Loan Approval DataSet

Problem Statment 1

By Shaik Meer Hussain

Exploratory Data Analysis

Problem statement 1

1. Data Understanding

1.We have 3 folders of data

- application_data.csv
  - It has information weather the client has payment difficulties
- previous_application.csv
  - It has data about the previous applications and includes
  - Approved
  - Canceled
  - refused
  - Unused
- columns_description.csv
  - Give an idea about all variables in the dataset

Some of the Basic operations are :

- head() - get the first part of the data frame
- info() - print a summary of the data frame
- describe() - it includes mean count STD percentile and min max values
- shape() - gives the shape of a data frame or vector or a matrix
- colums() - gives the names of the columns
- values() - gives the list of the values
- And so on....

In [ ]:
```
#reading the first colums of the data set
data.head()
```

sample operations on the data set

In [ ]:
```
#shape of the document gives rows and columns
data.shape
```

In [ ]:
```
#info gives the columns,dtypes
data.info()
```

In [ ]:
```
#getting all the info of the dataset
data.info(verbose = 1)
```

In [ ]:
```
#describing the data
data.describe()
```

In [ ]:
```
#finding the datatypes of the variables in the given dataset
data.dtypes
#difficult to access all BUT ALL OF THEM ARE IN CORRECT DATA TYPES
```

In [ ]:
```
#finding the count of the null values in the dataset

data.isnull().sum()
```

2. Data Cleaning and Manipulation

- This includes cleaning raw data and converting it into more handy and useful
- After redaing, look after for the unnecessary data such as null
- If you had an idea about the null points count and make it a percentile for that null values
- If the values are grater than 50 simply drop them and manage the required data by imputing mean, median and mode

```
Must include necessary import files
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

```
data = pd.read_csv("application_data.csv")
```

```
drop_col = data.isnull().sum()/len(data)*100
drop_col = drop_col[drop_col > 50.0]
```

```
data['AMT_ANNUITY'].fillna(value=filling, inplace = True)
```

For the cleaning process, Note all the null values

Find if there are any outliers[not mandatory]

2.1 Standardise the values

[cleaning process]

Renaming and converting raw data in variables

	SK_ID_CURR	TARGET	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_GOODS_PRICE	REGION_POPULATION_RELATIVE	YEARS_BIRT
count	307511.000000	307511.000000	307511.000000	3.075110e+05	3.075110e+05	3.072330e+05	307511.000000	307511.00000
mean	278180.518577	0.080729	0.417052	1.687979e+05	5.990260e+05	5.383962e+05	0.020868	43.93697
std	102790.175348	0.272419	0.722121	2.371231e+05	4.024908e+05	3.694465e+05	0.013831	11.95613
min	100002.000000	0.000000	0.000000	2.565000e+04	4.500000e+04	4.050000e+04	0.000290	20.51780
25%	189145.500000	0.000000	0.000000	1.125000e+05	2.700000e+05	2.385000e+05	0.010006	34.00821
50%	278202.000000	0.000000	0.000000	1.471500e+05	5.135310e+05	4.500000e+05	0.018850	43.15068
75%	367142.500000	0.000000	1.000000	2.025000e+05	8.086500e+05	6.795000e+05	0.028663	53.92328
max	456255.000000	1.000000	19.000000	1.170000e+08	4.050000e+06	4.050000e+06	0.072508	69.12054

As you can see the data is not fully cleaning the variable data is complex...convert it into simplest form

- Values such as AMT_INCOME_TOTAL has an complex output
- Convert it into simple forms by dividing it by 100000

data['AMT_INCOME_TOTAL']=data['AMT_INCOME_TOTAL']/100000

- And similarly go for all the remaining raw values

	DAYS_BIRTH	DAYS_EMPLOYED	DAYS_REGISTRATION	DAYS_ID_PUBLISH	DAYS_LAST_PHONE_CHANGE
count	307511.000000	307511.000000	307511.000000	307511.000000	307510.000000
mean	16036.995067	67724.742149	4986.120328	2994.202373	962.858788
std	4363.988632	139443.751806	3522.886321	1509.450419	826.808487
min	7489.000000	0.000000	0.000000	0.000000	0.000000
25%	12413.000000	933.000000	2010.000000	1720.000000	274.000000
50%	15750.000000	2219.000000	4504.000000	3254.000000	757.000000
75%	19682.000000	5707.000000	7479.500000	4299.000000	1570.000000
max	25229.000000	365243.000000	24672.000000	7197.000000	4292.000000

- After this the values in days are more complex making it into simple numbers by dividing it by 365

data[col_with_days]=data[col_with_days]/365

- After the conversion renaming the columns to years would give a better understanding

data.rename(columns={'DAYS_BIRTH':'YEARS_BIRTH','DAYS_EMPLOYED':'YEARS_EMPLOYED','DAYS_REGISTRATION':'YEARS_REGISTRATION','DAYS_ID_PUBLISH':'YEARS_ID_PUBLISH','DAYS_LAST_PHONE_CHANGE':'YEARS_LAST_PHONE_CHANGE'},inplace=True)

- After that check for the un recognised data types AND convert them into related data type
- I do have converted it into hybrid form such as
- category datatype which is more convenient and is simple to convert instead of turning it into numeric type by bumpy vales
- data[i]=data[i].astype(‘category')

```
SK_ID_CURR                      int64
TARGET                       category
NAME_CONTRACT_TYPE           category
CODE_GENDER                  category
FLAG_OWN_CAR                 category
FLAG_OWN_REALTY              category
CNT_CHILDREN                 category
AMT_INCOME_TOTAL              float64
AMT_CREDIT                    float64
AMT_ANNUITY                  category
AMT_GOODS_PRICE               float64
NAME_TYPE_SUITE              category
NAME_INCOME_TYPE             category
NAME_EDUCATION_TYPE          category
NAME_FAMILY_STATUS           category
NAME_HOUSING_TYPE            category
REGION_POPULATION_RELATIVE    float64
YEARS_BIRTH                   float64
YEARS_EMPLOYED                float64
YEARS_REGISTRATION            float64
YEARS_ID_PUBLISH              float64
FLAG_MOBIL                   category
CNT_FAM_MEMBERS              category
WEEKDAY_APPR_PROCESS_START   category
HOUR_APPR_PROCESS_START      category
```

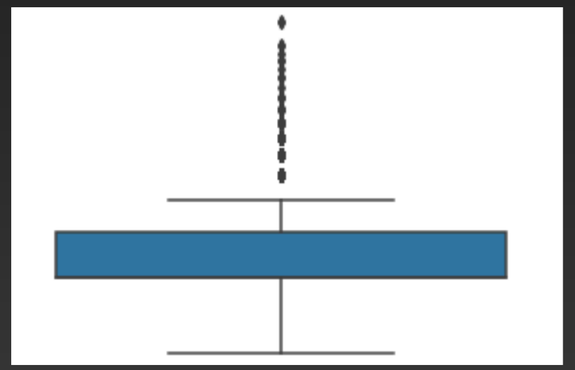
BINNING & OUTLIERS

- Binnig the salary ranges will give us the convinient data format

AMT_INCOME_RANGE	AMT_CREDIT_RANGE
0-25000	0-150000
0-25000	0-150000
0-25000	0-150000
0-25000	0-150000
0-25000	0-150000

- Outliers gives the out of the box values, In this we are just showing the outliers instead of handling the data

- The outliers can be shown below



3. Analysis

Data Analysis is **the process of systematically applying statistical and/or logical techniques to describe and illustrate, condense and recap, and evaluate data.**

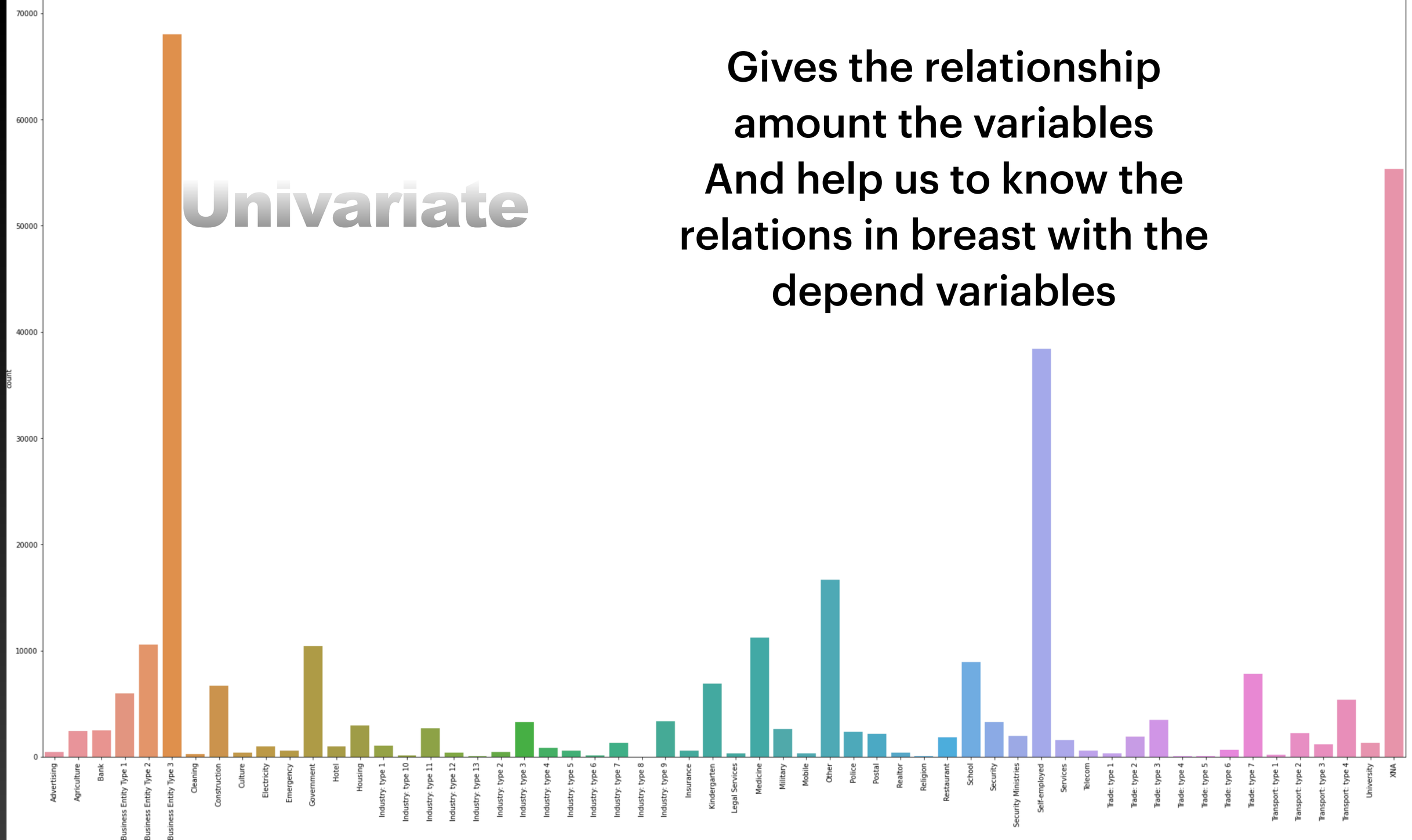
3.1 Univariate Analysis

Univariate analysis is **the simplest form of analyzing data**. Uni means one, so in other words the data has only one variable. Univariate data requires to analyze each variable separately. Data is gathered for the purpose of answering a question, or more specifically, a research question.

- We are using plain plots for the univariate analysis
- Plots such as
    - Box
    - Count
    - Heat
    - And dividing the plot areas using the subplot
    - Not to mention figure gives the shape and size to the chart

Univariate

Gives the relationship amount the variables
And help us to know the relations in breast with the depend variables

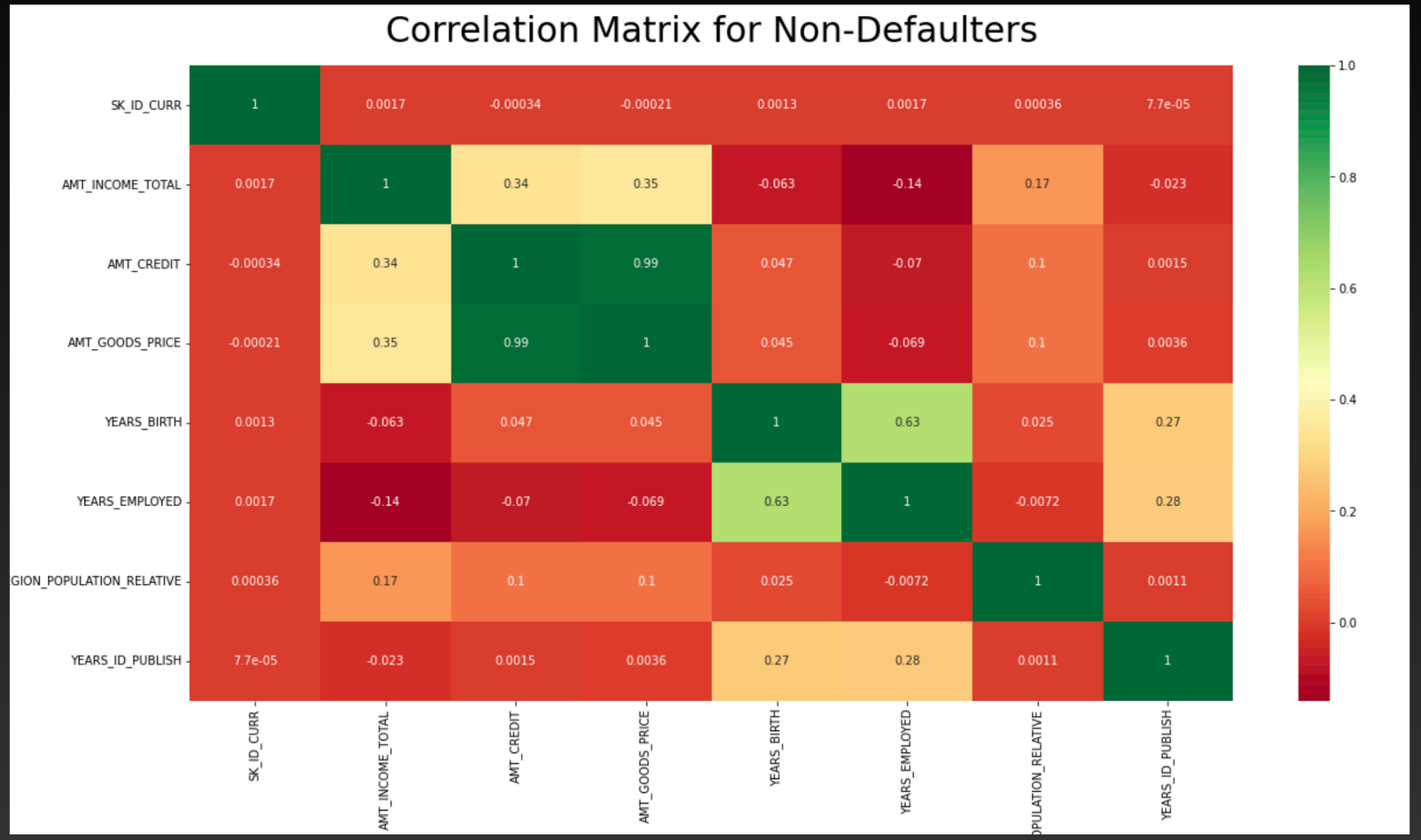


3.2 Bivariate Analysis

Classifying the data based on different variables is Bi-Variate Analysis

Lets look some of the plots of the bivariate analysis

Based on the target variable we are estimating possibilities of yes and no statements



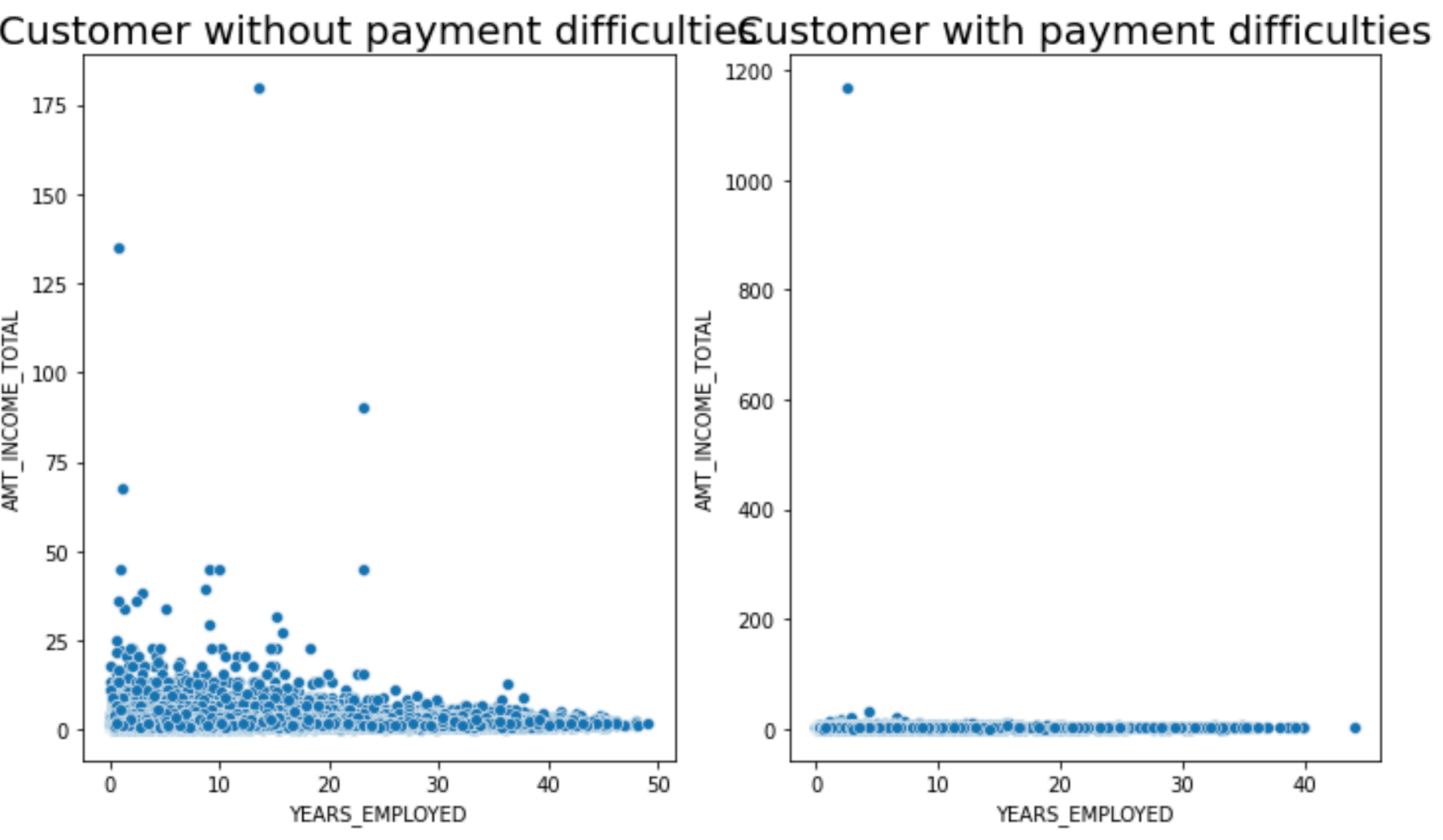
- Target variable has values ranging from 0 to 1
- Target_0 gives the rejected pieces
- Target_1 gives the accepted ones
- Assuming the concept we are doing the remaining process
- Even though its target 0 it has the values which is dependent on other values such as
- Labour,education,credit
- And so on.....

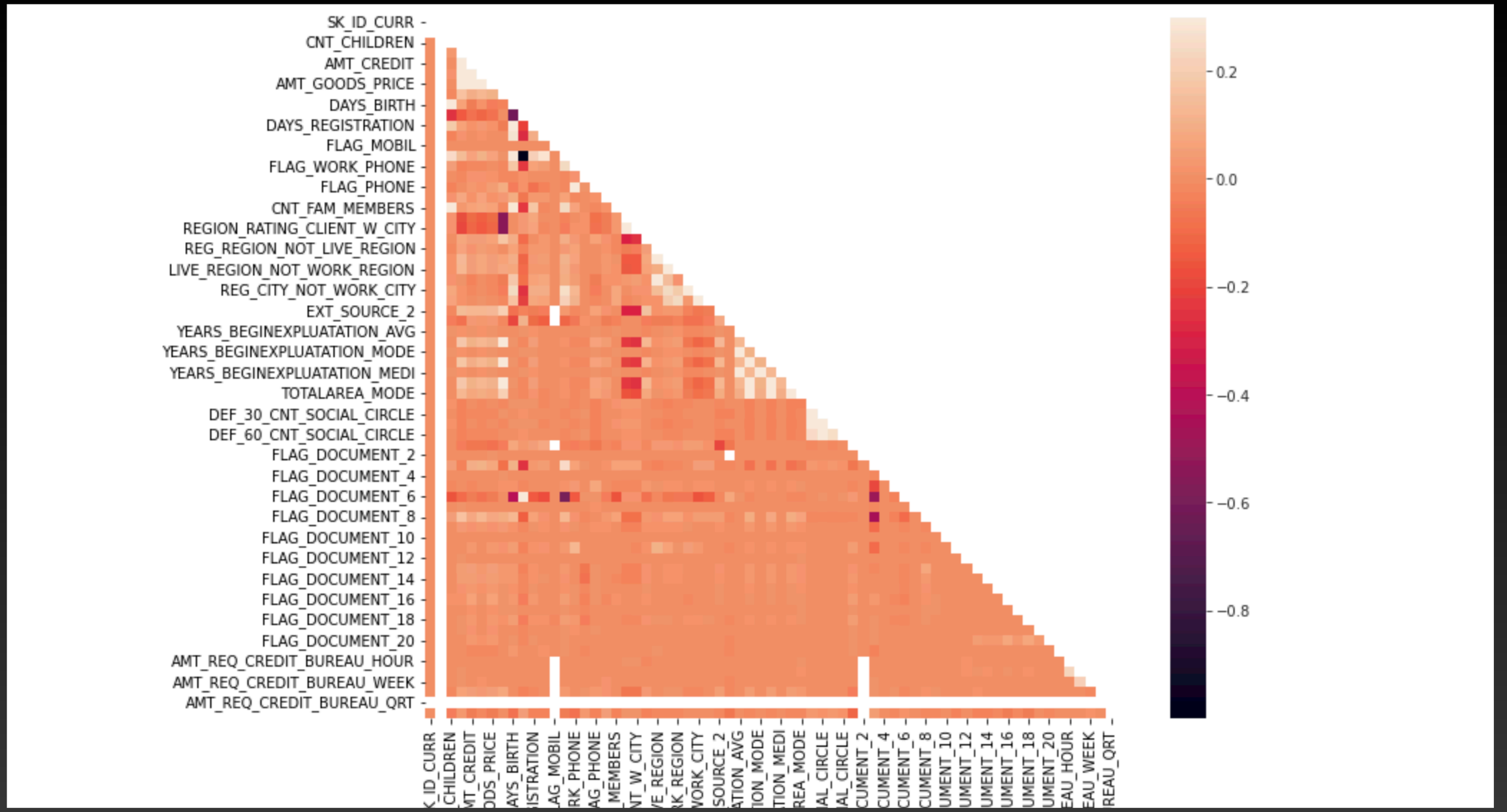
```
elements = data[[ 'AMT_INCOME_TOTAL','AMT_CREDIT']]

fig = plt.figure(figsize=(16,12))

for i in enumerate(elements):
plt.subplot(2,2,i[0]+1)
sns.distplot(Target_1[i[1]], hist=0,label ="Unable to pay persons")
sns.distplot(Target_0[i[1]], hist=True, label ="Able to pay persons")
plt.subplots_adjust(hspace=1)
plt.title(i[1], fontdict={'fontsize' : 15, 'fontweight' : 5})
plt.legend()


plt.show()
```


Customer without payment difficulties
Customer with payment difficulties
AMT_INCOME_TOTAL
YEARS_EMPLOYED
0
25
50
75
100
125
150
175
200
400
600
800
1000
1200
0
10
20
30
40
50


SK_ID_CURR
CNT_CHILDREN
AMT_CREDIT
AMT_GOODS_PRICE
DAYS_BIRTH
DAYS_REGISTRATION
FLAG_MOBIL
FLAG_WORK_PHONE
FLAG_PHONE
CNT_FAM_MEMBERS
REGION_RATING_CLIENT_W_CITY
REG_REGION_NOT_LIVE_REGION
LIVE_REGION_NOT_WORK_REGION
REG_CITY_NOT_WORK_CITY
EXT_SOURCE_2
YEARS_BEGINEXPLUATATION_AVG
YEARS_BEGINEXPLUATATION_MODE
YEARS_BEGINEXPLUATATION_MEDI
TOTALAREA_MODE
DEF_30_CNT_SOCIAL_CIRCLE
DEF_60_CNT_SOCIAL_CIRCLE
FLAG_DOCUMENT_2
FLAG_DOCUMENT_4
FLAG_DOCUMENT_6
FLAG_DOCUMENT_8
FLAG_DOCUMENT_10
FLAG_DOCUMENT_12
FLAG_DOCUMENT_14
FLAG_DOCUMENT_16
FLAG_DOCUMENT_18
FLAG_DOCUMENT_20
AMT_REQ_CREDIT_BUREAU_HOUR
AMT_REQ_CREDIT_BUREAU_WEEK
AMT_REQ_CREDIT_BUREAU_QRT
0.2
0.0
−0.2
−0.4
−0.6
−0.8

Conclusion

Problem statement 1

- At the end of this we have founded that
  - Labours are more likely becoming defaulters
  - And good credit holders are likely non- defaulters
  - Along education is also having a high impact
  - Apart of this, people ho are unable to pay and having difficulty to pay can be categorised based on the job experience along side with the living house
  - To be simple based on this application data we need to filter the persons
  - Who are less likely to become defaulters

PROBLEM STATEMENT II

PREVIOUS DATA MERGING

1.Data Understanding

1.read the data to have a clear understanding what the data is about

2. Take the necessary things I mean the columns that are similar to the application data

3.Know about the variables and data types

4.merge the csv files by giving necessary unique ids in both of them and do operations on the merged file

```
merged = pd.merge(data,prev_data,how='inner',on='SK_ID_CURR')
```

2.Cleaning and data manipulating

Cleaning is a step done before merging

- This cleaning includes dropping of necessary items and checking the data types
- This is clear in the file of problemstatement1 & 2.ipynb
- As the ppt also gives the idea of the cleaning if the
- Percentile is above 50% simply drop the columns but mentioning the columns as the index values
- Hence cleaning is done

In [*]:
```
#Getting the previous data of the values
prev_data=pd.read_csv("previous_application.csv")
```

In [*]:
```
#doing certain basic operations on the previous data to find about it
```

In [*]:
```
prev_data.info()
```

In [*]:
```
prev_data.describe()
```

In [*]:
```
prev_data.dtypes
```

In [*]:
```
prev_data.isnull()
```

In [*]:
```
prev_data.shape
```

2. DATA CLEANING

In [*]:
```
#prev_data.isnull().sum()
count = prev_data.isnull().sum()
```

In [*]:
```
percentile = count*100/len(prev_data)
```

In [*]:
```
percentile
```

In [*]:
```
#getting the values whose percentile is above 50%
above_50=percentile[percentile>=50]
```

In [*]:
```
len(above_50)
```

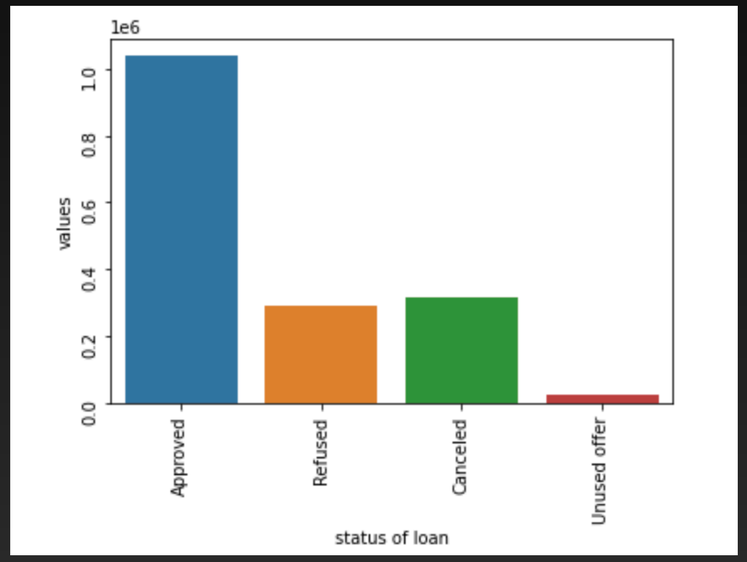
In [*]:
```
above_50
```

In [*]:
```
#getting the values whose percentile is below 20%
below_20 = percentile[percentile<=20]
```

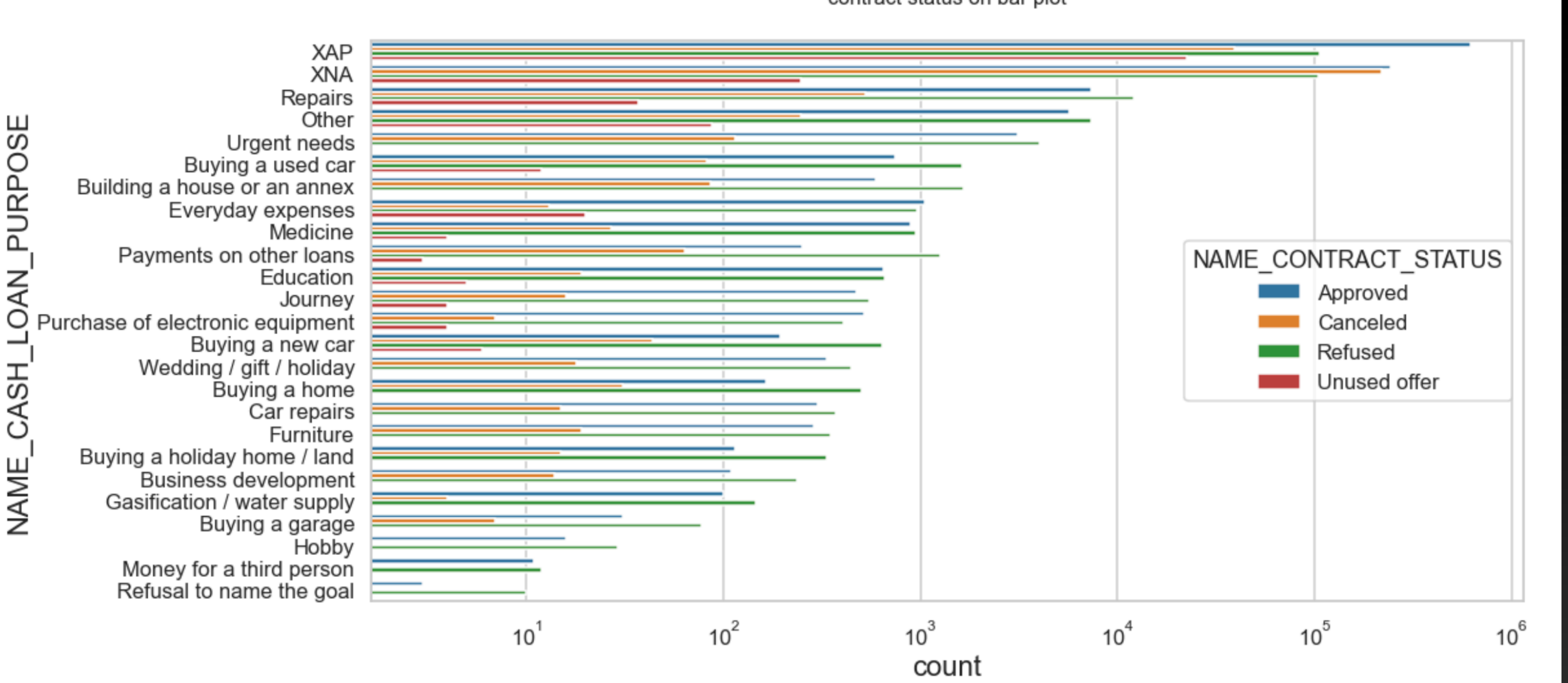
3.ANALYSIS

3.1 UNIVARIATE ANALYSIS

Performing operations on the merged files

- 

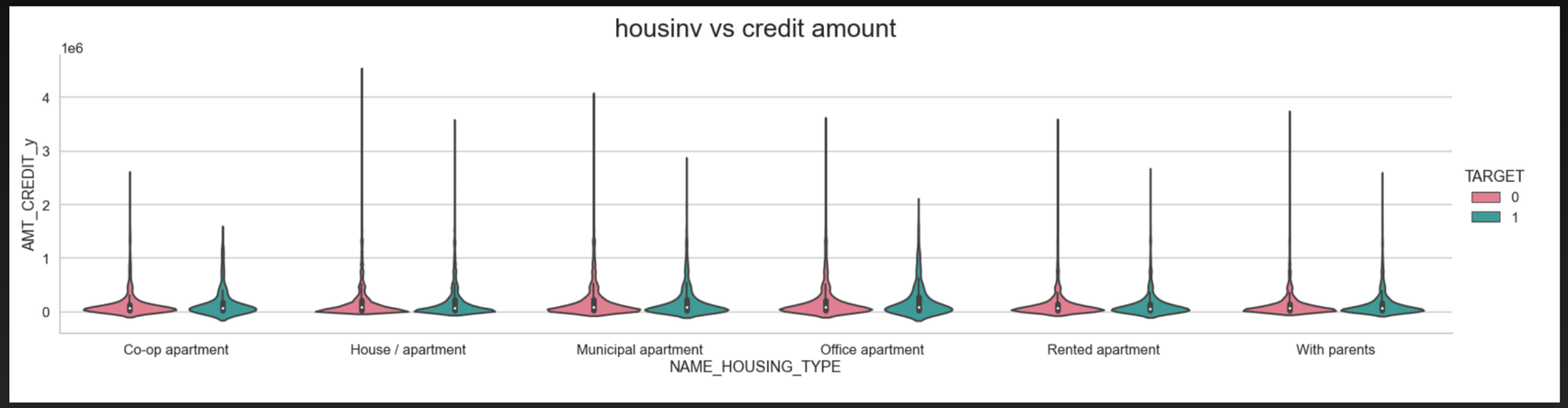
- In this process we can simply identify the values who are
- Refused, Approved, Cancelled and unused offer clients



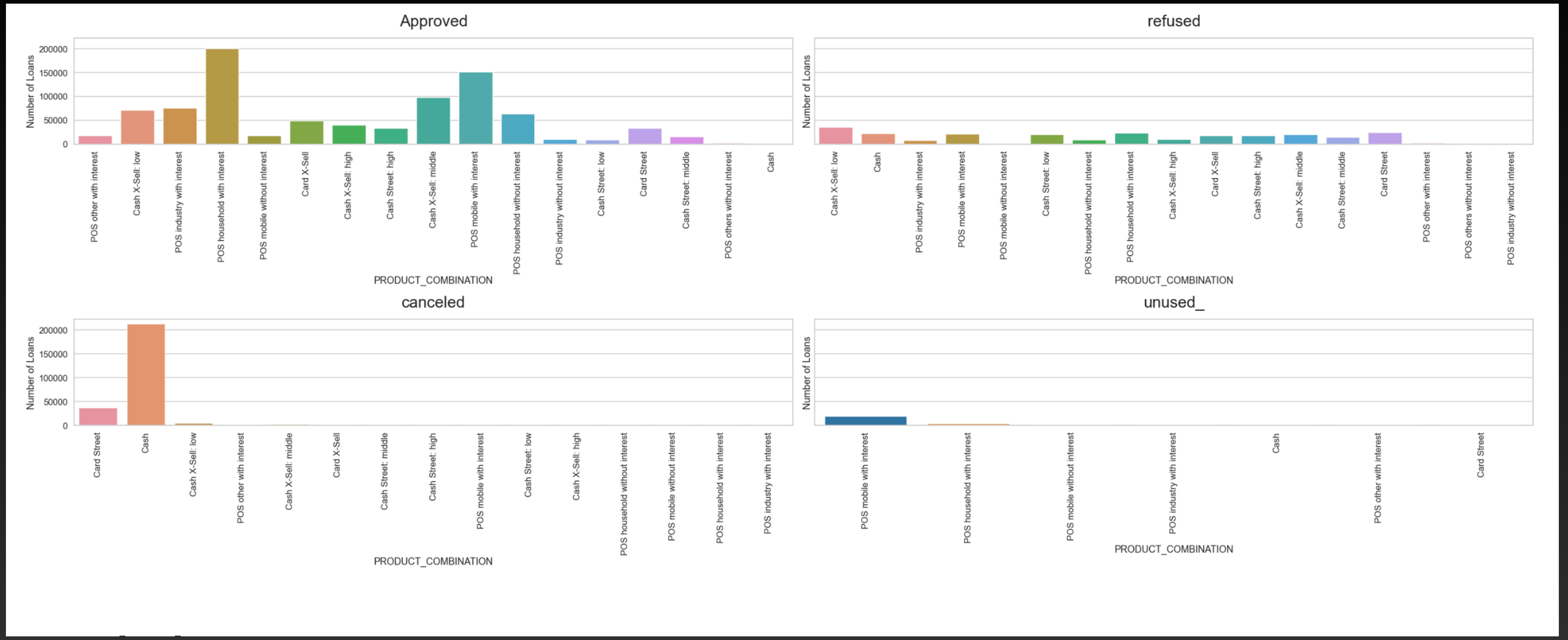
- Simply can understand that the variation of loan usage

3.2 Bivariate analysis

[merged file] merge

- 

- Can simply understand the housing people based on the credit vales



Count plot for all the variables

“Conclusion”

Focussing on the people with higher credit, own houses along with the good education rate will provide you the maximum success approval loan rates

Apart from this people with low education and no house scheme and with low credit income may have the chance of refusal

The people who may have the chance to pay or unable to pay must be recorded clearly and necessary time limits are given to that clients who is having the possibility to pay

And the people who can’t pay must be noted as defaulters and precautions are taken on their basis for not approving defaulters based on the application data of the clients

THANK YOU

YOURS SINCERLY SK.MEER HUSSAIN