

# Bank Loan Case Study

## Aim of the Project

The aim is to **identify patterns** which indicate if a client has difficulty paying their installments which may be used for taking actions such as denying the loan, reducing the amount of loan, lending (to risky applicants) at a higher interest rate, etc. This will ensure that the consumers capable of repaying the loan are not rejected. **Identification of such applicants using EDA** is the aim of this case study.

In other words, the company wants to understand the **driving factors** (or driver variables) behind loan default, i.e. the variables which are strong indicators of default. The company can utilize this knowledge for its portfolio and risk assessment.

*Q1. Present the overall approach of the **analysis**. Mention the problem statement and the analysis approach briefly*

## Approach

In this notebook, we are analyzing

- **previous\_application.csv** i.e. data about previous application of an applicant.
- **application\_data.csv** i.e. data about current application of loan.

For the Exploratory Data Analysis, mentioned steps have been followed.

--> **Importing Modules**

--> **Read the dataset**

--> **Data Cleaning**

1. Missing value handling
2. Fixing Rows and Columns - removing unnecessary rows/columns (through missing value handling and correlation)
3. Handling Outliers

--> Uni-variate **Analysis**

--> Bi-variate **Analysis**

--> **Conclusion**

## 1. Importing Modules

Modules imported - numpy, pandas, seaborn, matplotlib.

## 2. Reading the dataset

Read the dataset using `pd.read_csv('dataset')`

## 3. Data cleaning

**Q2. Identify the missing data and use appropriate method to deal with it.**

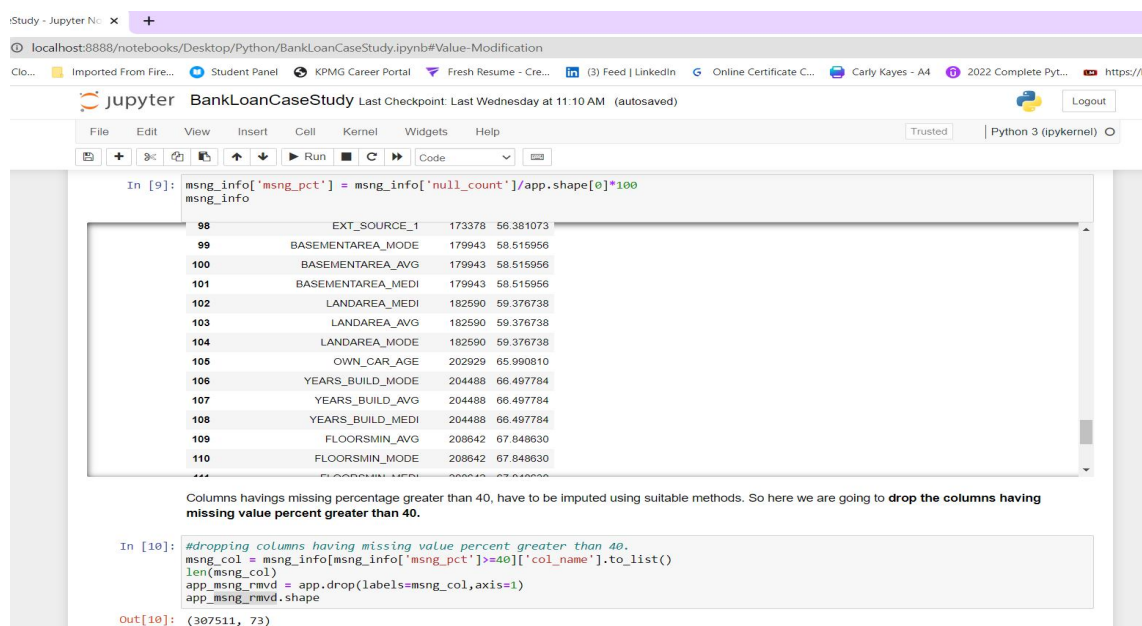
Data cleaning had 3 major steps, they are:

- Missing value handling
- Fixing Rows and Columns - removing unnecessary rows/columns (through missing value imputation, Value Modification and correlation)
- Handling Outliers

### ➤ Missing Value Handling:

This is for the Application Dataset:

Columns were sorted based on the percentage of null values they hold and all columns having missing percentage greater than 40 were dropped.



```
In [9]: msgng_info['msgng_pct'] = msgng_info['null_count']/app.shape[0]*100
msgng_info
```

98	EXT_SOURCE_1	173378	56.381073	
99	BASEMENTAREA_MODE	179943	58.515956	
100	BASEMENTAREA_AVG	179943	58.515956	
101	BASEMENTAREA_MEDI	179943	58.515956	
102	LANDAREA_MEDI	182590	59.376738	
103	LANDAREA_AVG	182590	59.376738	
104	LANDAREA_MODE	182590	59.376738	
105	OWN_CAR_AGE	202929	65.990810	
106	YEARS_BUILD_MODE	204488	66.497784	
107	YEARS_BUILD_AVG	204488	66.497784	
108	YEARS_BUILD_MEDI	204488	66.497784	
109	FLOORSMIN_AVG	208642	67.848630	
110	FLOORSMIN_MODE	208642	67.848630	

Columns havings missing percentage greater than 40, have to be imputed using suitable methods. So here we are going to **drop the columns having missing value percent greater than 40.**

```
In [10]: #dropping columns having missing value percent greater than 40.
msgng_col = msgng_info[msgng_info['msgng_pct']>40]['col_name'].to_list()
len(msgng_col)
app_msgng_rmvd = app.drop(labels=msgng_col,axis=1)
app_msgng_rmvd.shape
```

```
Out[10]: (307511, 73)
```

After dropping these columns, we are left with 73 columns for data cleaning.

This is for the Previous Application Dataset:

Columns were sorted based on the percentage of null values they hold and all columns having missing percentage greater than 40 were dropped.

22 columns are left for further analysis after dropping 15 columns.

#### Previous application dataset

```
In [125]: #columns having missing percentage greater than 40
null_count = pd.DataFrame(prev_app.isnull().sum().sort_values(ascending=False)/p

null_count
var_msng_ge_40 = list(null_count[null_count['count_pct']>=40]['var'])
var_msng_ge_40

Out[125]: ['RATE_INTEREST_PRIVILEGED',
'RATE_INTEREST_PRIMARY',
'AMT_DOWN_PAYMENT',
'RATE_DOWN_PAYMENT',
'NAME_TYPE_SUITE',
'NFLAG_INSURED_ON_APPROVAL',
'DAYS_TERMINATION',
'DAYS_LAST_DUE',
'DAYS_LAST_DUE_1ST_VERSION',
'DAYS_FIRST_DUE',
'DAYS_FIRST_DRAWING']

In [126]: #dropping columns
nva_cols = var_msng_ge_40+['WEEKDAY_APPR_PROCESS_START', 'HOUR_APPR_PROCESS_START
len(nva_cols)

Out[126]: 15
```

### ➤ Fixing Rows and Columns:

Here the unwanted columns are removed by analyzing them through missing value imputation and correlation. The correlation of the columns with the target column is checked and decided whether to keep the columns or drop it. Heatmaps are drawn to find the correlation

#### Through Correlation:

1. Flag Columns: The number of flag columns were found to be 28. A few of the flag columns contained object data and few had numeric data.

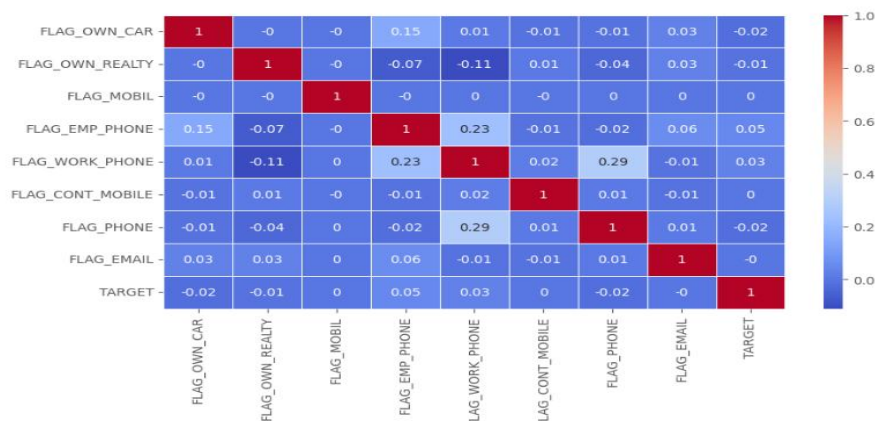
A subplot was obtained for the distribution of columns with Target as hue. It was observed that most of the flag documents were not submitted and most turned out to be non defaulters and had the same distribution except Flag document 3. Hence we can drop these columns too.

The other flag columns, 'FLAG\_OWN\_CAR', 'FLAG\_OWN\_REALTY', 'FLAG\_MOBIL', 'FLAG\_EMP\_PHONE', 'FLAG\_WORK\_PHONE', 'FLAG\_CONT\_MOBILE', 'FLAG\_PHONE', 'FLAG\_EMAIL', with object data was converted to numeric data to find the correlation with the target variable. A heat map was obtained.

```
In [20]: corr_df = round(flag_corr_df.corr(),2)

plt.figure(figsize=(10,5))
sns.heatmap(corr_df,cmap='coolwarm',linewidths=.5,annot=True)

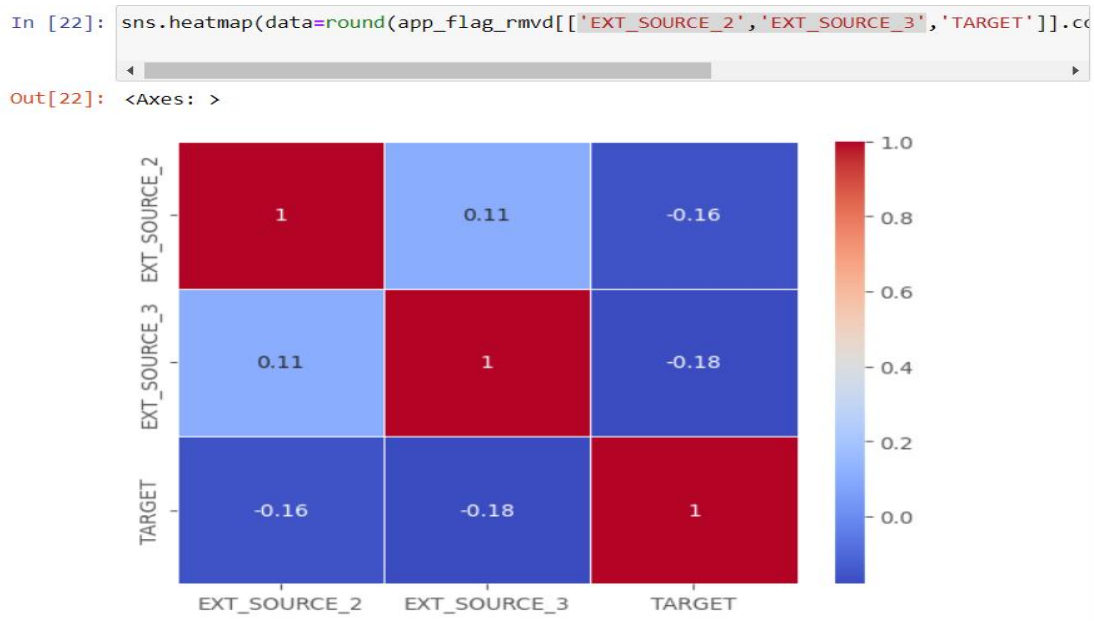
Out[20]: <Axes: >
```



Correlation coefficient values less than +0.8 or greater than -0.8 are not considered significant. From the above plot we get to observe that the flag columns do not have good correlation with the target variable. So we can drop all the flag columns due to their poor linearity .

Now 45 columns are left for analysis.

2. External Columns: The correlation between external sources , 'EXT\_SOURCE\_2','EXT\_SOURCE\_3' and Target variable is drawn and it is observed to have poor correlation. Hence these columns are dropped.



43 columns are left for analysis.

#### Through Missing Value Imputation:

Here the missing values in the columns are filled using the suitable mean, median, mode values and later the columns are converted to zero missing data columns and are not dropped.

This is for the Application Dataset:

The null values in the columns are filled by the respective methods:

CNT\_FAM\_MEMBERS - Mode

OCCUPATION\_TYPE - Mode

NAME\_TYPE\_SUITE - Mode

AMT\_ANNUITY - Mean

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 - Median

AMT\_GOODS\_PRICE - Median

*This is for the Previous Application Dataset:*

AMT\_GOODS\_PRICE - Median

AMT\_ANNUITY - Median

PRODUCT\_COMBINATION - Mode

CNT\_PAYMENT - fillna(0)

### **Through Value Modification:**

Any negative values in the columns are converted to positive, i.e, Absolute Value Conversion.

The following columns undergo value modification:

DAYS\_BIRTH  
DAYS\_EMPLOYED  
DAYS\_REGISTRATION  
DAYS\_ID\_PUBLISH  
DAYS\_LAST\_PHONE\_CHANGE

### ➤ **Handling Outliers:**

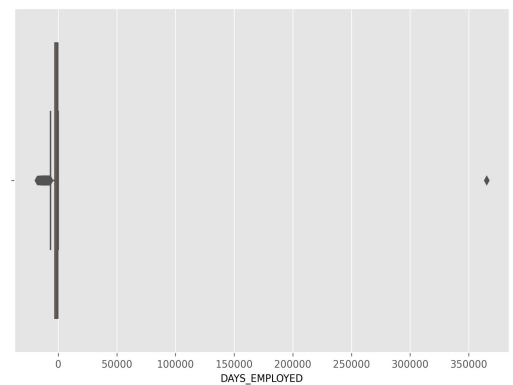
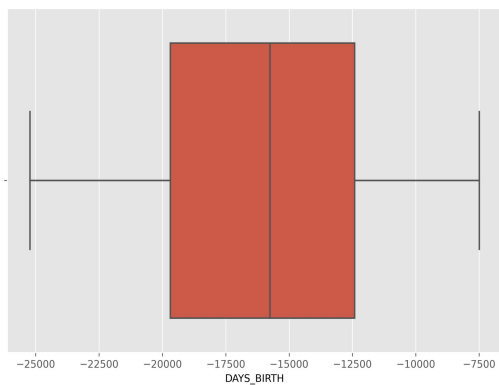
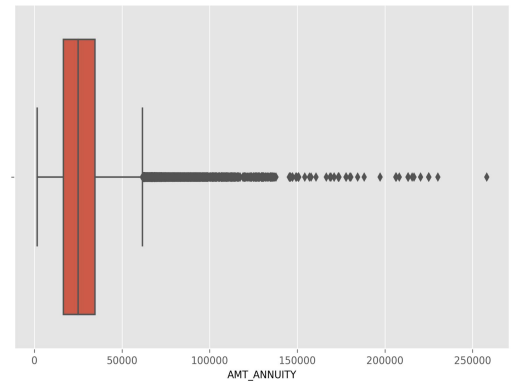
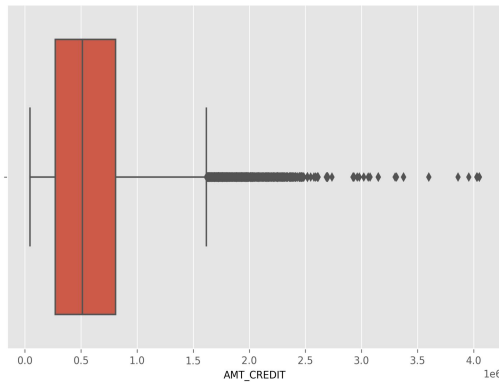
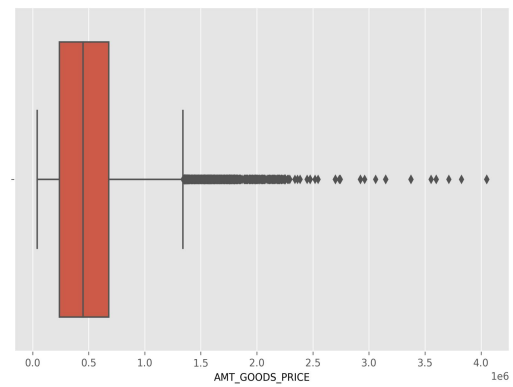
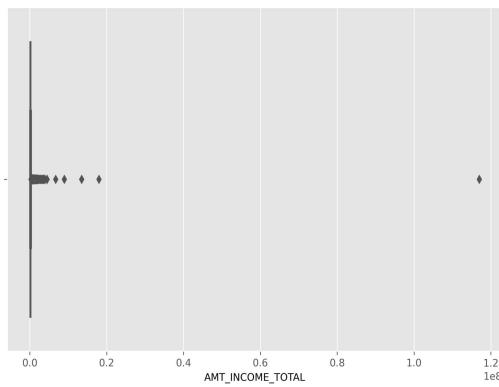
*Q3. Identify if there are **outliers** in the dataset. Also, mention why do you think it is an outlier.*

Some columns have unique values greater than 1000, hence such columns require binning.

AMT_GOODS_PRICE	1002
AMT_INCOME_TOTAL	2548
DAYS_LAST_PHONE_CHANGE	3773
AMT_CREDIT	5603
DAYS_ID_PUBLISH	6168
DAYS_EMPLOYED	12574
AMT_ANNUITY	13673
DAYS_REGISTRATION	15688
DAYS_BIRTH	17460
SK_ID_CURR	307511

Quantiles are obtained at different points of the data and binning is made for the following columns.

AMT\_GOODS\_PRICE  
AMT\_INCOME\_TOTAL  
AMT\_CREDIT  
AMT\_ANNUITY  
DAYS\_EMPLOYED & DAYS\_BIRTH



- ◆ IQR for AMT\_INCOME\_TOTAL is very slim and it has large number of outliers.
- ◆ AMT\_GOODS\_PRICE large number of outliers are present after 150000.
- ◆ Third quartile of AMT\_CREDIT is larger as compared to First quartile which means that most of the Credit amount of the loan of customers are present in the third quartile. And there are large number of outliers present in AMT\_CREDIT.
- ◆ Third quartile of AMT\_ANNUITY is slightly larger than First quartile and there are large number of outliers.
- ◆ DAYS\_BIRTH have no outlier.
- ◆ IQR for DAYS EMPLOYED is very slim. Most of the outliers are present below 25000. And a outlier is present 375000.

## Data Analysis

Q4. Explain the **results of uni-variate, segmented uni-variate, bi-variate analysis, etc.**

### Uni variate Analysis :

Analyzing the columns of **object data type**

```
In [89]: obj_var
Out[89]: Index(['NAME_CONTRACT_TYPE', 'CODE_GENDER', 'NAME_TYPE_SUITE',
              'NAME_INCOME_TYPE', 'NAME_EDUCATION_TYPE', 'NAME_FAMILY_STATUS',
              'NAME_HOUSING_TYPE', 'OCCUPATION_TYPE', 'WEEKDAY_APPR_PROCESS_START',
              'ORGANIZATION_TYPE'],
              dtype='object')

In [90]: plt.figure(figsize=(25,60))

for i, var in enumerate(obj_var):

    data_pct = app_score_col_rmvd[[var, 'TARGET']].groupby([var], as_index=False).mean().sort_values(by='TARGET', ascending=False)
    data_pct['PCT'] = data_pct['TARGET']*100

    plt.subplot(10,2,i+1)
    plt.subplots_adjust(wspace=0.1, hspace=1)
    sns.countplot(data=app_score_col_rmvd, x=var, hue='TARGET')
    plt.xticks(rotation=90)

    plt.subplot(10,2,i+2)
    sns.barplot(data=data_pct, x=var, y='PCT', palette='coolwarm')
    plt.xticks(rotation=90)
```

#### 1. NAME\_CONTRACT\_TYPE -

Most of the customers have taken cash loan

Customers who have taken cash loans are less likely to default

#### 2. CODE\_GENDER -

Most of the loans have been taken by female customers.

Default rate of females are just ~7% which is safer and lesser than male.

#### 3. NAME\_TYPE\_SUITE -

Unaccompanied people had taken most of the loans and the default rate is ~8.5%.

#### 4. NAME\_INCOME\_TYPE -

The safest segments are working, commercial associates and pensioners.

#### 5. NAME\_EDUCATION\_TYPE -

Higher education is the safest segment to give the loan with a default rate of less than 5%

#### 6. NAME\_HOUSING\_TYPE -

People having house/apartment are safe to give the loan with default rate of ~8%

#### 7. OCCUPATION\_TYPE -

Low skilled laborers and drivers are highest defaulters

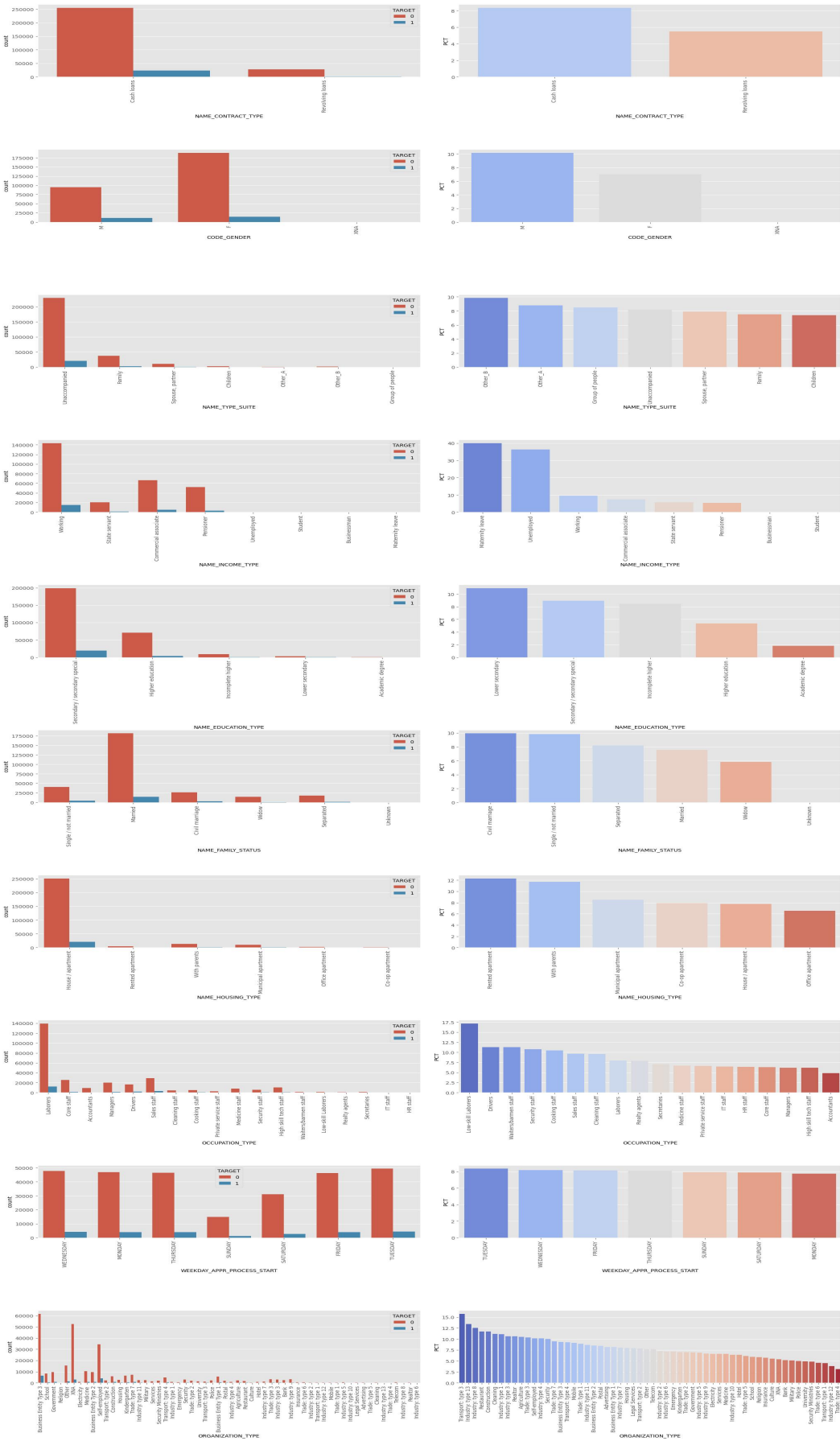
Accountants are less defaulters

Core staff, managers and laborers are safer to target with a default rate of <= 7.5 to 10%

#### 8. ORGANIZATION\_TYPE -

Transport type 3 has highest defaulters

Others, Business Entity type 3, Self Employed are good to go with default rate around 10%

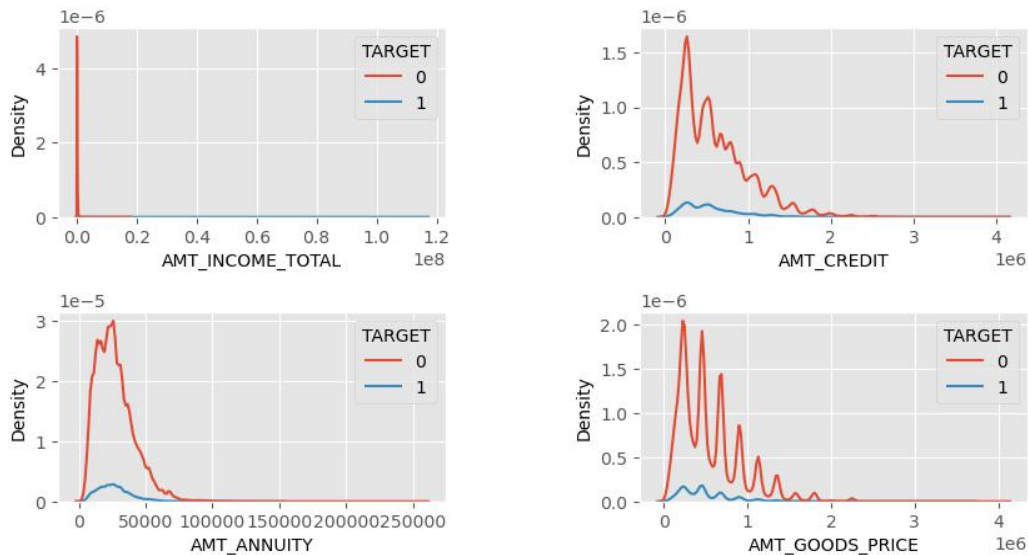




## Analyzing the columns of **numeric data type**

```
In [99]: amt_var = ['AMT_INCOME_TOTAL', 'AMT_CREDIT', 'AMT_ANNUITY', 'AMT_GOODS_PRICE']
```

```
In [101]: plt.figure(figsize=(10,5))  
  
for i, col in enumerate(amt_var):  
    plt.subplot(2,2,i+1)  
    sns.kdeplot(data=num_data,x=col,hue='TARGET')  
    plt.subplots_adjust(wspace=0.5,hspace=0.5)
```



### 1. AMT\_INCOME\_TOTAL -

Mostly the customers have income between 0 to 1M.

### 2. AMT\_GOODS\_PRICE -

Most of the loans were given for the goods price ranging between 0 to 1M.

### 3. AMT\_CREDIT -

Most of the loans were given for the credit amount of 0 to 1M.

### 4. AMT\_ANNUITY -

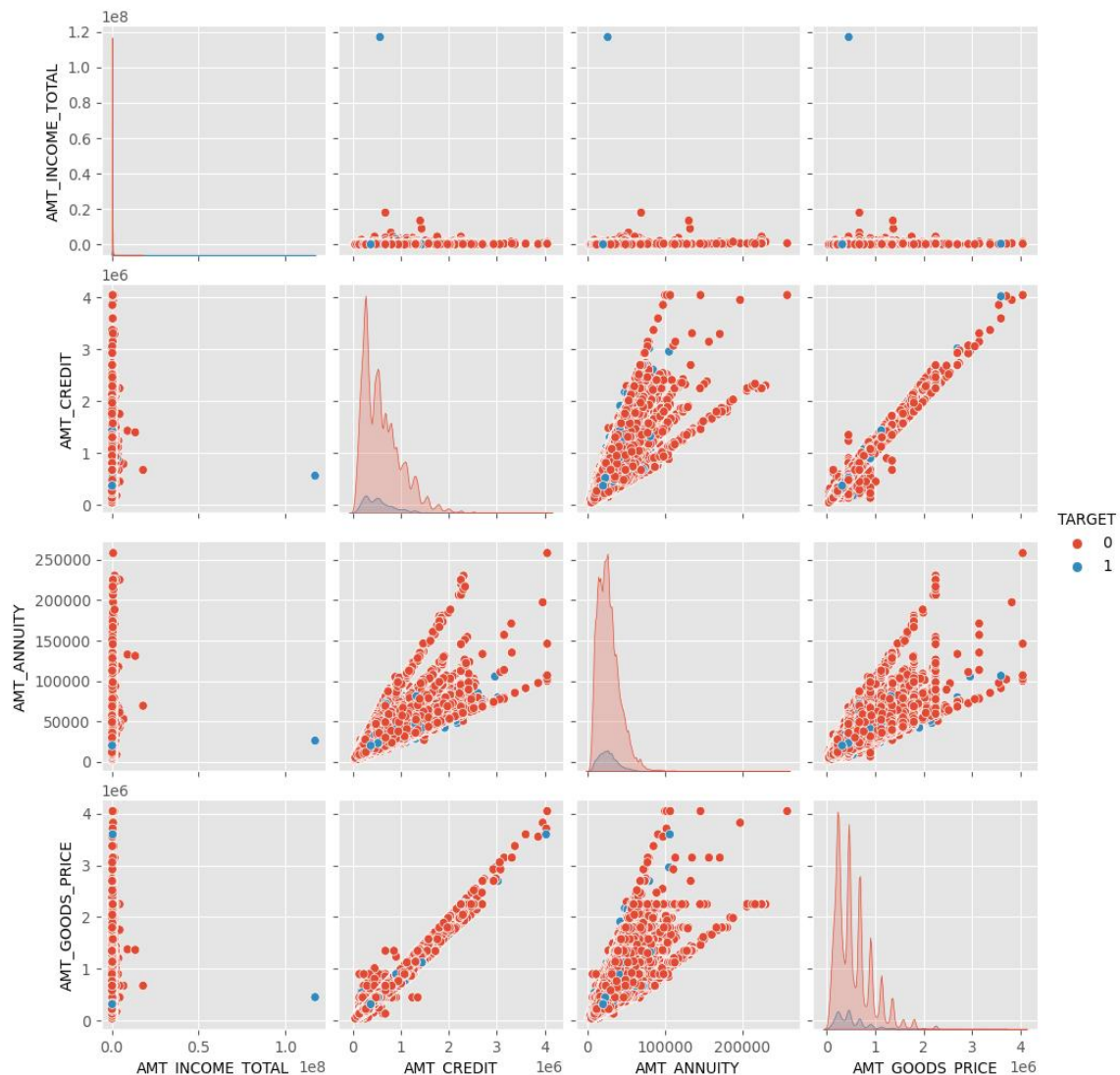
Most of the customers are paying annuity of 0 to 50 K.

## Bi-variate Analysis

```
In [115]: amt_var = num_data[['AMT_INCOME_TOTAL', 'AMT_CREDIT', 'AMT_ANNUITY', 'AMT_GOODS_PRICE', 'TARGET']]
```

```
In [116]: sns.pairplot(data=amt_var, hue='TARGET')
```

```
Out[116]: <seaborn.axisgrid.PairGrid at 0x16dcc4735e0>
```



>> AMT\_CREDIT and AMT\_GOODS\_PRICE are linearly correlated, if the AMT\_CREDIT increases the defaulters are decreasing.

>> People having income less than or equals to 1 Million, are more likely to take loans out of which who are taking loan of less than 1.5 Million, could turn out to be defaulters. We can target income below 1 Million and loan amount greater than 1.5 Million.

>> People having children 1 to less than 5 are safer to give the loan.

>> People who can pay the annuity of 100K are more like to get the loan and that's up-to less than 2M (safer segment).

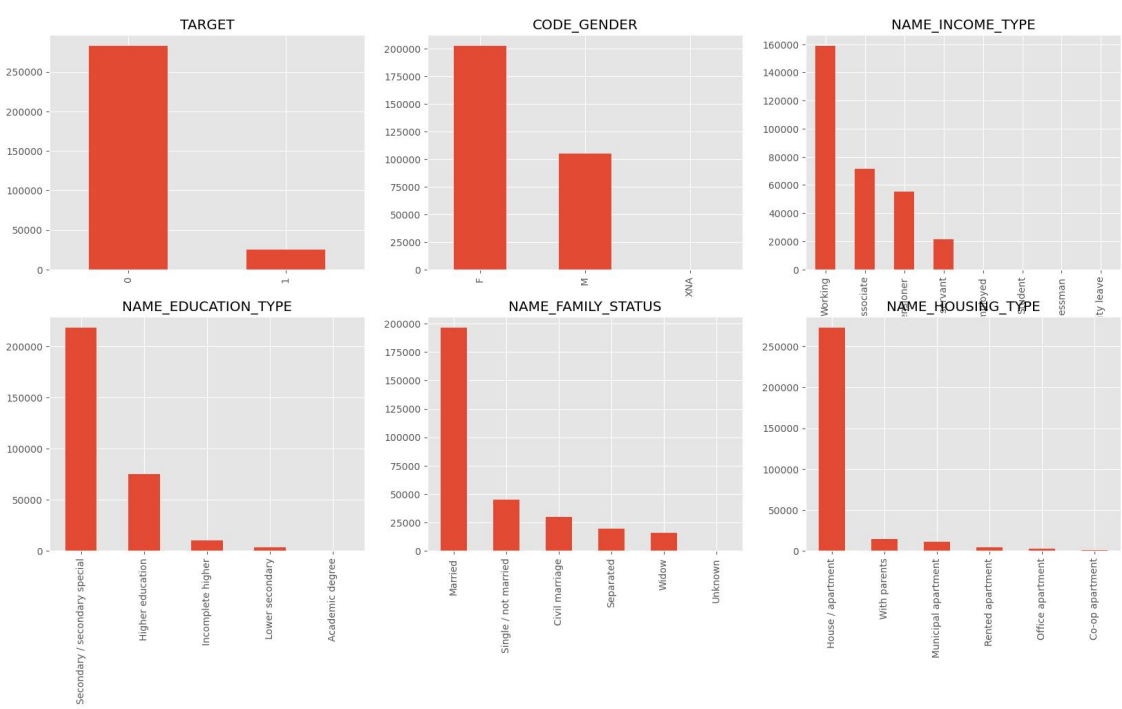
Q5. Identify if there is data imbalance in the data. Find the ratio of data imbalance.

```
In [174]: # Listing out columns to check data imbalance in them
col_list = ['TARGET', 'CODE_GENDER', 'NAME_INCOME_TYPE', 'NAME_EDUCATION_TYPE', 'NAME_FAMILY_STATUS', 'NAME_HOUSING_TYPE']

# Plotting Data Imbalance for different columns
k=0
plt.figure(figsize=(20,15))
for col in col_list:
    k=k+1
    plt.subplot(3, 3,k)
    app[col].value_counts().plot(kind='bar');
    plt.title(col)

In [179]: #We can clearly see the Imbalance in Target Value
Target0 = app.loc[app["TARGET"]==0] #Repayers
Target1 = app.loc[app["TARGET"]==1] #Defaulters

In [180]: #calculating ratio
print('Data Imbalance Ratio:',round(len(Target0)/len(Target1),2))
Data Imbalance Ratio: 11.39
```

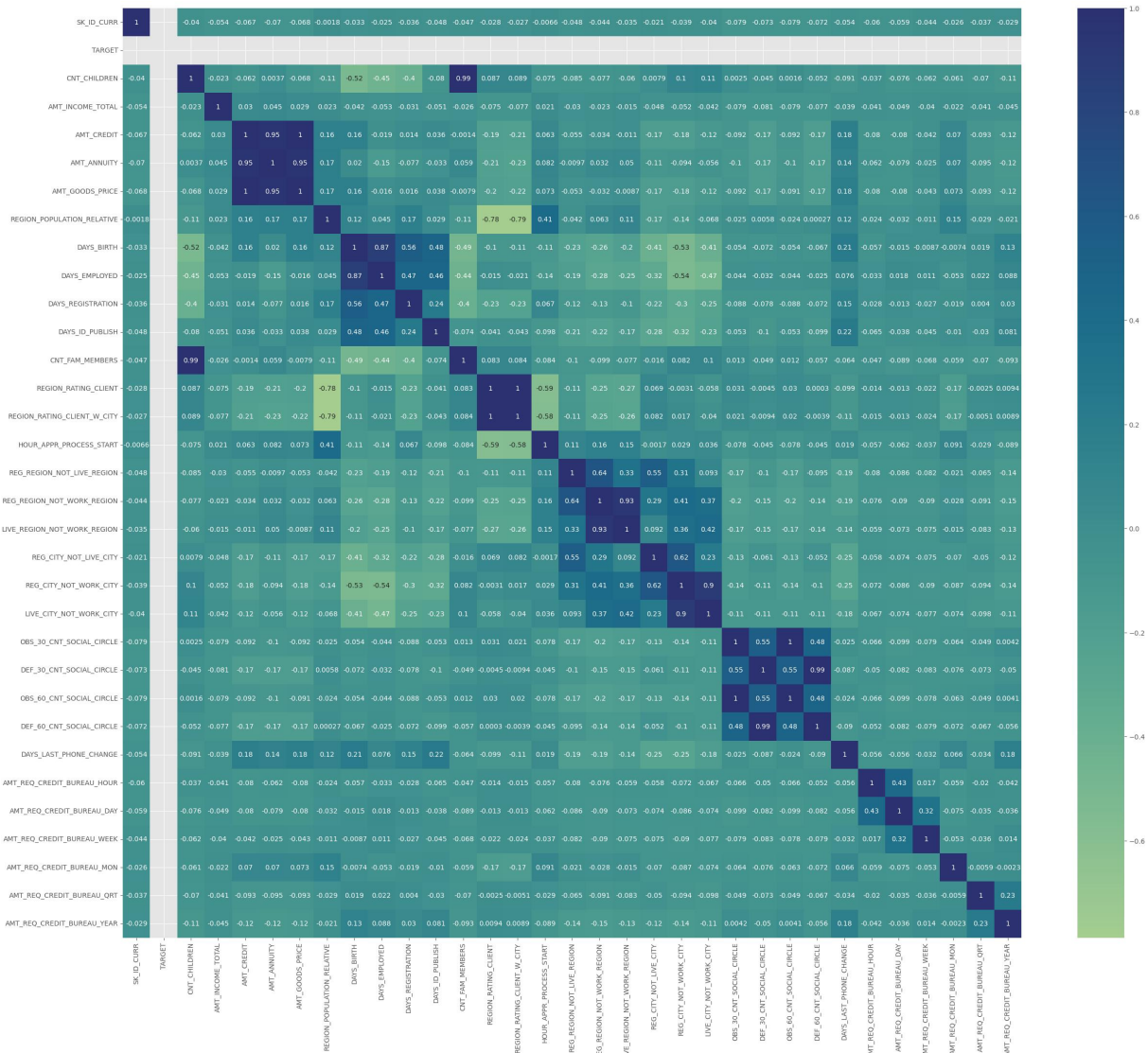


Q6. Find the top 10 **correlation** for the Client with payment difficulties and all other cases (Target variable).

Defaulter's correlation

	var1	var2	corr
814	OBS_60_CNT_SOCIAL_CIRCLE	OBS_30_CNT_SOCIAL_CIRCLE	0.998269
202	AMT_GOODS_PRICE	AMT_CREDIT	0.982783
475	REGION_RATING_CLIENT_W_CITY	REGION_RATING_CLIENT	0.956637
398	CNT_FAM_MEMBERS	CNT_CHILDREN	0.885484
848	DEF_60_CNT_SOCIAL_CIRCLE	DEF_30_CNT_SOCIAL_CIRCLE	0.868994
611	LIVE_REGION_NOT_WORK_REGION	REG_REGION_NOT_WORK_REGION	0.847885
713	LIVE_CITY_NOT_WORK_CITY	REG_CITY_NOT_WORK_CITY	0.778540
203	AMT_GOODS_PRICE	AMT_ANNUITY	0.752295

	var1	var2	corr
169	AMT_ANNUITY	AMT_CREDIT	0.752195
305	DAYS_EMPLOYED	DAYS_BIRTH	0.582185

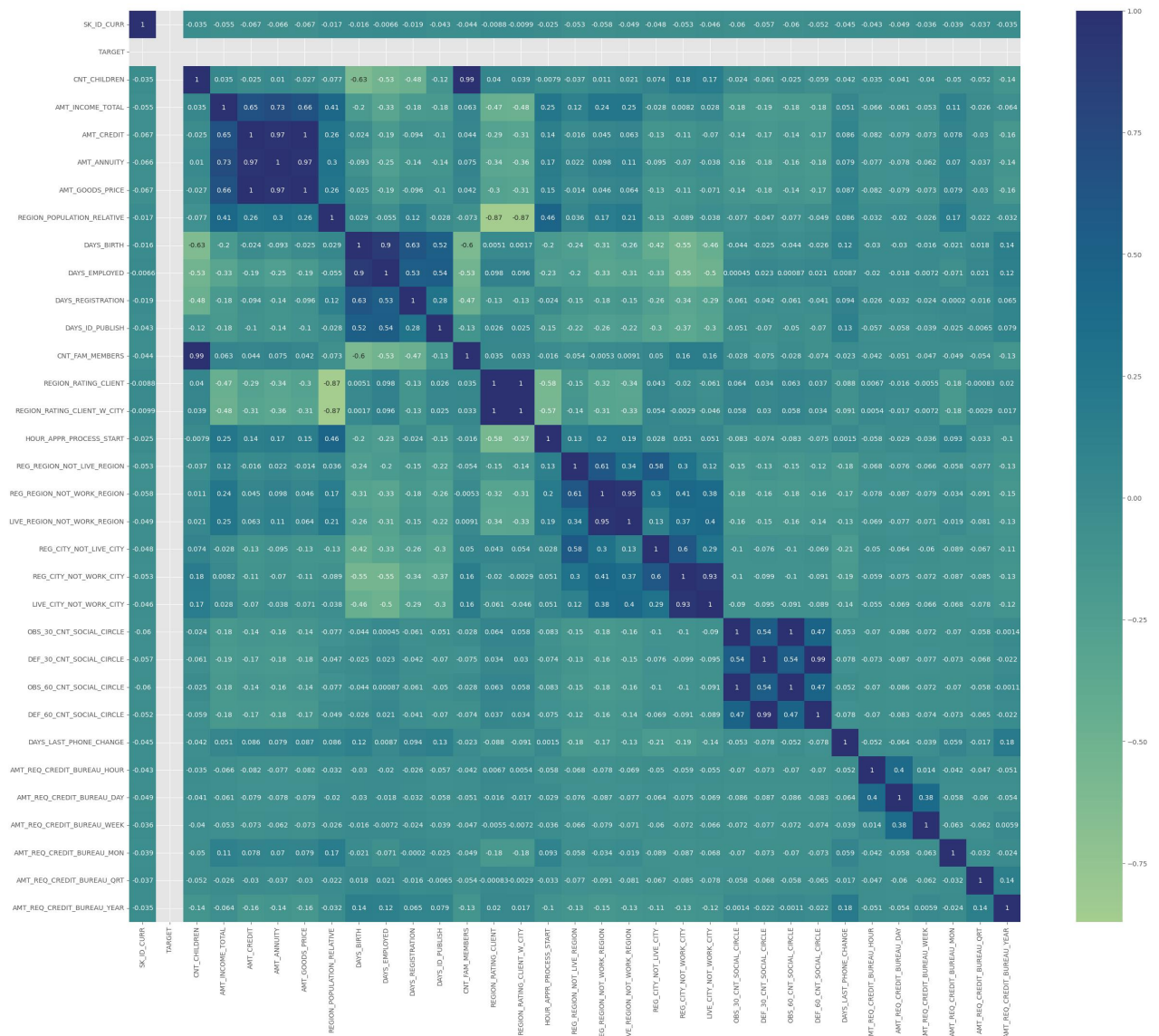


Correlation coefficient values less than +0.8 or greater than -0.8 are not considered significant. Hence LIVE\_CITY\_NOT\_WORK\_CITY, AMT\_GOODS\_PRICE, AMT\_ANNUITY, DAYS\_EMPLOYED are not significant.

## Re-payer's Correlation

	var1	var2	corr
814	OBS_60_CNT_SOCIAL_CIRCLE	OBS_30_CNT_SOCIAL_CIRCLE	0.998508
202	AMT_GOODS_PRICE	AMT_CREDIT	0.987022
475	REGION_RATING_CLIENT_W_CITY	REGION_RATING_CLIENT	0.950149
398	CNT_FAM_MEMBERS	CNT_CHILDREN	0.878571
611	LIVE_REGION_NOT_WORK_REGION	REG_REGION_NOT_WORK_REGION	0.861861

	var1	var2	corr
848	DEF_60_CNT_SOCIAL_CIRCLE	DEF_30_CNT_SOCIAL_CIRCLE	0.859332
713	LIVE_CITY_NOT_WORK_CITY	REG_CITY_NOT_WORK_CITY	0.830381
203	AMT_GOODS_PRICE	AMT_ANNUITY	0.776421
169	AMT_ANNUITY	AMT_CREDIT	0.771297
305	DAYS_EMPLOYED	DAYS_BIRTH	0.626114



Correlation coefficient values less than +0.8 or greater than -0.8 are not considered significant.

Hence AMT\_GOODS\_PRICE, AMT\_ANNUITY, DAYS\_EMPLOYED are not significant

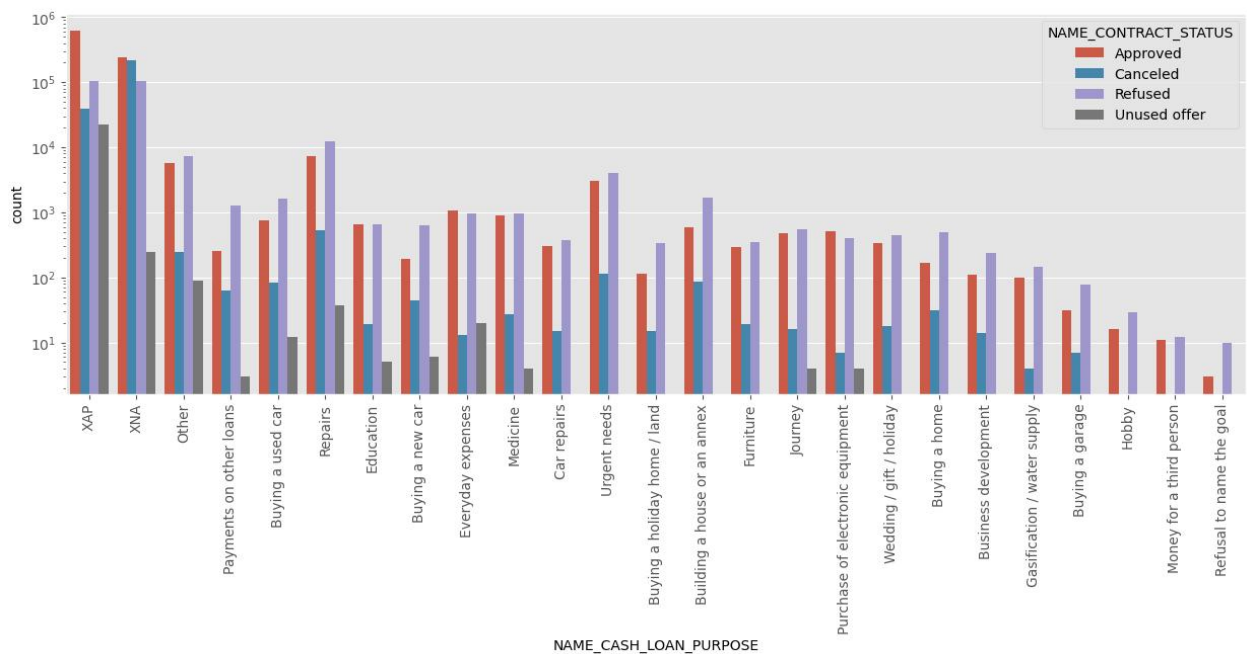


## Merging Previous application and Application datasets

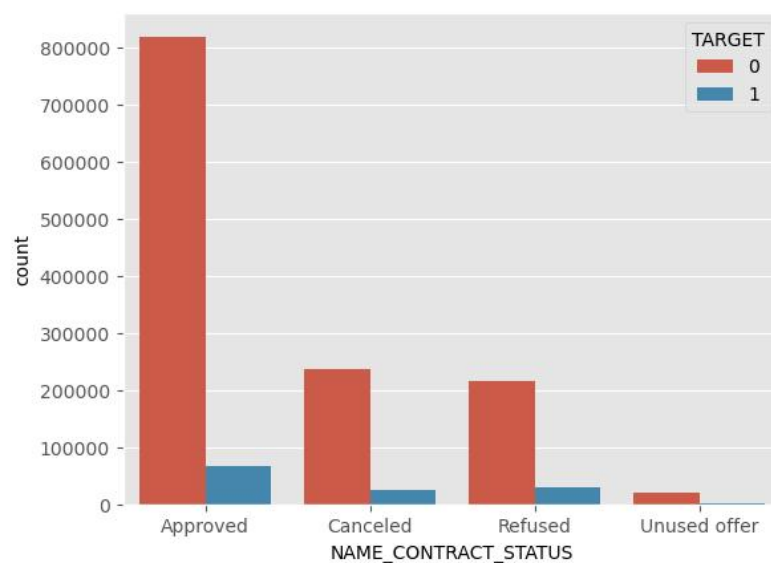
```
In [166]: #merging Previous application and Application datasets
merged_df = pd.merge(app_score_col_rmvd,prev_app_nva_col_rmvd,how='inner',on='SK_ID_CURR')
```

```
In [153]: plt.figure(figsize=(15,5))

sns.countplot(data=merged_df,x='NAME_CASH_LOAN_PURPOSE',hue='NAME_CONTRACT_STATUS')
plt.xticks(rotation=90)
plt.yscale('log')
```



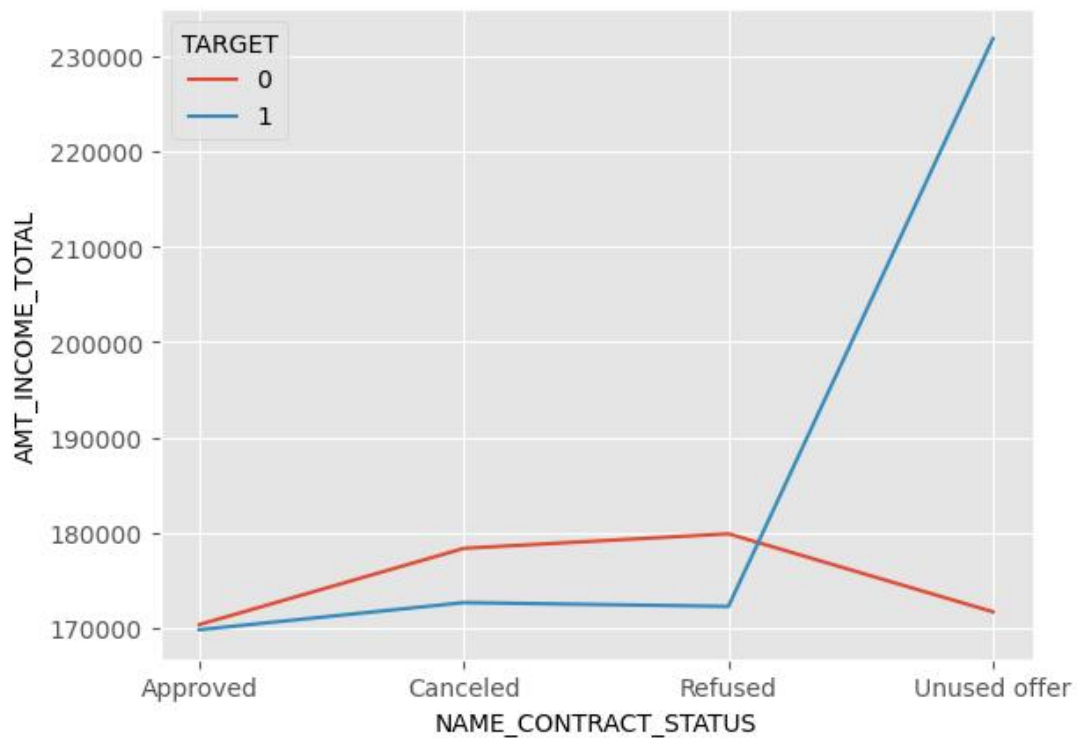
For repairing purpose customers had applied mostly previously and the same purpose has most number of cancellations.



Most of the applications which were previously canceled or refused , 80-90% of them are re-payers in the current data.

	NAME_CONTRACT_STATUS	TARGET	counts_x	counts_y	pct
0	Approved	0	818856	886099	92.41
1	Approved	1	67243	886099	7.59
2	Canceled	0	235641	259441	90.83
3	Canceled	1	23800	259441	9.17
4	Refused	0	215952	245390	88.00
5	Refused	1	29438	245390	12.00
6	Unused offer	0	20892	22771	91.75
7	Unused offer	1	1879	22771	8.25

0 means non - defaulters  
1 means defaulters



Offers which were unused previously, now have maximum number of defaulters despite of having high income band customers.

## Final Conclusions

### Bank should target the customers :

- Having income below 1 million
- Working in others, business entity type 3, self employed org. Type
- Working as Accountants, Core staff, Managers and Labourer
- Having house/ apartment
- Married and are having children not more than 5
- Highly educated
- Preferably female
- Unaccompanied people can be safer - default rate is ~8.5%

### Amount segment recommended :

- The credit amount should not be more than 1 million
- Annuity can be made of 50K depending on the eligibility
- Income could be below 1 million
- 80 - 90% of the customers whose application were previously canceled/ refused were re-payers

### Precaution :

- Org. Transport type 3 should be avoided
- Low skilled laborers and drivers should be avoided
- Offers which were unused previously and high income band customers must be avoided

GitHub link for Jupyter file:

<https://github.com/PriyankaPrabhu7/BankLoanCaseStudy>

Loom video link:

<https://www.loom.com/share/17c8ac0152944fa49474cafe4c4777d9>



