EDA ANALYSIS

CREDIT EDA ASSIGNMENT ON CREDIT DATA

READING AND EXPLORING THE DATA

- Importing the required libraries such as numpy, Pandas and Seaborn.
- Reading the data with read_csv command and viewing the head of the data.
- Calculating the percentage of null value in the dataset.
- Dropping the columns which has more than 35% of null values And also dropping he irrelevant columns

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
pd.set_option('display.max_columns',None)
pd.set_option('display.max_rows',None)
```

```
df = pd.read_csv("./assignment/application_data.csv")
df.head()
```

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

```
#droping the columns which are more than 35% of null values
selected_columns = df.columns[(df.isnull().sum() / len(df) * 100 < 35) == Tru
df1 = df[selected_columns].copy()
```

```
# calculaing the percentage of null values in he dataset
df.isnull().sum() / len(df) * 100
```

HANDLING NULL VALUES

- As we can see there are multiple null values in various fields.
- We will be looking for the mean and median of the Columns if it is a numerical column else we look for mode
- We fill the values with the mode or median w.r.t the Column type.

```
# Calculating the null values
df1.isnull().sum()
```

```
SK_ID_CURR                     0
TARGET                         0
NAME_CONTRACT_TYPE             0
CODE_GENDER                    0
FLAG_OWN_CAR                   0
FLAG_OWN_REALTY                0
CNT_CHILDREN                   0
AMT_INCOME_TOTAL               0
AMT_CREDIT                     0
AMT_ANNUITY                   12
AMT_GOODS_PRICE              278
NAME_TYPE_SUITE             1292
NAME_INCOME_TYPE               0
NAME_EDUCATION_TYPE            0
NAME_FAMILY_STATUS             0
NAME_HOUSING_TYPE              0
REGION_POPULATION_RELATIVE     0
DAYS_BIRTH                     0
DAYS_EMPLOYED                  0
DAYS_REGISTRATION              0
DAYS_ID_PUBLISH                0
FLAG_MOBIL                     0
OCCUPATION_TYPE            96391
CNT_FAM_MEMBERS                2
REGION_RATING_CLIENT           0
ORGANIZATION_TYPE              0
EXT_SOURCE_2                 660
EXT_SOURCE_3               60965
dtype: int64
```

```
df1.isnull().sum()
```

```
SK_ID_CURR                    0
TARGET                        0
NAME_CONTRACT_TYPE            0
CODE_GENDER                   0
FLAG_OWN_CAR                  0
FLAG_OWN_REALTY               0
CNT_CHILDREN                  0
AMT_INCOME_TOTAL              0
AMT_CREDIT                    0
AMT_ANNUITY                   0
AMT_GOODS_PRICE               0
NAME_TYPE_SUITE               0
NAME_INCOME_TYPE              0
NAME_EDUCATION_TYPE           0
NAME_FAMILY_STATUS            0
NAME_HOUSING_TYPE             0
REGION_POPULATION_RELATIVE    0
DAYS_BIRTH                    0
DAYS_EMPLOYED                 0
DAYS_REGISTRATION             0
DAYS_ID_PUBLISH               0
FLAG_MOBIL                    0
OCCUPATION_TYPE               0
CNT_FAM_MEMBERS               0
REGION_RATING_CLIENT          0
ORGANIZATION_TYPE             0
EXT_SOURCE_2                  0
EXT_SOURCE_3                  0
dtype: int64
```

FIXING THE IRREGULAR VALUES

- Some of the columns has ‘XNA’ as the irregular value.
- We will be dropping the values if they are completely missing at random
- Converting the days columns to years
- Converting the Flags into binary for ease of use.

```
df1[df1.CODE_GENDER == 'XNA']
```

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER
35657	141289	0	Revolving loans	XNA
38566	144669	0	Revolving loans	XNA
83382	196708	0	Revolving loans	XNA
189640	319880	0	Revolving loans	XNA

```
# AS they are missing completely at random we can drop the va
```

```
df1 = df1[~(df1.CODE_GENDER == 'XNA')]
```

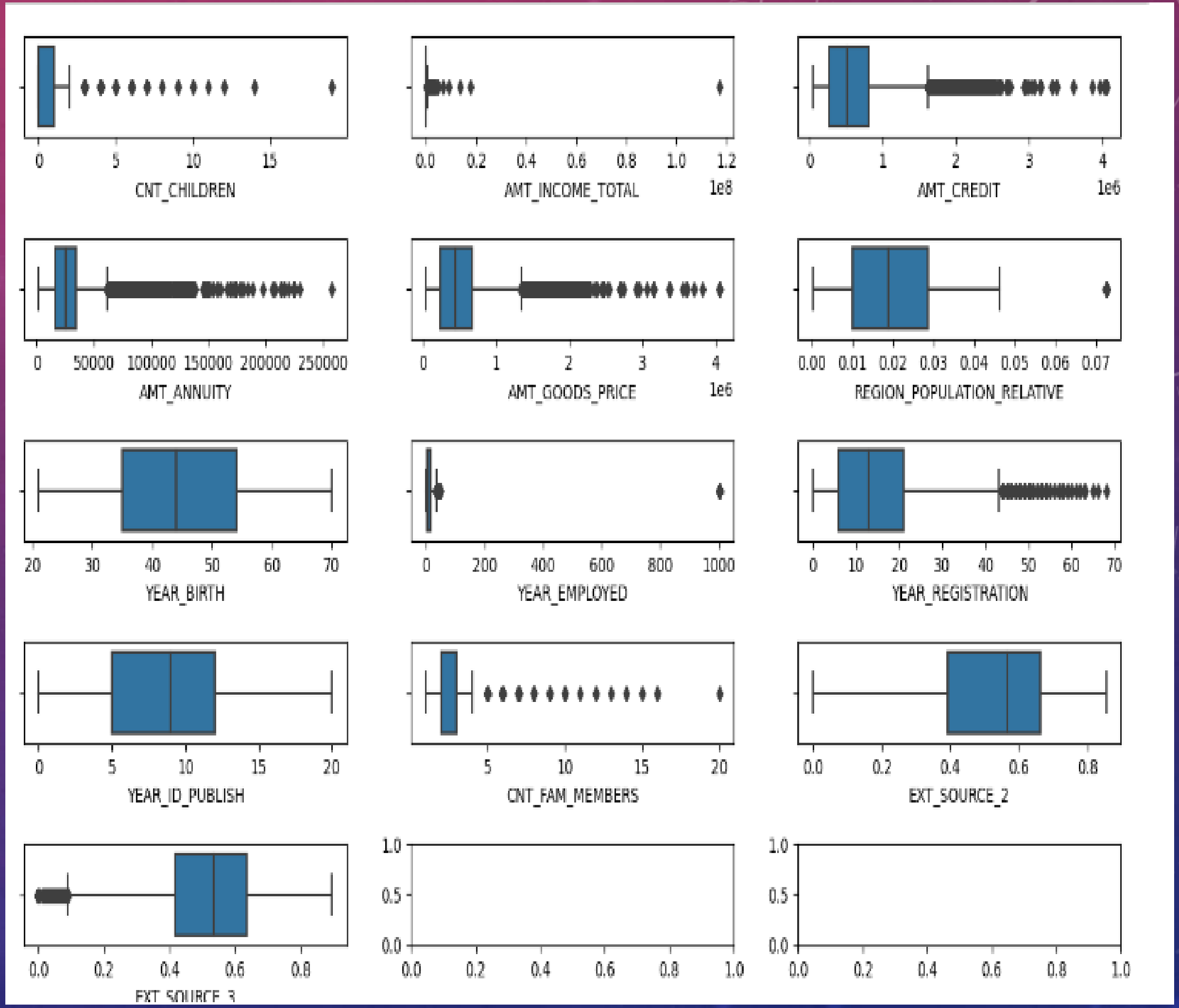
```
df1[['FLAG_OWN_CAR', 'FLAG_OWN_REALTY']].head()
```

	FLAG_OWN_CAR	FLAG_OWN_REALTY
0	0	1
1	0	0
2	1	1
3	0	1
4	0	1

```
cols = ["DAYS_BIRTH","DAYS_EMPLOYED","DAYS_REGISTRATION","DAYS_ID_PUBLISH"]

df1[cols] = abs(df1[cols]//365)
df1[cols].head()
```

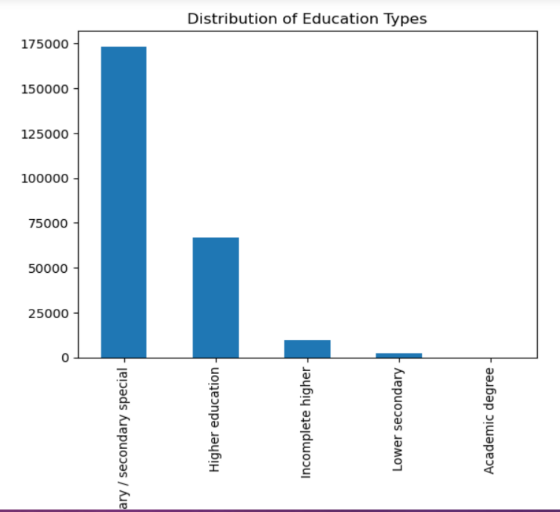
OUTLIER ANALYSIS

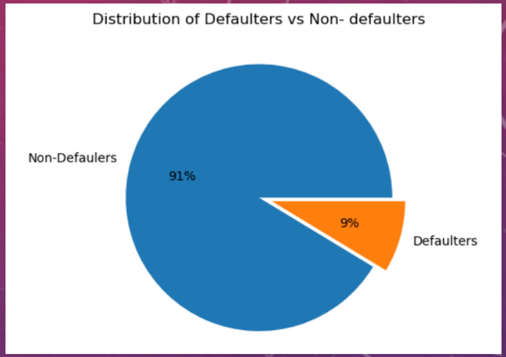
- There is a huge outlier in the Total income Column but, We will be ignoring it because people can have high income.
- There is continuous outliers in the Credit, Annuity, Goods Price, External Source 3
- There is an unrealistic outlier in employment year, So we will be removing that values

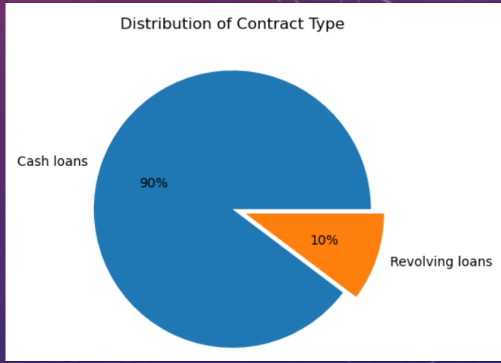


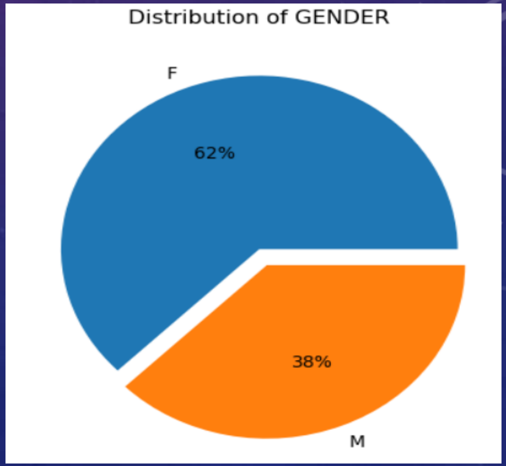
UNIVARIATE ANALYSIS

- Distribution of Defaulters & Non-Defaulters
  - About 91% of the people are non defaulters. There is a notable imbalance in the distribution of the data.
- Distribution of Contract Type
  - About 90% of the applications are of cash loans.
- Distribution of Gender
  - Here we can see almost a ratio of 3:2 of female to male.
- Distribution of Age
  - Most people have an age of 35-45.
  - Age of applicants sharply increased from 20 to 45 and gradually decreased from 50.







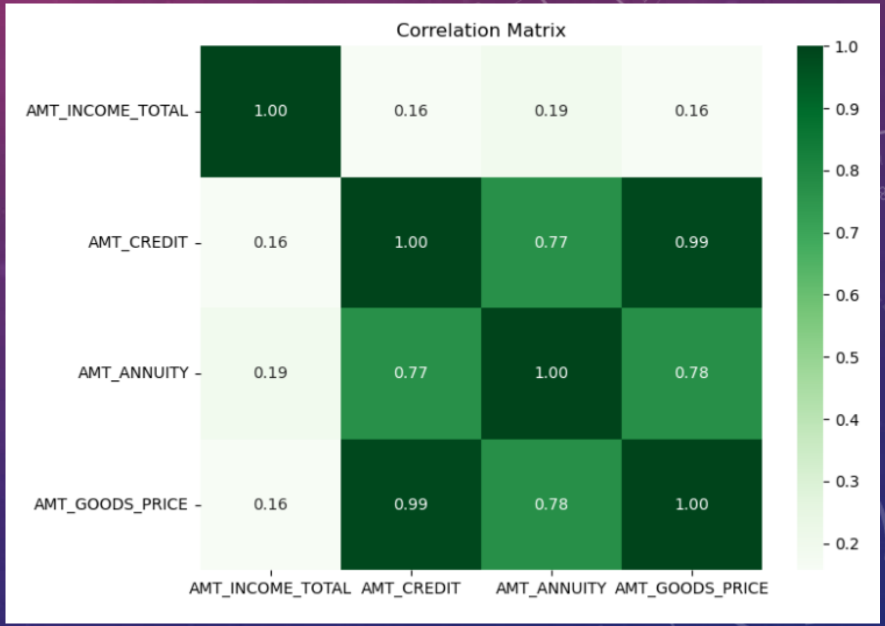


BIVARIATE AND MULTIVARIATE ANALYSIS

NUMERIC – NUMERIC ANALYSIS

- There is a strong correlation between Credit amount and Goods Price
- Also there is a significant correlation between Credit and Annuity,

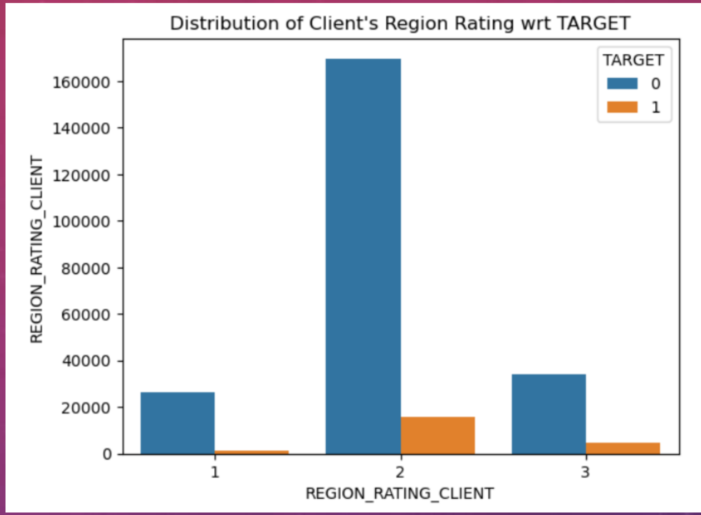
  Goods Price and Annuity.

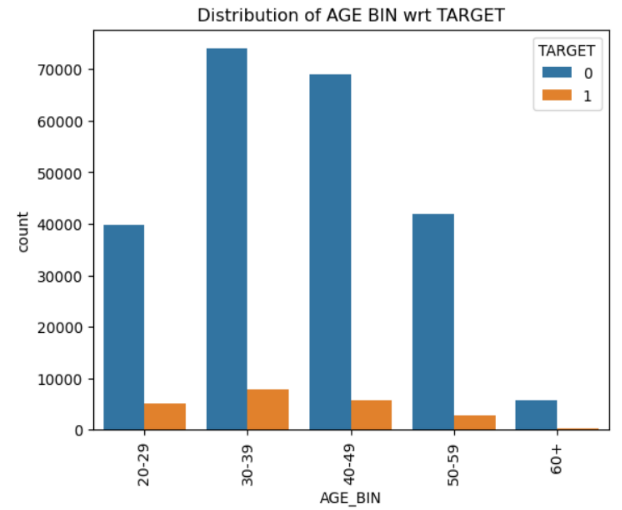


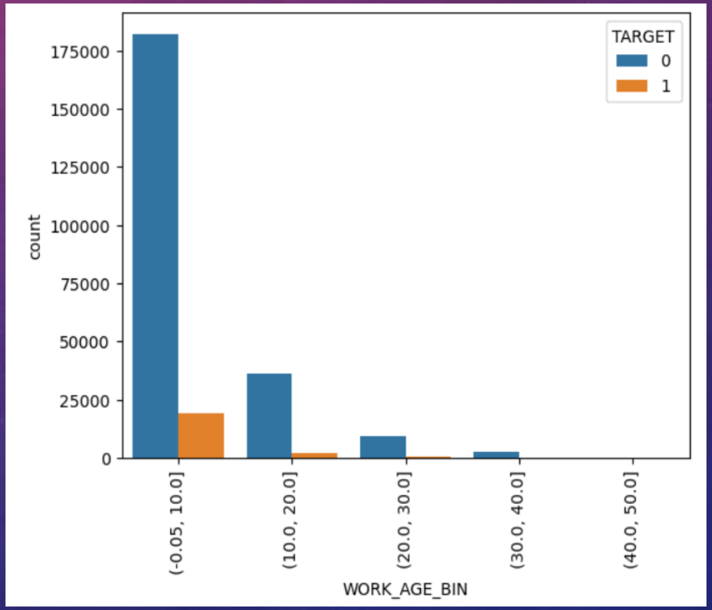
NUMERIC – CATEGORICAL ANALYSIS

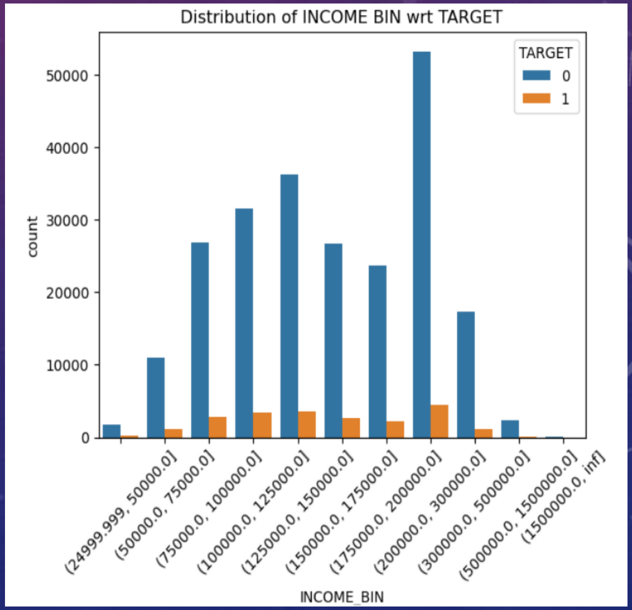
Defaulters	Non Defaulters
Most of them are from 2 rating region	Most of them are from 2 rating region
Most of them are from 0-10 years of work exp	Most of them are from 0-10 years of work exp
Most of them are of age bin 30-39	Most of them are of age bin 30-49
Almost every income bin has equal defaulters	1.75 – 2 lakh income group has more non defaulters

- People from different categories who took loans the most are also the people who defaulted the loans in the categories respectively



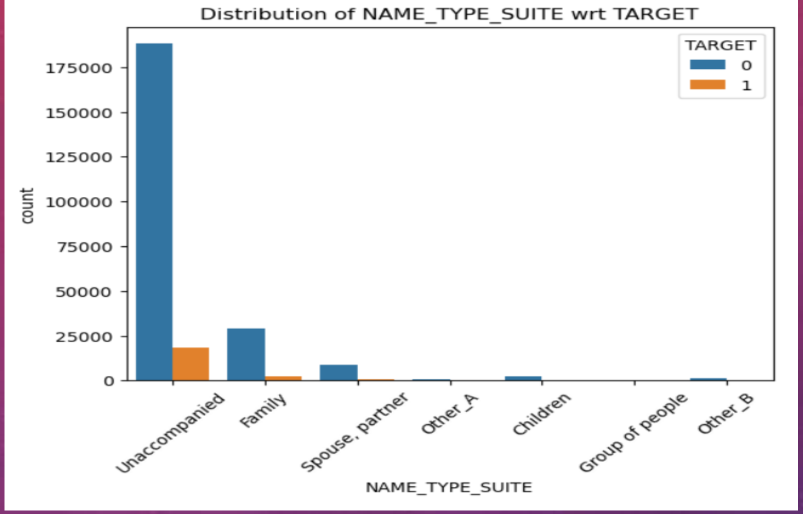


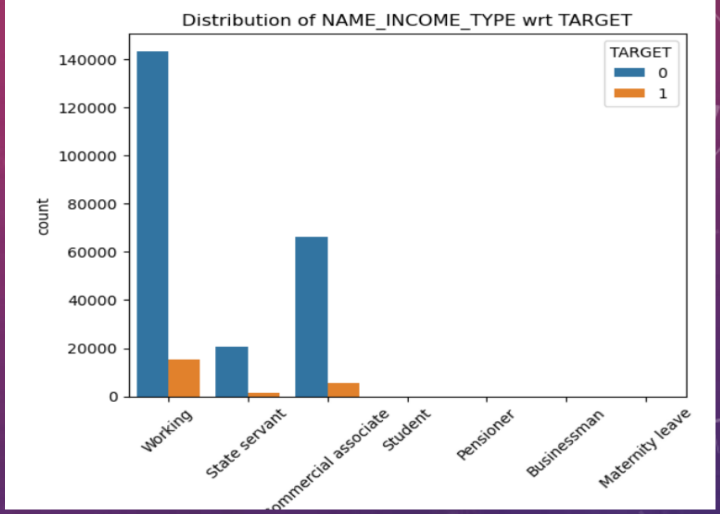


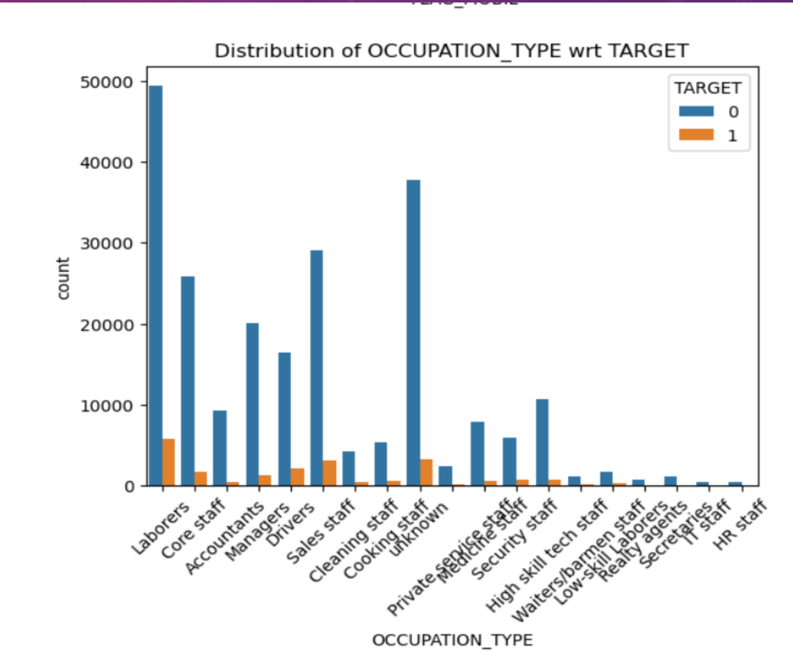


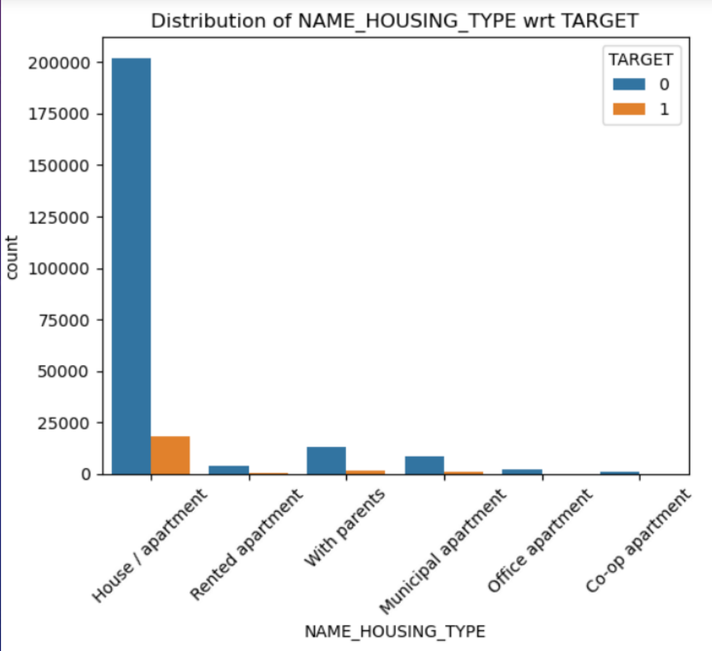
CATEGORICAL – CATEGORICAL ANALYSIS

Defaulters	Non Defaulters
Most of them were unaccompanied while applying	Most of them were of working class
Most of them have house/apartment	Most of them also have house
Most of them are from Laborers	Most of them are also unaccompanied while applying
Most of them are from working class	Most of them are from working class

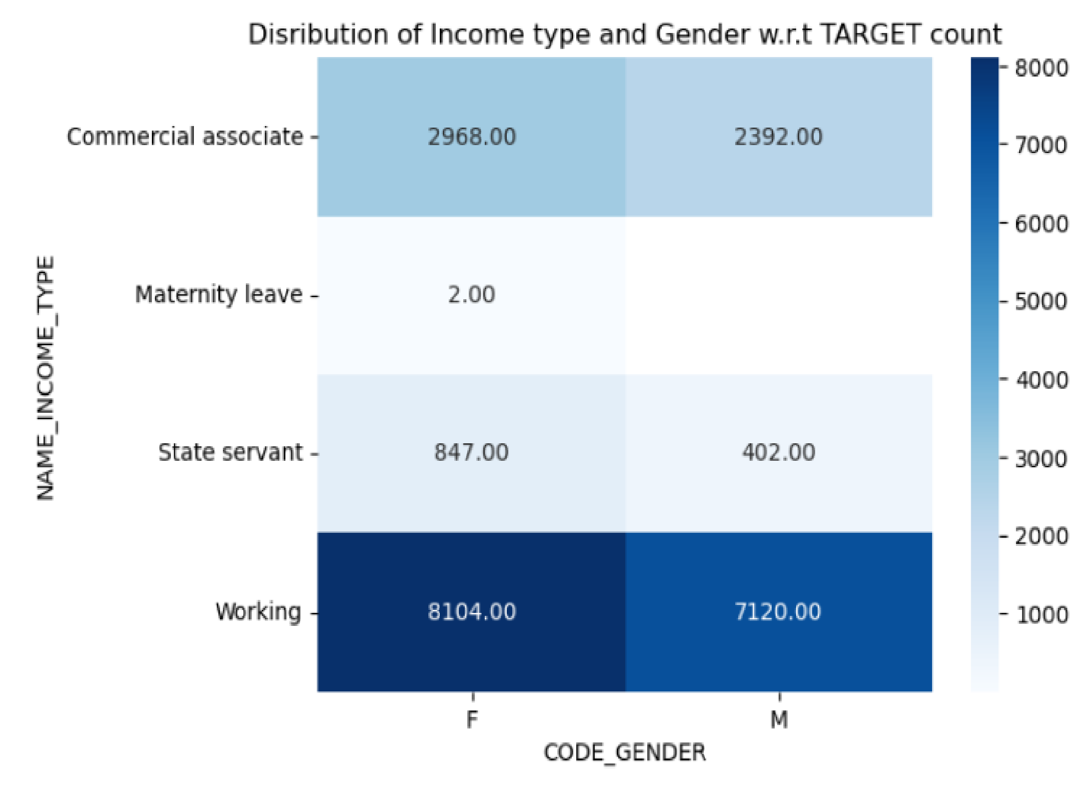


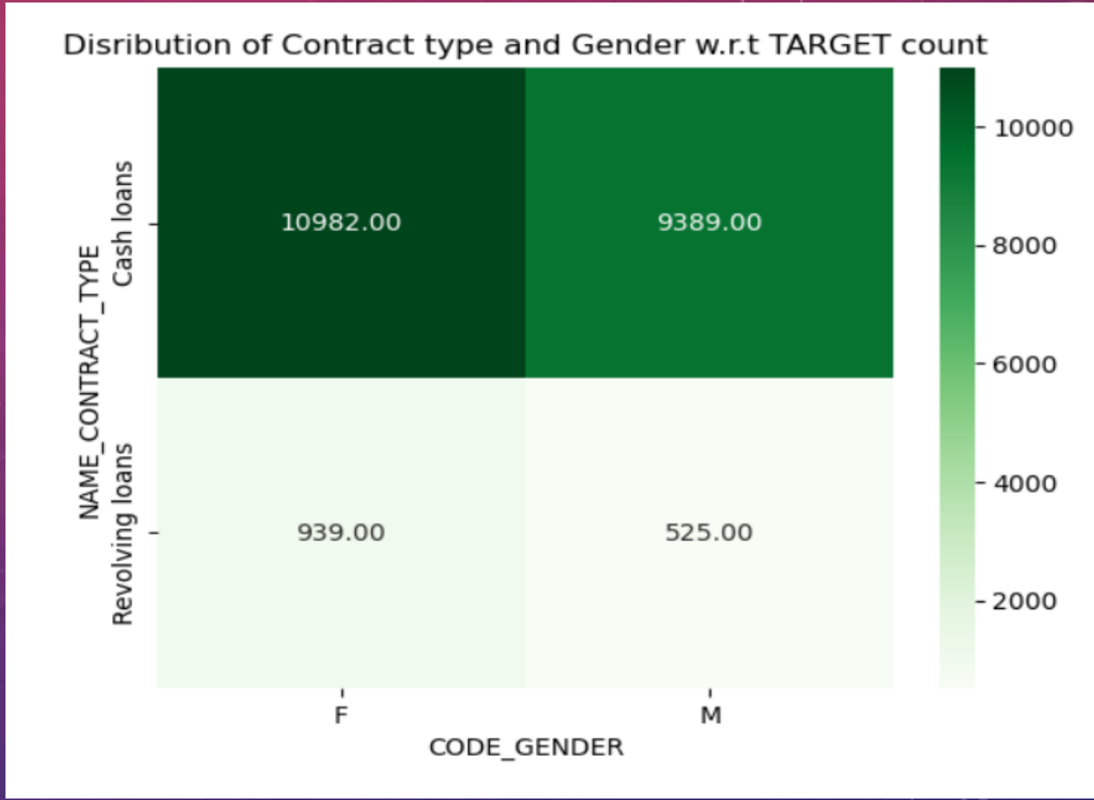






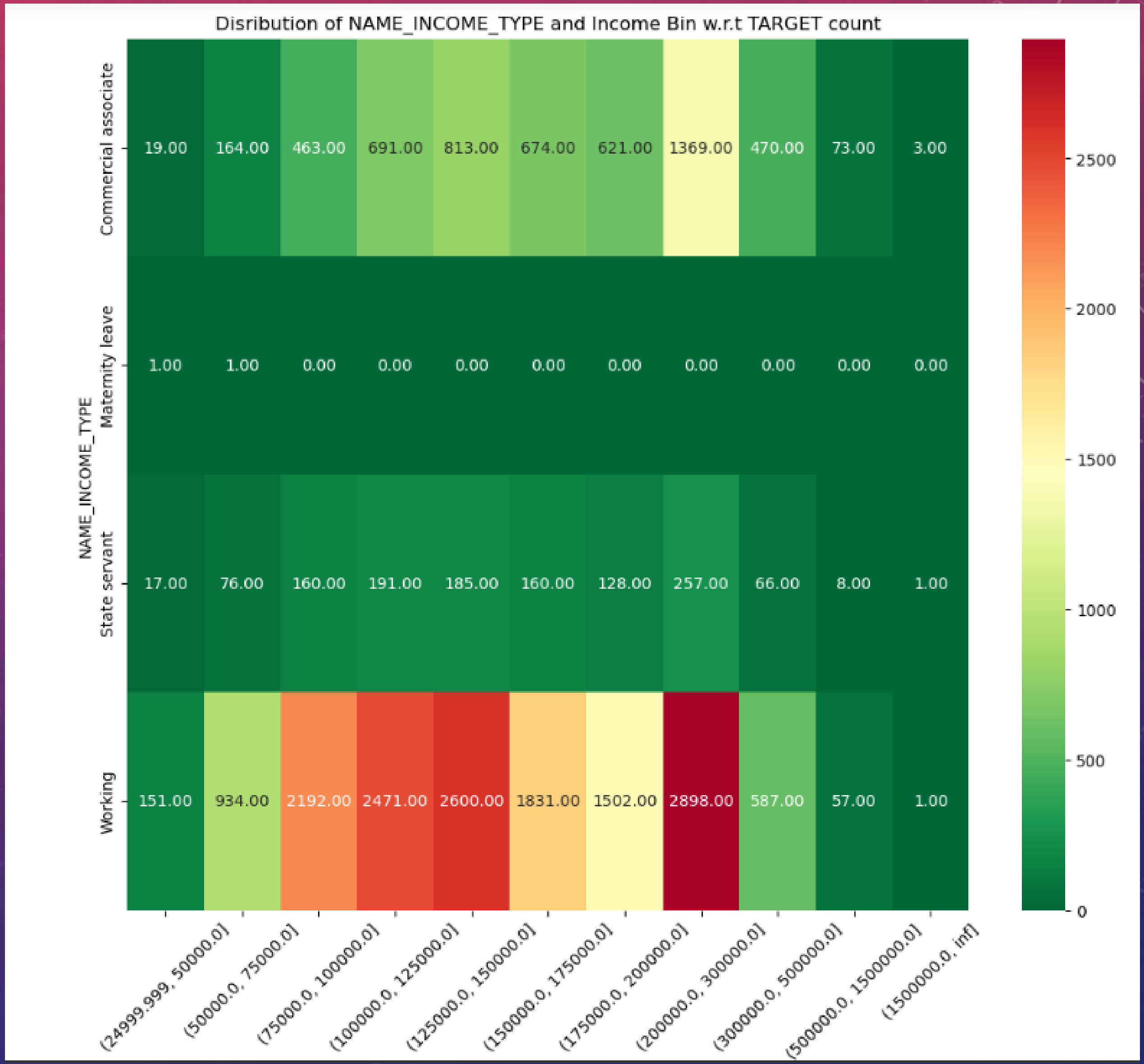
MULTIVARIATE ANALYSIS



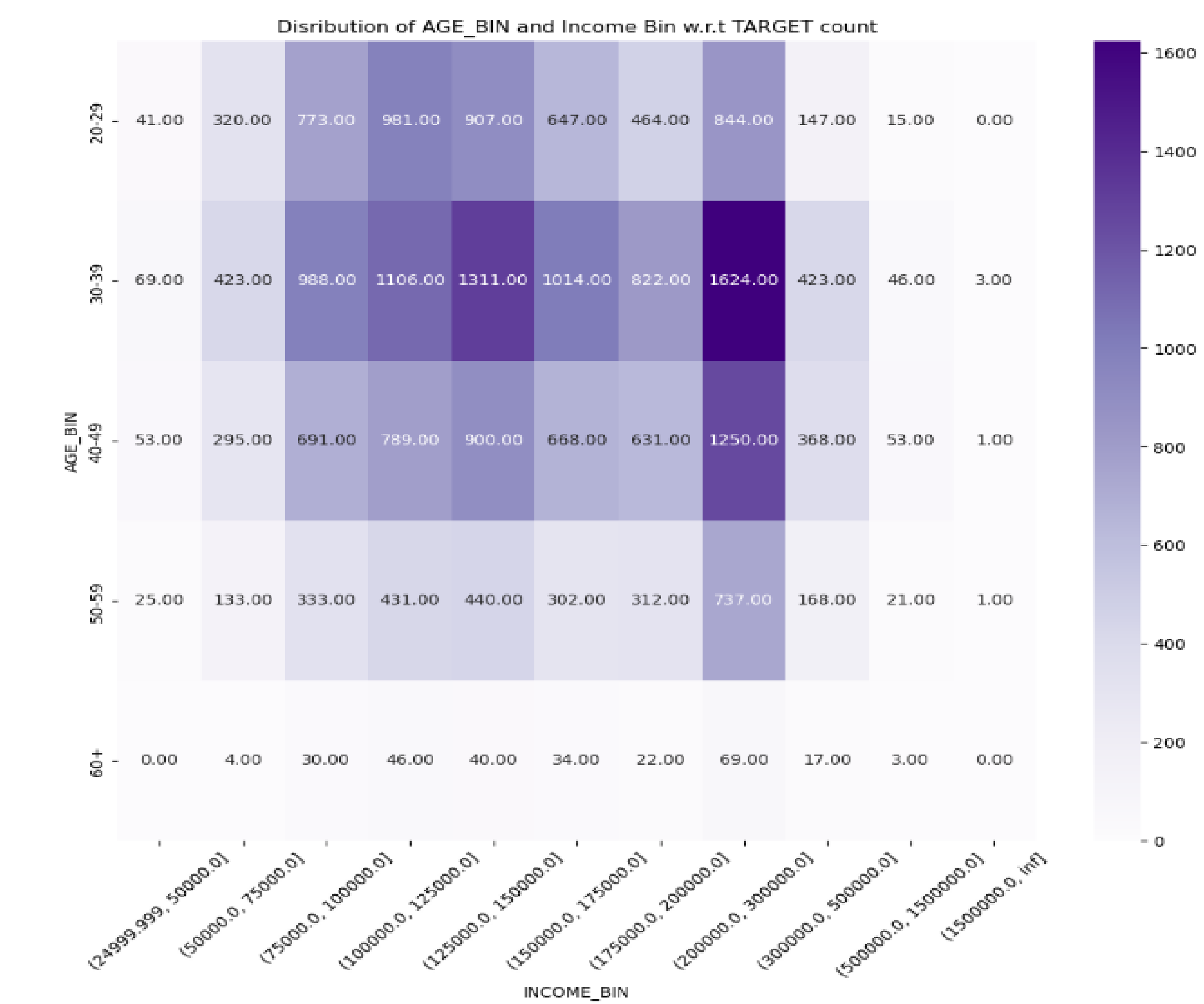


- Most the defaulters are females who took cash loans
- Males and females who defaulted are almost same
- Working class males and females defaulted the most.

- There were only 2 defaulters while people were on maternity leave
- Most of the defaulters are working class people from 1.75 -2 lakh and 1-1.25 lakh income bin.
- There is one defaulter who is in income bin of 15lakh+



- Most of the defaulters are form age group 30-39,40-49 from the income bin 1.75-2 lakh
- Second highest defaulters are from 1-1.25 lakh income range and are in the age group 30-39.



READING AND EXPLORING PREVIOUS DATASET

- Reading the previous dataset and checking all the columns.
- Replacing all the redundant values such as XNA and XAP
- Checking the null values as dropping the columns with null values more than 25%
- Dropped the irrelevant columns as well

```
for i in null_cols.index:
    if null_cols.loc[i] > 25:
        prev.drop(columns = i ,axis = 1, inplace = True)
```

```
null_cols = prev.isnull().sum()/len(prev)*100
null_cols
```

```
SK_ID_PREV                   0.000000
SK_ID_CURR                   0.000000
NAME_CONTRACT_TYPE           0.020716
AMT_ANNUITY                 22.286665
AMT_APPLICATION              0.000000
AMT_CREDIT                   0.000060
AMT_GOODS_PRICE             23.081773
```

```
# dropping the irrelevant columns
cols=['WEEKDAY_APPR_PROCESS_START','HOUR_APPR_PROCES
prev.drop(columns=cols,axis=1,inplace=True)
```

```
prev = pd.read_csv("C:/Users/chpsy/Downloads/assignment/previ
```

```
prev.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1670214 entries, 0 to 1670213
Data columns (total 37 columns):
 #   Column                      Non-Null Count    Dtype
---  ------                      --------------    -----
 0   SK_ID_PREV                  1670214 non-null  int64
 1   SK_ID_CURR                  1670214 non-null  int64
 2   NAME_CONTRACT_TYPE          1670214 non-null  object
 3   AMT_ANNUITY                 1297979 non-null  float64
 4   AMT_APPLICATION             1670214 non-null  float64
 5   AMT_CREDIT                  1670213 non-null  float64
 6   AMT_DOWN_PAYMENT            774370 non-null   float64
 7   AMT_GOODS_PRICE             1284699 non-null  float64
 8   WEEKDAY_APPR_PROCESS_START  1670214 non-null  object
 9   HOUR_APPR_PROCESS_START     1670214 non-null  int64
 10  FLAG_LAST_APPL_PER_CONTRACT 1670214 non-null  object
```

```
prev.replace('XNA',np.nan,inplace=True)
prev.replace('XAP',np.nan,inplace=True)
```

```
null_cols = prev.isnull().sum()/len(prev)*100
null_cols
```

```
SK_ID_PREV                   0.000000
SK_ID_CURR                   0.000000
NAME_CONTRACT_TYPE           0.020716
AMT_ANNUITY                 22.286665
AMT_APPLICATION              0.000000
AMT_CREDIT                   0.000060
AMT_DOWN_PAYMENT            53.636480
AMT_GOODS_PRICE             23.081773
WEEKDAY_APPR_PROCESS_START   0.000000
```

HANDLING NULL VALUES

- Checking all the null values in the columns
- Replacing them with the mean and mode of respective columns

```
prev.CNT_PAYMENT.fillna(prev.CNT_PAYMENT.median() , inplace = True)
```

```
prev.CNT_PAYMENT.isnull().sum()
```

```
0
```

```
# Filling with mode values as it is a categorical variable
```

```
prev.NAME_PORTFOLIO.fillna(prev.NAME_PORTFOLIO.mode()[0], inplace = True)
```

```
prev.NAME_PORTFOLIO.isnull().sum()
```

```
0
```

```
prev.AMT_GOODS_PRICE.fillna(prev.AMT_GOODS_PRICE.median(),inplace = True)
```

```
prev.AMT_GOODS_PRICE.isnull().sum()
```

```
0
```

```
prev.AMT_ANNUITY.fillna(prev.AMT_ANNUITY.median(),inplace = True)
```

```
prev.AMT_ANNUITY.isnull().sum()
```

```
0
```

Handling missing values in PRODUCT_

AMT_CREDIT & NAME_CONTRACT_TY

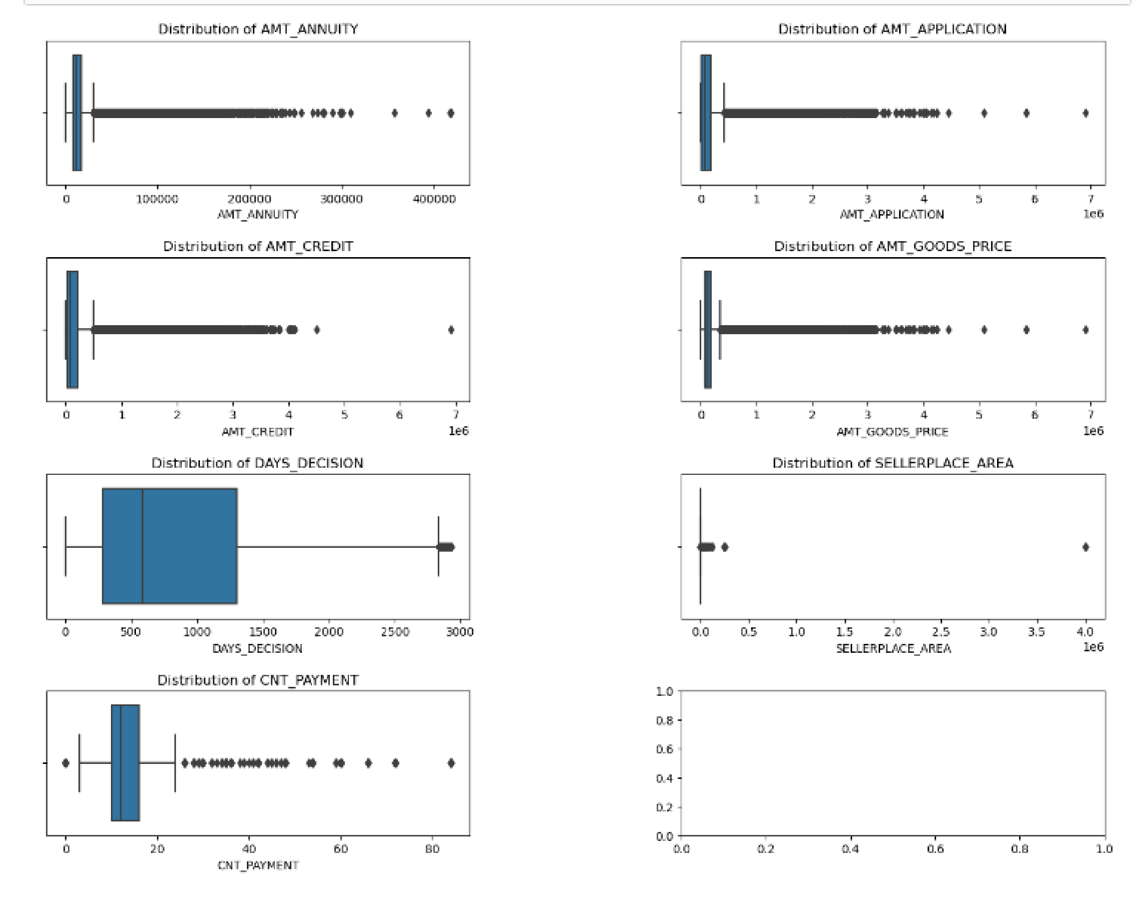
```
t = ["PRODUCT_COMBINATION", "NAME_
for i in t:
    prev  = prev[~(prev[i].isna())
```

```
prev.isna().sum()
```

```
SK_ID_PREV                  0
SK_ID_CURR                  0
NAME_CONTRACT_TYPE          0
AMT_ANNUITY            370849
AMT_APPLICATION             0
AMT_CREDIT                  0
AMT_GOODS_PRICE        384163
NAME_CONTRACT_STATUS        0
DAYS_DECISION               0
NAME_CLIENT_TYPE            0
NAME_PORTFOLIO         370844
CHANNEL_TYPE                0
SELLERPLACE_AREA            0
CNT_PAYMENT            370844
PRODUCT_COMBINATION         0
dtype: int64
```

OUTLIER ANALYSIS & MERGING BOTH DATASETS

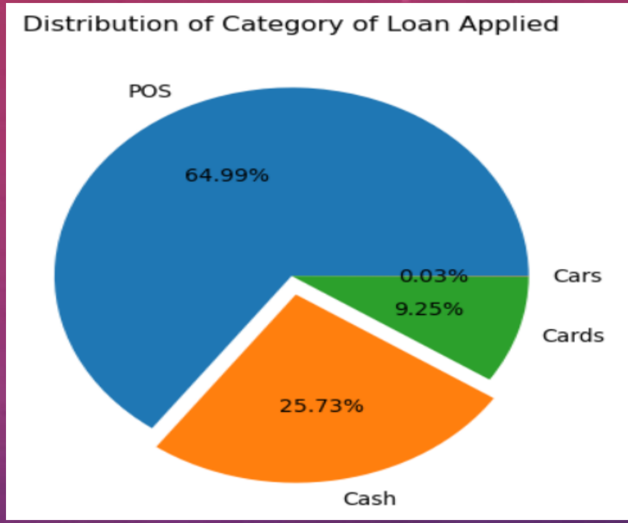
- We can see there are outliers in almost all the columns
- The Annuity , Credit, Application amount, Goods price all have linear outliers.
- We will not be ignoring or deleting the outliers because the data is related to finance
- We are dropping the common columns and are merging both the datasets.

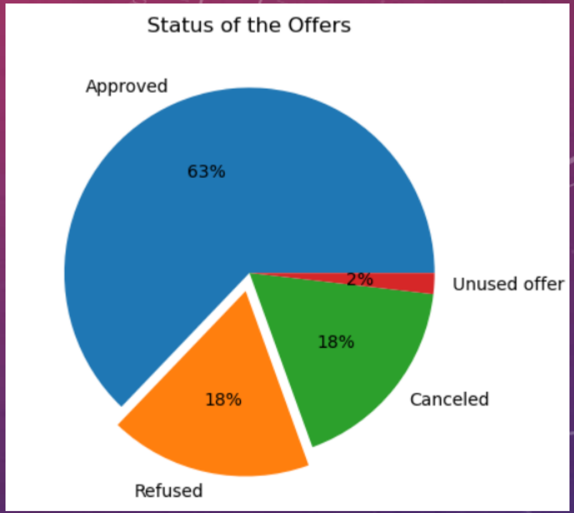


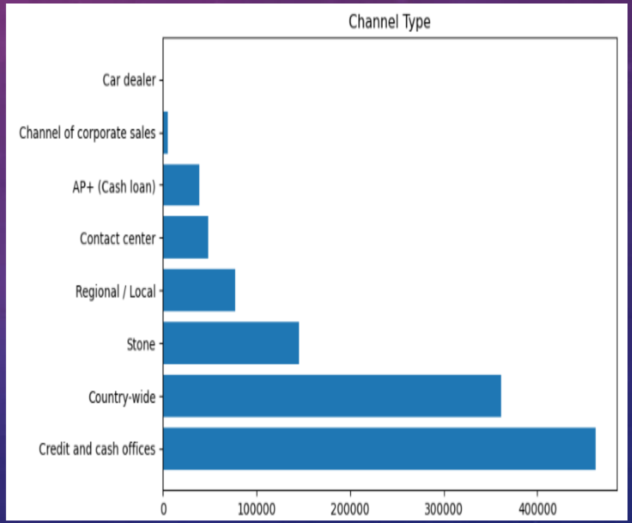

```python
#we will be dropping the common columns in both datasets as we will be
# doing inner join from prev dataset

t = ['NAME_CONTRACT_TYPE', 'AMT_ANNUITY', 'AMT_CREDIT', 'AMT_GOODS_PRICE']
right = prev.drop(columns=t)

merged=pd.merge(left=df2,right=right,how='inner',on='SK_ID_CURR')
merged.head()
```

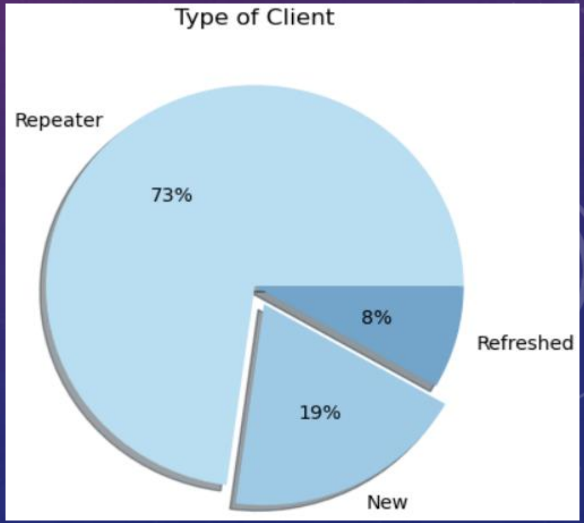
UNIVARIATE ANALYSIS

- We can see major loans are of POS category with 65% of the total loans
- About 62% of the loan applications are approve and 2% of the loans approved are not used.
- About 73% of the clients were repeaters.
- Most of the applicants took loan from credit and cash offices

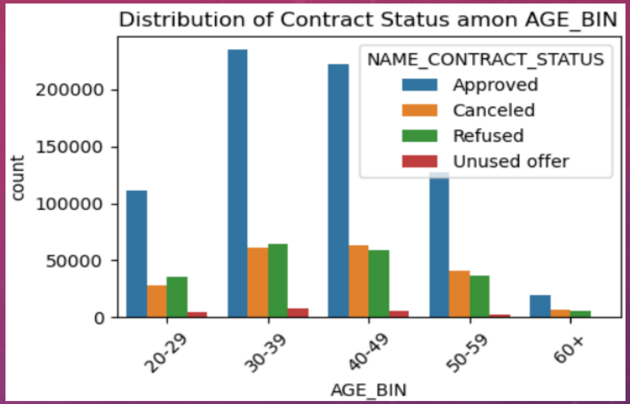


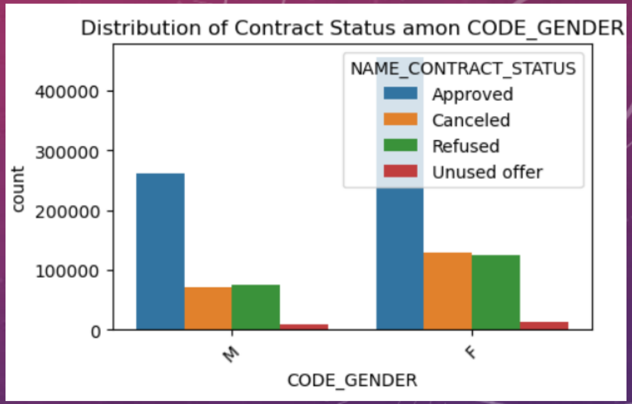




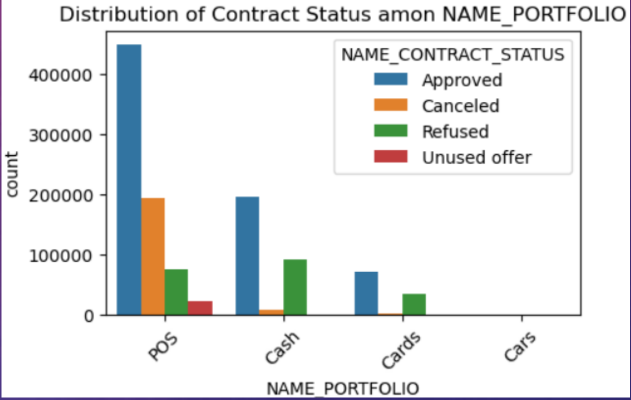


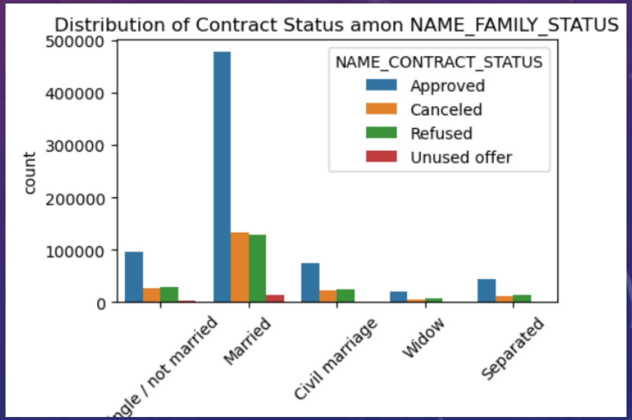
BIVARIATE ANALYSIS





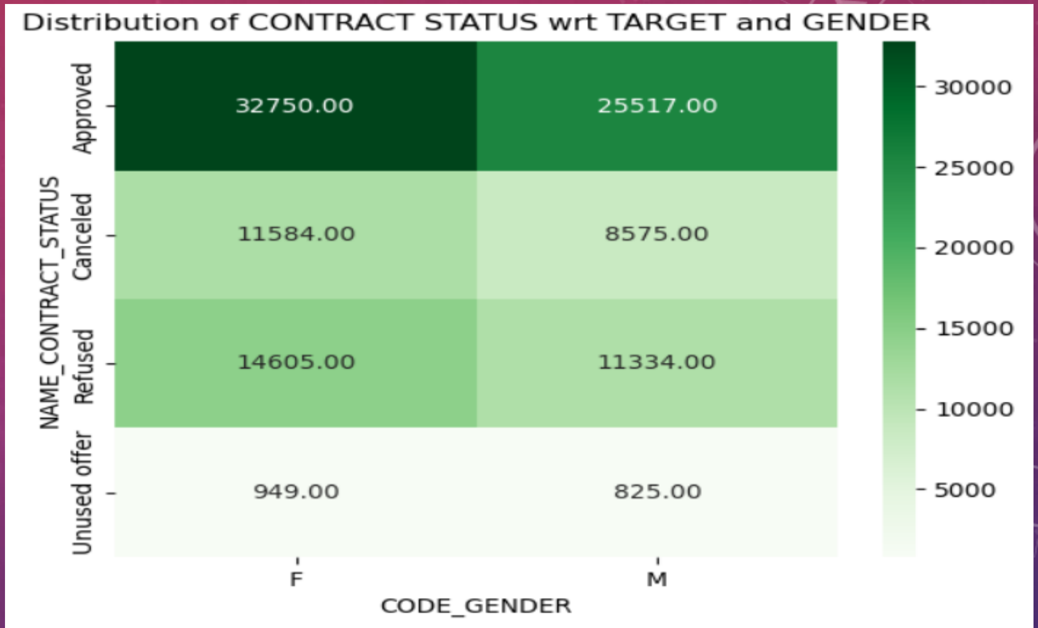
- Most of the approved loans are from 30-39, 40-49 age group.
- Most of the approved loan portfolios are POS
- Most of the approved loans are from Married people.

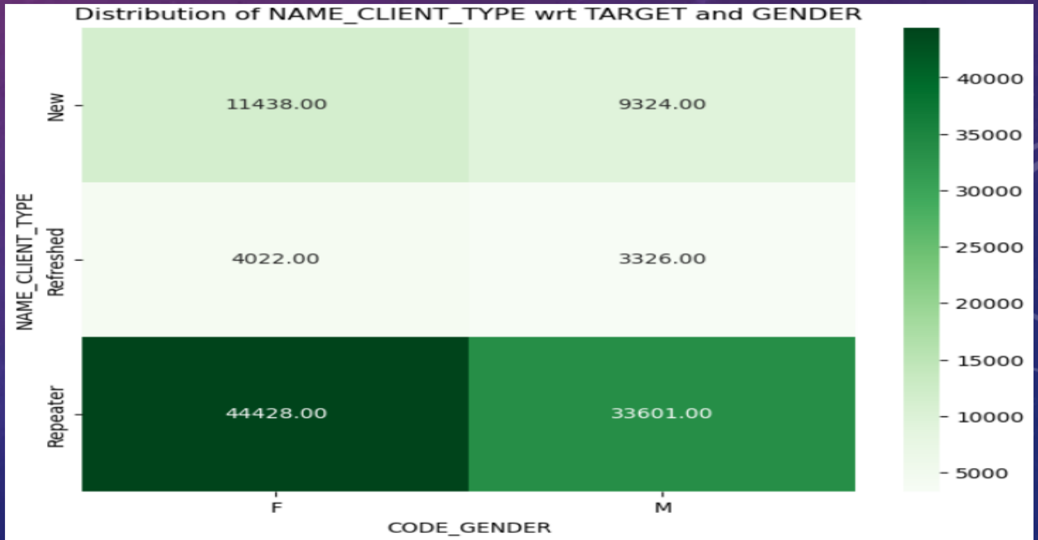




MULTIVARIATE ANALYSIS

- 32750 applications of Females were approved, whereas 25517 applications of Males were approved.
- Most of the repeaters were females.
- Most of the refused applicants were also females.





MULTIVARIATE ANALYSIS

- Most of the females and males who defaulted took Credit from cash and credit offices and country wide.
- Most of the defaulters from females took cash loans and POS mobile with interest