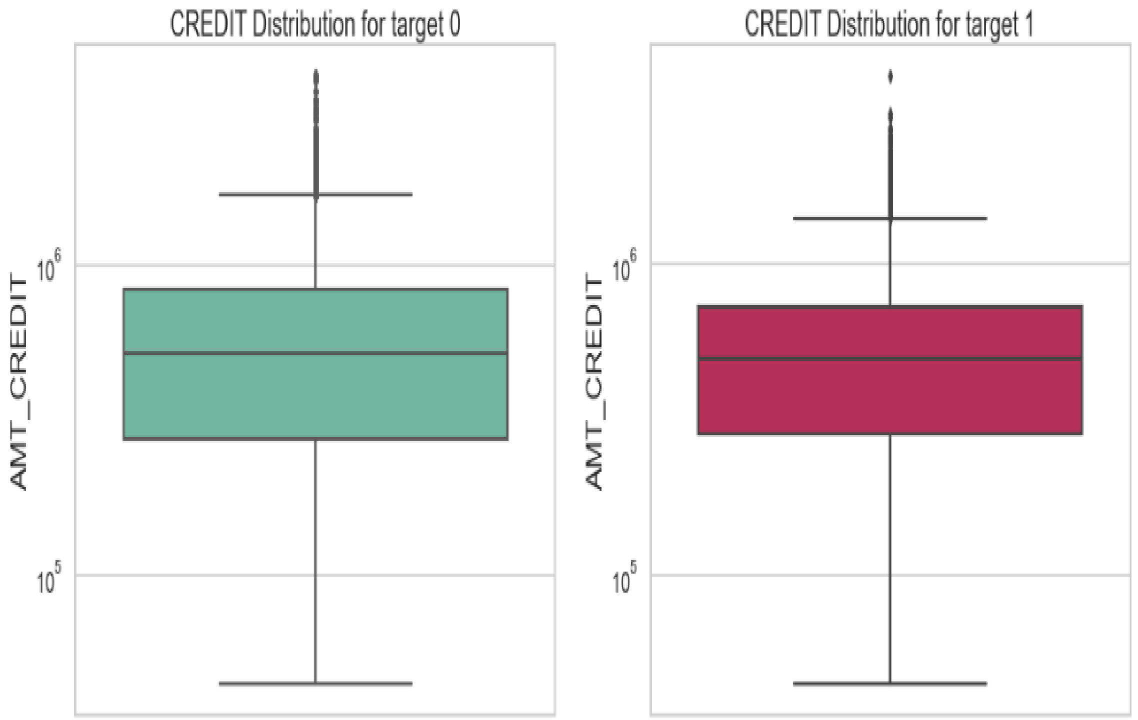


Data Analysis
Numerical variable
Univariate Analysis

Distribution of **Credit Amount**

Inferences:

- Amount of Credit is higher for the clients which are less likely default
- The first quartile is bigger than third quartile for Credit amount which means most of the Credit clients are from first quartile for both the cases i.e clients who are likely to default and unlikely to default .
- Outliers for credit amount are present for clients which are likely to default.

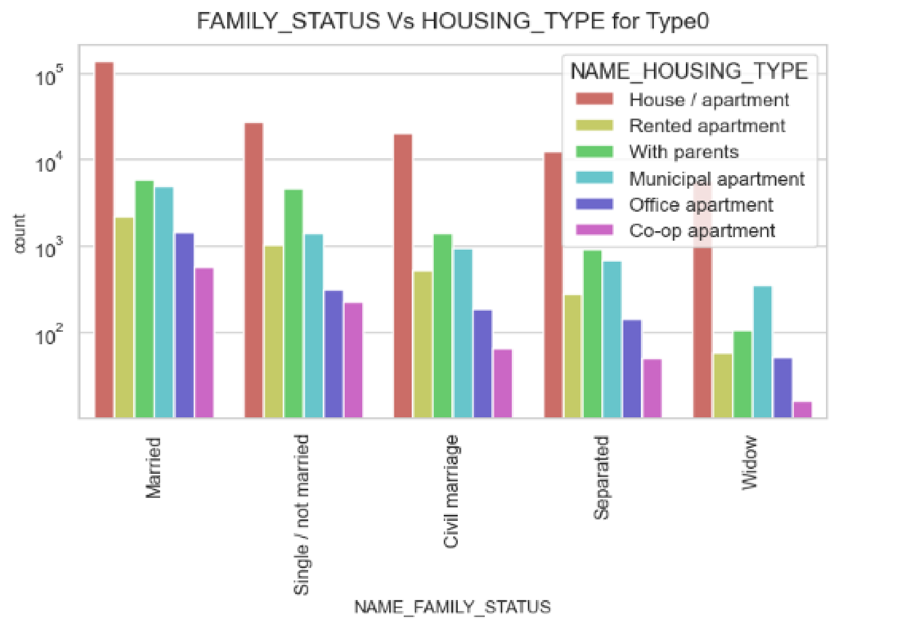


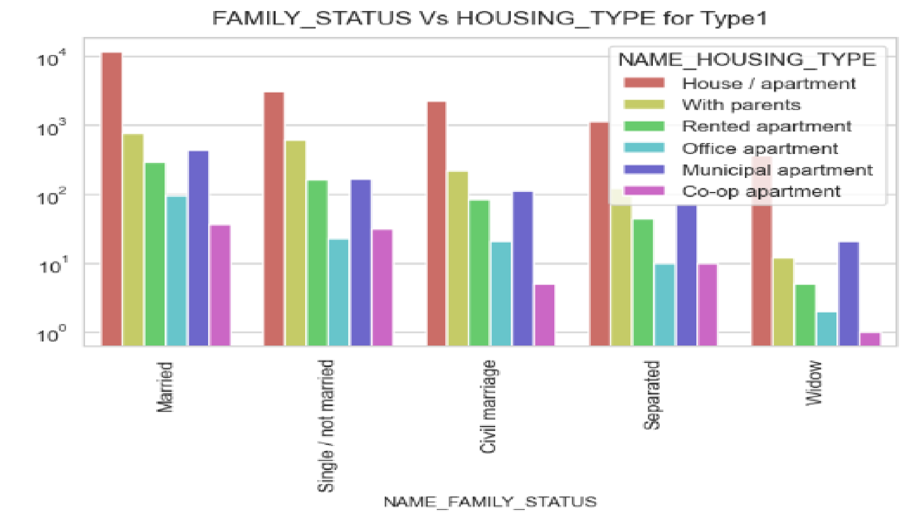
Data Analysis
Bivariate Analysis

Distribution of **FAMILY_STATUS vs HOUSING_TYPE**

Inferences:

- Most people live in House/Apartment irrespective of their family status
- Count of people living in Co-op Apartment is least



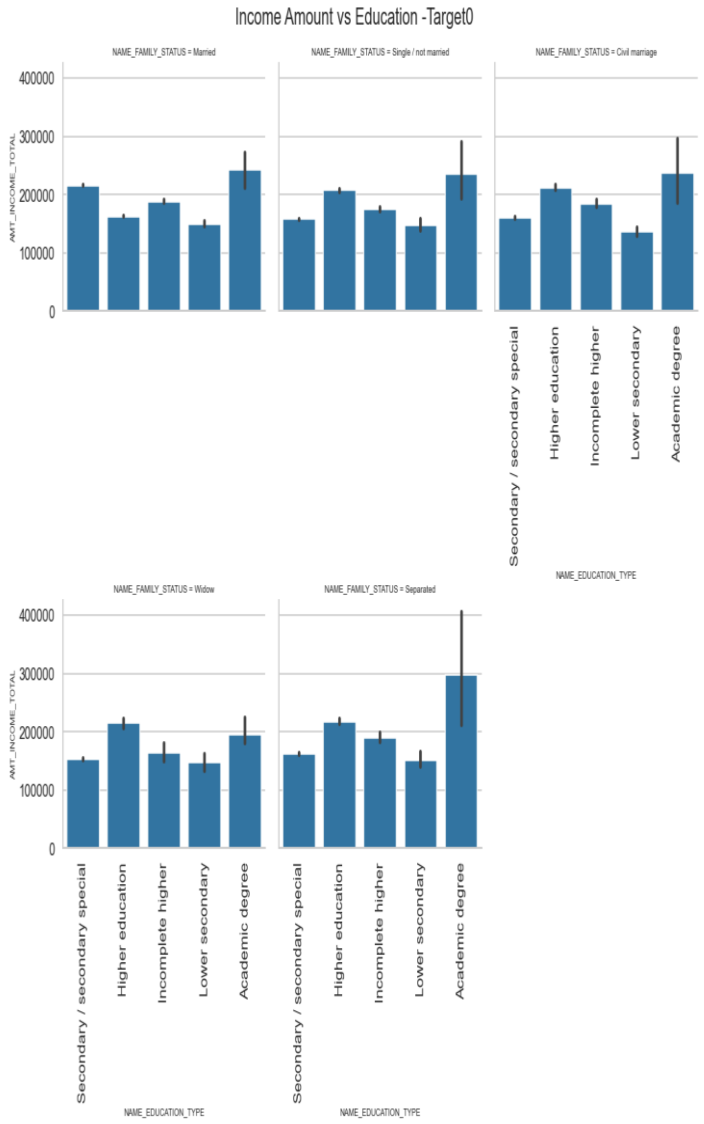


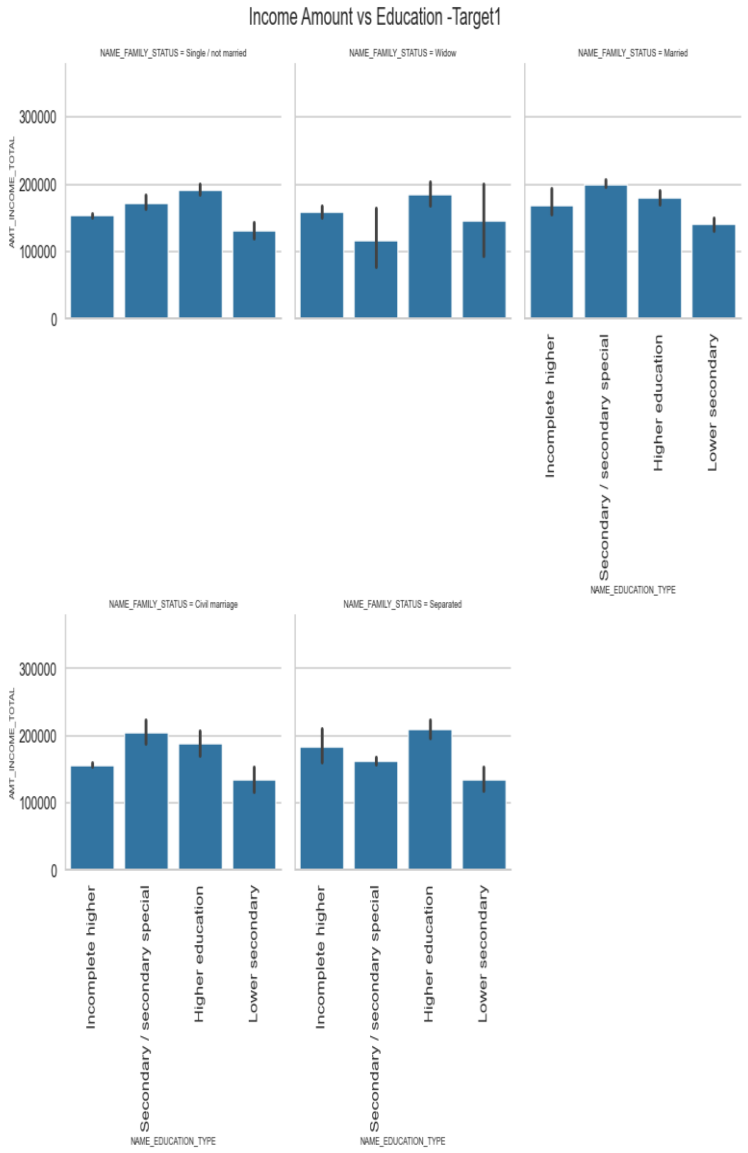
Data Analysis
Bivariate Analysis

Distribution of **vs Education & Family Status**

Inferences:

- For Education type 'Higher education' the income amount is mostly equal across all family status in both target-0 and target-1 category
- Academic degree has more income in target-0 category
- For target 1 – highest income range is for education type Secondary/secondary special and family status civil marriage and married
- Lower secondary of civil marriage family status have less income amount than others.



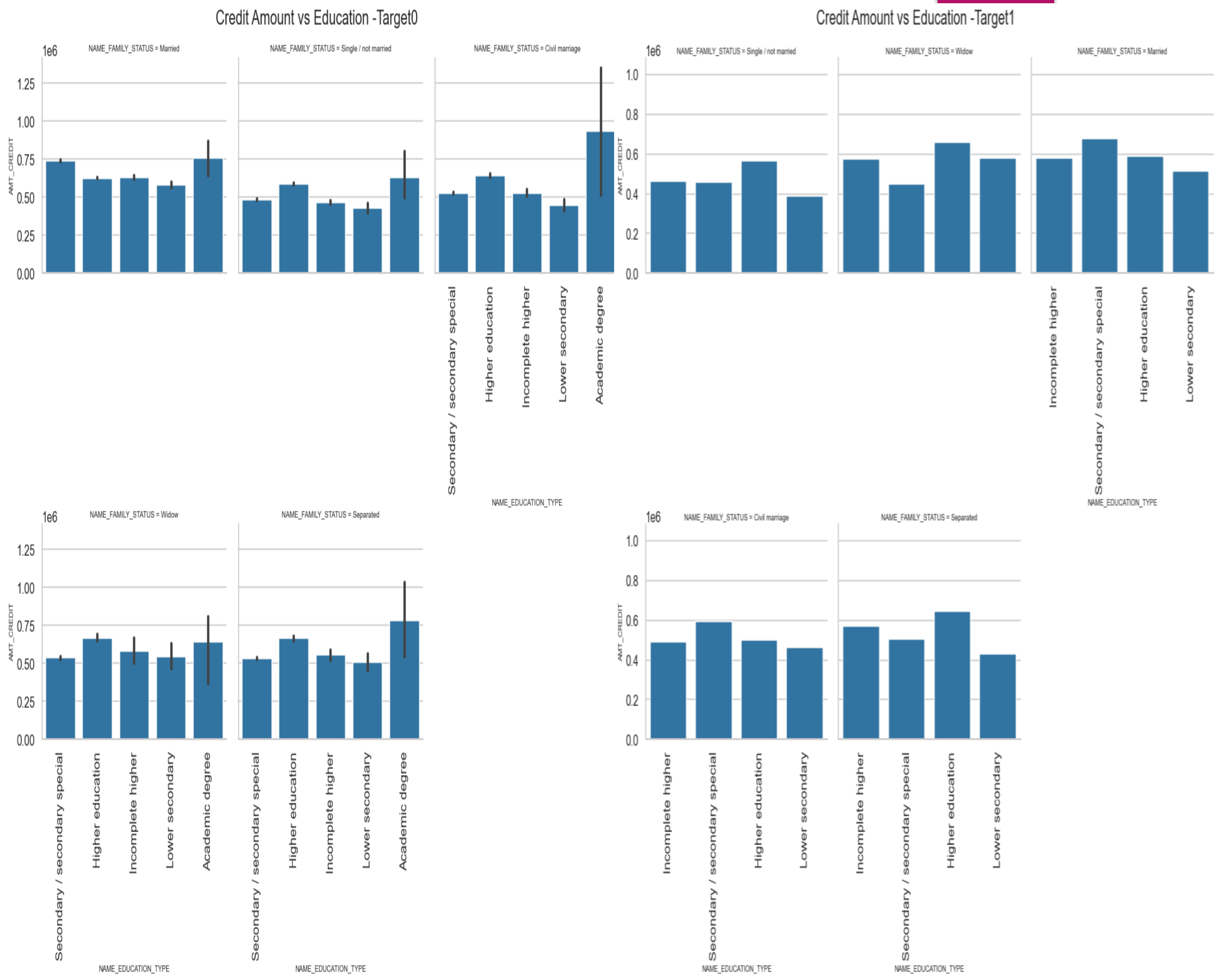


Data Analysis
Bivariate Analysis

Distribution of **Credit Amount vs Education & Family Status**

Inferences:

- Even though Credit Range is higher for clients with Family status as 'civil marriage', 'marriage' and 'separated' and education as "Academic degree" than others, these clients are least likely to default as income is also higher in type0 category
- credit Range is higher for clients with family status as 'widow' ,'married'& ,'separated' and education as 'secondary/secondary special' & higher education' these clients are more likely to default



Merging Previous Application Data With Current Application Data

- Merging data from previous application and current application based on SK_ID_CURR

```
Merged_Application_Data=pd.merge(left=application_data,right=previous_application_data,how='inner',on='SK_ID_CURR',suffixes='_x')
```

- Renaming common columns appropriately for better analysis

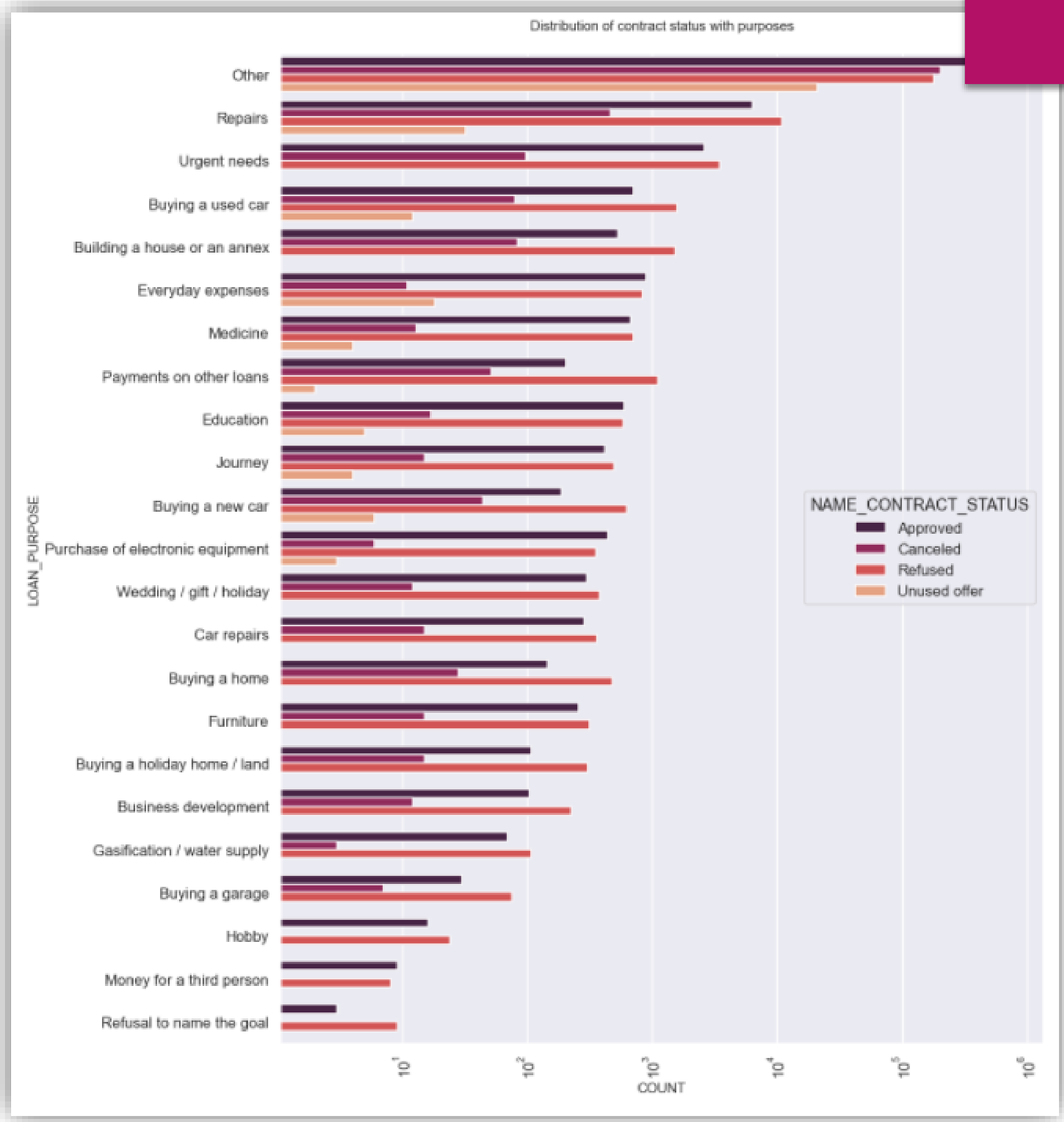
```
Merged_Application_Data = Merged_Application_Data.rename({'NAME_CONTRACT_TYPE_' : 'NAME_CONTRACT_TYPE',
                                                         'AMT_CREDIT_':'AMT_CREDIT','AMT_ANNUITY_':'AMT_ANNUITY',
                         'AMT_GOODS_PRICE_':'AMT_GOODS_PRICE','NAME_CONTRACT_TYPEx':'NAME_CONTRACT_TYPE_PREV',
                         'AMT_CREDITx':'AMT_CREDIT_PREV','AMT_ANNUITYx':'AMT_ANNUITY_PREV',
                         'AMT_GOODS_PRICEx':'AMT_GOODS_PRICE_PREV'}, axis=1)
```

Data Analysis on Merged Data Univariate Analysis For Variables

Distribution of contract status w.r.t purpose

Inference

1. Count of refused loans is greatest for loans applied for purpose of repair
2. Count of approved and refused loans for Education purpose is same
3. In general count of loans which are refused is more across all purpose categories
4. There are no cancelled or Unused offer for loans applied for purpose of Hobby, Money for a third person or Refusal to name the goal
5. Count of approved loan is highest for purpose as Others

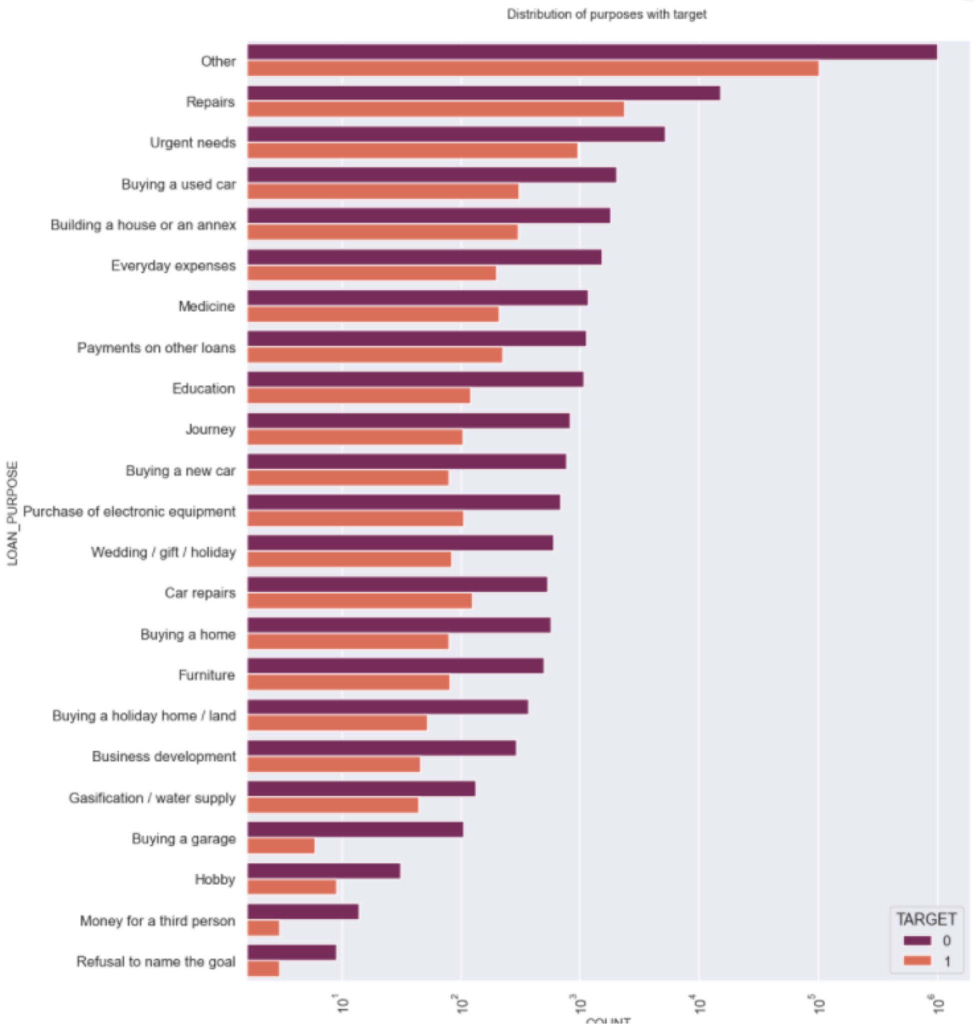


Data Analysis on Merged Data Univariate Analysis For Variables

Distribution of Purpose with respect to Target Variable

Inference

1. Loans taken for Purpose : Others are most likely to default
2. Loans taken for buying a home and Furniture are almost equally likely to default
3. In general count of loans which are likely to default is less than the loans which will be repayed



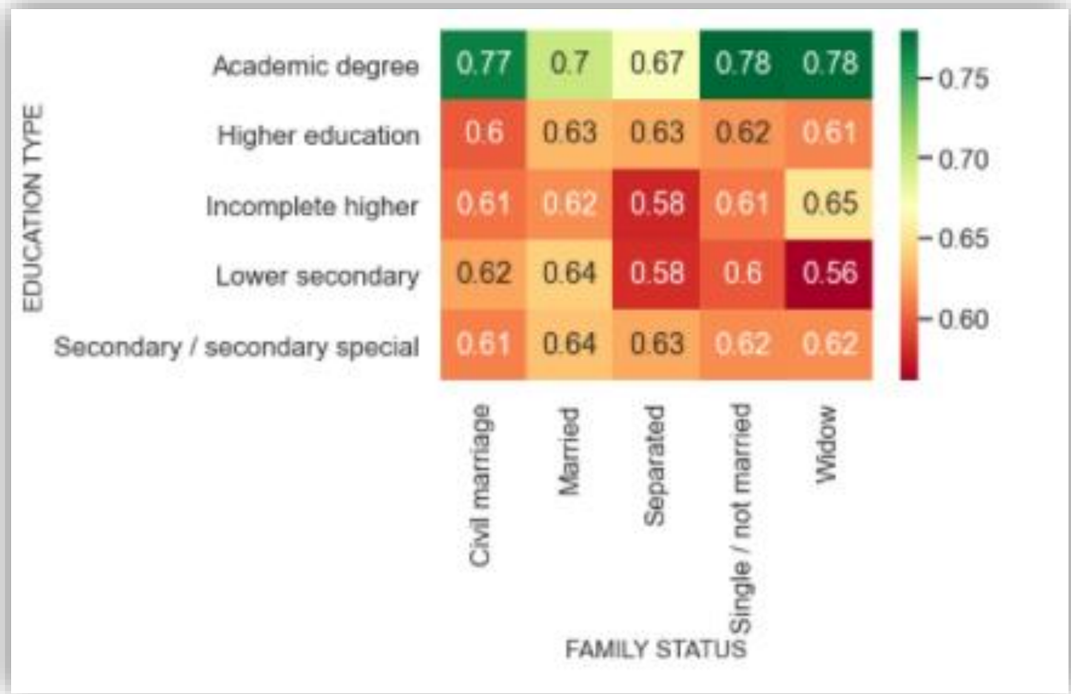
Data Analysis on Merged Data

Heat map of Education Type vs Family Status vs Approved Flag

Assumption : Anything which is not approved is marked as flag 0 (Refused, Cancelled, Unused offer)

Inference

1. Count of Loans approved which are applied by people having Academic degree is highest across all Family Status
2. Count of Loans approved which are applied by widow with lower secondary education is least
3. Count of Loans approved which are applied by Single/Not Married and Window people having Academic degree is highest



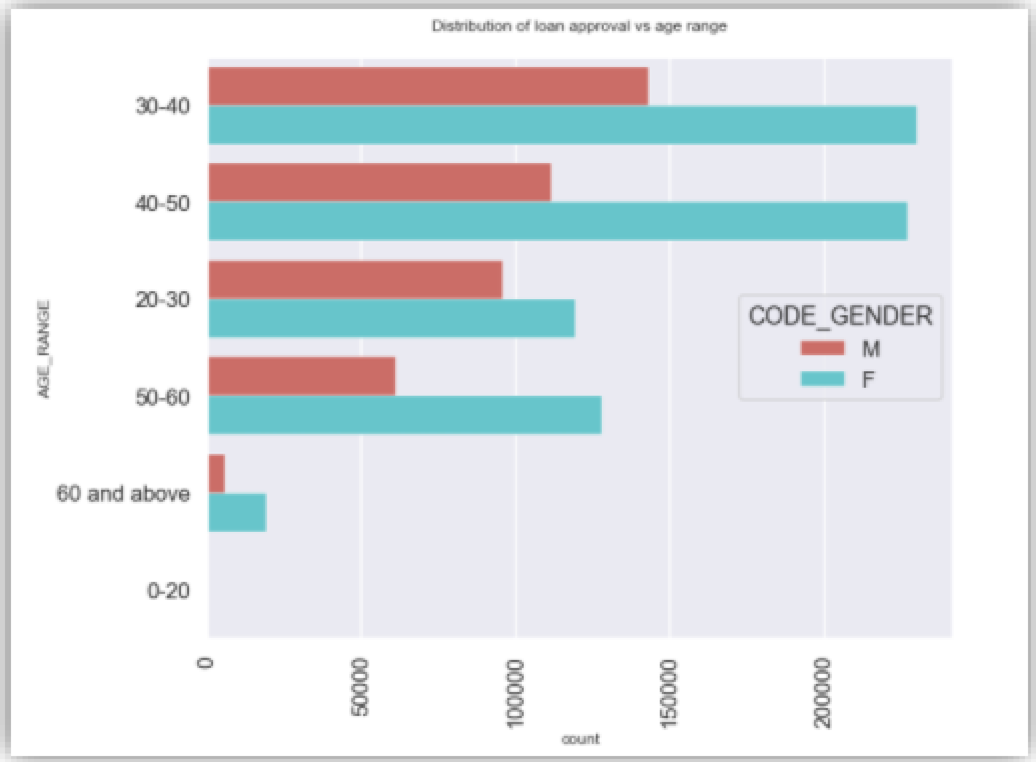
Data Analysis on Merged Data

Count values of approved applications for each age group

Assumption : Anything which is not approved is marked as flag 0 (Refused, Cancelled, Unused offer)

Inference

- Count of Loans approved is highest for Females in age group of 30-40
- Count of Loans approved is in general higher for Females
- No loan is approved for people who are less than 20 yrs

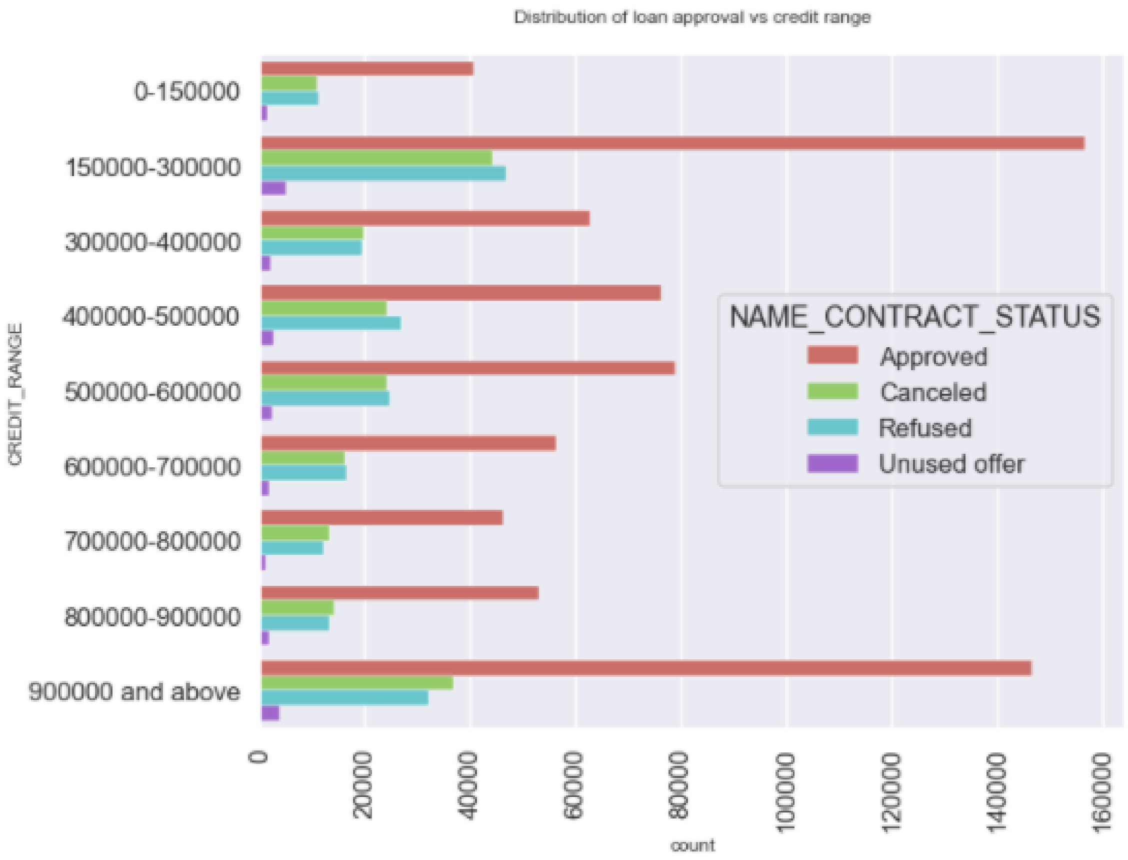


Data Analysis on Merged Data

Count values of applications status across various credit amount range

Inference

- *Count of Loans approved is highest credit range 150000-300000 and 900000 and above*
- *Count of Loans refused is lowest for credit amount range 0-150000 and 700000-900000*

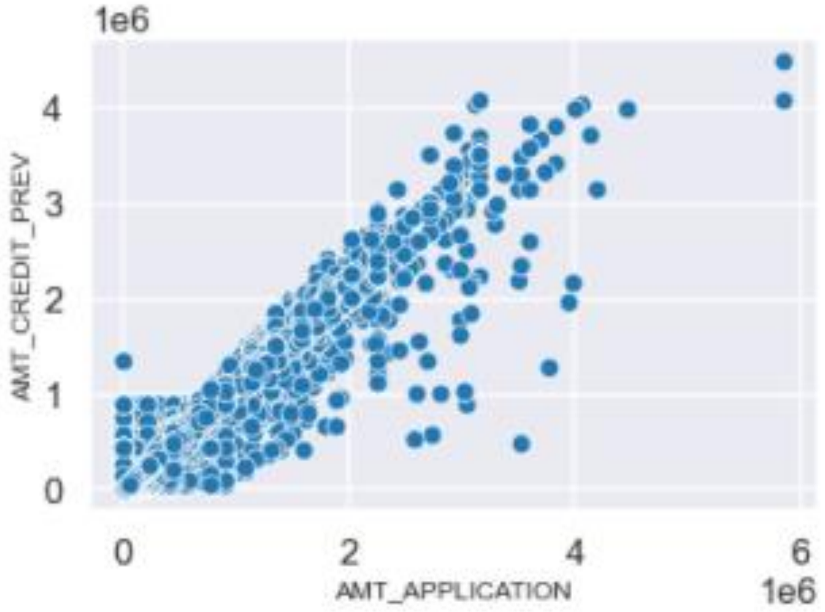


Data Analysis on Merged Data

Relation between Amount of application and Amount credit in previous application

Inference

- *There is a strong positive correlation between Amount of Loan and Amount which was credited*



Conclusions

- People in the age group of 30-40 having secondary or lower secondary education in organization types Business/Self Employed/Trade are most likely to default.
- Females are more likely to default in lower income range while males are more likely to default at higher income range of > 1.5 Lakhs
- Loans for purpose of 'Others' or Repairs are most likely to default
- In general Females have applied for more loans and as a result more loans are approved for females
- In general cash loans are more popular than revolving loans
- Income for people having Secondary/Lower secondary education is least irrespective of family status
- People in age group of >50 Yrs. are least likely to default
- People with academic degree are least likely to default also they have highest income range
- Student, Businessman, Pensioners less default, possibly because of less number of applications
- Number of loans approved for widow with lower secondary education is least