Bank Loan Case Study

Bussari Jashwanth Jee

Description

- In this project Bank loan case study analysis has been performed.
- This project will give insight and answers to the below Questions
- **Identify Missing Data and Deal with it Appropriately:**
- **Identify Outliers in the Dataset**
- **Analyze Data Imbalance**
- **Perform Univariate, Segmented Univariate, and Bivariate Analysis**
- **Identify Top Correlations for Different Scenarios**

Approach

- First I downloaded the dataset.
- Read the dataset un jupyter notebook.
- Using pandas data analysis has been done.
- Identified outliers and filled missing values by suitable methods.
- Done univariate and bivariate analysis.
- Used seaborn and matplotlib for graphical representation.

Tech-Stack Used

Jupyter Notebook

- This tool is used to clean data and remove unnecessary columns and filling null values using Pandas library .

Identify Missing Data

- Finding and Handling missing data.
- Removing unwanted columns.
- Dropped the columns which has null value percent more than 40%.
- Standardized values.
- Data imputation
- Initial data shape.

```
: data1.shape
: (49999, 73)
```

Finding and Handling missing data

Finding and handling missing data(dataset-1)

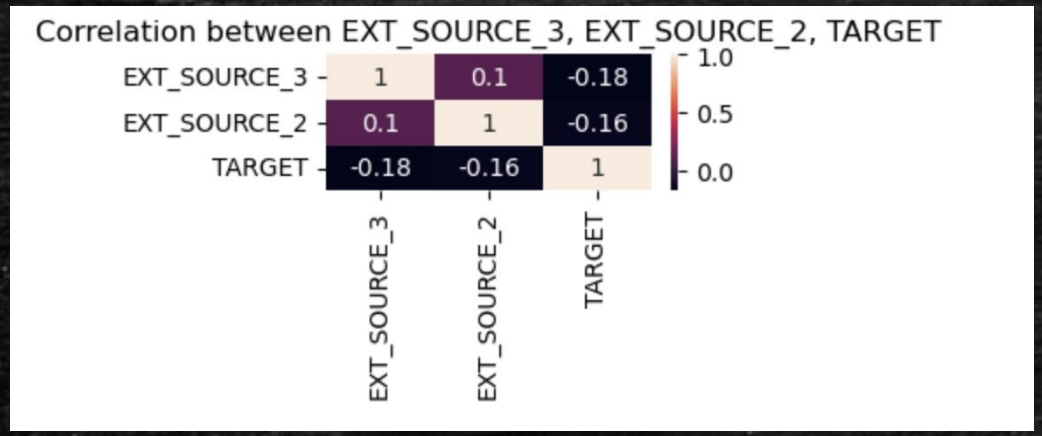
- There are 49 columns with more than 40% of null values.
- After removing these columns
  updated shape of the data is

```
data1.shape
```

```
(49999, 73)
```

```
following columns have null value more than 40%
Index(['COMMONAREA_MEDI', 'COMMONAREA_AVG', 'COMMONAREA_MODE',
       'NONLIVINGAPARTMENTS_MODE', 'NONLIVINGAPARTMENTS_AVG',
       'NONLIVINGAPARTMENTS_MEDI', 'LIVINGAPARTMENTS_MODE',
       'LIVINGAPARTMENTS_AVG', 'LIVINGAPARTMENTS_MEDI', 'FONDKAPREMONT_MODE',
       'FLOORSMIN_AVG', 'FLOORSMIN_MODE', 'FLOORSMIN_MEDI', 'YEARS_BUILD_MEDI',
       'YEARS_BUILD_MODE', 'YEARS_BUILD_AVG', 'OWN_CAR_AGE', 'LANDAREA_MEDI',
       'LANDAREA_MODE', 'LANDAREA_AVG', 'BASEMENTAREA_MEDI',
       'BASEMENTAREA_AVG', 'BASEMENTAREA_MODE', 'EXT_SOURCE_1',
       'NONLIVINGAREA_MODE', 'NONLIVINGAREA_AVG', 'NONLIVINGAREA_MEDI',
       'ELEVATORS_MEDI', 'ELEVATORS_AVG', 'ELEVATORS_MODE',
       'WALLSMATERIAL_MODE', 'APARTMENTS_MEDI', 'APARTMENTS_AVG',
       'APARTMENTS_MODE', 'ENTRANCES_MEDI', 'ENTRANCES_AVG', 'ENTRANCES_MODE',
       'LIVINGAREA_AVG', 'LIVINGAREA_MODE', 'LIVINGAREA_MEDI',
       'HOUSETYPE_MODE', 'FLOORSMAX_MODE', 'FLOORSMAX_MEDI', 'FLOORSMAX_AVG',
       'YEARS_BEGINEXPLUATATION_MODE', 'YEARS_BEGINEXPLUATATION_MEDI',
       'YEARS_BEGINEXPLUATATION_AVG', 'TOTALAREA_MODE', 'EMERGENCYSTATE_MODE'],
      dtype='object')
No. of columns with more than 40% missing value: 49
```

As in the dictionary we see a normalised data set of ext source 1, because ext source 2 and ext source 3 have no linear correlation with target the the column ext source 2 and ext source 3 are dropped
After dropping we have 71 columns



Remaining null values

OCCUPATION_TYPE	31.31
EXT_SOURCE_3	19.89
AMT_REQ_CREDIT_BUREAU_YEAR	13.47
AMT_REQ_CREDIT_BUREAU_QRT	13.47
AMT_REQ_CREDIT_BUREAU_MON	13.47
AMT_REQ_CREDIT_BUREAU_WEEK	13.47
AMT_REQ_CREDIT_BUREAU_DAY	13.47
AMT_REQ_CREDIT_BUREAU_HOUR	13.47
NAME_TYPE_SUITE	0.38
OBS_30_CNT_SOCIAL_CIRCLE	0.34
DEF_30_CNT_SOCIAL_CIRCLE	0.34
OBS_60_CNT_SOCIAL_CIRCLE	0.34
DEF_60_CNT_SOCIAL_CIRCLE	0.34
EXT_SOURCE_2	0.25
AMT_GOODS_PRICE	0.08

dtype: float64

Updated data shape

```
data1.shape
```

```
(49999, 71)
```

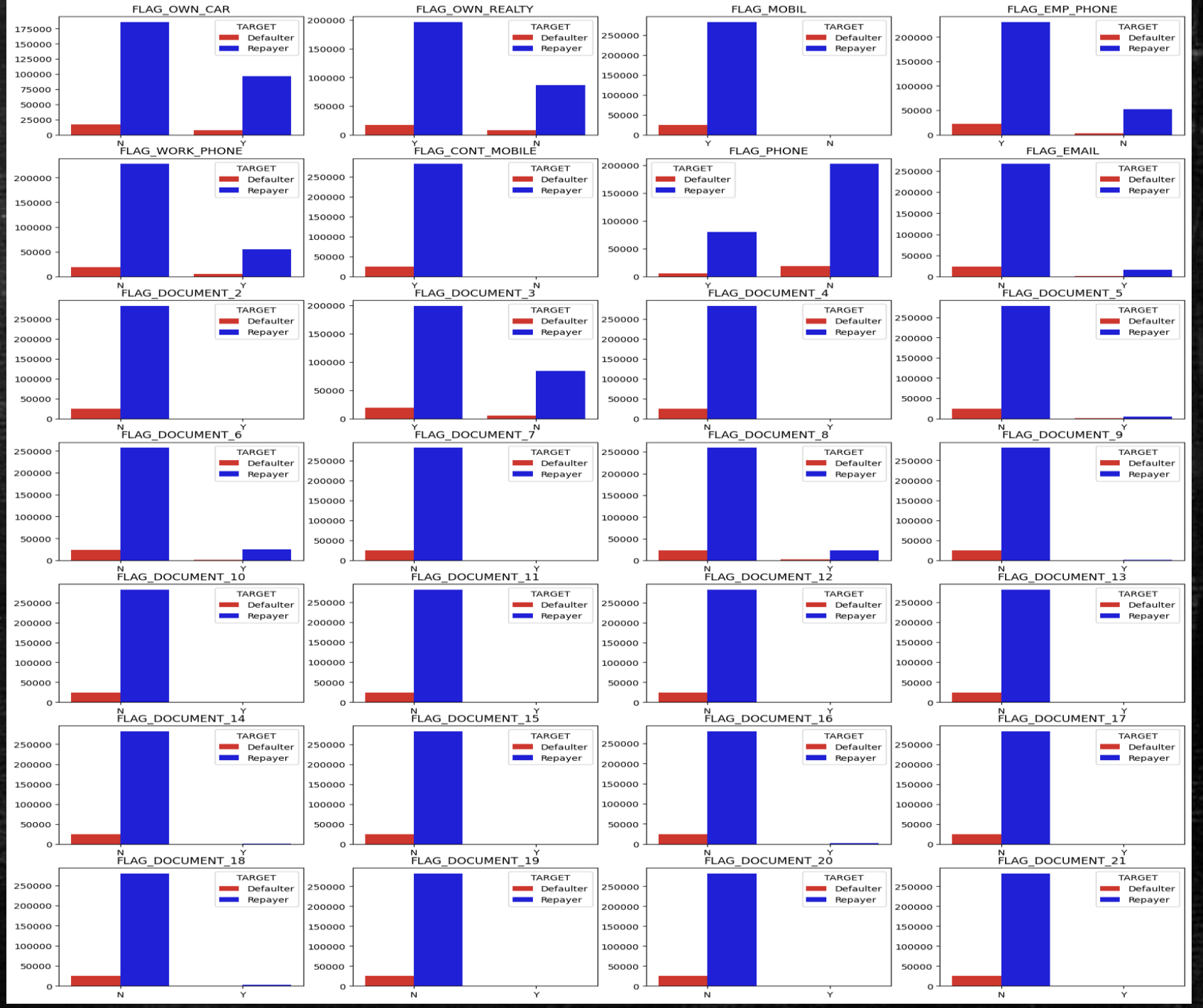
FLAG_DOCUMENT_3,
FLAG_OWN_REALTY,
FLAG_MOBIL,
FLAG_OWN_CAR

Make more sense thus we can include these columns and remove all other FLAG columns for further analysis.

Hence we drop these columns and after removing all unnecessary columns now we have 47 relevant columns

```
data1.shape
```

```
(49999, 47)
```



Data Imputation(d-1)

- •In the column occupation type there are 31.35% null values hence we will change those to 'Unknown', this unknown column has the highest percentage.
- •Similarly in the column Name Type Suite, the missing values will be replaced with "Unaccompanied" which is the mode of the data.

```
data1["OCCUPATION_TYPE"].value_counts()
```

```
Laborers                 8952
Sales staff              5160
Core staff               4434
Managers                 3489
Drivers                  3044
High skill tech staff    1852
Accountants              1621
Medicine staff           1403
Security staff           1140
Cooking staff             963
Cleaning staff            739
Private service staff     447
Low-skill Laborers        357
Waiters/barmen staff      228
Secretaries               212
Realty agents             123
HR staff                  101
IT staff                   80
Name: OCCUPATION_TYPE, dtype: int64
```

```
data1["OCCUPATION_TYPE"]=data1["OCCUPATION_TYPE"].fillna("Unknown")
```

```
AMT=["AMT_REQ_CREDIT_BUREAU_YEAR","A
```

```
data1[AMT].median()
```

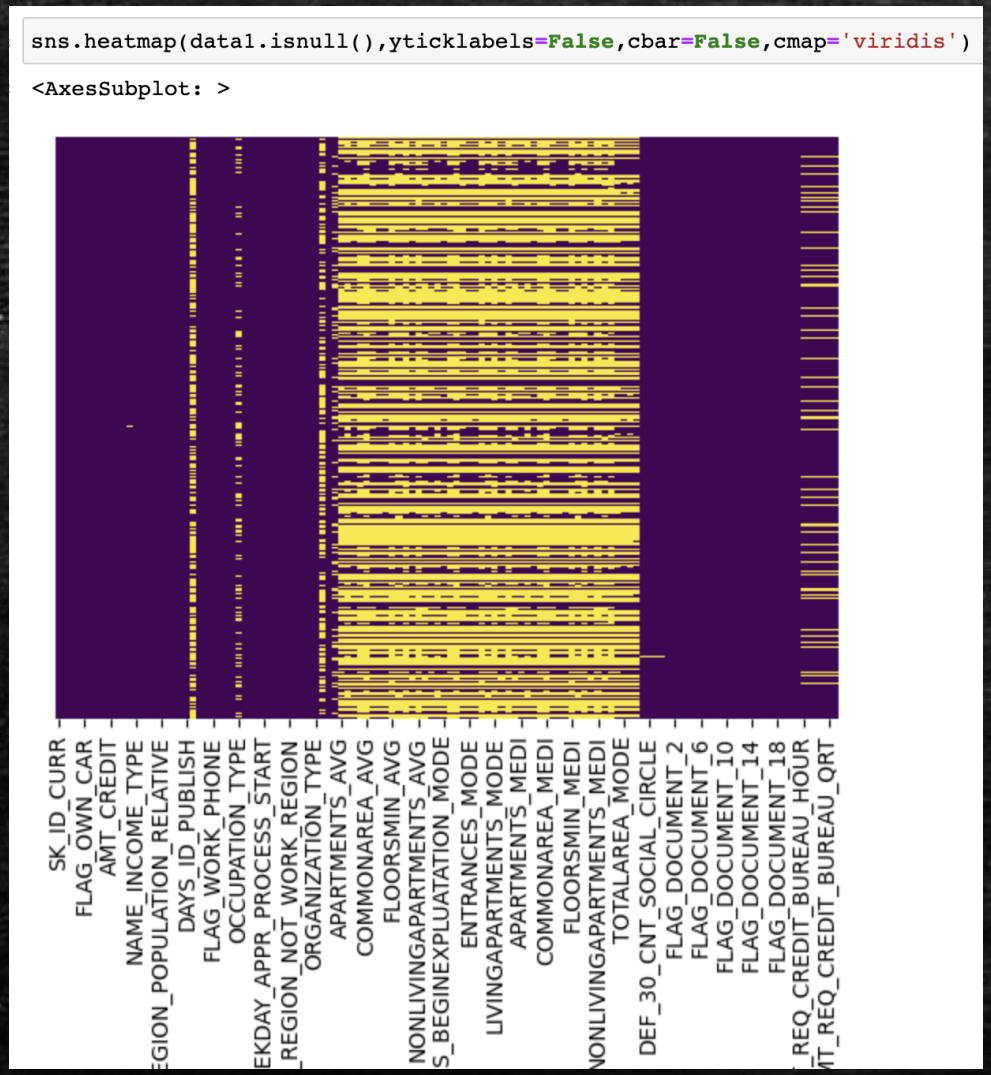
```
AMT_REQ_CREDIT_BUREAU_YEAR    1.0
AMT_REQ_CREDIT_BUREAU_QRT     0.0
AMT_REQ_CREDIT_BUREAU_MON     0.0
AMT_REQ_CREDIT_BUREAU_WEEK    0.0
AMT_REQ_CREDIT_BUREAU_DAY     0.0
AMT_REQ_CREDIT_BUREAU_HOUR    0.0
dtype: float64
```

For rest of the columns with null value we will replace them with their median values.

Null values before and after dataset - 1

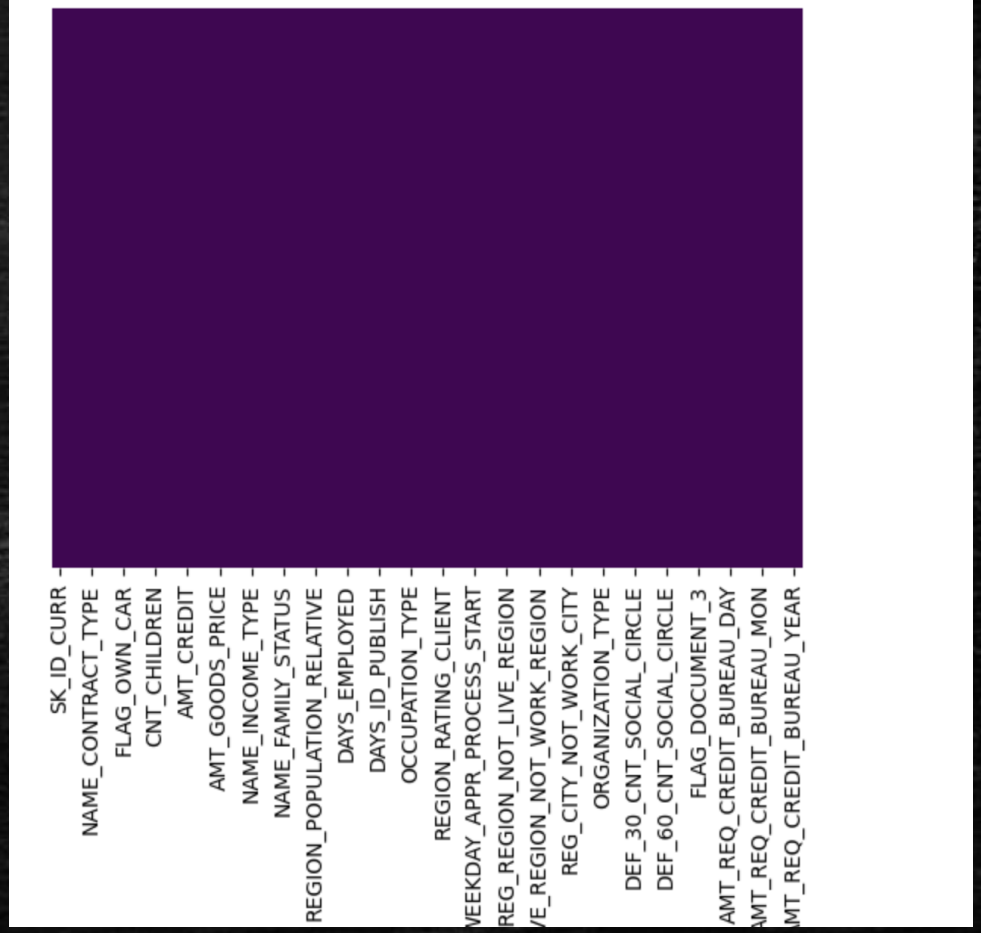
```
sns.heatmap(data1.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

```
<AxesSubplot: >
```




```
sns.heatmap(data1.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

```
<AxesSubplot: >
```



Standardizing values Dataset-1

columns AMT_INCOME_TOTAL, AMT_CREDIT, AMT_GOODS_PRICE have very high values, thus will make these numerical columns in categorical columns for better understanding by BINNING

```
data1['AMT_INCOME_TOTAL']=data1['AMT_INCOME_TOTAL']/100000
```

```
data1['AMT_INCOME_TOTAL'].max()
```

```
1170.0
```

```
bins = [0,1,2,3,4,5,6,7,8,9,10,1170]#1170 is the largest value
slot = ['0-1L','1L-2L', '2L-3L','3L-4L','4L-5L','5L-6L','6L-7L','7L-8L','8L-9L','9L-10L','10L Above']

data1['AMT_INCOME_RANGE']=pd.cut(data1['AMT_INCOME_TOTAL'],bins,labels=slot)
```

```
data1['AMT_GOODS_PRICE']=data1['AMT_GOODS_PRICE']/100000
data1['AMT_GOODS_PRICE'].max()
```

```
40.5
```

```
bins = [0,1,2,3,4,5,6,7,8,9,10,41] #40.5 is the highest value
slots = ['0-1L','1L-2L', '2L-3L','3L-4L','4L-5L','5L-6L','6L-7L','7L-8L','8L-9L','9L-10L','10L Above']

data1['AMT_GOODS_PRICE_RANGE']=pd.cut(data1['AMT_GOODS_PRICE'],bins=bins,labels=slots)
```

```
days_col = ["DAYS_BIRTH", "DAYS_EMPLOYED", "DAYS_REGISTRATION", "DAYS_ID_PUBLISH", "DAYS_LAST_PHONE_CHANGE"]
```

```
data1[days_col]=abs(data1[days_col])
```

```
data1["AGE"] = abs(data1["DAYS_BIRTH"]/365)
```

```
data1["AGE"].max()
```

```
68.9972602739726
```

```
bins = [0,20,25,30,35,40,45,50,55,60,100]
slots = ["0-20","20-25","25-30","30-35","35-40","40-45","45-50","50-55","55-60","60 Above"]
data1["AGE_GROUP"] = pd.cut(data1["AGE"], bins=bins, labels=slots)
```

Finding and handling missing data(dataset-2)

- We will drop those columns with missing value % greater than 50%
- Initial data shape

```
data2.shape
```

```
(49999, 37)
```

```
nullval(data2)[nullval(data2)>0]
```

```
RATE_INTEREST_PRIVILEGED      99.67
RATE_INTEREST_PRIMARY         99.67
RATE_DOWN_PAYMENT             50.40
AMT_DOWN_PAYMENT              50.40
NAME_TYPE_SUITE               48.49
NFLAG_INSURED_ON_APPROVAL     38.32
DAYS_FIRST_DRAWING            38.32
DAYS_FIRST_DUE                38.32
DAYS_LAST_DUE_1ST_VERSION     38.32
DAYS_LAST_DUE                 38.32
DAYS_TERMINATION              38.32
AMT_GOODS_PRICE               21.49
AMT_ANNUITY                   21.18
CNT_PAYMENT                   21.18
PRODUCT_COMBINATION            0.02
dtype: float64
```

Removed 4 columns with null values greater than 50% , now there are 33 columns left

RATE_INTEREST_PRIVILEGED	99.67
RATE_INTEREST_PRIMARY	99.67
RATE_DOWN_PAYMENT	50.40
AMT_DOWN_PAYMENT	50.40

Data imputation

- In column Name_type_suit, unknown values filled
- In column N_FLAG_INSURED_ON_APPROVAL null values were filled with mode i.e. 0
- For AMT_ANNUITY as the kde plot is skewed with 1peak, null values were replaced by median

```
data2["NAME_TYPE_SUITE"] = data2["NAME_TYPE_SUITE"].fillna("Unknown")
```

```
data2['NFLAG_INSURED_ON_APPROVAL'].value_counts()
```

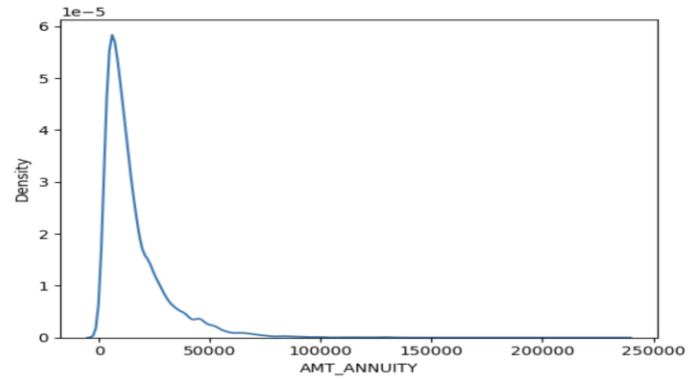
```
0.0    20898
1.0     9941
Name: NFLAG_INSURED_ON_APPROVAL, dtype: int64
```

```
data2["NFLAG_INSURED_ON_APPROVAL"] = data2["NFLAG_INSURED_ON_APPROVAL"].fillna(0)
```

```
data2['NFLAG_INSURED_ON_APPROVAL'].value_counts()
```

```
0.0    40058
1.0     9941
Name: NFLAG_INSURED_ON_APPROVAL, dtype: int64
```

```
sns.kdeplot(data2['AMT_ANNUITY'])
plt.show()
```




```
data2['AMT_ANNUITY'].fillna(data2['AMT_ANNUITY'].median(),inplace = True)
```

Standardizing values

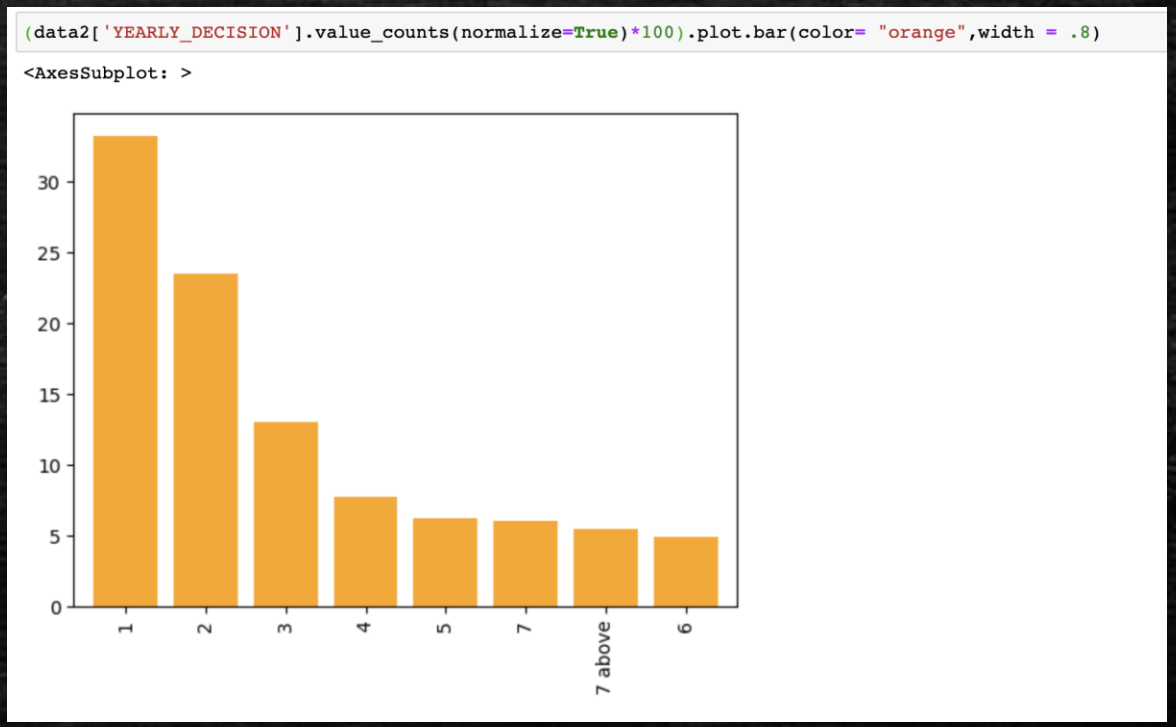
- Standardizing values for days_decision column by converting them into years by binning, for better understanding of data after replacing the null values with mode.

```
#days group calculation

bins = [0,1*365,2*365,3*365,4*365,5*365,6*365,7*365,10*365]
slots = ["1","2","3","4","5","6","7","7 above"]
data2['YEARLY_DECISION'] = pd.cut(data2['DAYS_DECISION'],bins,labels=slots)
```

```
(data2['YEARLY_DECISION'].value_counts(normalize=True)*100).plot.bar(color= "orange",width = .8)
```

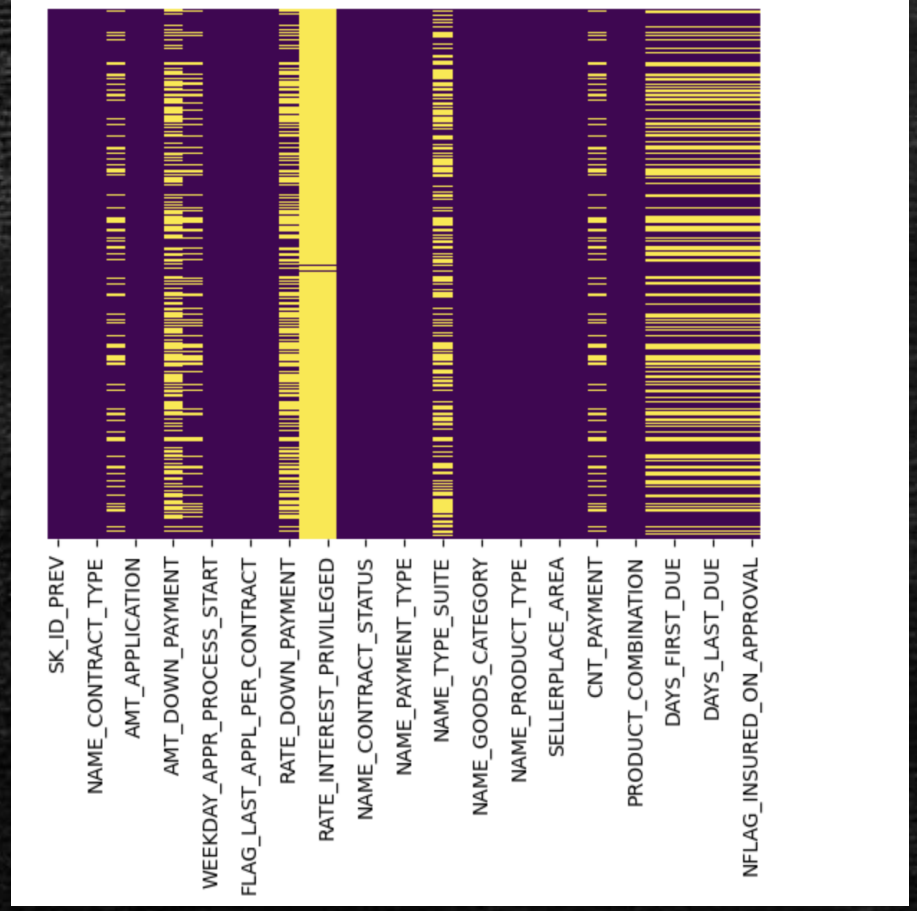
```
<AxesSubplot: >
```



Data cleaning dataset-2 conclusion

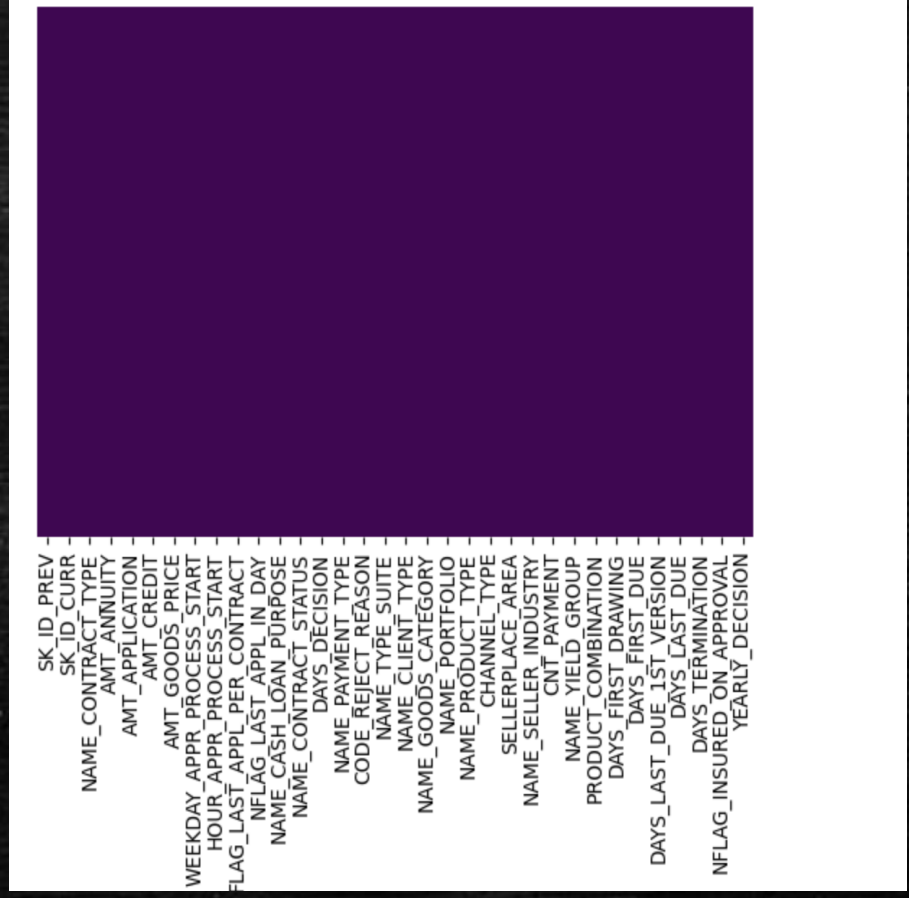
```
sns.heatmap(data2.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

```
<AxesSubplot: >
```




```
sns.heatmap(data2.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

```
<AxesSubplot: >
```



Data cleaning conclusion

- Dataset-1 initial

```
print(data1.shape)
```

```
(49999, 122)
```

- Dataset-1 final

```
data1.shape
```

```
(49999, 47)
```

- Dataset-2 initial

```
data2.shape
```

```
(49999, 37)
```

- Dataset-2 final

```
data2.shape
```

```
(49999, 34)
```

Outlier Identification

OUTLIER IDENTIFICATION
DATA SET-1 APPLICATION DATA

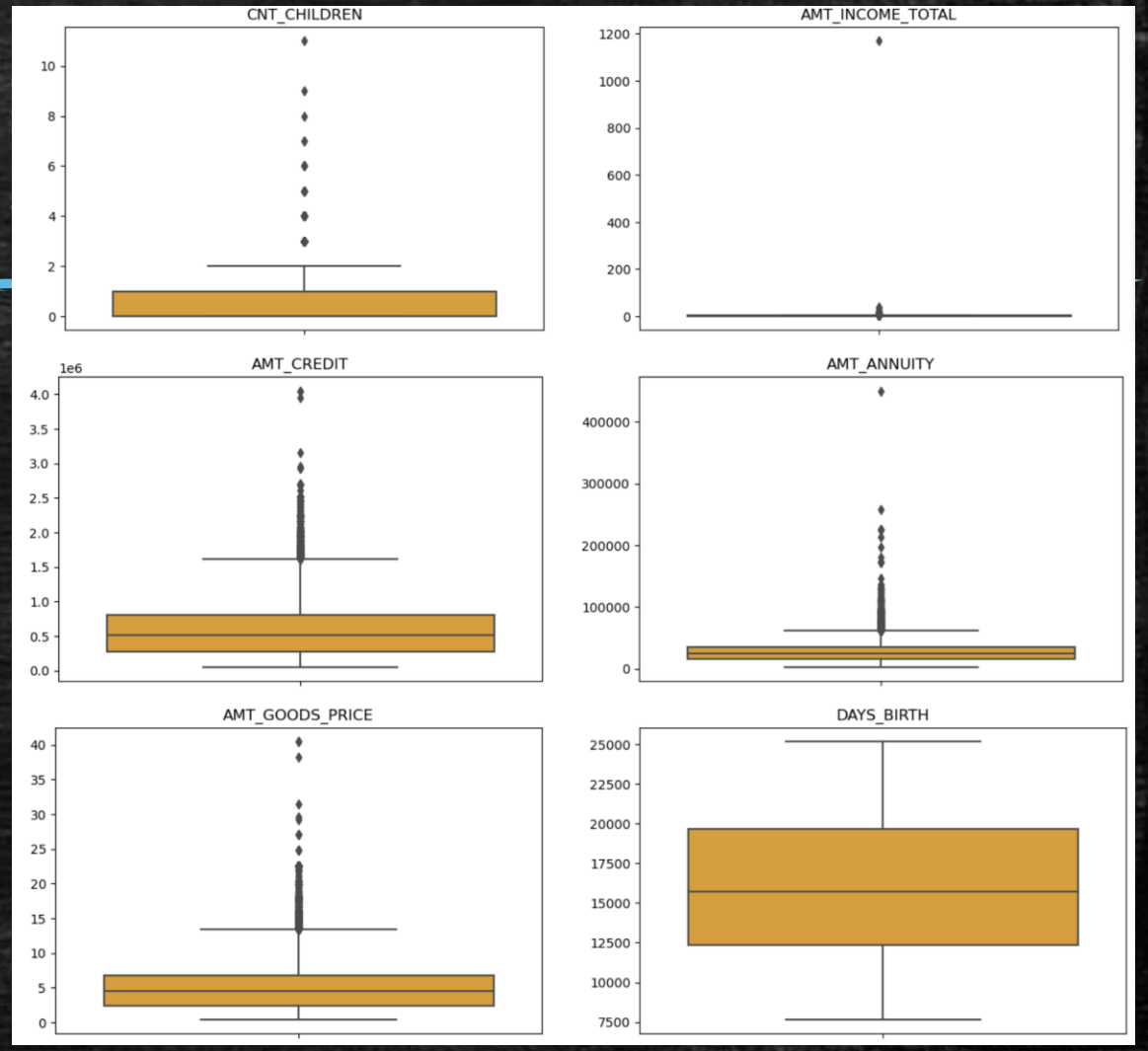
- **From describe we could find all the columns those who have high difference between max and 75 percentile and the ones which makes no sense having max value to be so high are captured below:**

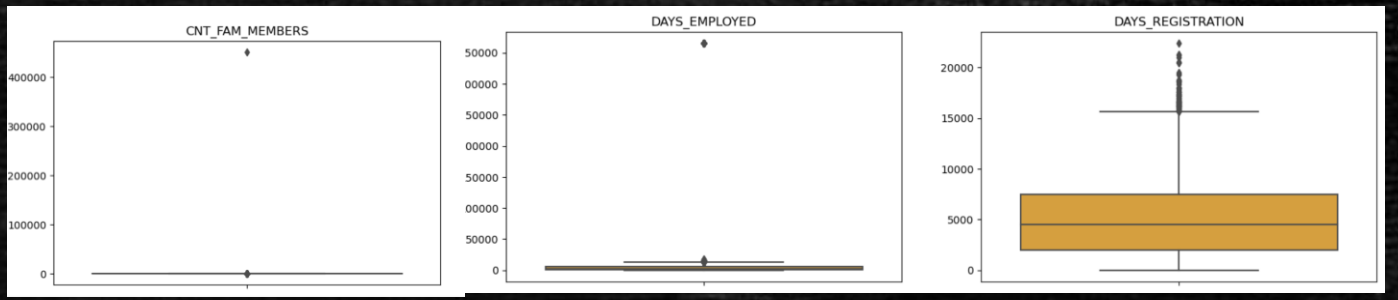
```
outlier_col = ["CNT_CHILDREN","AMT_INCOME_TOTAL", "AMT_CREDIT", "AMT_ANNUITY", "AMT_GOODS_PRICE",
               "DAYS_BIRTH", "DAYS_EMPLOYED", "DAYS_REGISTRATION", "CNT_FAM_MEMBERS"]
```

- The box plot of these columns will reflect if they have outliers or not.

Boxplots for outliers

- AMT_ANNUITY, AMT_CREDIT, AMT_GOODS_PRICE,CNT_CHILDREN have some number of outliers.
- AMT_INCOME_TOTAL has huge number of outliers which indicate that few of the loan applicants have high income when compared to the others.
- DAYS_BIRTH has no outliers which means the data available is reliable.
- DAYS_EMPLOYED has outlier values around 350000(days) which is around 958 years which is impossible and hence this has to be incorrect entry.
- CNT_FAM_MEMBERS has outlier value more than 45000 which is impossible hence it





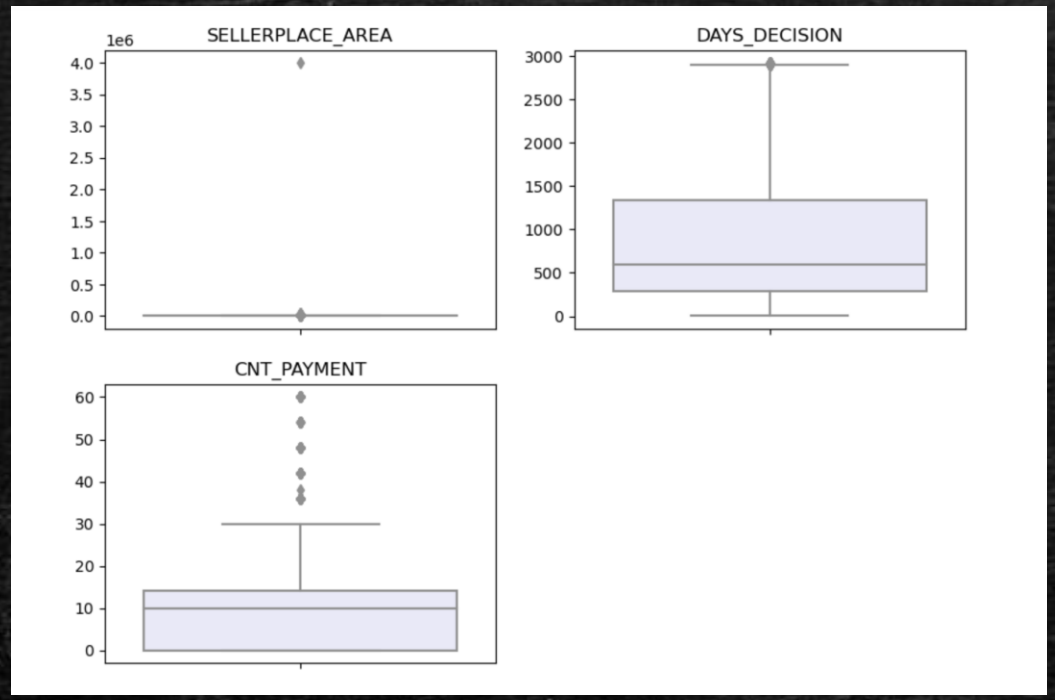
OUTLIER IDENTIFICATION
DATA SET-2 PREVIOUS APPLICATION

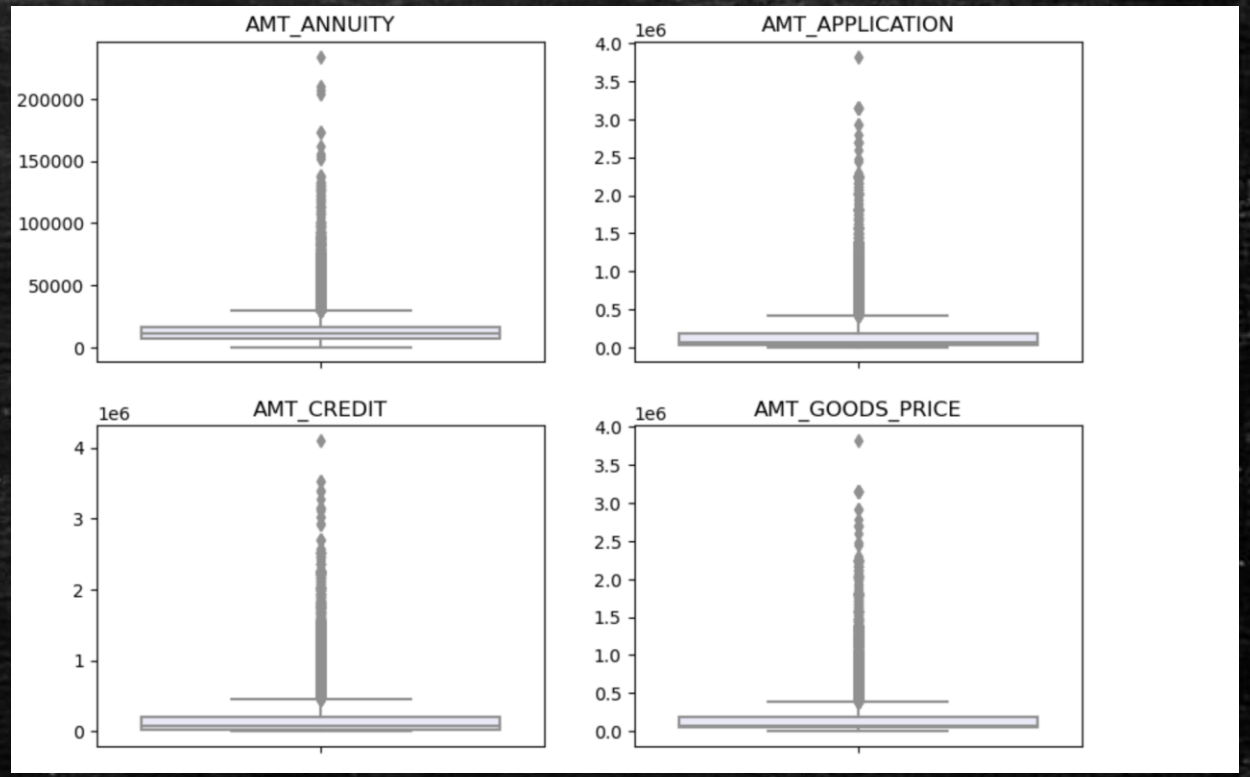
- **From describe we could find all the columns those who have high difference between max and 75 percentile and the ones which makes no sense having max value to be so high are**

```
p_outlier_col = ['AMT_ANNUITY','AMT_APPLICATION','AMT_CREDIT','AMT_GOODS_PRICE',
                 'SELLERPLACE_AREA','DAYS_DECISION','CNT_PAYMENT']
```

- The box plot of these columns will reflect if they have outliers or not.

Boxplot for outliers dataset-2

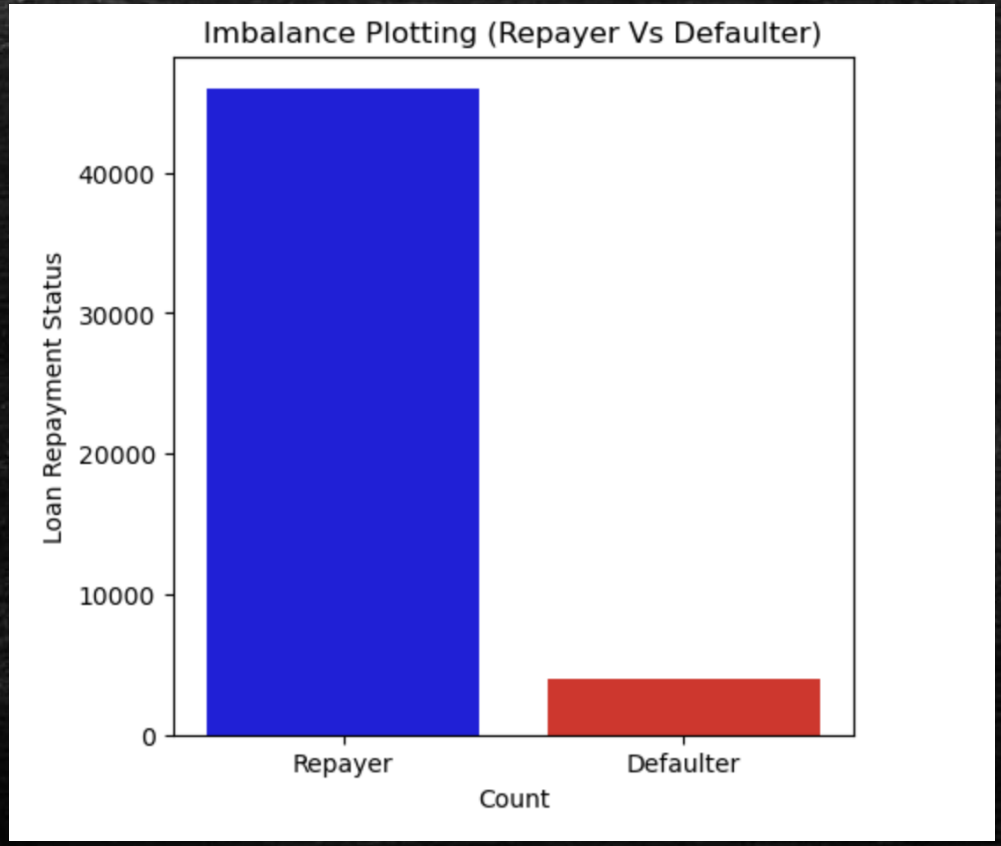




Data Imbalance

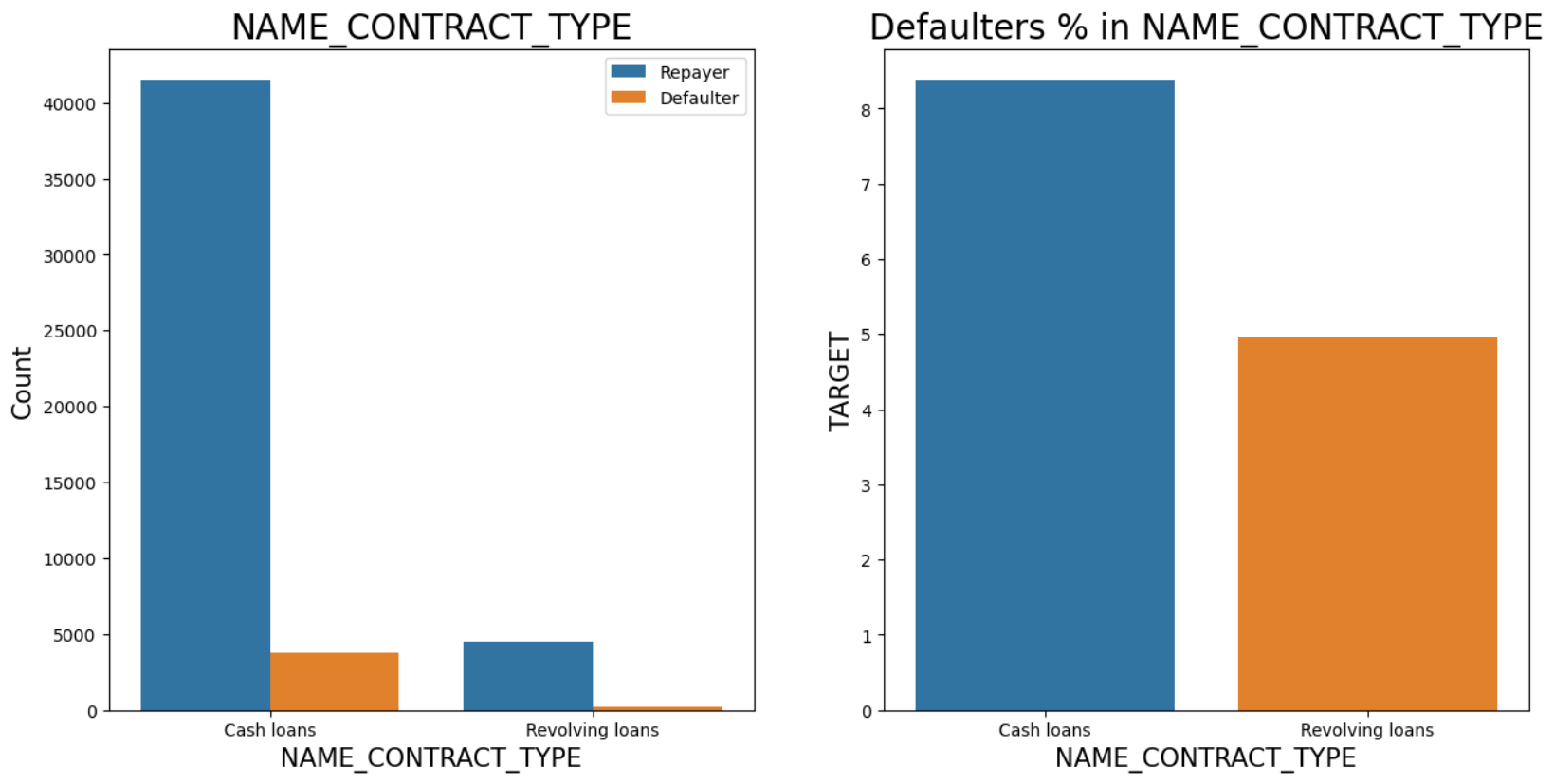
Reporting data imbalance

```
Repayer Percentage is 91.95%
Defaulter Percentage is 8.05%
Imbalance Ratio with respect to Repayer and Defaulter is given: 11.42/1 (approx)
```



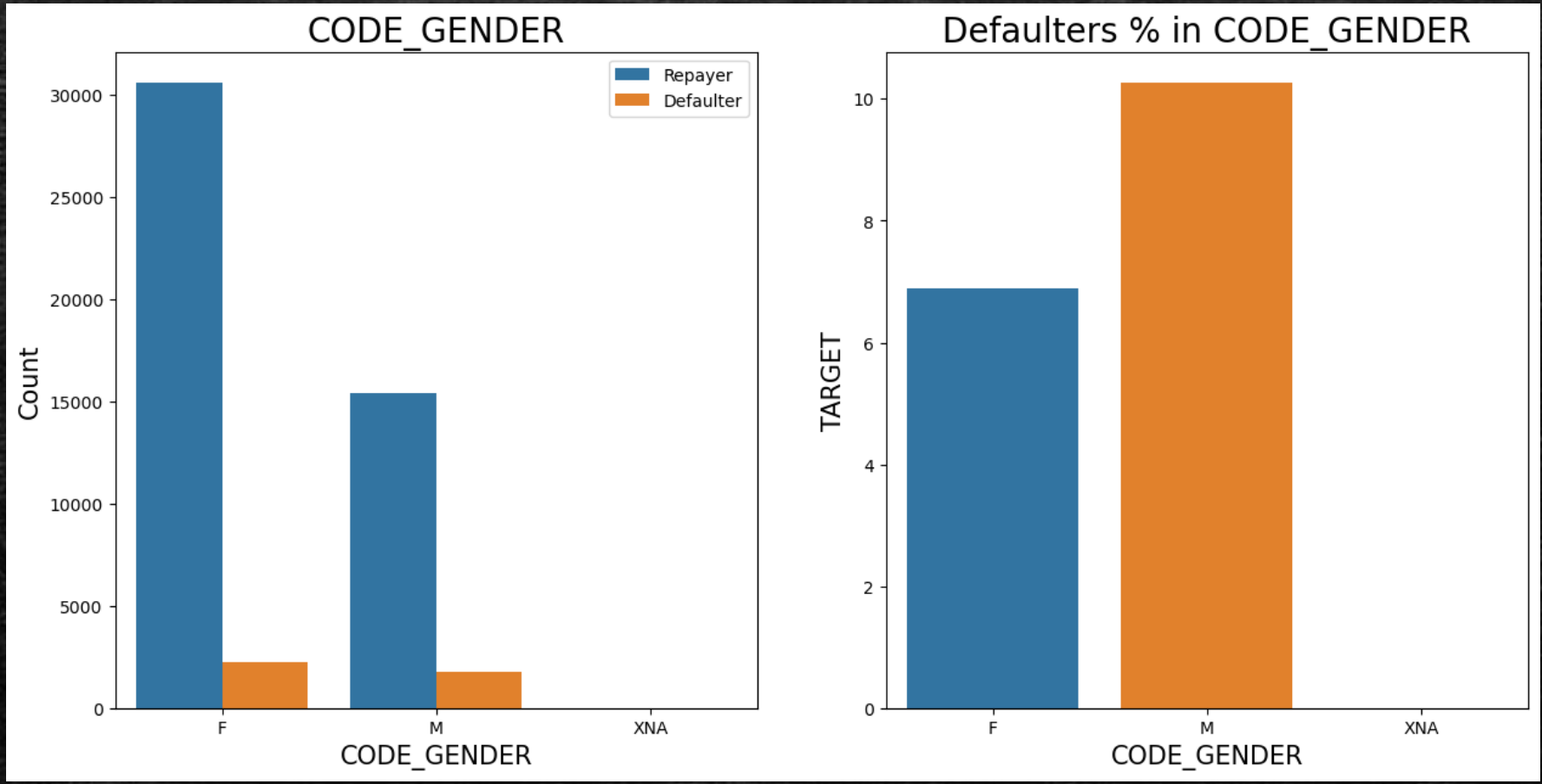
REPORTING RESULTS OF UNIVARIATE,SEGMENTED UNIVARAITE AND BIVARIATE ANALYSIS

Contract type

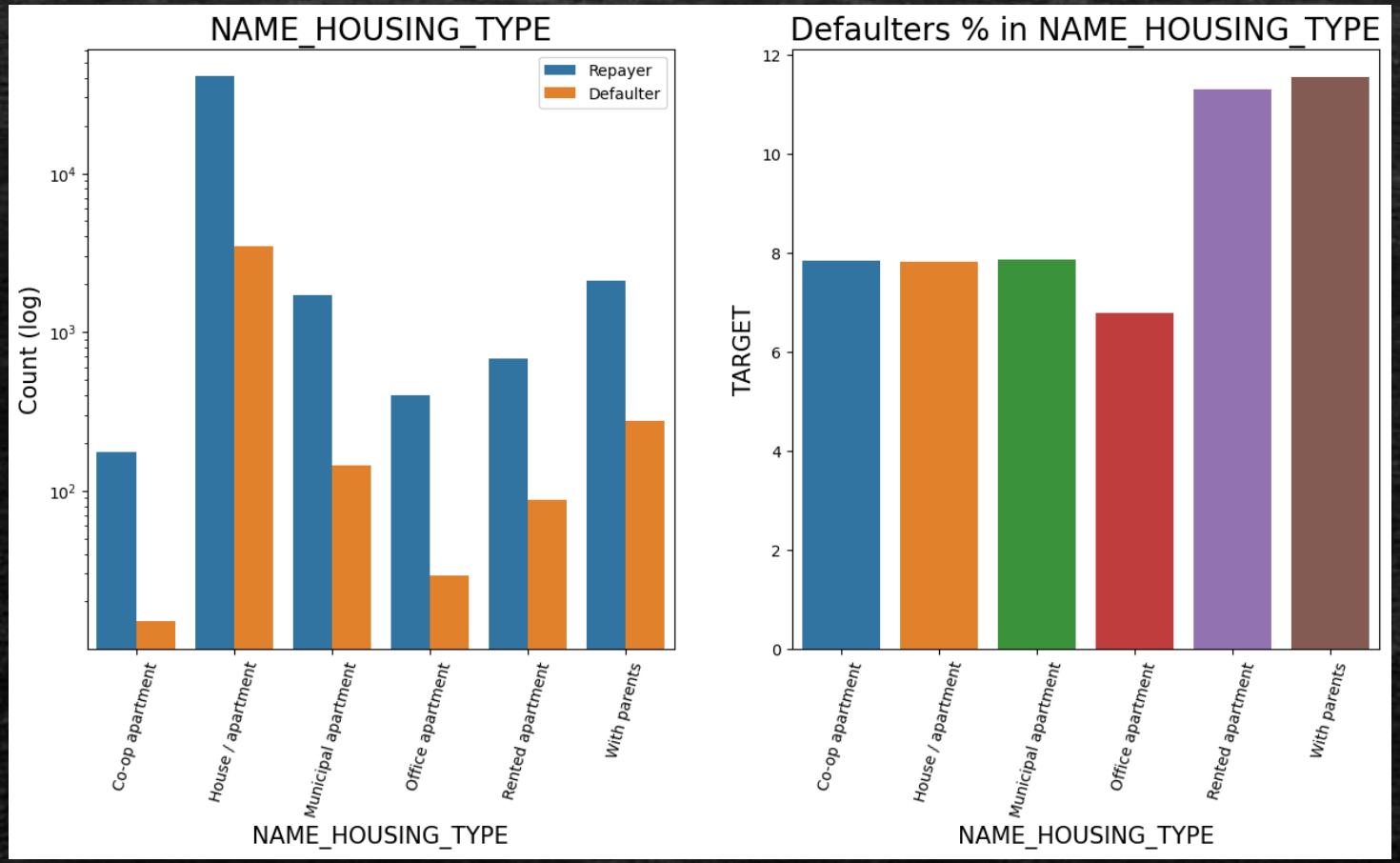


- Revolving loans are just a small fraction (10%) from the total number of loans.
- Around 8-9% Cash loan applicants and 5-6% Revolving loan applicant are in defaulters.

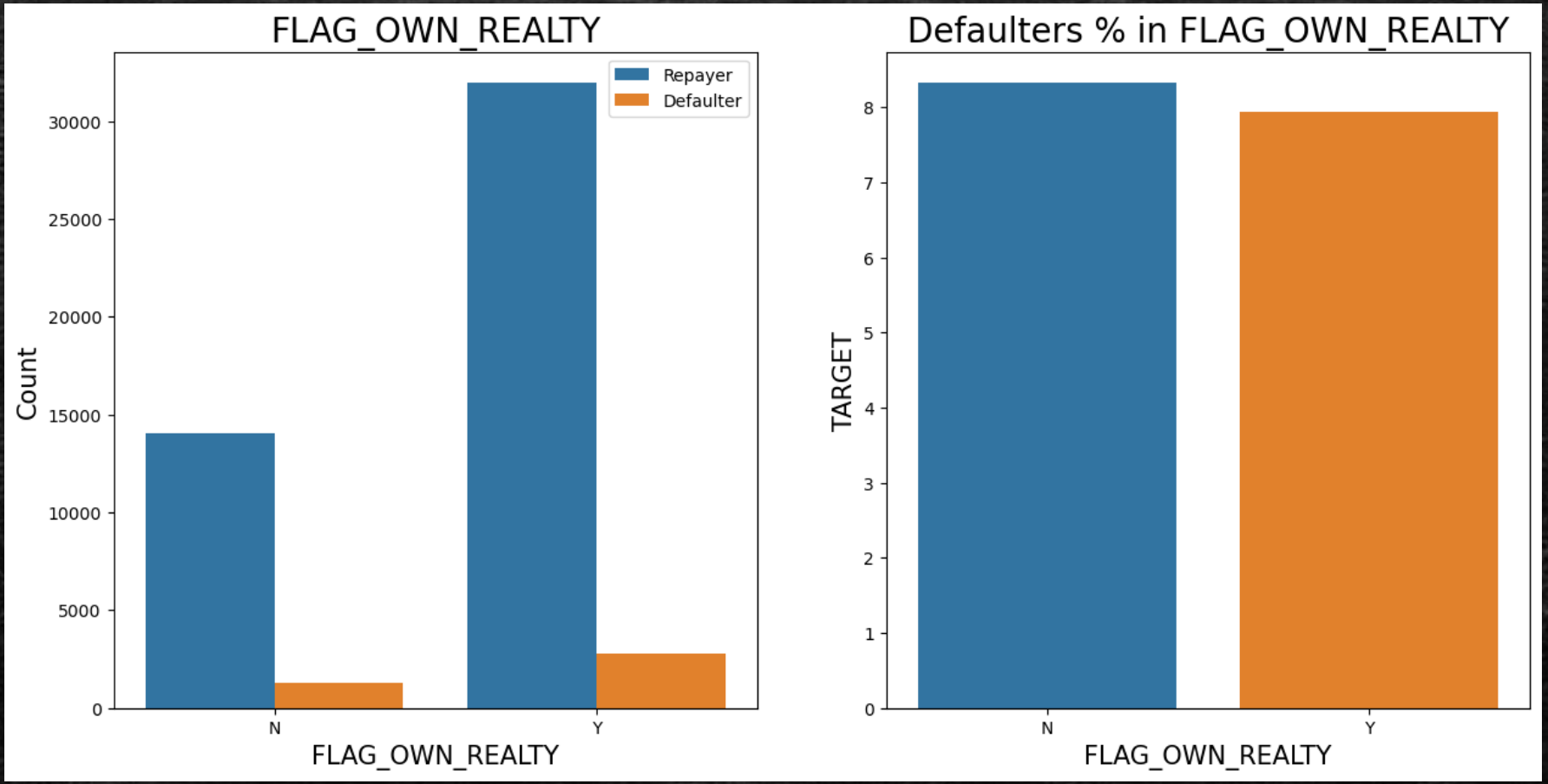
Gender Wise


CODE_GENDER
Repayer
Defaulter
30000
25000
20000
15000
10000
5000
0
Count
F
M
XNA
CODE_GENDER
Defaulters % in CODE_GENDER
10
8
6
4
2
0
TARGET
F
M
XNA
CODE_GENDER

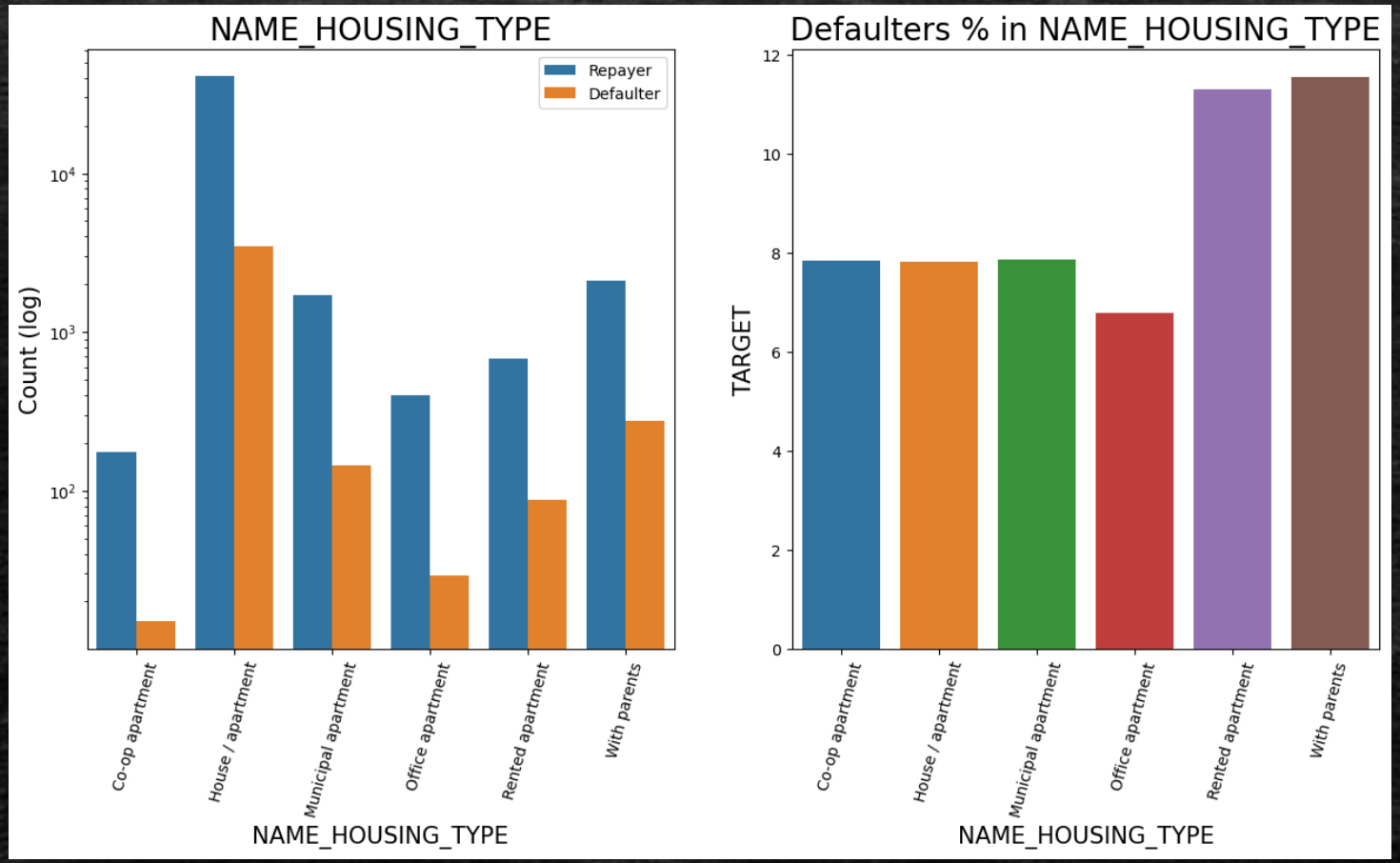
House type


NAME_HOUSING_TYPE
Repayer
Defaulter
Count (log)
10^4
10^3
10^2
Co-op apartment
House / apartment
Municipal apartment
Office apartment
Rented apartment
With parents
NAME_HOUSING_TYPE
Defaulters % in NAME_HOUSING_TYPE
TARGET
12
10
8
6
4
2
0
Co-op apartment
House / apartment
Municipal apartment
Office apartment
Rented apartment
With parents
NAME_HOUSING_TYPE

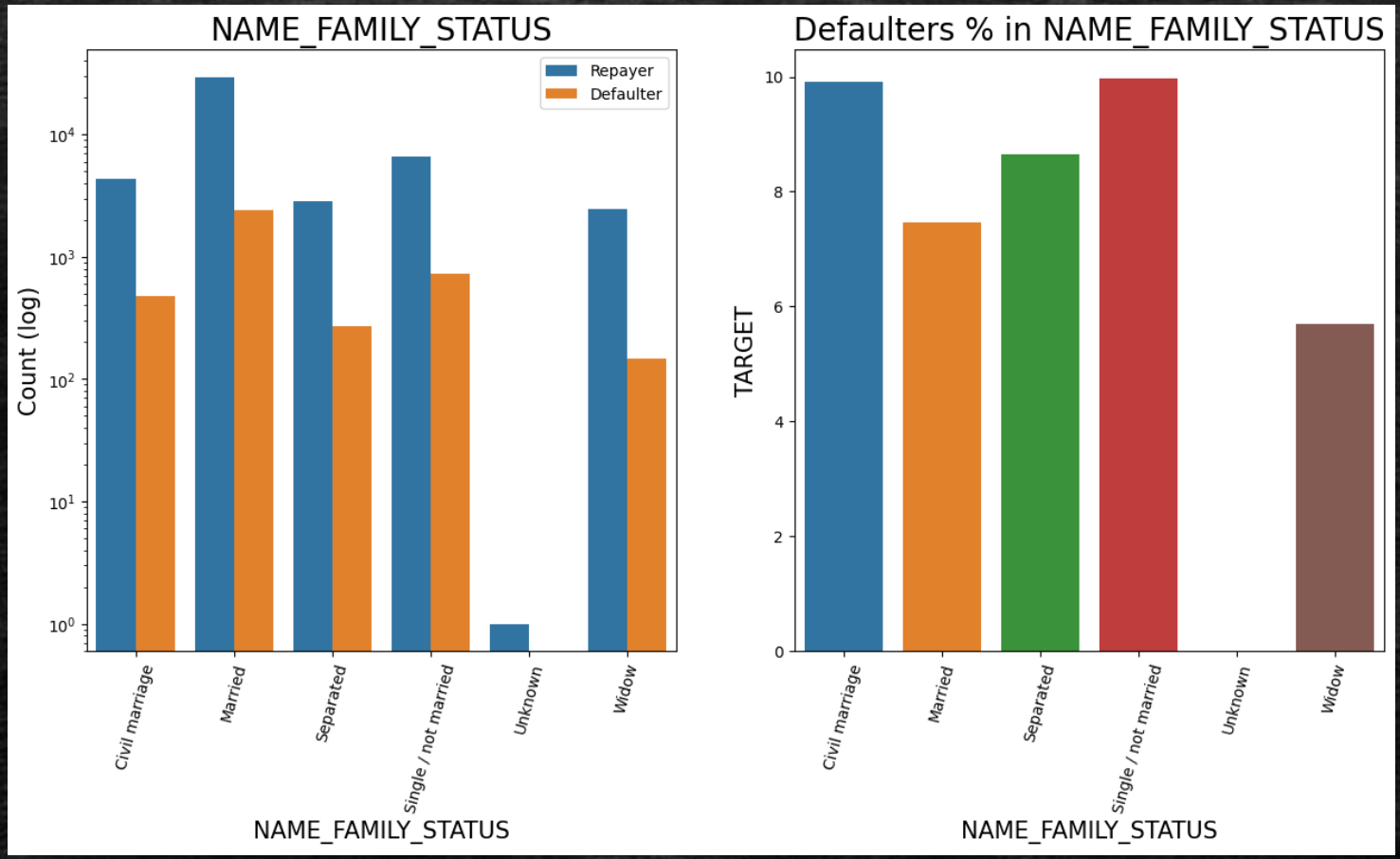
Own Real-estate



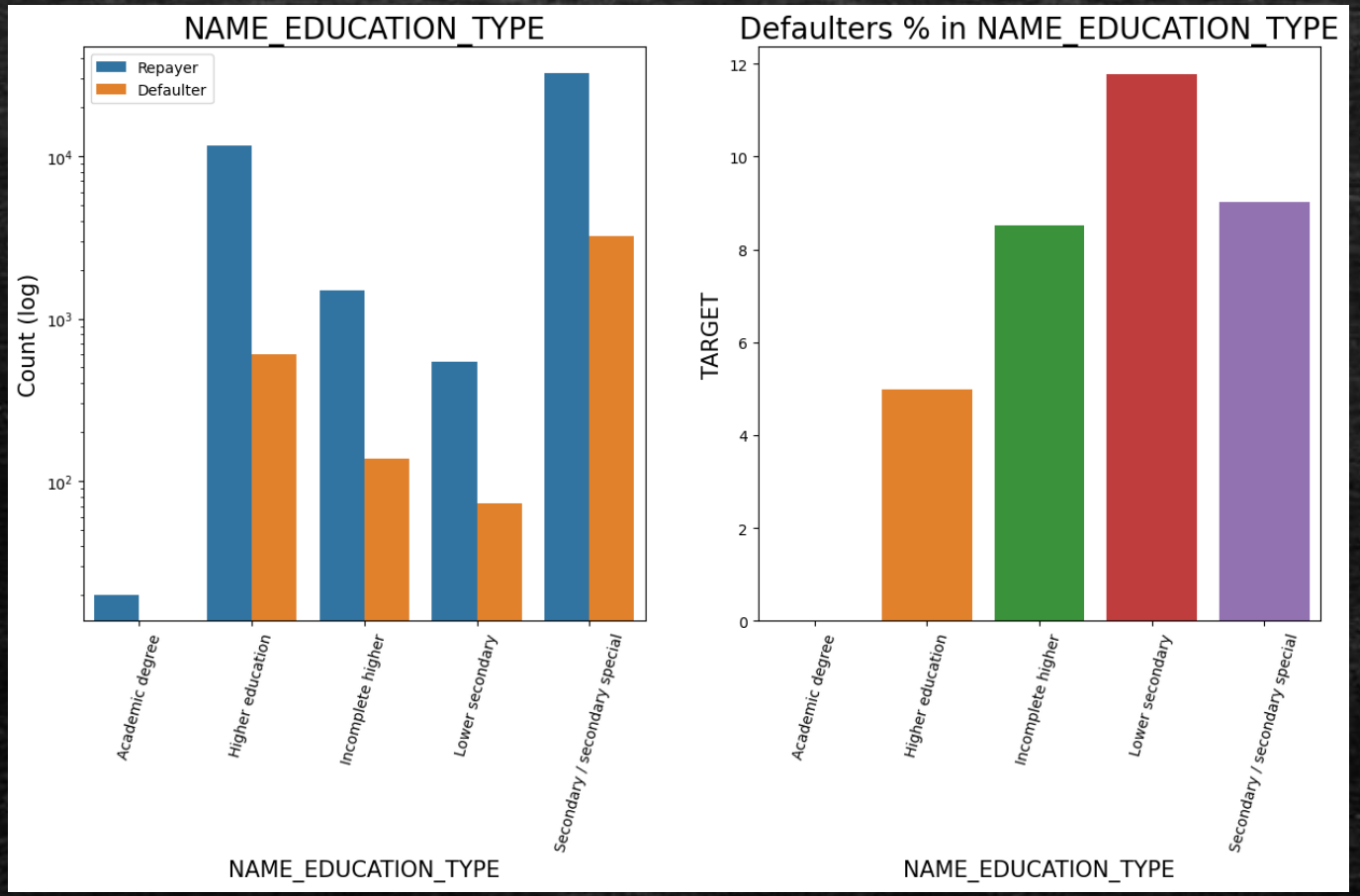
Housing type



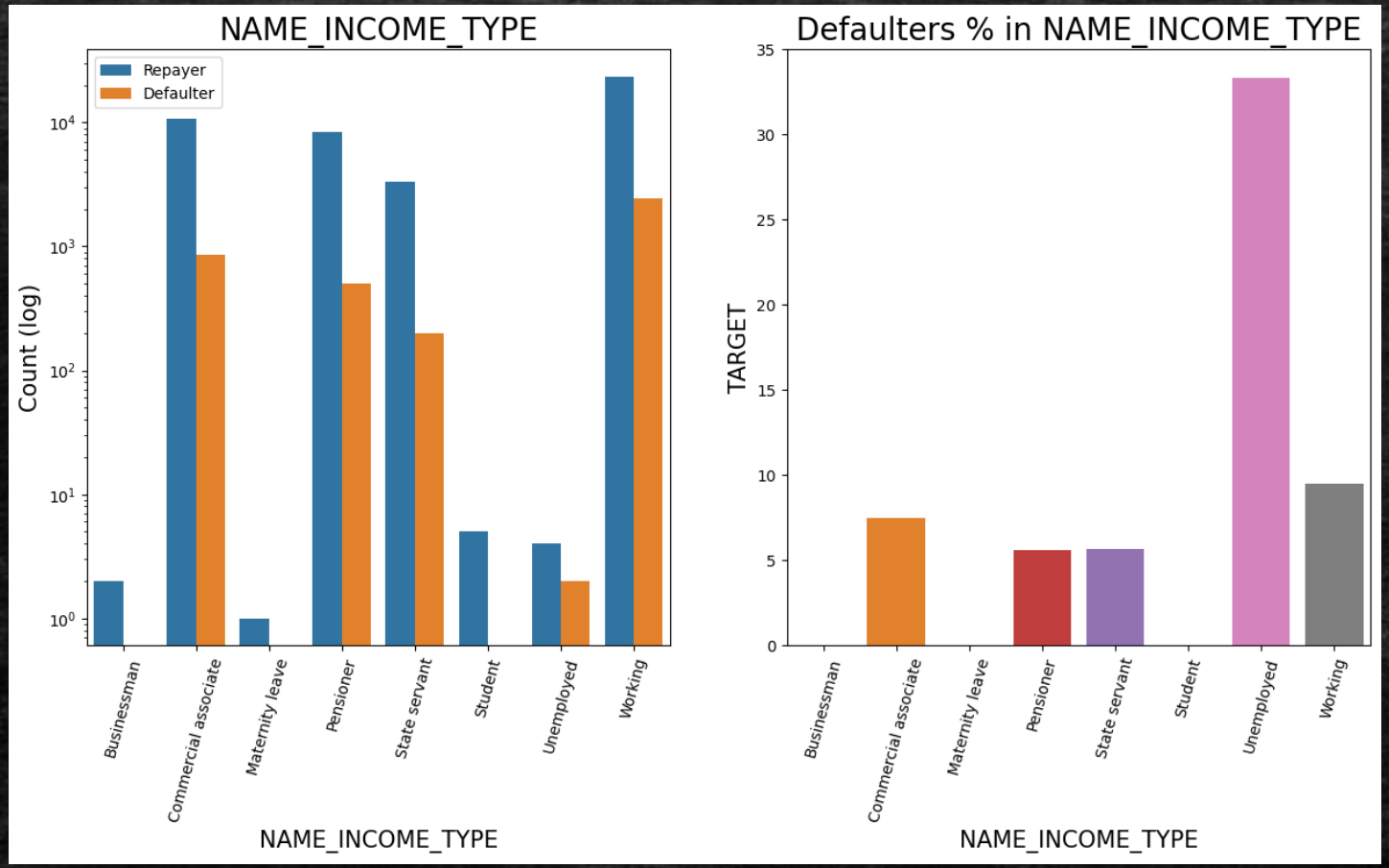
Family Type


NAME_FAMILY_STATUS
Repayer
Defaulter
Count (log)
10^4
10^3
10^2
10^1
10^0
Civil marriage
Married
Separated
Single / not married
Unknown
Widow
NAME_FAMILY_STATUS
Defaulters % in NAME_FAMILY_STATUS
TARGET
10
8
6
4
2
0
Civil marriage
Married
Separated
Single / not married
Unknown
Widow
NAME_FAMILY_STATUS

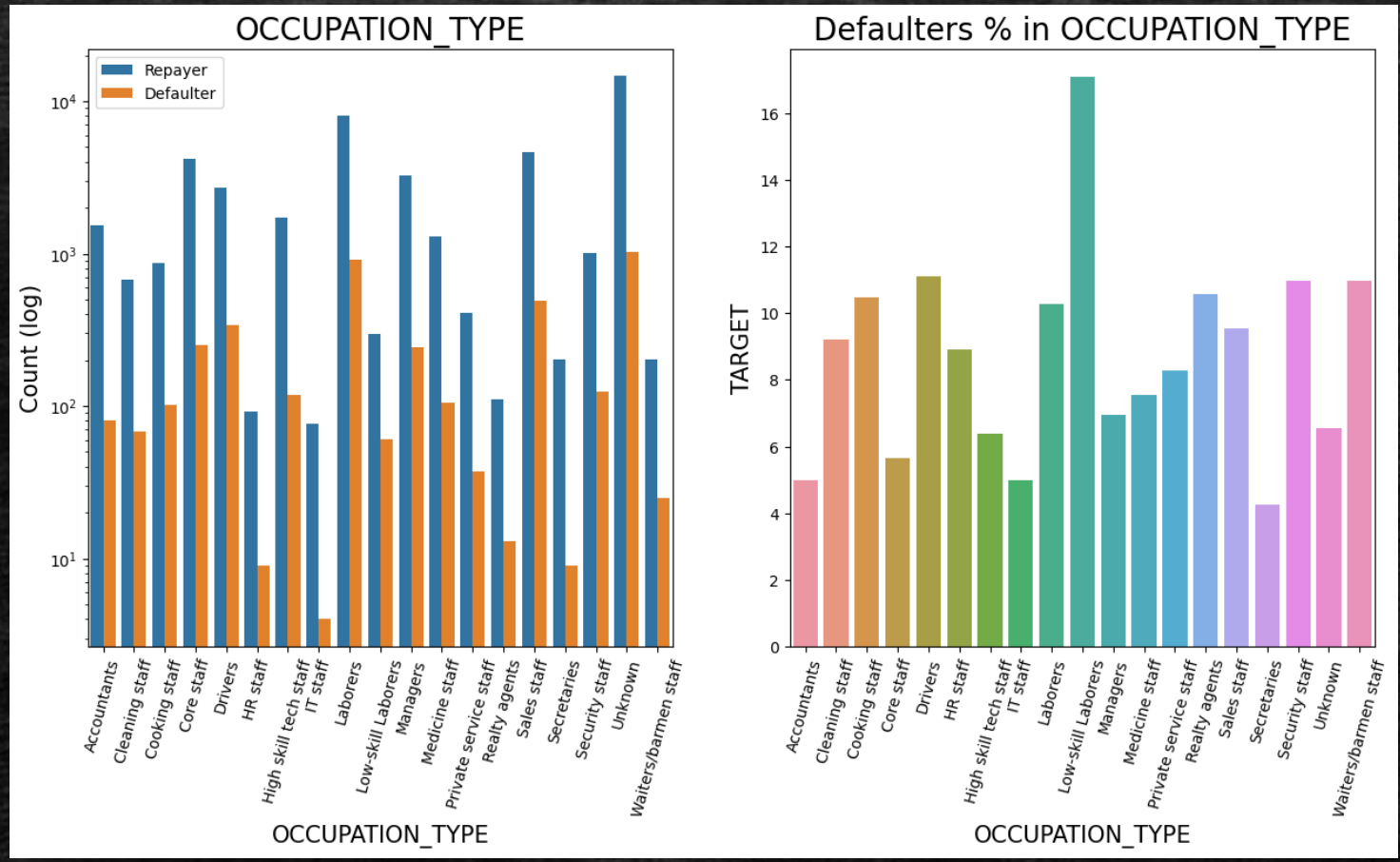
Education Type



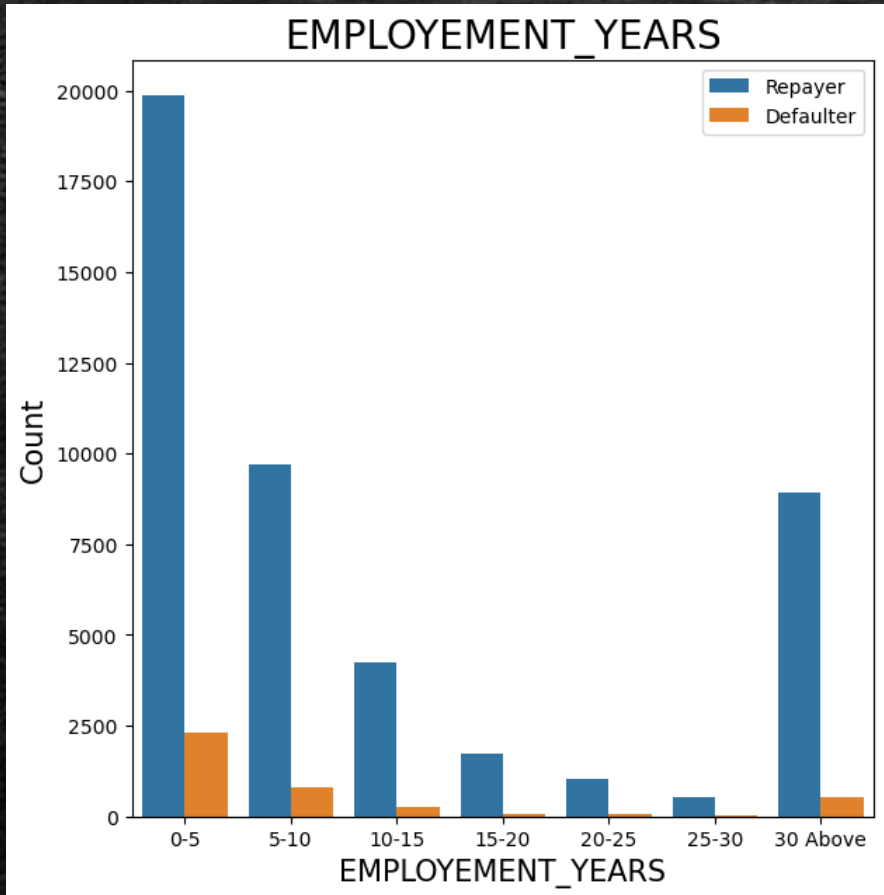
Income Type

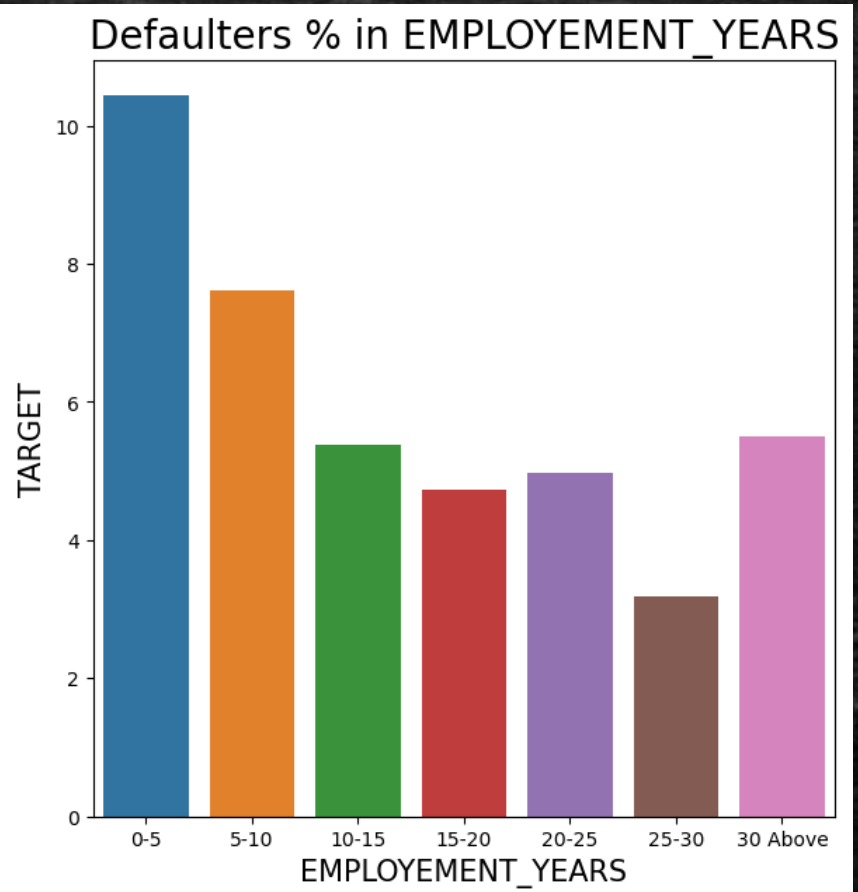

NAME_INCOME_TYPE
Repayer
Defaulter
Count (log)
10^0
10^1
10^2
10^3
10^4
Businessman
Commercial associate
Maternity leave
Pensioner
State servant
Student
Unemployed
Working
NAME_INCOME_TYPE
Defaulters % in NAME_INCOME_TYPE
TARGET
0
5
10
15
20
25
30
35
Businessman
Commercial associate
Maternity leave
Pensioner
State servant
Student
Unemployed
Working
NAME_INCOME_TYPE

Occupation Type

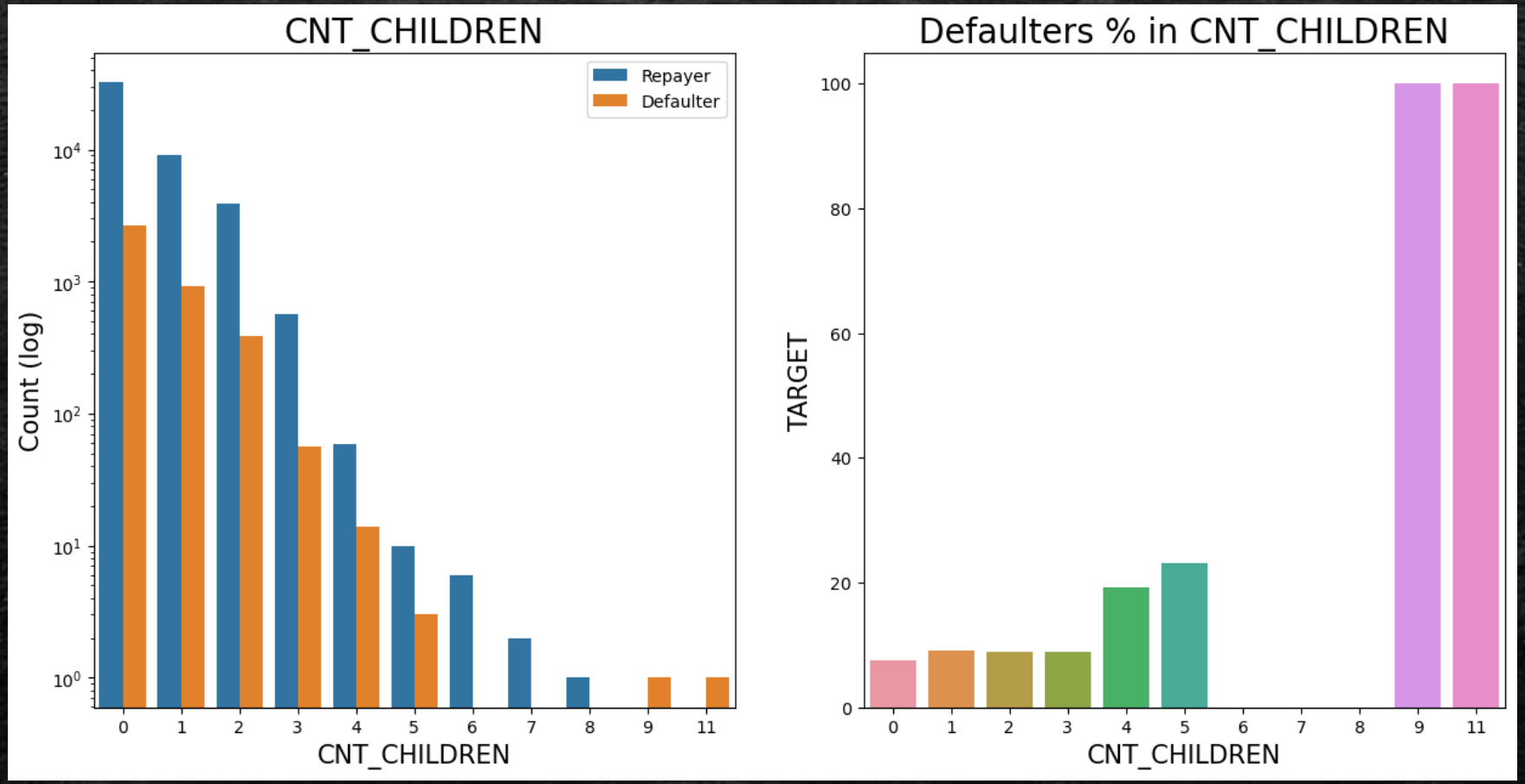


Employment years

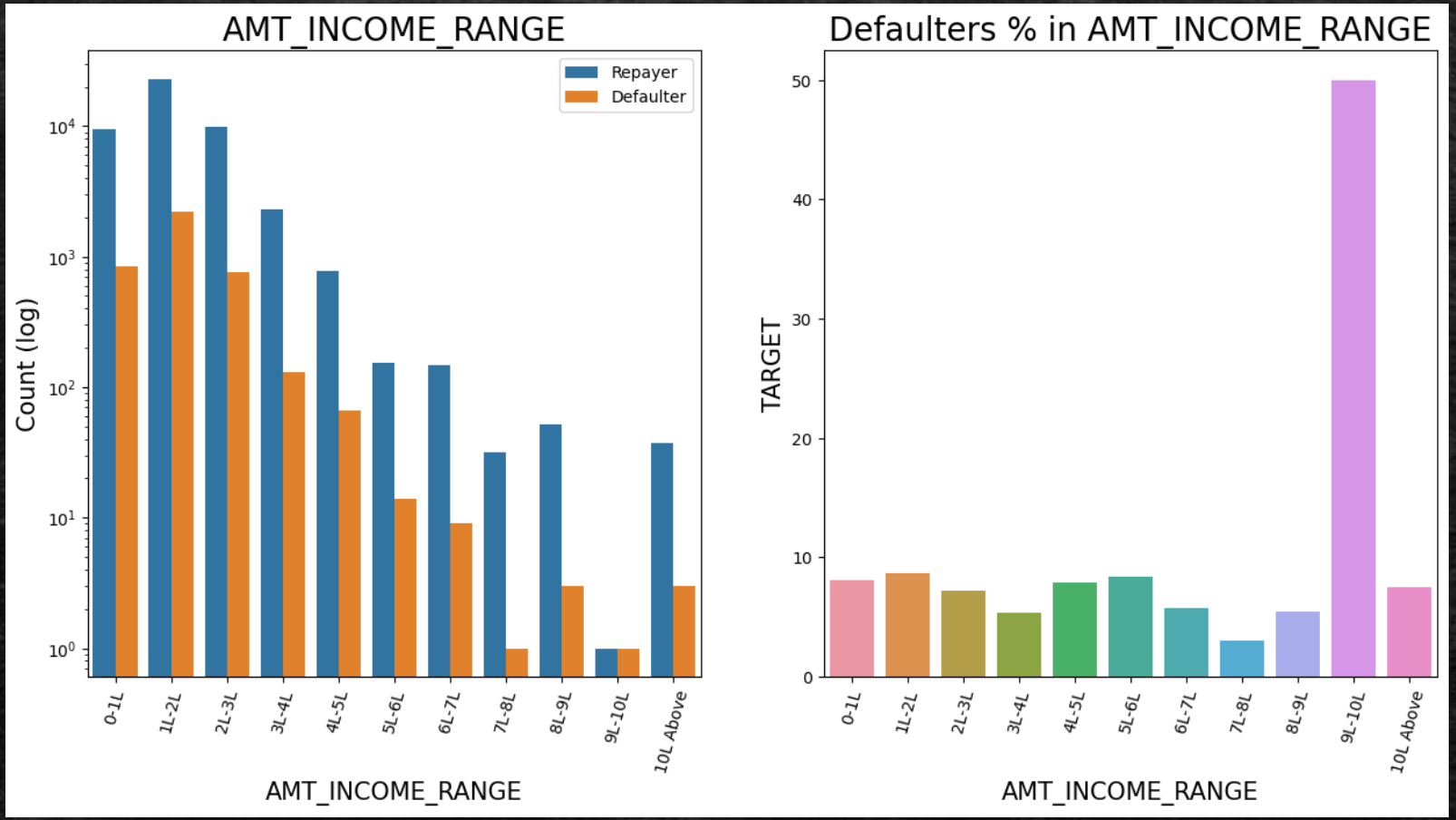




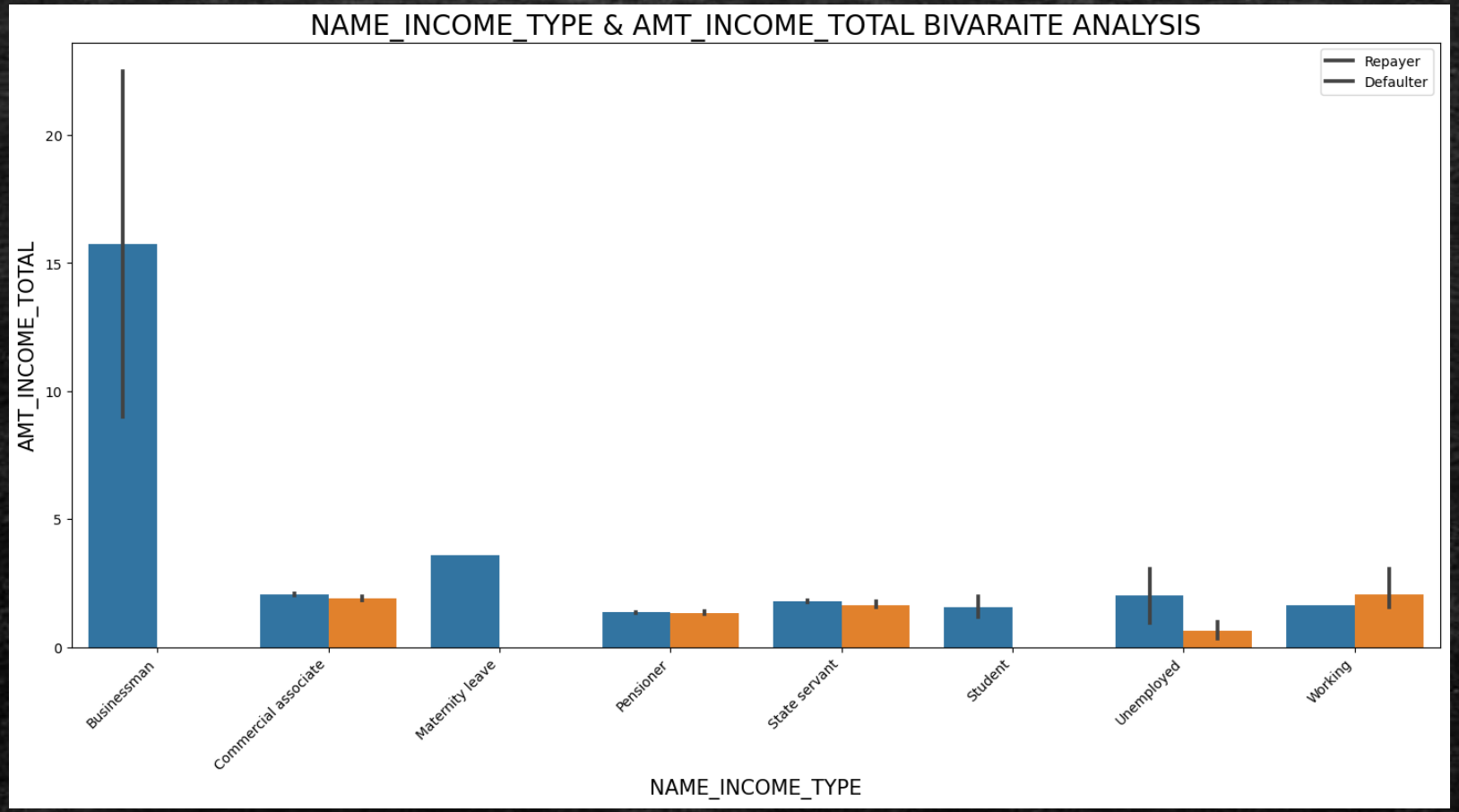
Child count


CNT_CHILDREN
Repayer
Defaulter
Count (log)
10^4
10^3
10^2
10^1
10^0
0 1 2 3 4 5 6 7 8 9 11
CNT_CHILDREN
Defaulters % in CNT_CHILDREN
TARGET
100
80
60
40
20
0
0 1 2 3 4 5 6 7 8 9 11
CNT_CHILDREN

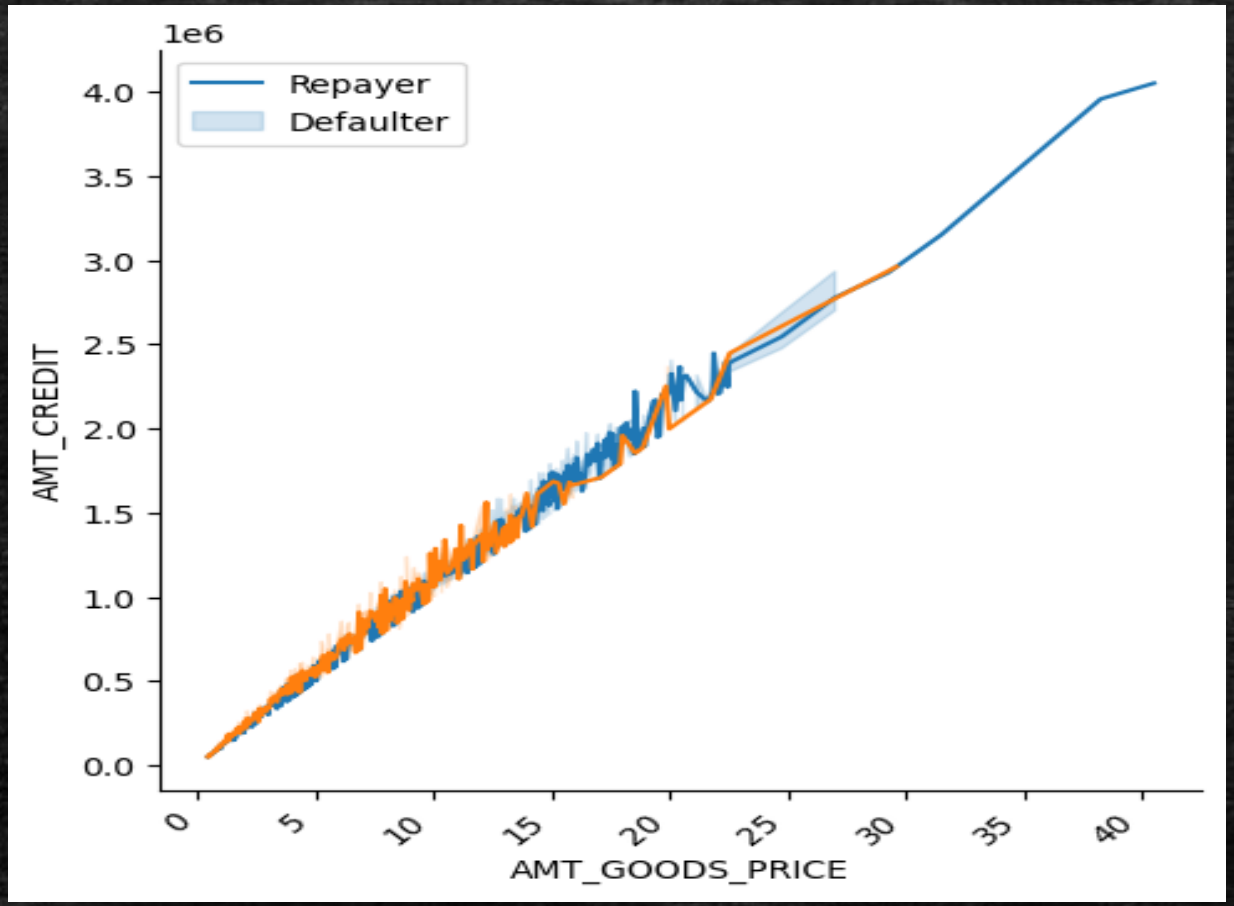
Income range



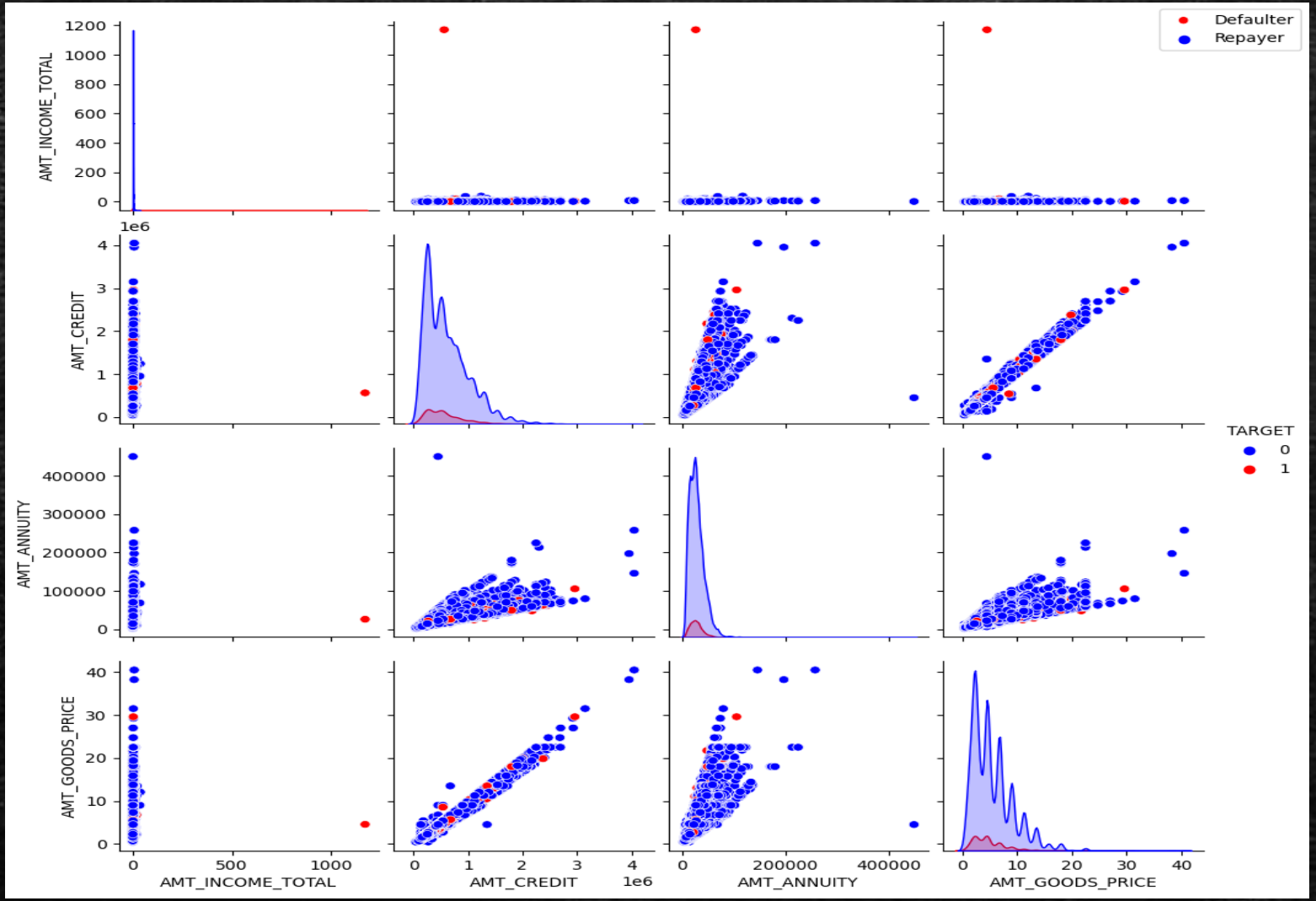
Bivariate of income amount and income type



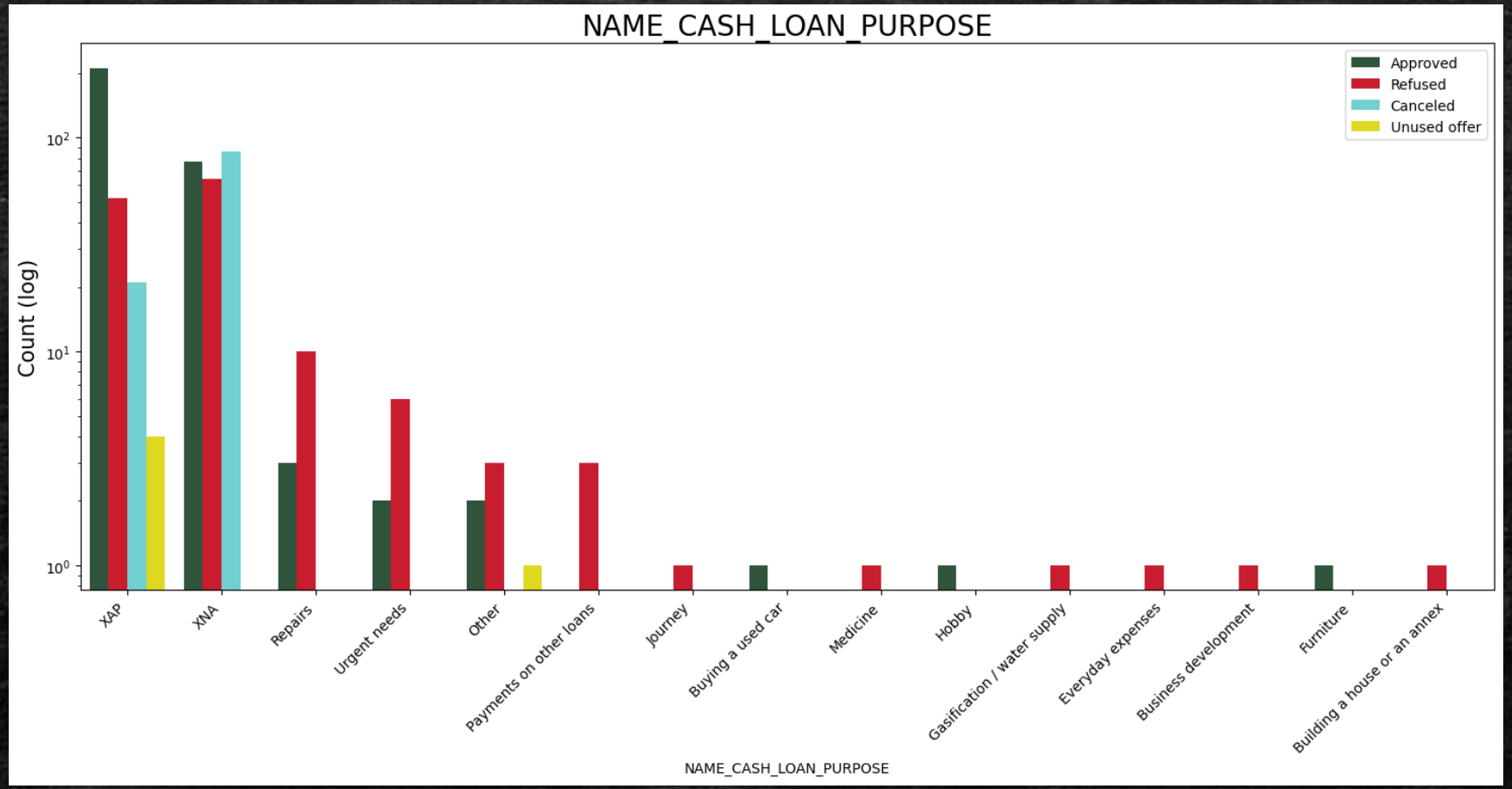
Bivariate of numerical type



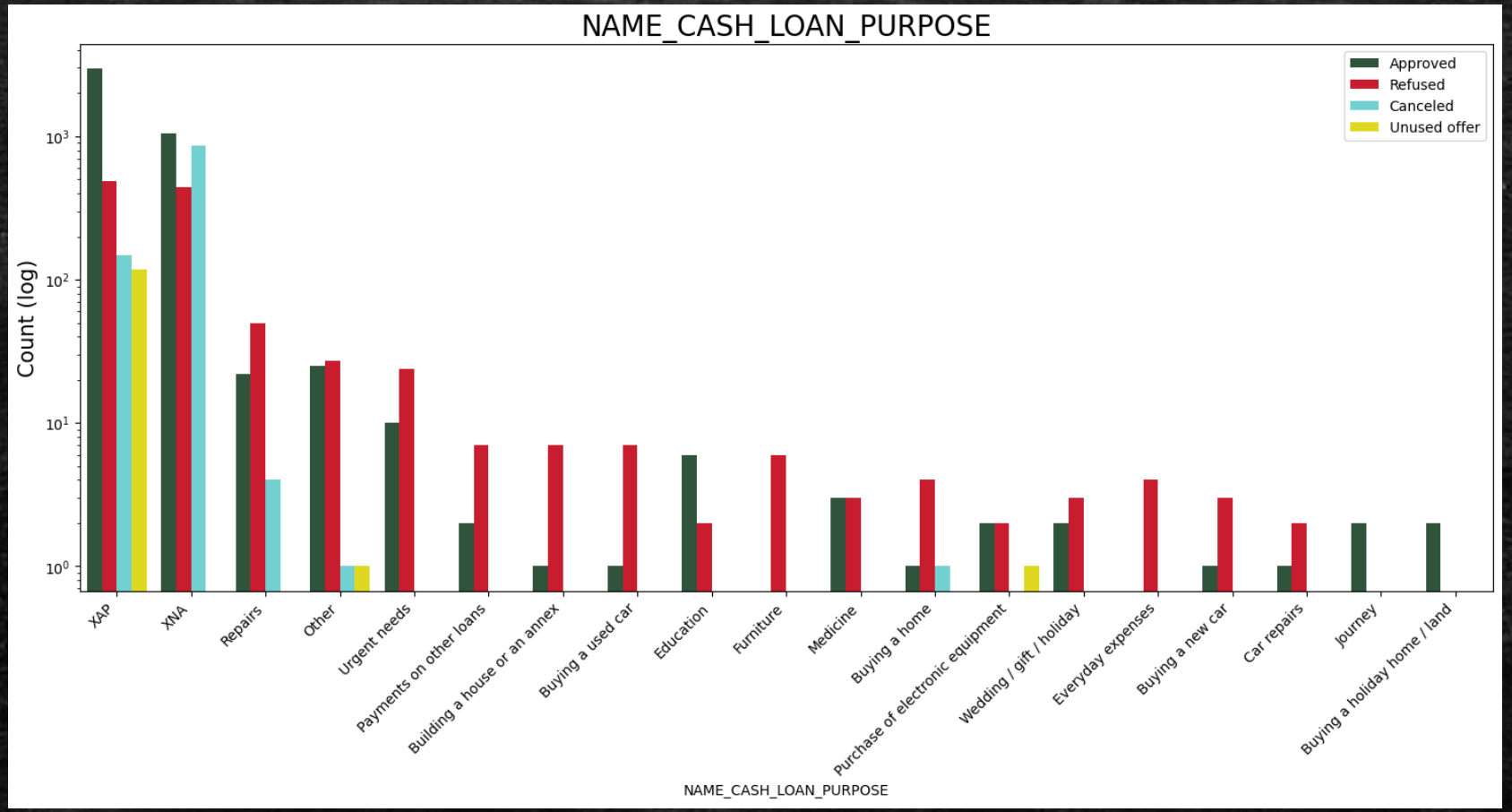
Multivariate pair plot



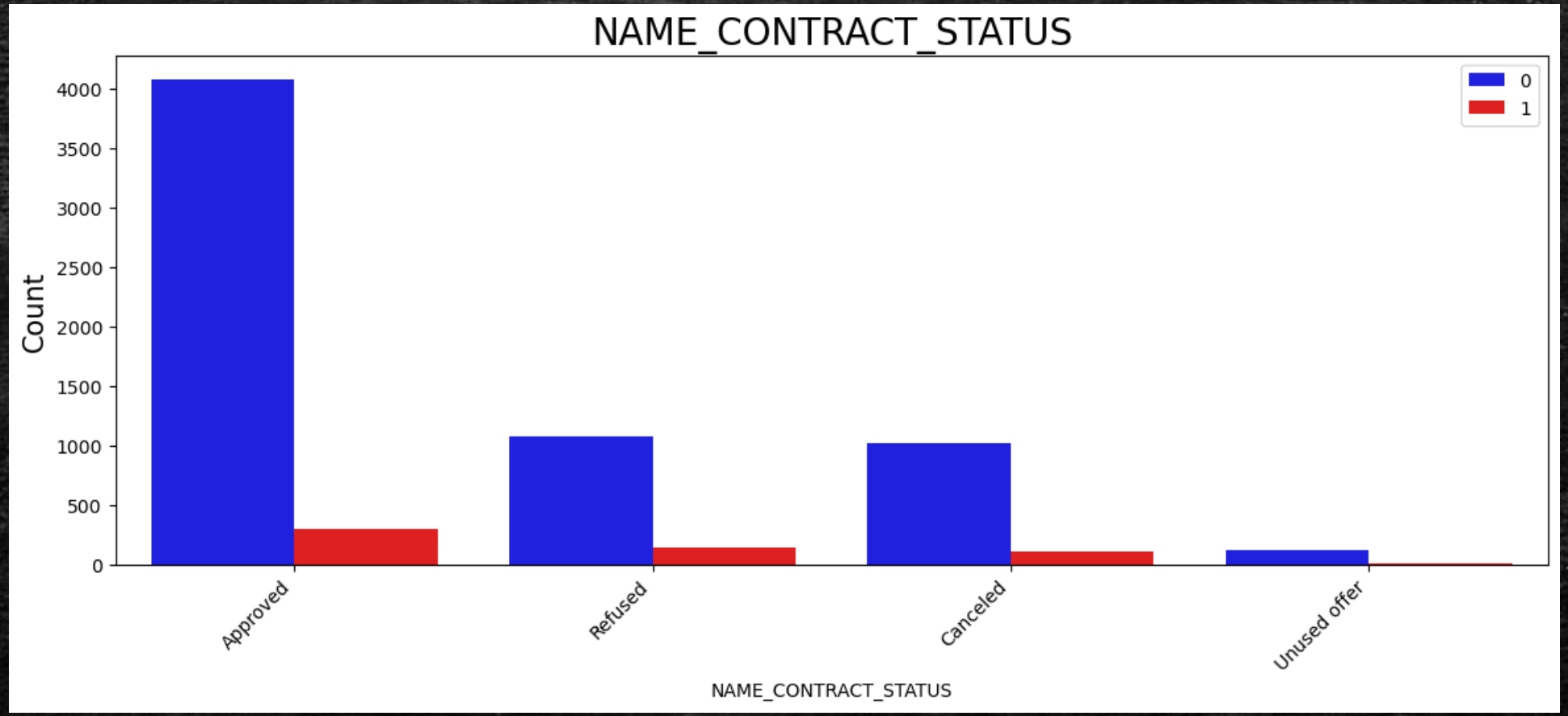
Merged univariate for defaulters



Merged univariate for Repayer

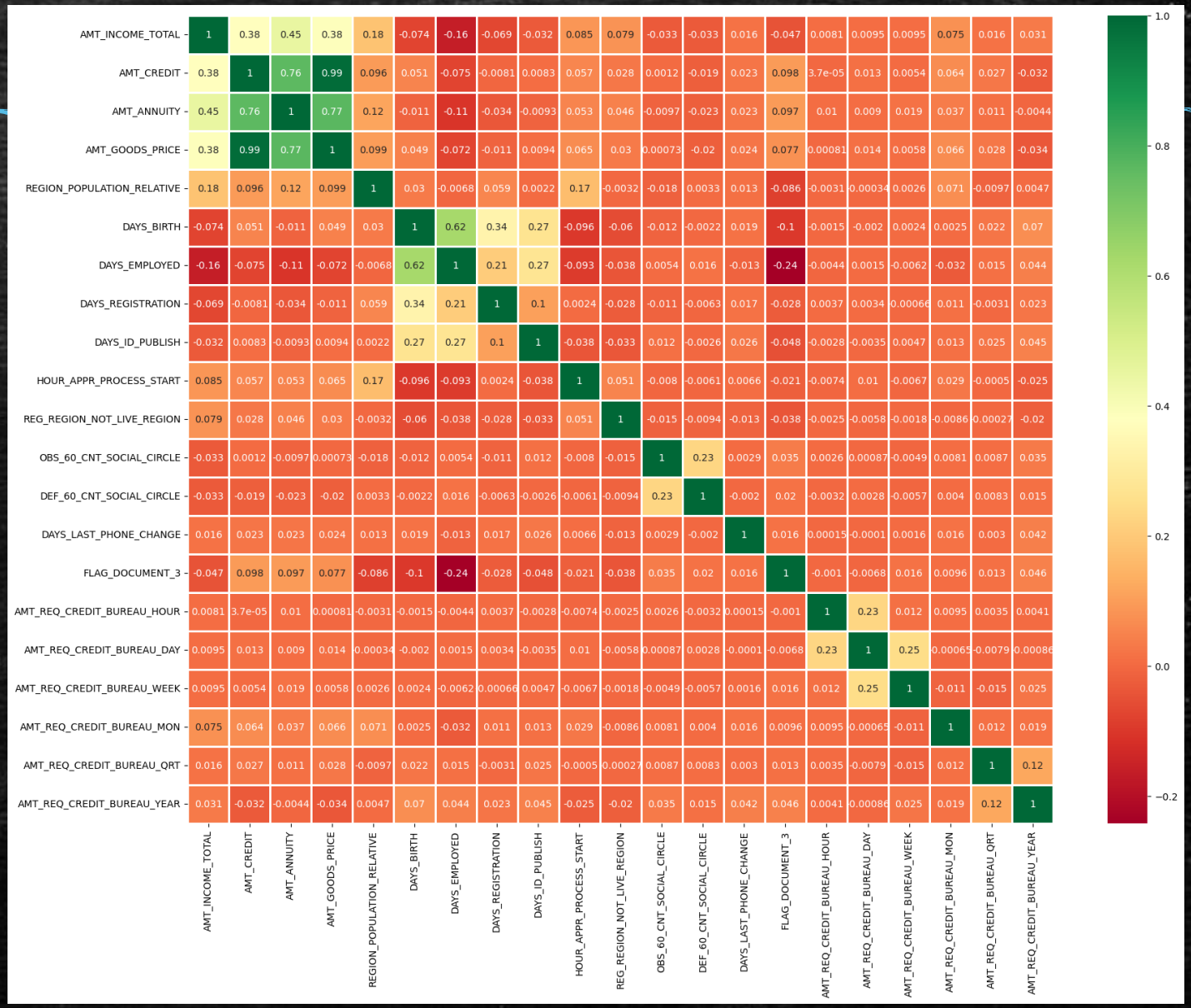


Contact status by TARGET



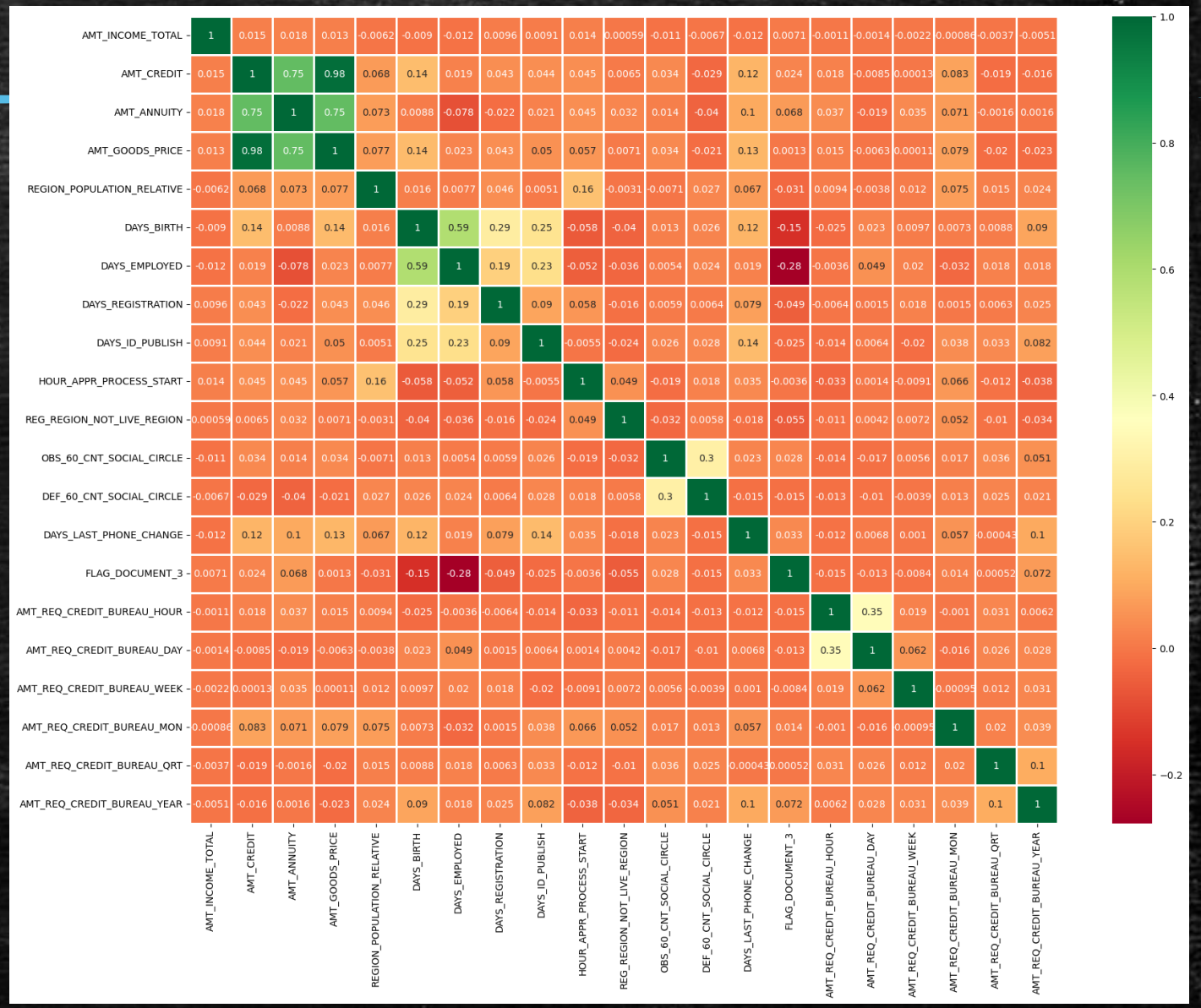
CORRELATION BETWEEN VARIABLES WHEN LOAN REPAID

	VAR1	VAR2	Correlation
64	AMT_GOODS_PRICE	AMT_CREDIT	0.987000
65	AMT_GOODS_PRICE	AMT_ANNUITY	0.768833
43	AMT_ANNUITY	AMT_CREDIT	0.763737
131	DAYS_EMPLOYED	DAYS_BIRTH	0.623475
42	AMT_ANNUITY	AMT_INCOME_TOTAL	0.447222
63	AMT_GOODS_PRICE	AMT_INCOME_TOTAL	0.384576
21	AMT_CREDIT	AMT_INCOME_TOTAL	0.377966
152	DAYS_REGISTRATION	DAYS_BIRTH	0.335028
174	DAYS_ID_PUBLISH	DAYS_EMPLOYED	0.274516
173	DAYS_ID_PUBLISH	DAYS_BIRTH	0.270073



CORRELATION BETWEEN VARIABLES WHEN LOAN DEFAULT

	VAR1	VAR2	Correlation
64	AMT_GOODS_PRICE	AMT_CREDIT	0.982268
43	AMT_ANNUITY	AMT_CREDIT	0.749665
65	AMT_GOODS_PRICE	AMT_ANNUITY	0.749504
131	DAYS_EMPLOYED	DAYS_BIRTH	0.588243
351	AMT_REQ_CREDIT_BUREAU_DAY	AMT_REQ_CREDIT_BUREAU_HOUR	0.351200
263	DEF_60_CNT_SOCIAL_CIRCLE	OBS_60_CNT_SOCIAL_CIRCLE	0.301421
152	DAYS_REGISTRATION	DAYS_BIRTH	0.288438
300	FLAG_DOCUMENT_3	DAYS_EMPLOYED	0.277264
173	DAYS_ID_PUBLISH	DAYS_BIRTH	0.247897
174	DAYS_ID_PUBLISH	DAYS_EMPLOYED	0.232662



Factors of repayer

- NAME_EDUCATION_TYPE: Academic degree has less defaults.
- NAME_INCOME_TYPE: Student and Businessmen have no defaults.
- REGION_RATING_CLIENT: RATING 1 is safer.
- DAYS_BIRTH: People above age of 50 have low probability of defaulting.
- DAYS_EMPLOYED: Clients with 40+ year experience having less than 1% default rate.
- AMT_INCOME_TOTAL: Applicant with Income more than 700,000 are less likely to default.
- NAME_CASH_LOAN_PURPOSE: Loans bought for Hobby, Buying garage are being repaid mostly.
- CNT_CHILDREN: People with zero to two children tend to repay the loans.

Factors of Defaulter

- CODE_GENDER: Men are at relatively higher default rate.
- NAME_FAMILY_STATUS : People who have civil marriage or who are single default a lot.
- NAME_EDUCATION_TYPE: People with Lower Secondary & Secondary education.
- NAME_INCOME_TYPE: Clients who are either at Maternity leave OR Unemployed default a lot.
- OCCUPATION_TYPE: Avoid Low-skill Laborers, Drivers and Waiters/barmen staff, Security staff, Laborers and Cooking staff as their default rate is huge.
- DAYS_BIRTH: Avoid young people who are in age group of 20-40 as they have higher probability of defaulting.
- DAYS_EMPLOYED: People who have less than 5 years of employment have high default rate.
- CNT_CHILDREN & CNT_FAM_MEMBERS: Client who have children equal to or more than 9 default 100% and hence their applications are to be rejected.
- AMT_GOODS_PRICE: When the credit amount goes beyond 3lakhs, there is an increase in defaulters.

Suggestions

- 90% of the previously cancelled client have actually repaid the loan. Record the reason for cancellation which might help the bank to determine and negotiate terms with these repaying customers in future for increase business opportunity.
- 88% of the clients who were refused by bank for loan earlier have now turned into a repaying client. Hence documenting the reason for rejection could mitigate the business loss and these clients could be contacted for further loans.

Result

- I Have learned a project.
- This project **strengthen** my knowledge in pandas and matplotlib.
- This project helped me learn data cleaning and data visualization.
- Datasets are huge and complex learned a lot from cleaning them.
- IPYNB file link.
- Drive link.
- Thank you.