Bank Loan Case Study Project

NIHARIKA SAI LALITHA PRESENTATION BEGINS

Table of contents

01 Project Description

02 Approach

03 Tech-Stack Used

04 Insights

05 Results

01
Project Description

Description of Bank Loan Case Study Project

This case study aims to give an idea of applying EDA in a real business scenario. In this case study, apart from applying the techniques that you have learnt in the EDA module, you will also develop a basic understanding of risk analytics in banking and financial services and understand how data is used to minimize the risk of losing money while lending to customers.

—Description

- ★ The loan providing companies find it hard to give loans to the people due to their insufficient or non-existent credit history. Because of that, some consumers use it as their advantage by becoming a defaulter. Suppose you work for a consumer finance company which specialises in lending various types of loans to urban customers. You have to use EDA to analyse the patterns present in the data. This will ensure that the applicants capable of repaying the loan are not rejected.

- ★ When the company receives a loan application, the company has to decide for loan approval based on the applicant's profile. Two types of risks are associated with the bank's decision:
  - If the applicant is likely to repay the loan, then not approving the loan results in a loss of business to the company.
  - If the applicant is not likely to repay the loan, i.e. he/she is likely to default, then approving the loan may lead to a financial loss for the company

—Business Understanding

- ★ The loan providing companies find it hard to give loans to the people due to their insufficient or non-existent credit history. Because of that, some consumers use it as their advantage by becoming a defaulter. Suppose you work for a consumer finance company which specialises in lending various types of loans to urban customers. You have to use EDA to analyse the patterns present in the data. This will ensure that the applicants capable of repaying the loan are not rejected.
- ★ When the company receives a loan application, the company has to decide for loan approval based on the applicant's profile. Two types of risks are associated with the bank's decision:
    - If the applicant is likely to repay the loan, then not approving the loan results in a loss of business to the company.
    - If the applicant is not likely to repay the loan, i.e. he/she is likely to default, then approving the loan may lead to a financial loss for the company

—Business Understanding (Continuation)

—Business Understanding (Continuation)

★ The data given below contains the information about the loan application at the time of applying for the loan. It contains two types of scenarios:
  - The client with payment difficulties: he/she had late payment more than X days on at least one of the first Y instalments of the loan in our sample
  - All other cases: All other cases when the payment is paid on time.
  - If the applicant is likely to repay the loan, then not approving the loan results in a loss of business to the company.

★ When a client applies for a loan, there are four types of decisions that could be taken by the client/company:
  1. **Approved:** The company has approved loan application
  2. **Cancelled:** The client cancelled the application sometime during approval. Either the client changed her/his mind about the loan or in some cases due to a higher risk of the client he received worse pricing which he did not want.
  3. **Refused:** The company had rejected the loan (because the client does not meet their requirements etc.).
  4. **Unused Offer:** Loan has been cancelled by the client but on different stages of the process.

In this case study, you will use EDA to understand how consumer attributes and loan attributes influence the tendency of default.

- It aims 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.
- To develop your understanding of the domain, you are advised to independently research a little about **risk analytics** – understanding the types of variables and their significance should be enough).

—Business Objective

Three csv files given under dataset section:

1. **`application_data.csv`** contains all the information of the client at the time of application.
   The data is about whether a client has payment difficulties.
2. **`previous_application.csv`** contains information about the client's previous loan data. It contains the data whether the previous application had been Approved, Cancelled, Refused or Unused offer.
3. **`columns_descrption.csv`** is data dictionary which describes the meaning of the variables.

—Data Understanding

02
Approach

Approach to the Case Study

Our Approach

Taking datasets into consideration and analysed, applied EDA steps as a best approach to remove irrelevant columns, filled missed data according to the column. Finally furnished the dataset flawlessly in csv files to get relevant solution for the problem

03

Tech-Stack Used

Technologies used in this Case Study

Tech-Stack

PPT

It's the PPT for documenting the problem and solution with the given datasets

Google sheets and Excel

To see the data and used filter when necessary

Jupiter

Installed jupiter to run the python code and presented analysis using chart

04
Insight

Insight for business needs

Insight

Gained knowledge on how to analyze each given problem and find relevant solutions to it using

- Learned matplotlib library
- Got a good drip on seaborn
- All EDA steps
- Learned how to analyse univariate, Bivariant, segment univariant , . . . analysis
- Formatting the charts

At last learned to give correct insights to improve business success growth

05
Result

Result of dataset analysis

Application Data Analysis

```
df_app.duplicated().sum()
```

```
0
```

```
df_app.shape
```

```
(307511, 122)
```

```
df_app.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
```

```
df_app.head()
```

1 to 5 of 5 entries Filter

index	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	A
0	100002	1	Cash loans	M	N	Y	0	202500.0	406597.5	24700.5	
1	100003	0	Cash loans	F	N	N	0	270000.0	1293502.5	35698.5	
2	100004	0	Revolving loans	M	Y	Y	0	67500.0	135000.0	6750.0	
3	100006	0	Cash loans	F	N	Y	0	135000.0	312682.5	29686.5	
4	100007	0	Cash loans	M	N	Y	0	121500.0	513000.0	21865.5	

```
df_app.tail()
```

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	...	FLAG_DOCU
307506	456251	0	Cash loans	M	N	N	0	157500.0	254700.0	27558.0	...	
307507	456252	0	Cash loans	F	N	Y	0	72000.0	269550.0	12001.5	...	
307508	456253	0	Cash loans	F	N	Y	0	153000.0	677664.0	29979.0	...	
307509	456254	1	Cash loans	F	N	Y	0	171000.0	370107.0	20205.0	...	
307510	456255	0	Cash loans	F	N	N	0	157500.0	675000.0	49117.5	...	

5 rows × 122 columns

Application Data Analysis

```
df_app.describe()
```

	SK_ID_CURR	TARGET	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_PRICE	REGION_POPULATION_RELATIVE	DAYS_BIRTH	DAYS_E
count	307511.000000	307511.000000	307511.000000	3.075110e+05	3.075110e+05	307499.000000	3.072330e+05	307511.000000	307511.000000	307511
mean	278180.518577	0.080729	0.417052	1.687979e+05	5.990260e+05	27108.573909	5.383962e+05	0.020868	-16036.995067	63815
std	102790.175348	0.272419	0.722121	2.371231e+05	4.024908e+05	14493.737315	3.694465e+05	0.013831	4363.988632	141275
min	100002.000000	0.000000	0.000000	2.565000e+04	4.500000e+04	1615.500000	4.050000e+04	0.000290	-25229.000000	-17912
25%	189145.500000	0.000000	0.000000	1.125000e+05	2.700000e+05	16524.000000	2.385000e+05	0.010006	-19682.000000	-2760
50%	278202.000000	0.000000	0.000000	1.471500e+05	5.135310e+05	24903.000000	4.500000e+05	0.018850	-15750.000000	-1213
75%	367142.500000	0.000000	1.000000	2.025000e+05	8.086500e+05	34596.000000	6.795000e+05	0.028663	-12413.000000	-289
max	456255.000000	1.000000	19.000000	1.170000e+08	4.050000e+06	258025.500000	4.050000e+06	0.072508	-7489.000000	365243

8 rows × 106 columns

```
df_app.dtypes
```

```
SK_ID_CURR                     int64
TARGET                         int64
NAME_CONTRACT_TYPE            object
CODE_GENDER                   object
FLAG_OWN_CAR                  object
                               ...
AMT_REQ_CREDIT_BUREAU_DAY    float64
AMT_REQ_CREDIT_BUREAU_WEEK   float64
AMT_REQ_CREDIT_BUREAU_MON    float64
AMT_REQ_CREDIT_BUREAU_QRT    float64
AMT_REQ_CREDIT_BUREAU_YEAR   float64
Length: 122, dtype: object
```

```
df_app.select_dtypes(include=["int64","float64"]).columns
```

```
Index(['SK_ID_CURR', 'TARGET', 'CNT_CHILDREN', 'AMT_INCOME_TOTAL',
       'AMT_CREDIT', 'AMT_ANNUITY', 'AMT_GOODS_PRICE',
       'REGION_POPULATION_RELATIVE', 'DAYS_BIRTH', 'DAYS_EMPLOYED',
```

```
df_app.select_dtypes(include="object").columns
```

```
Index(['NAME_CONTRACT_TYPE', 'CODE_GENDER', 'FLAG_OWN_CAR', 'FLAG_OWN_REALTY',
       'NAME_TYPE_SUITE', 'NAME_INCOME_TYPE', 'NAME_EDUCATION_TYPE',
       'NAME_FAMILY_STATUS', 'NAME_HOUSING_TYPE', 'OCCUPATION_TYPE',
```

```
App_data=round(df_app.isnull().mean(axis=0).sort_values(ascending=False)*100,2)
print((App_data))
```

```
COMMONAREA_MEDI               69.87
COMMONAREA_AVG                69.87
COMMONAREA_MODE               69.87
NONLIVINGAPARTMENTS_MODE      69.43
NONLIVINGAPARTMENTS_AVG       69.43
                               ...
NAME_HOUSING_TYPE              0.00
NAME_FAMILY_STATUS             0.00
NAME_EDUCATION_TYPE            0.00
NAME_INCOME_TYPE               0.00
SK_ID_CURR                     0.00
Length: 122, dtype: float64
```

COUNTING THE PERCENTAGE OF NULL VALUES IN EACH COLUMN

```
App_data = df_app.loc[:,App_data<32]
print(len(App_data.columns))
```

```
73
```

STORING DATA WHERE COLUMNS HAVE <32% OF NULL VALUES INTO APP_DATA

```
curr_to_drop = ['OBS_30_CNT_SOCIAL_CIRCLE', 'DEF_30_CNT_SOCIAL_CIRCLE','OBS_60_CNT_SOCIAL_CIRCLE', 'DEF_60_CNT_SOCIAL_CIRCLE',
                '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', 'REGION_POPULATION_RELATIVE', 'FLAG_MOBIL',
                'FLAG_EMP_PHONE', 'FLAG_WORK_PHONE', 'FLAG_CONT_MOBILE', 'FLAG_PHONE',
                'FLAG_EMAIL', 'WEEKDAY_APPR_PROCESS_START', 'HOUR_APPR_PROCESS_START',
                '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', 'EXT_SOURCE_2','AMT_REQ_CREDIT_BUREAU_YEAR','AMT_REQ_CREDIT_BUREAU_QRT','AMT_REQ_CREDIT_BUREAU_MON',
                'AMT_REQ_CREDIT_BUREAU_WEEK',
                'AMT_REQ_CREDIT_BUREAU_DAY',
                'AMT_REQ_CREDIT_BUREAU_HOUR']
App_data = App_data.drop(curr_to_drop, axis=1)
print((App_data.columns))
```

```
Index(['SK_ID_CURR', 'TARGET', 'NAME_CONTRACT_TYPE', 'CODE_GENDER',
       'FLAG_OWN_CAR', 'FLAG_OWN_REALTY', 'CNT_CHILDREN', 'AMT_INCOME_TOTAL',
       'AMT_CREDIT', 'AMT_ANNUITY', 'AMT_GOODS_PRICE', 'NAME_TYPE_SUITE',
       'NAME_INCOME_TYPE', 'NAME_EDUCATION_TYPE', 'NAME_FAMILY_STATUS',
       'NAME_HOUSING_TYPE', 'DAYS_BIRTH', 'DAYS_EMPLOYED', 'DAYS_REGISTRATION',
       'DAYS_ID_PUBLISH', 'OCCUPATION_TYPE', 'CNT_FAM_MEMBERS',
       'REGION_RATING_CLIENT', 'REGION_RATING_CLIENT_W_CITY',
       'ORGANIZATION_TYPE', 'EXT_SOURCE_3'],
      dtype='object')
```

REMOVING UNNECESSARY COLUMNS FROM APP_DATA

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

```
F     202452
M     105059
```

AS 'F' ARE IN LARGE NUMBER CHANGED 'XNA' TO 'F' TO REMOVE INCORRECT DATA

```
App_data['OCCUPATION_TYPE'].fillna(value = 'Unknown', inplace = True)
App_data['OCCUPATION_TYPE'].value_counts()
```

```
Unknown                  96391
Laborers                 55186
Sales staff              32102
Core staff               27570
Managers                 21371
Drivers                  18603
High skill tech staff    11380
Accountants               9813
Medicine staff            8537
```

FILLED ALL THE NULL/EMPTY VALUES TO 'UNKNOWN' TO AVOID INCORRECT DATA IN OCCUPATION_TYPE COLUMN

```
App_data['NAME_TYPE_SUITE'].fillna(value = 'unknown', inplace = True)
App_data.NAME_TYPE_SUITE.value_counts()
```

```
Unaccompanied      248526
Family              40149
Spouse, partner     11370
Children             3267
Other_B              1770
unknown              1292
Other_A               866
Group of people       271
```

FILLED ALL THE NULL/EMPTY VALUES TO 'UNKNOWN' TO AVOID INCORRECT DATA IN NAME_TYPE_SUITE COLUMN

CHANGED ALL THE NEGATIVE TO POSITIVE VALUES AND CONVERTED DAYS TO YEAR

```
App_data['DAYS_BIRTH'] = round(App_data['DAYS_BIRTH'].abs()/365,2)
App_data['DAYS_EMPLOYED'] = round(App_data['DAYS_EMPLOYED'].abs()/365,2)
App_data['DAYS_REGISTRATION'] = round(App_data['DAYS_REGISTRATION'].abs()/365,2)
App_data['DAYS_ID_PUBLISH'] = round(App_data['DAYS_ID_PUBLISH'].abs()/365,2)
App_data.head()
```

DIT	AMT_ANNUITY	...	DAYS_BIRTH	DAYS_EMPLOYED	DAYS_REGISTRATION	DAYS_ID_PUBLISH
7.5	24700.5	...	25.92	1.75	9.99	5.81
2.5	35698.5	...	45.93	3.25	3.25	0.80
0.0	6750.0	...	52.18	0.62	11.67	6.93
2.5	29686.5	...	52.07	8.33	26.94	6.68
0.0	21865.5	...	54.61	8.32	11.81	9.47

Previous Application Data Analysis

```
df_prev.duplicated().sum()
```

```
0
```

```
df_prev.shape
```

```
(1670214, 37)
```

```
df_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
```

```
df_prev.head()
```

	SK_ID_PREV	SK_ID_CURR	NAME_CONTRACT_TYPE	AMT_ANNUITY	AMT_APPLICATION	AMT_CREDIT	AMT_DOWN_PAYMENT	AMT_GOODS_PRICE	WEEKDAY_APPR_PROCESS_START	HOUR_APPR_P
0	2030495	271877	Consumer loans	1730.430	17145.0	17145.0	0.0	17145.0	SATURDAY	
1	2802425	108129	Cash loans	25188.615	607500.0	679671.0	NaN	607500.0	THURSDAY	
2	2523466	122040	Cash loans	15060.735	112500.0	136444.5	NaN	112500.0	TUESDAY	
3	2819243	176158	Cash loans	47041.335	450000.0	470790.0	NaN	450000.0	MONDAY	
4	1784265	202054	Cash loans	31924.395	337500.0	404055.0	NaN	337500.0	THURSDAY	

5 rows × 37 columns

```
df_prev.tail()
```

	SK_ID_PREV	SK_ID_CURR	NAME_CONTRACT_TYPE	AMT_ANNUITY	AMT_APPLICATION	AMT_CREDIT	AMT_DOWN_PAYMENT	AMT_GOODS_PRICE	WEEKDAY_APPR_PROCESS_START	H
1670209	2300464	352015	Consumer loans	14704.290	267295.5	311400.0	0.0	267295.5	WEDNESDAY	
1670210	2357031	334635	Consumer loans	6622.020	87750.0	64291.5	29250.0	87750.0	TUESDAY	
1670211	2659632	249544	Consumer loans	11520.855	105237.0	102523.5	10525.5	105237.0	MONDAY	
1670212	2785582	400317	Cash loans	18821.520	180000.0	191880.0	NaN	180000.0	WEDNESDAY	
1670213	2418762	261212	Cash loans	16431.300	360000.0	360000.0	NaN	360000.0	SUNDAY	

5 rows × 37 columns

Previous Application Data Analysis

```
df_prev.describe()
```

	SK_ID_PREV	SK_ID_CURR	AMT_ANNUITY	AMT_APPLICATION	AMT_CREDIT	AMT_DOWN_PAYMENT	AMT_GOODS_PRICE	HOUR_APPR_PROCESS_START	NFLAG_LAST_APPL_IN_DAY
count	1.670214e+06	1.670214e+06	1.297979e+06	1.670214e+06	1.670213e+06	7.743700e+05	1.284699e+06	1.670214e+06	1.670214e+06
mean	1.923089e+06	2.783572e+05	1.595512e+04	1.752339e+05	1.961140e+05	6.697402e+03	2.278473e+05	1.248418e+01	9.964675e-01
std	5.325980e+05	1.028148e+05	1.478214e+04	2.927798e+05	3.185746e+05	2.092150e+04	3.153966e+05	3.334028e+00	5.932963e-02
min	1.000001e+06	1.000010e+05	0.000000e+00	0.000000e+00	0.000000e+00	-9.000000e-01	0.000000e+00	0.000000e+00	0.000000e+00
25%	1.461857e+06	1.893290e+05	6.321780e+03	1.872000e+04	2.416050e+04	0.000000e+00	5.084100e+04	1.000000e+01	1.000000e+00
50%	1.923110e+06	2.787145e+05	1.125000e+04	7.104600e+04	8.054100e+04	1.638000e+03	1.123200e+05	1.200000e+01	1.000000e+00
75%	2.384280e+06	3.675140e+05	2.065842e+04	1.803600e+05	2.164185e+05	7.740000e+03	2.340000e+05	1.500000e+01	1.000000e+00
max	2.845382e+06	4.562550e+05	4.180581e+05	6.905160e+06	6.905160e+06	3.060045e+06	6.905160e+06	2.300000e+01	1.000000e+00

8 rows × 21 columns

```
df_prev.dtypes
```

```
SK_ID_PREV                    int64
SK_ID_CURR                    int64
NAME_CONTRACT_TYPE           object
AMT_ANNUITY                 float64
AMT_APPLICATION             float64
AMT_CREDIT                  float64
AMT_DOWN_PAYMENT            float64
AMT_GOODS_PRICE             float64
WEEKDAY_APPR_PROCESS_START   object
```

```
df_prev.select_dtypes(include=["int64","float64"]).columns
```

```
Index(['SK_ID_PREV', 'SK_ID_CURR', 'AMT_ANNUITY', 'AMT_APPLICATION',
       'AMT_CREDIT', 'AMT_DOWN_PAYMENT', 'AMT_GOODS_PRICE',
       'HOUR_APPR_PROCESS_START', 'NFLAG_LAST_APPL_IN_DAY',
       'RATE_DOWN_PAYMENT', 'RATE_INTEREST_PRIMARY',
```

```
df_prev.select_dtypes(include = "object").columns
```

```
Index(['NAME_CONTRACT_TYPE', 'WEEKDAY_APPR_PROCESS_START',
       'FLAG_LAST_APPL_PER_CONTRACT', 'NAME_CASH_LOAN_PURPOSE',
       'NAME_CONTRACT_STATUS', 'NAME_PAYMENT_TYPE', 'CODE_REJECT_REASON',
```

```python
Prev_data=round(df_prev.isnull().mean(axis=0).sort_values(ascending=False)*100,2)
print(len(Prev_data))
Prev_data = df_prev.loc[:,Prev_data<24]
print(len(Prev_data.columns),(Prev_data.columns))
```

```
37
26 Index(['SK_ID_PREV', 'SK_ID_CURR', 'NAME_CONTRACT_TYPE', 'AMT_ANNUITY',
       'AMT_APPLICATION', 'AMT_CREDIT', 'AMT_GOODS_PRICE',
       'WEEKDAY_APPR_PROCESS_START', 'HOUR_APPR_PROCESS_START',
       'FLAG_LAST_APPL_PER_CONTRACT', 'NFLAG_LAST_APPL_IN_DAY',
       'NAME_CASH_LOAN_PURPOSE', 'NAME_CONTRACT_STATUS', 'DAYS_DECISION',
       'NAME_PAYMENT_TYPE', 'CODE_REJECT_REASON', 'NAME_CLIENT_TYPE',
       'NAME_GOODS_CATEGORY', 'NAME_PORTFOLIO', 'NAME_PRODUCT_TYPE',
       'CHANNEL_TYPE', 'SELLERPLACE_AREA', 'NAME_SELLER_INDUSTRY',
       'CNT_PAYMENT', 'NAME_YIELD_GROUP', 'PRODUCT_COMBINATION'],
      dtype='object')
```

COUNTING THE PERCENTAGE OF NULL VALUES IN EACH COLUMN AND STORING DATA WHERE COLUMNS HAVE <24% OF NULL VALUES INTO PREV_DATA

```python
prev_to_drop = ['WEEKDAY_APPR_PROCESS_START', 'HOUR_APPR_PROCESS_START','FLAG_LAST_APPL_PER_CONTRACT',
                'NFLAG_LAST_APPL_IN_DAY', 'NAME_PORTFOLIO', 'NAME_PRODUCT_TYPE', 'CHANNEL_TYPE']
Prev_data = Prev_data.drop(prev_to_drop, axis=1)
Prev_data.columns
```

```
Index(['SK_ID_PREV', 'SK_ID_CURR', 'NAME_CONTRACT_TYPE', 'AMT_ANNUITY',
       'AMT_APPLICATION', 'AMT_CREDIT', 'AMT_GOODS_PRICE',
       'NAME_CASH_LOAN_PURPOSE', 'NAME_CONTRACT_STATUS', 'DAYS_DECISION',
       'NAME_PAYMENT_TYPE', 'CODE_REJECT_REASON', 'NAME_CLIENT_TYPE',
       'NAME_GOODS_CATEGORY', 'SELLERPLACE_AREA', 'NAME_SELLER_INDUSTRY',
       'CNT_PAYMENT', 'NAME_YIELD_GROUP', 'PRODUCT_COMBINATION'],
      dtype='object')
```

REMOVING UNNECESSARY COLUMNS FROM PREV_DATA

```
print(Prev_data.PRODUCT_COMBINATION.isnull().sum())
Prev_data['PRODUCT_COMBINATION'].fillna(value = 'Cash', inplace = True)
print(Prev_data.PRODUCT_COMBINATION.isnull().sum())
```

```
346
0
```

AS 'CASH' IS IN LARGE NUMBER FILLED NULL/EMPTY' TO 'CASH' TO REMOVE INCORRECT DATA

```
print(Prev_data['DAYS_DECISION'].head())
Prev_data['DAYS_DECISION'] = Prev_data['DAYS_DECISION'].abs()
Prev_data['DAYS_DECISION'].head()
```

```
0     -73
1    -164
2    -301
3    -512
4    -781
Name: DAYS_DECISION, dtype: int64
0      73
1     164
2     301
3     512
4     781
Name: DAYS_DECISION, dtype: int64
```

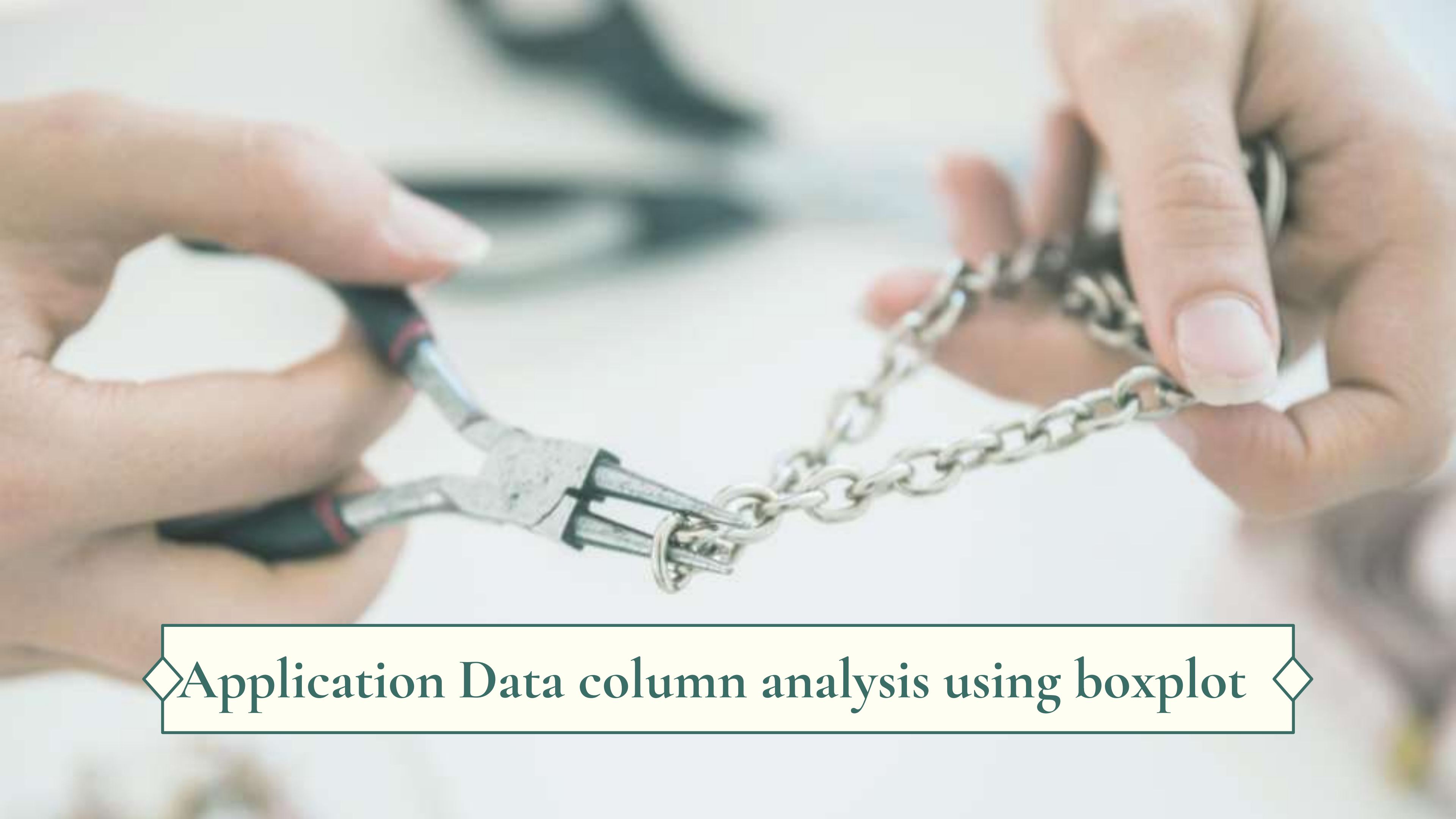
CHANGED ALL THE NEGATIVE TO POSITIVE VALUES

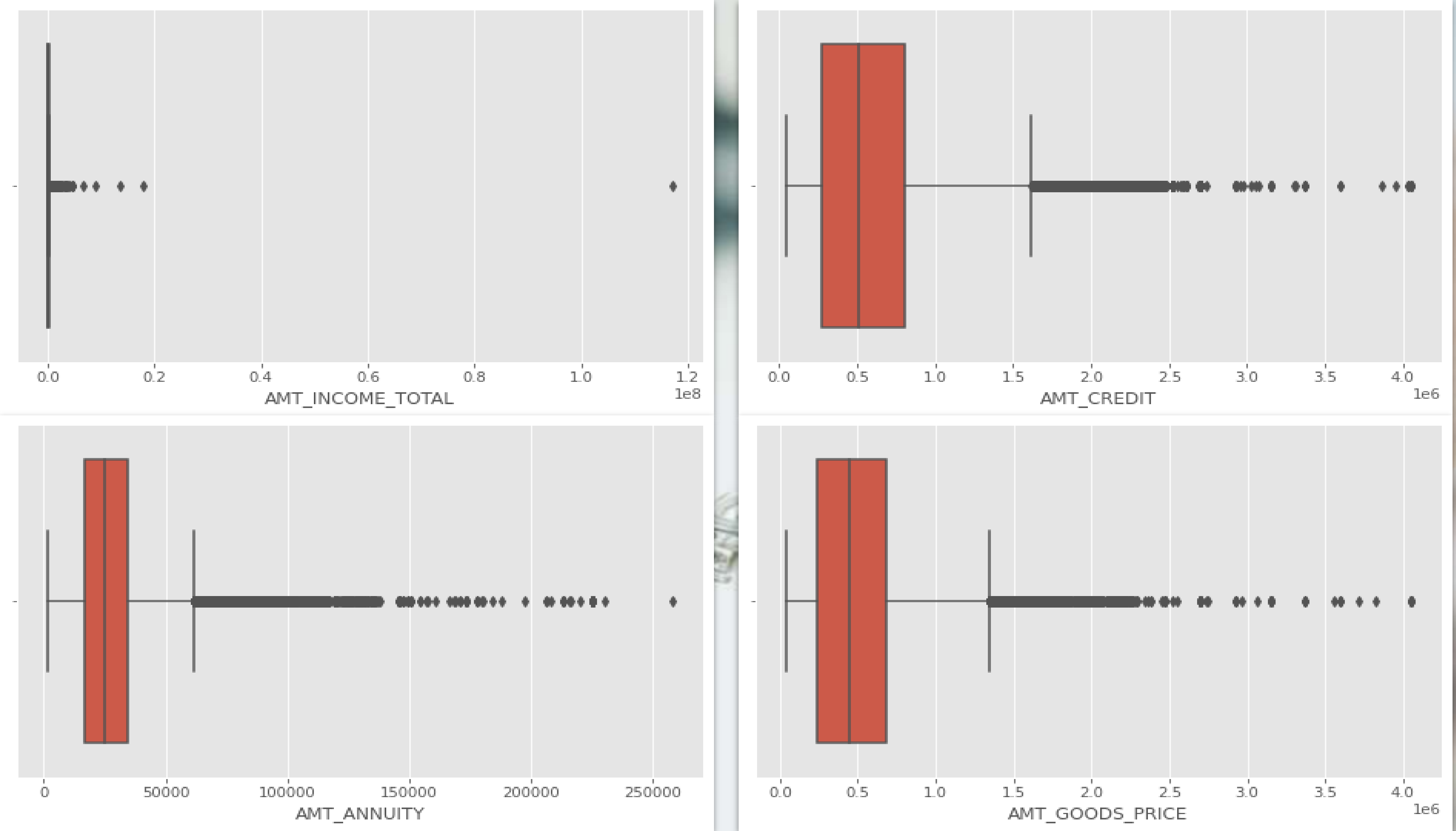
```
print(Prev_data['NAME_CLIENT_TYPE'].value_counts())
Prev_data.loc[Prev_data['NAME_CLIENT_TYPE']=='XNA']='Repeater'
Prev_data['NAME_CLIENT_TYPE'].value_counts(normalize=True)
```

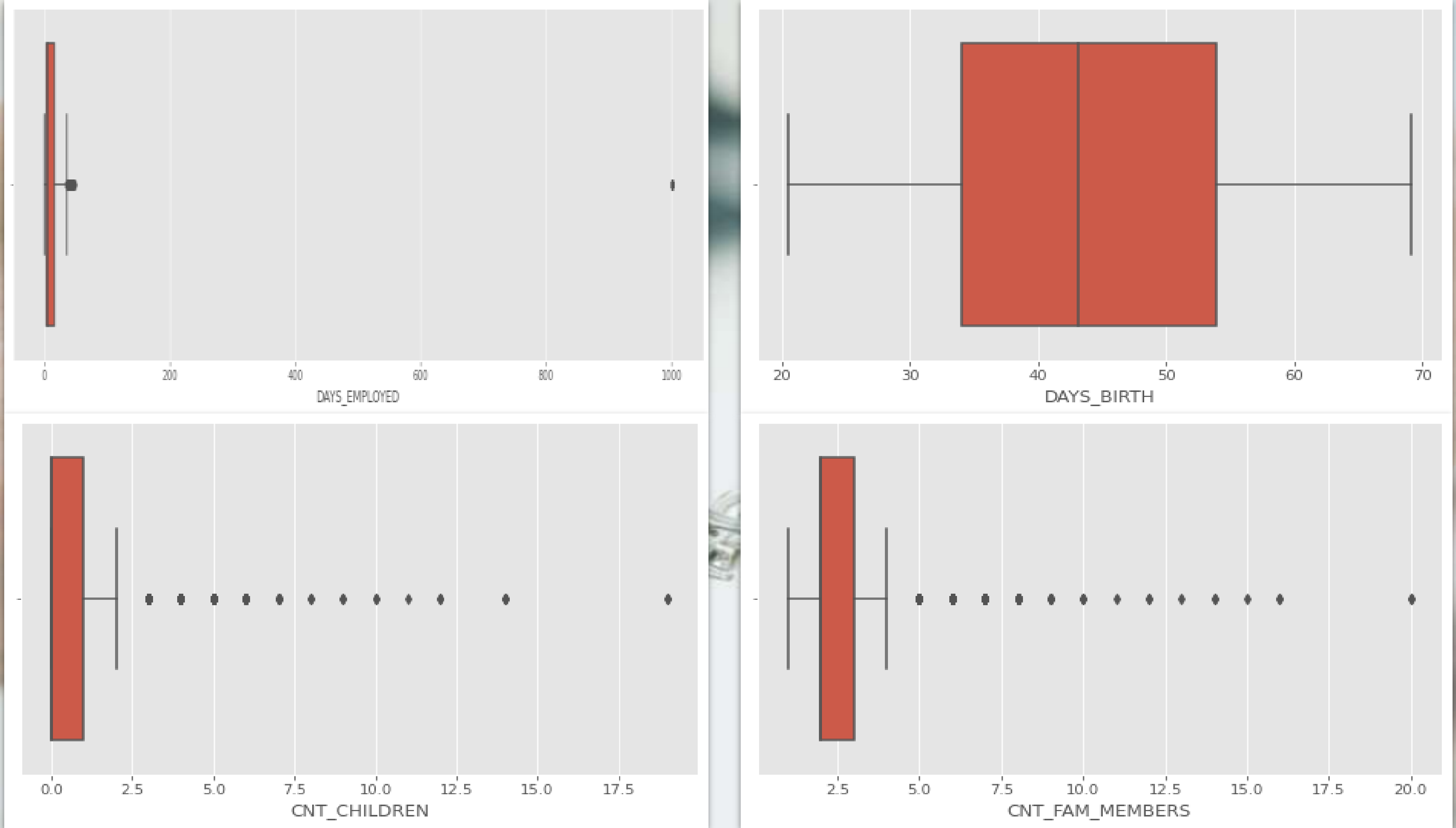
```
Repeater     1231261
New           301363
Refreshed     135649
XNA             1941
Name: NAME_CLIENT_TYPE, dtype: int64
Repeater     0.738350
New          0.180434
Refreshed    0.081217
Name: NAME_CLIENT_TYPE, dtype: float64
```

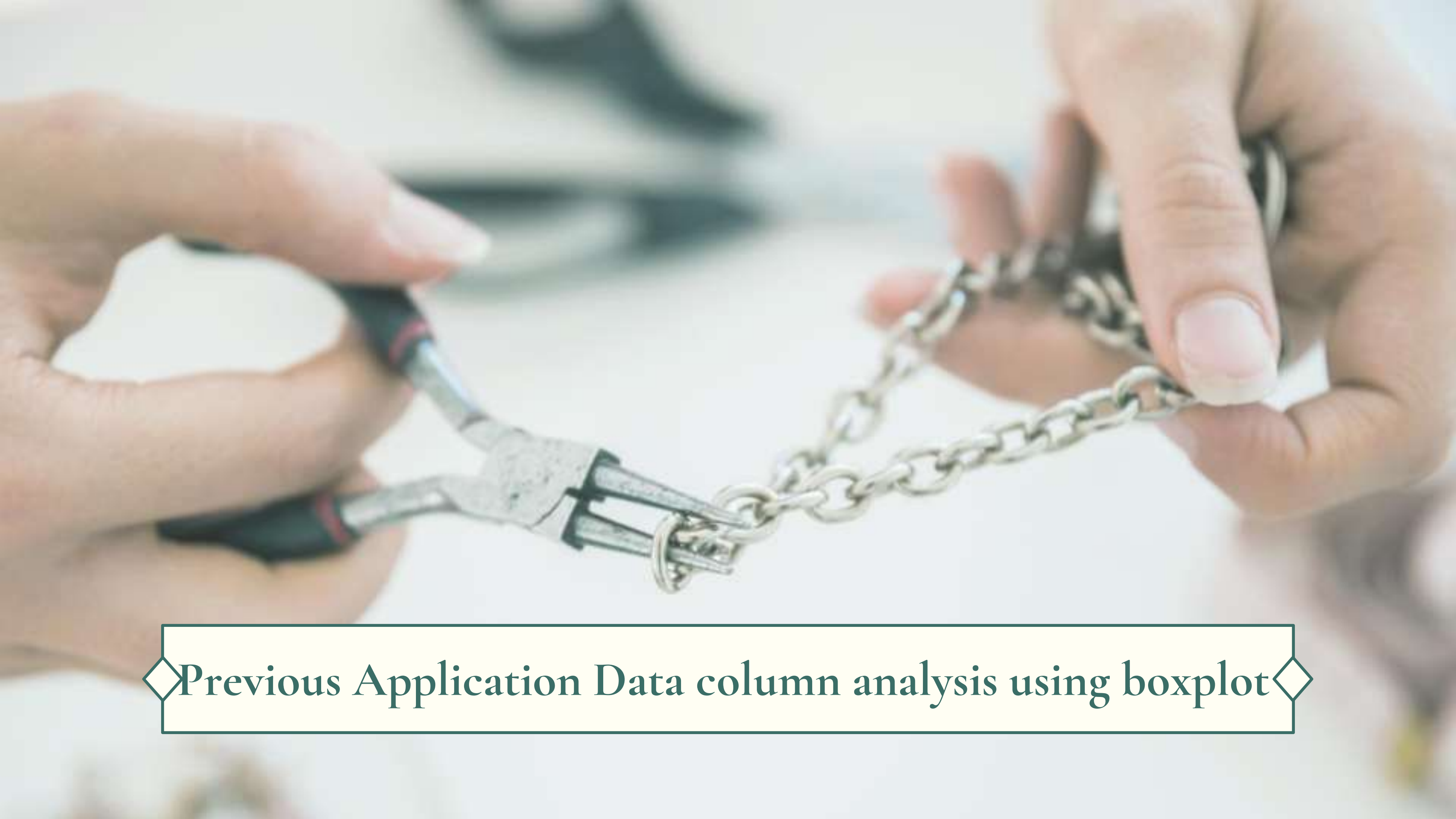
AS 'REPEATER' ARE IN LARGE NUMBER CHANGED 'XNA' TO 'REPEATER' TO REMOVE INCORRECT DATA

Application Data column analysis using boxplot

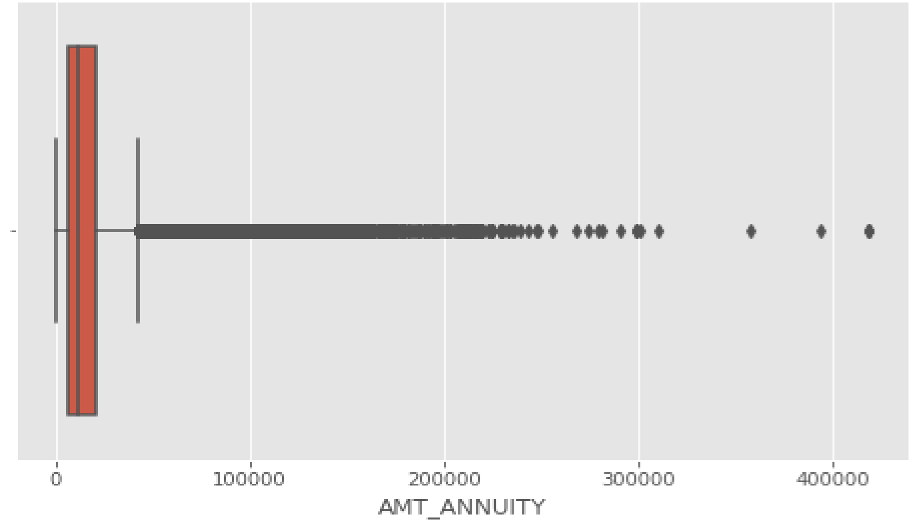
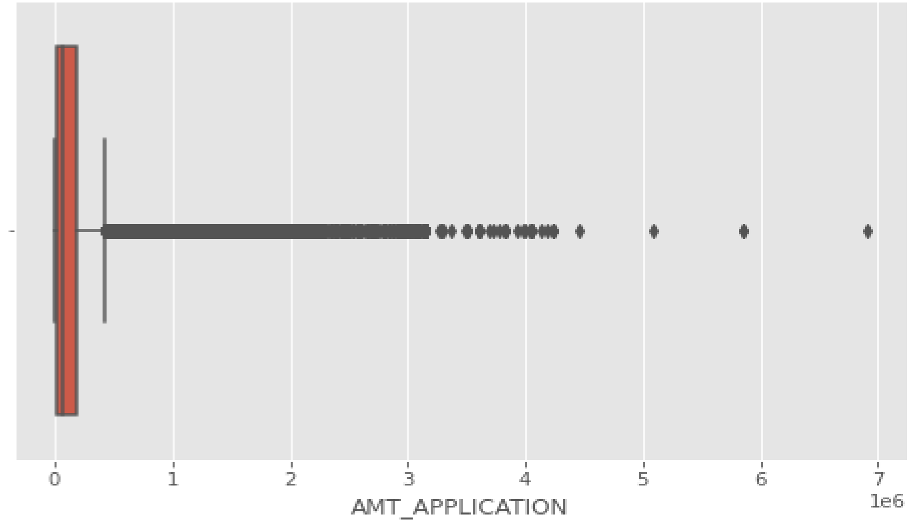
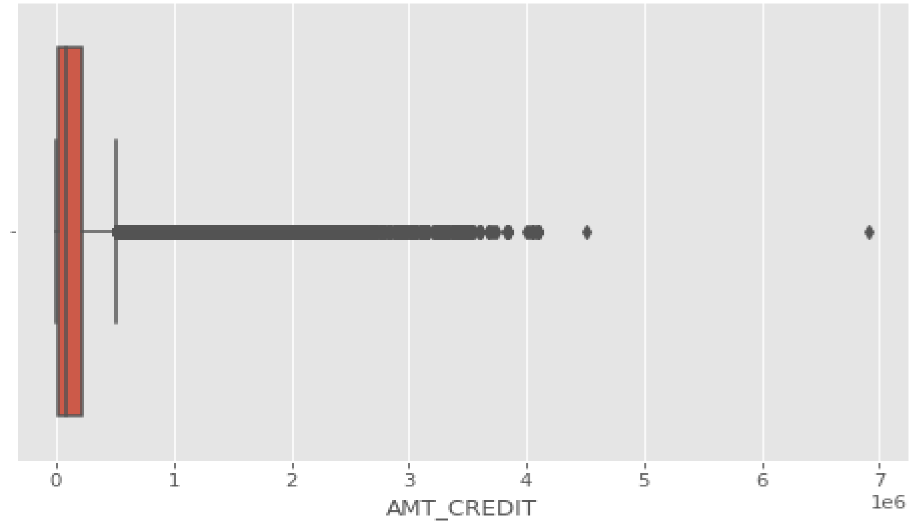
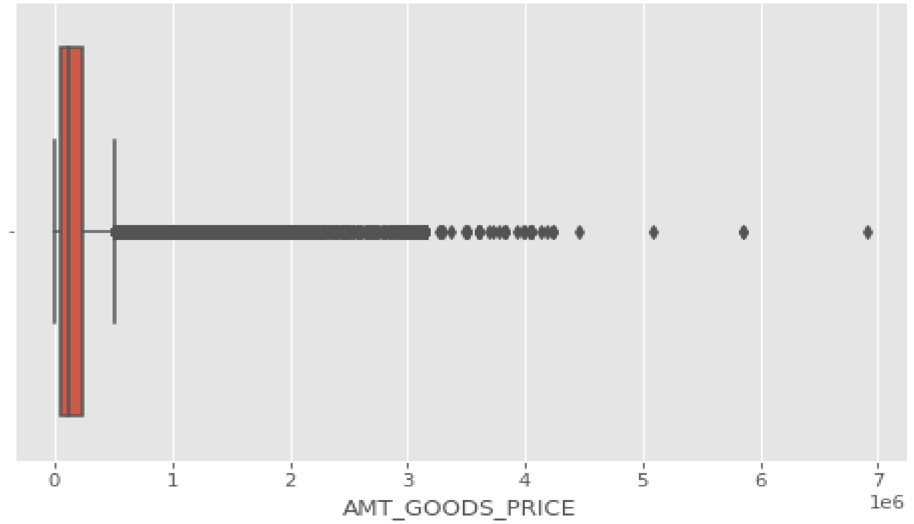


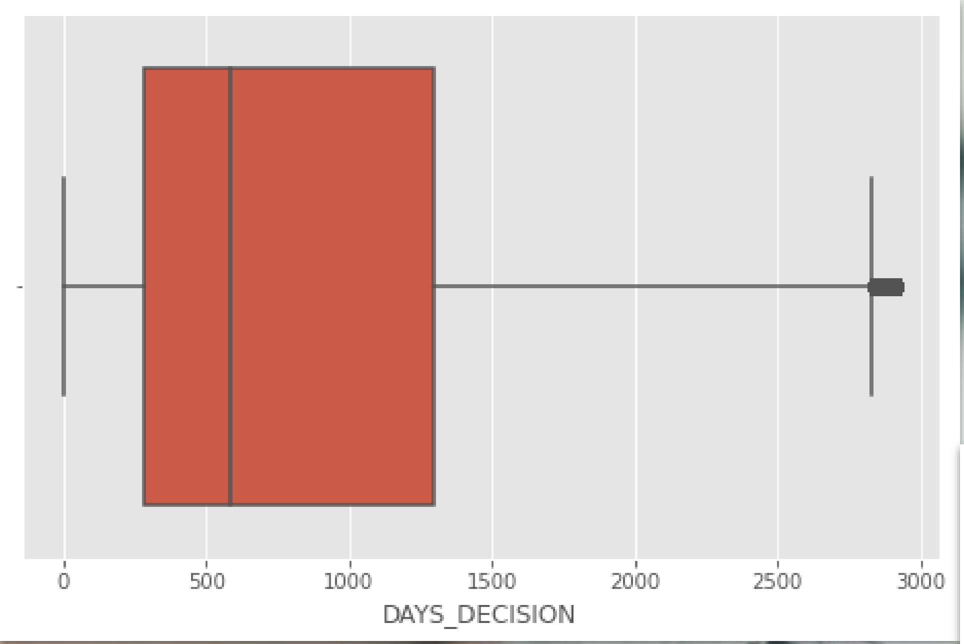

AMT_INCOME_TOTAL
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1e8
AMT_CREDIT
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
1e6
AMT_ANNUITY
0
50000
100000
150000
200000
250000
AMT_GOODS_PRICE
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
1e6

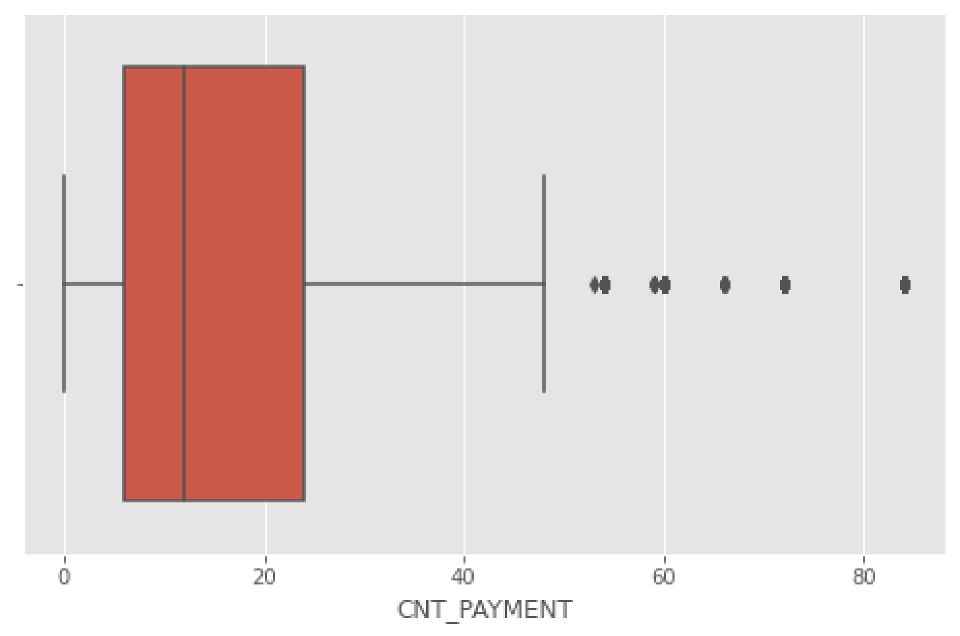

0
200
400
600
800
1000
DAYS_EMPLOYED
20
30
40
50
60
70
DAYS_BIRTH
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
CNT_CHILDREN
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
CNT_FAM_MEMBERS



Previous Application Data column analysis using boxplot


0
100000
200000
300000
400000
AMT_ANNUITY

0
1
2
3
4
5
6
7
1e6
AMT_APPLICATION

0
1
2
3
4
5
6
7
1e6
AMT_CREDIT

0
1
2
3
4
5
6
7
1e6
AMT_GOODS_PRICE


0
500
1000
1500
2000
2500
3000
DAYS_DECISION


0
20
40
60
80
CNT_PAYMENT

OUTLIERS

```
Outlier:
        AMT_INCOME_TOTAL -> Min: -22500.0  Max: 337500.0
        AMT_CREDIT -> Min: -537975.0  Max: 1616625.0
        AMT_ANNUITY -> Min: -10584.0  Max: 61704.0
        AMT_GOODS_PRICE -> Min: -423000.0  Max: 1341000.0
        YEARS_EMPLOYED -> Min: -17.060000000000002  Max: 35.26000000000
        YEARS_BIRTH -> Min: 4.144999999999925  Max: 83.78500000000001
        CNT_CHILDREN -> Min: -1.5  Max: 2.5
        CNT_FAM_MEMBERS -> Min: 0.5  Max: 4.5
```

```
Outlier:
        AMT_ANNUITY -> Min: -15183.18  Max: 42163.38
        AMT_APPLICATION -> Min: -223740.0  Max: 422820.0
        AMT_CREDIT -> Min: -264226.5  Max: 504805.5
        AMT_GOODS_PRICE -> Min: -223897.5  Max: 508738.5
        DAYS_DECISION -> Min: -2830.0  Max: 1250.0
        CNT_PAYMENT -> Min: -21.0  Max: 51.0
```

- Applicants with (AMT_ICOME_TOTAL) Income above 337500 are outliers
- Applicants with AMT_CREDIT above 1616625.0 (calculated using IQR) are outliers
- Applicants with AMT_ANNUITY above 61704 (calculated using IQR) are outliers
- Applicants with AMT_GOODS_PRICE above 1341000(calculated using IQR) are outliers
- Applicants with 3 or more children are outlier cases
- Applicants with 5 or more family members are clearly outliers

- Prev_Applicants with AMT_ANNUITY above 42163.38 (calculated using IQR) are outliers
- Prev_Applicants with AMT_APPLICATION above 422820.0 (calculated using IQR) are outliers
- Prev_Applicants with AMT_CREDIT above 504805.5 (calculated using IQR) are outliers
- Prev_Applicants with AMT_GOODS_PRICE above 508738.5 (cal using IQR) are outliers
- Prev_Applicants with above 51

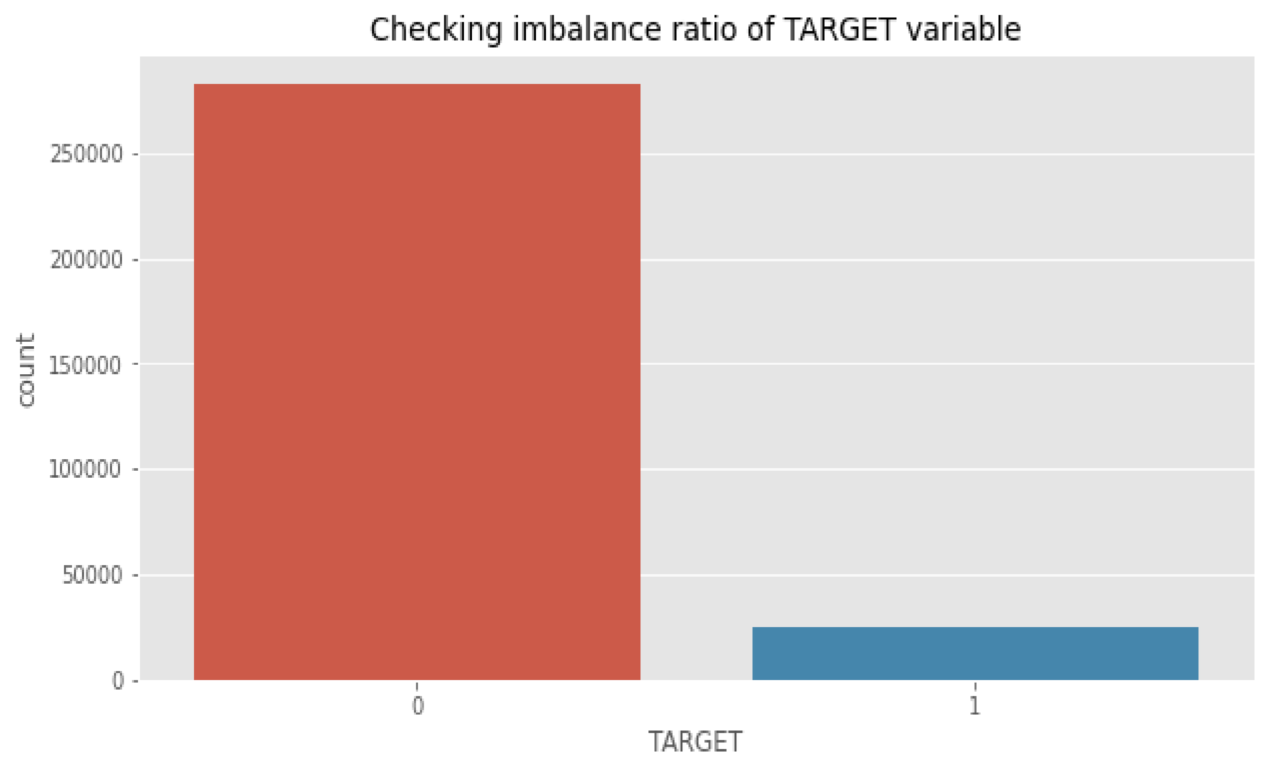
Imbalance

Value count and percentage in Target column

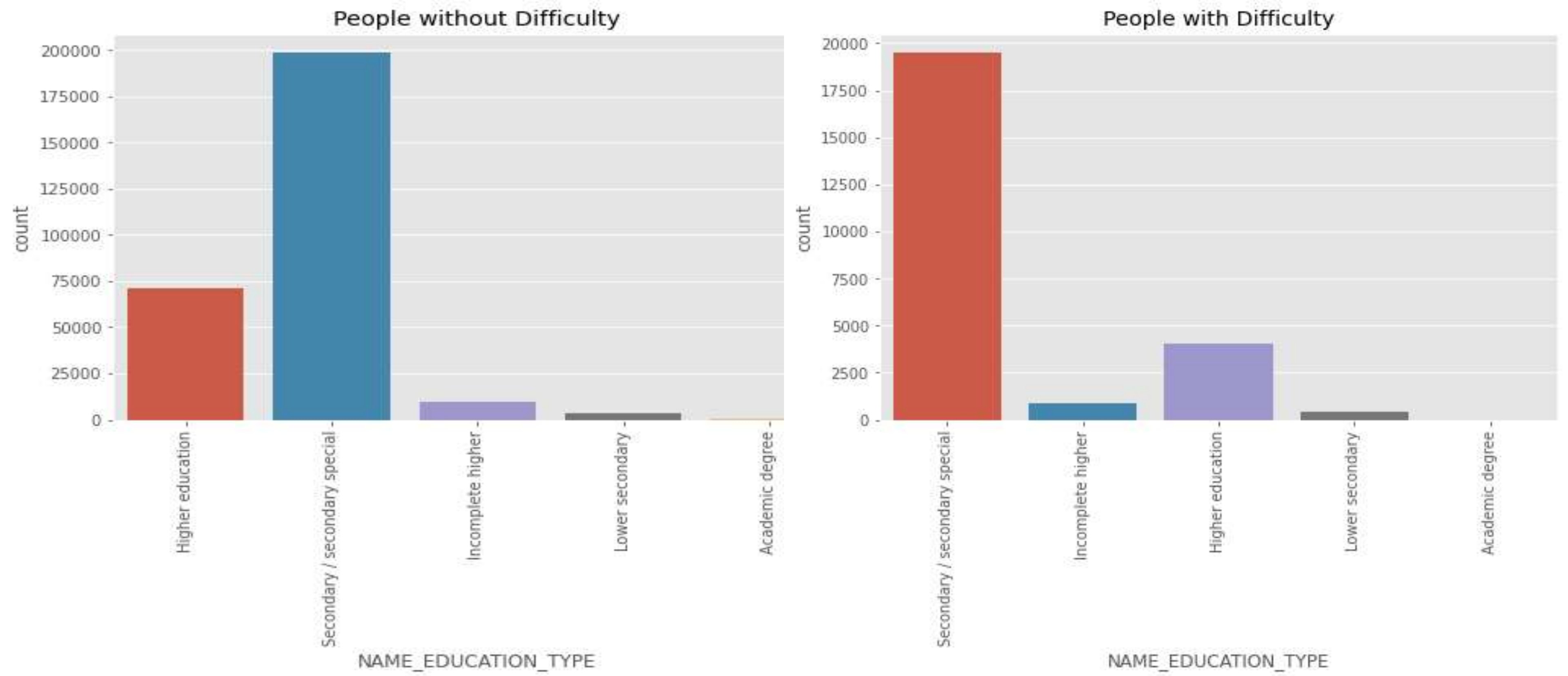
	Count	Percentage
0	282686	91.93
1	24825	8.07

```
Imbalance_ratio_T0: 91.93
Imbalance_ratio_T1: 8.07
```

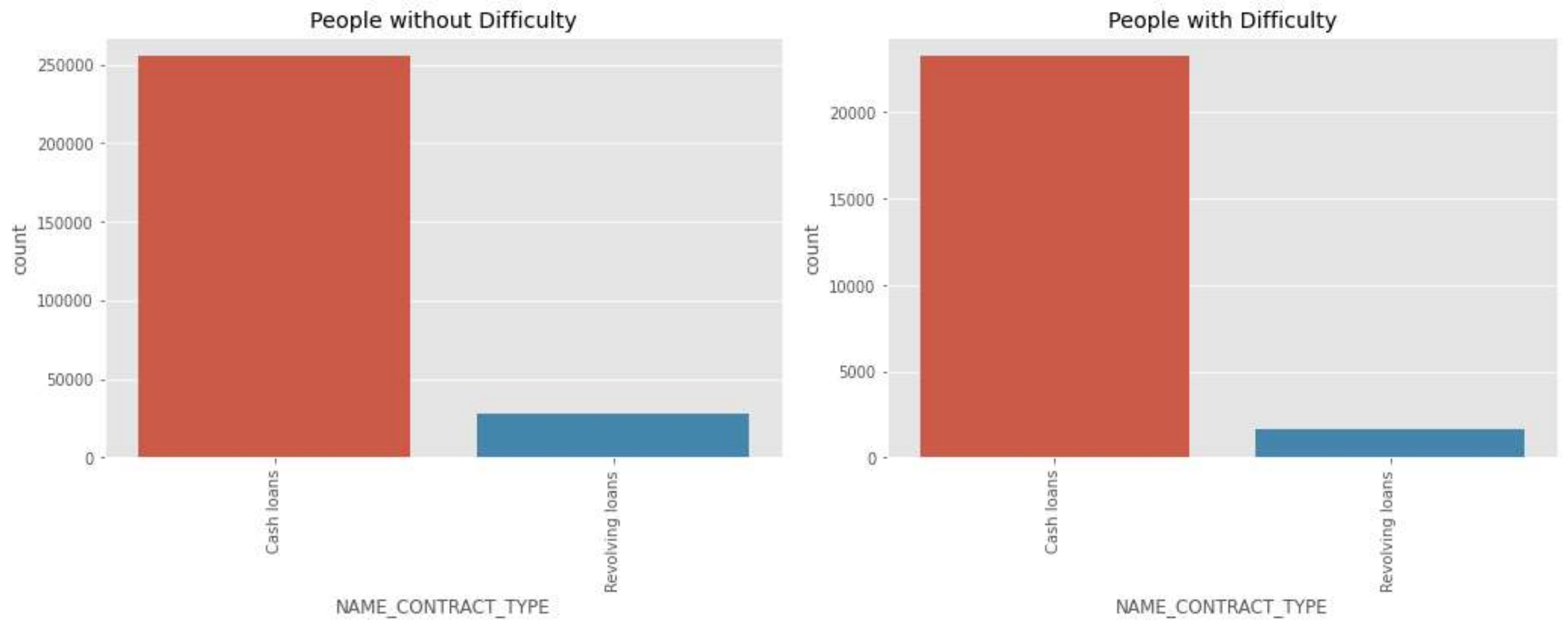
Imbalance ratio for values in Target column



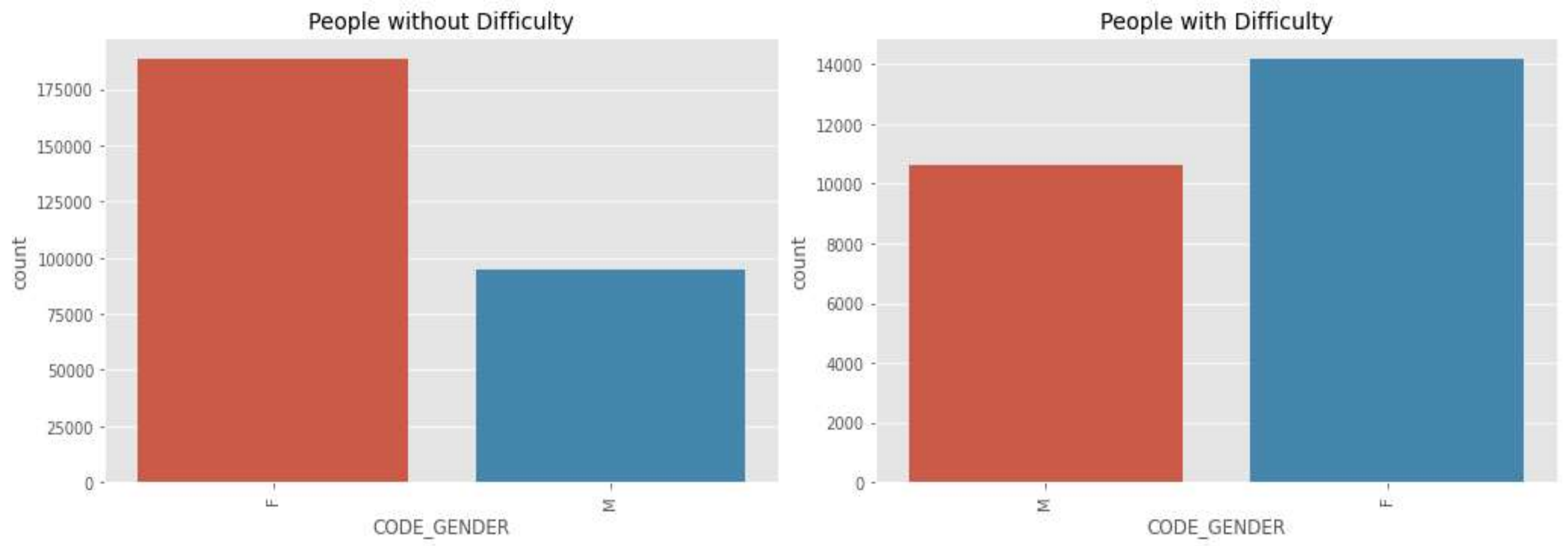
Application Data Analysis



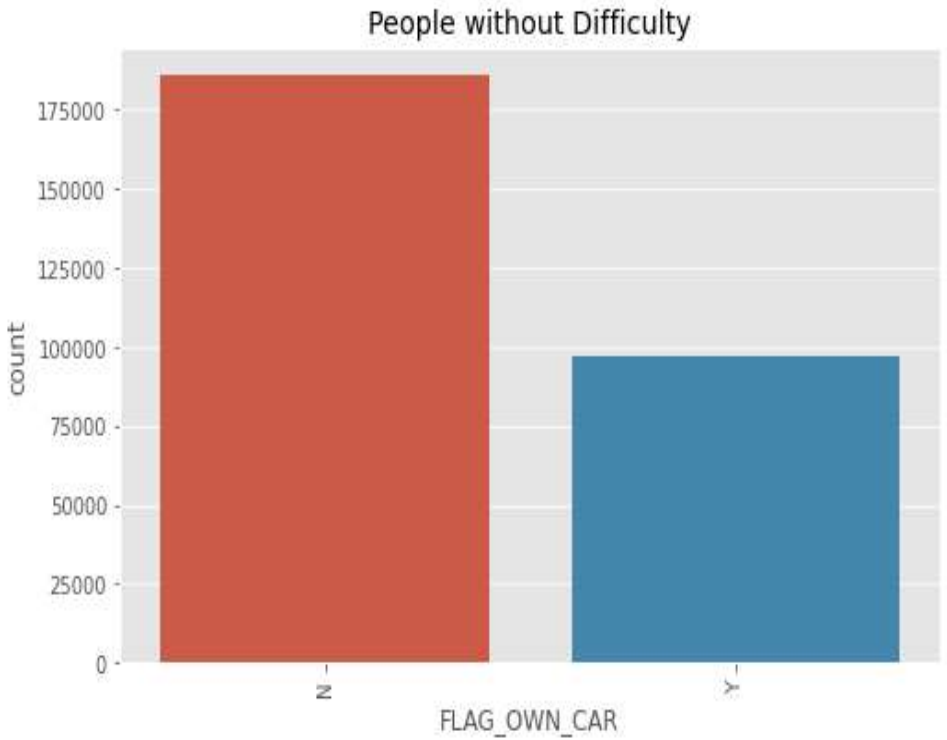
- Clients with 'Higher education' have better with payment difficulty than without payment difficulties
- Remaining categories don't provide any conclusive results

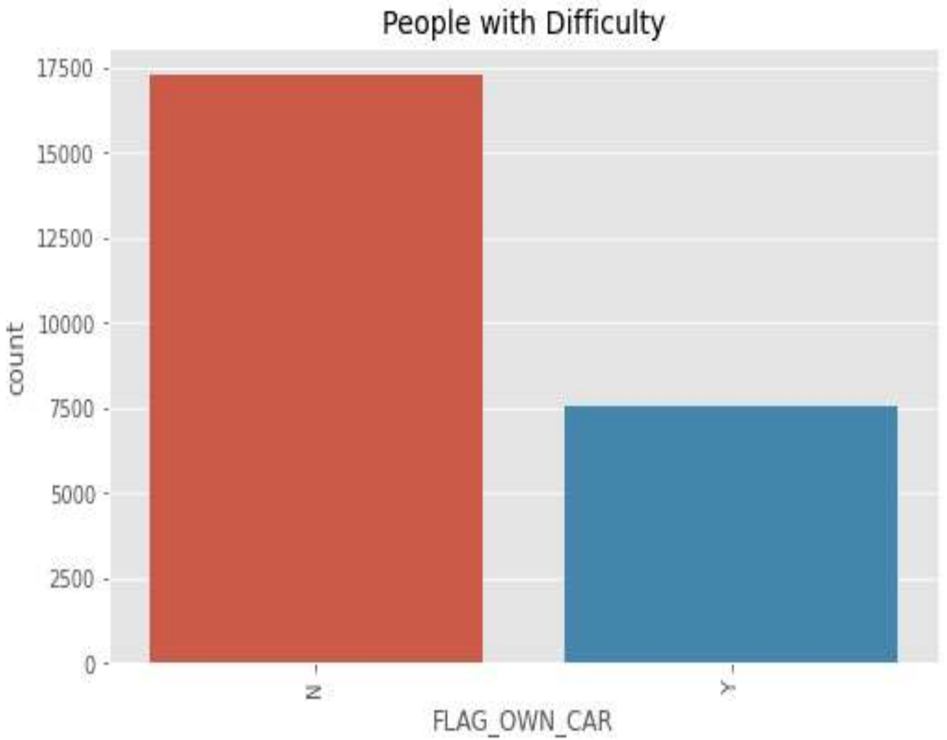


- **`NAME_CONTRACT_TYPE`** column does not provide any conclusive evidence in favor of clients with payment difficulties OR on-time payments because of no significant difference

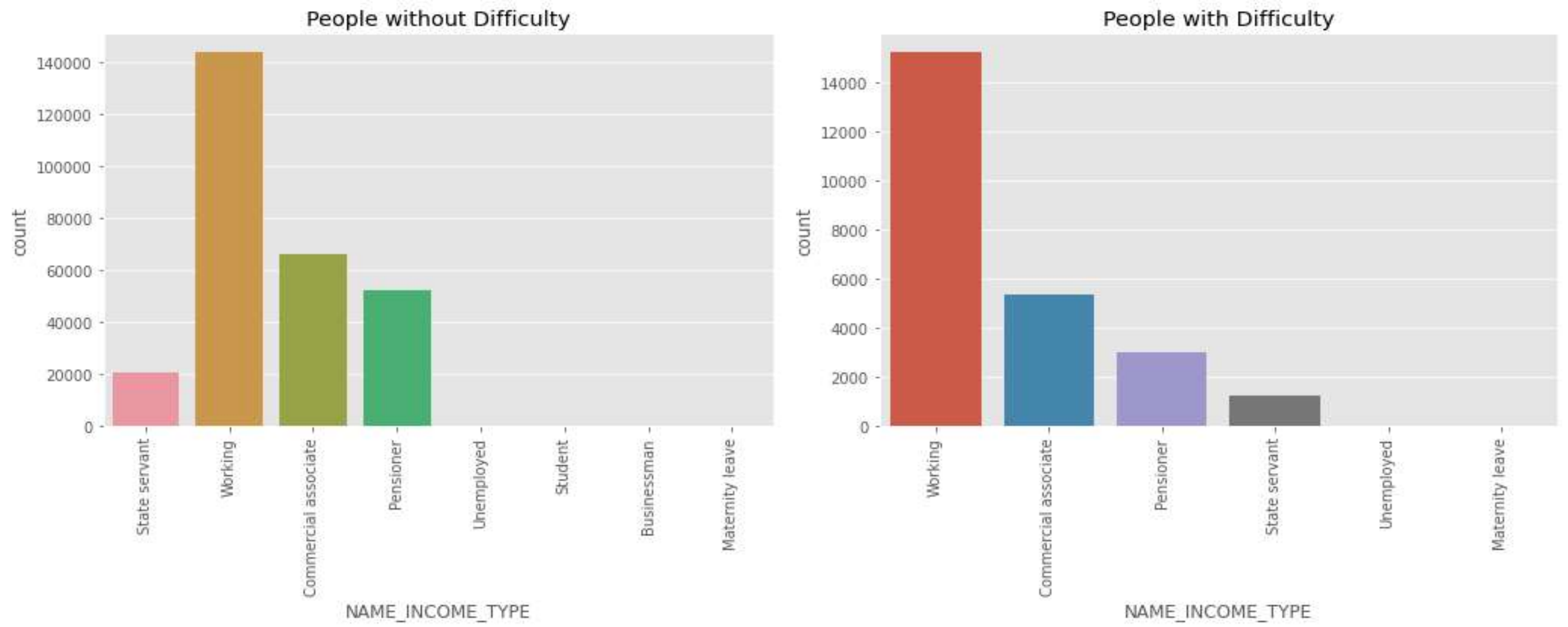


- **`CODE_GENDER`** column provides a weak inference that "Male" clients have more payment difficulties

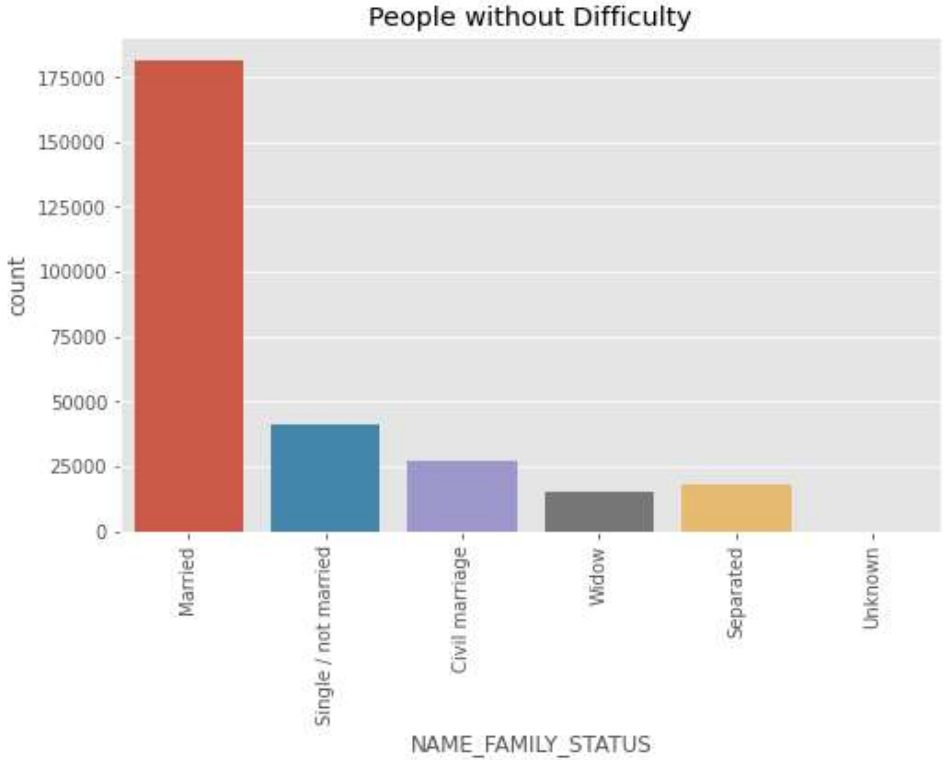


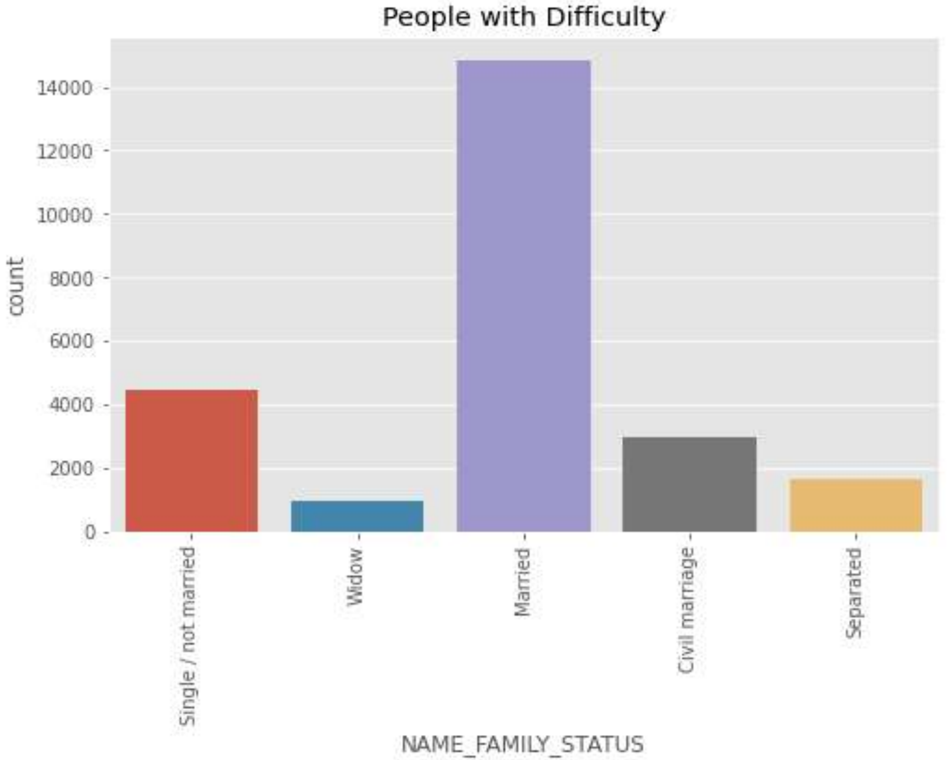


- **`FLAG_OWN_CAR`** column does not provide any conclusive evidence in favor of clients with payment difficulties OR without payment difficulties as there is no significant difference.

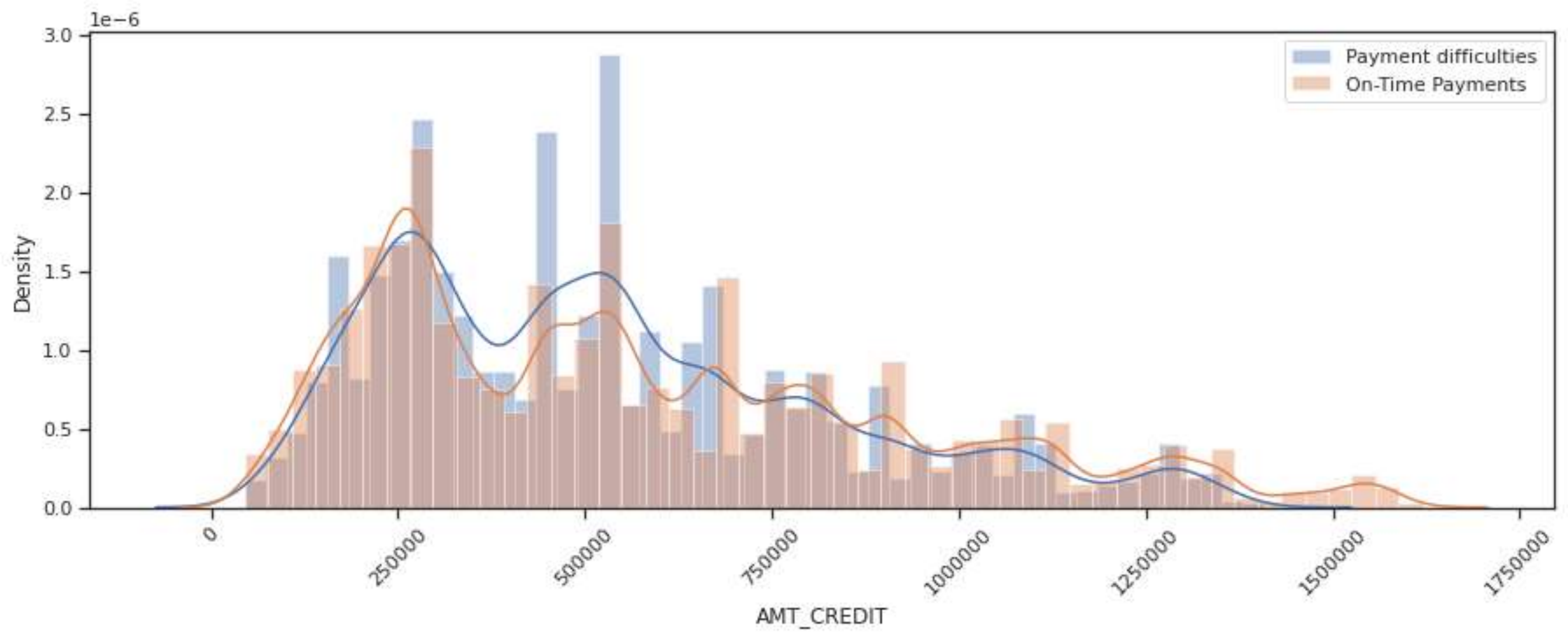


- Working people are most affected in both difficulties

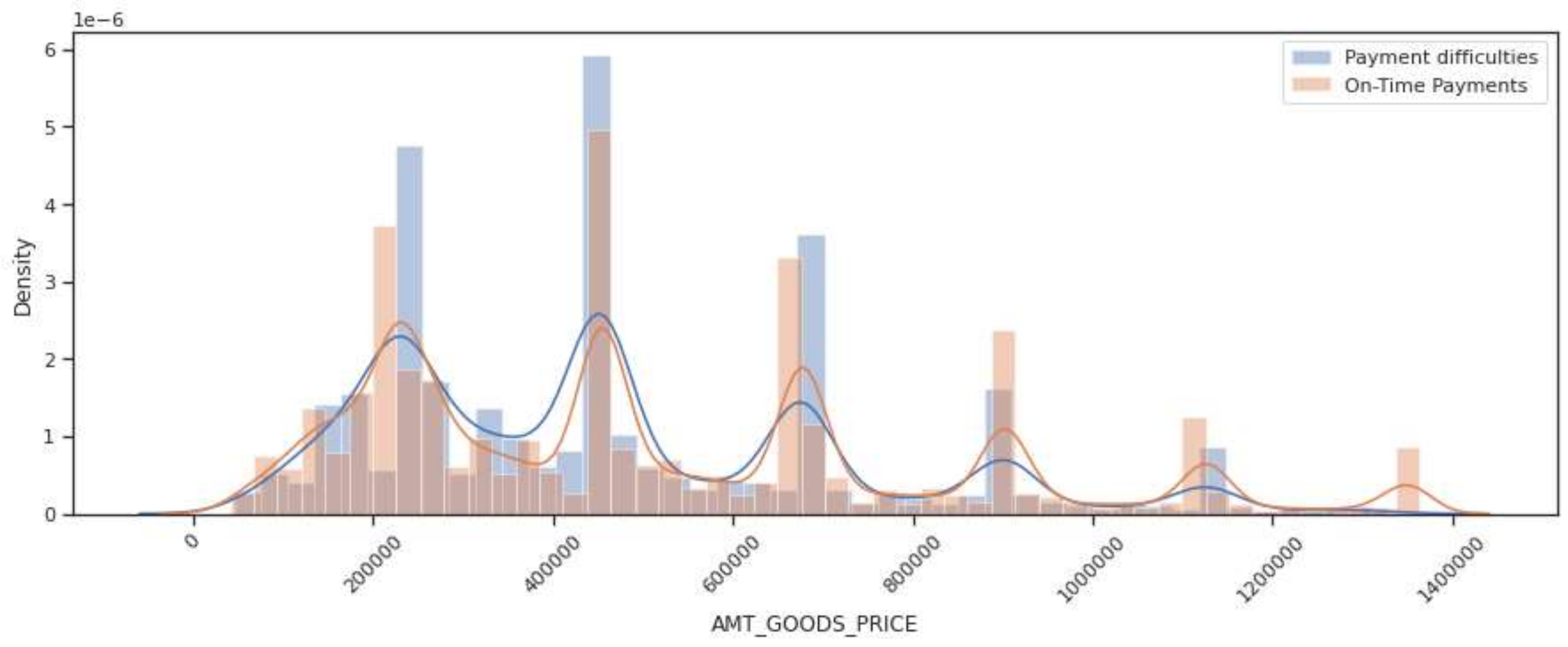




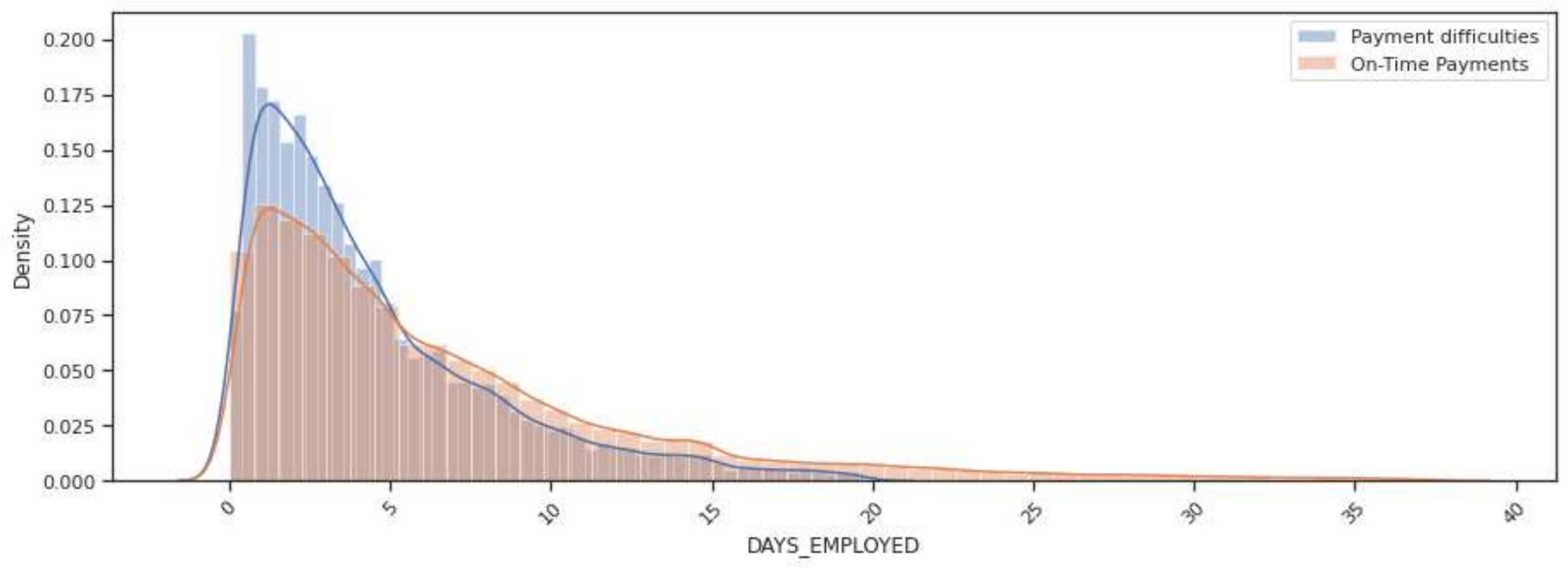
- Clients who are 'Married' OR 'Widow' do on-time payments better comparatively
- Clients who are 'Single/not married' have more difficulties with on-time payments comparatively



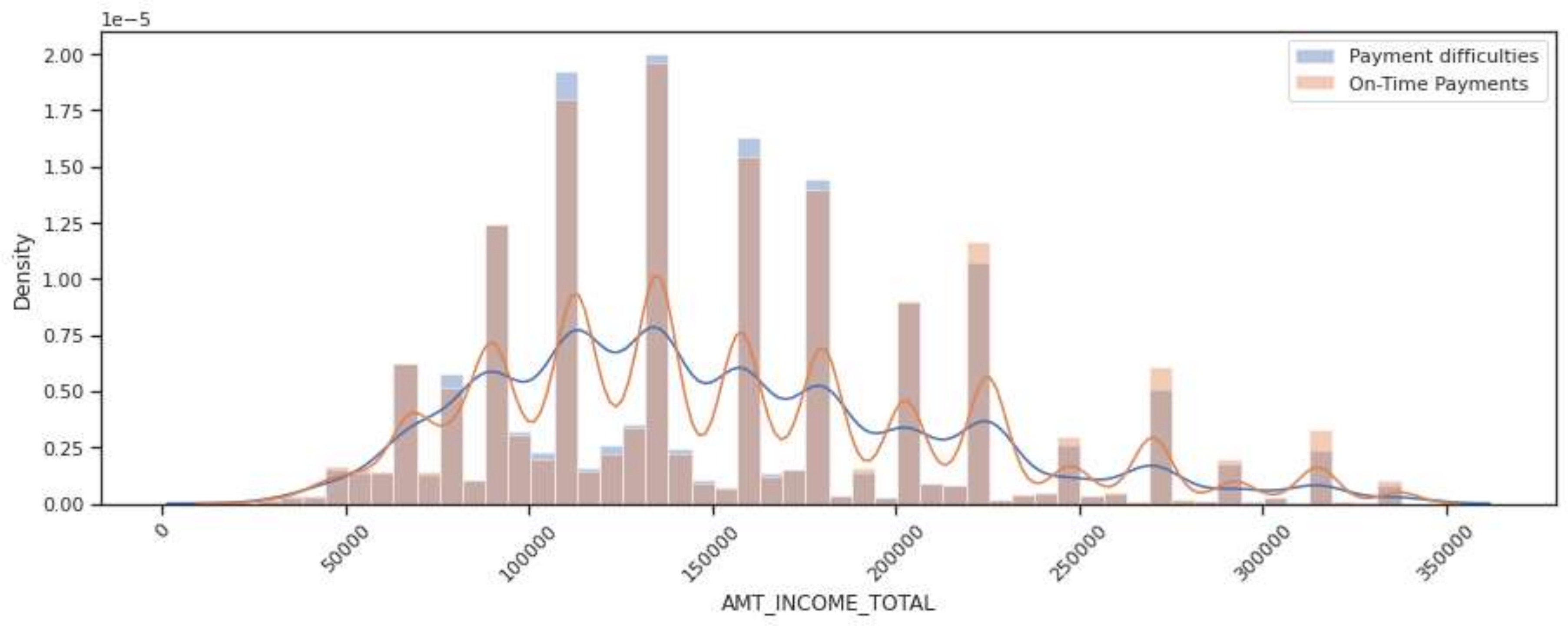
- For **AMT_CREDIT** between 250000 and approximately 650000, there are more clients with Payment difficulties
- For **AMT_CREDIT** > 750000 , there are more clients with On-Time Payments



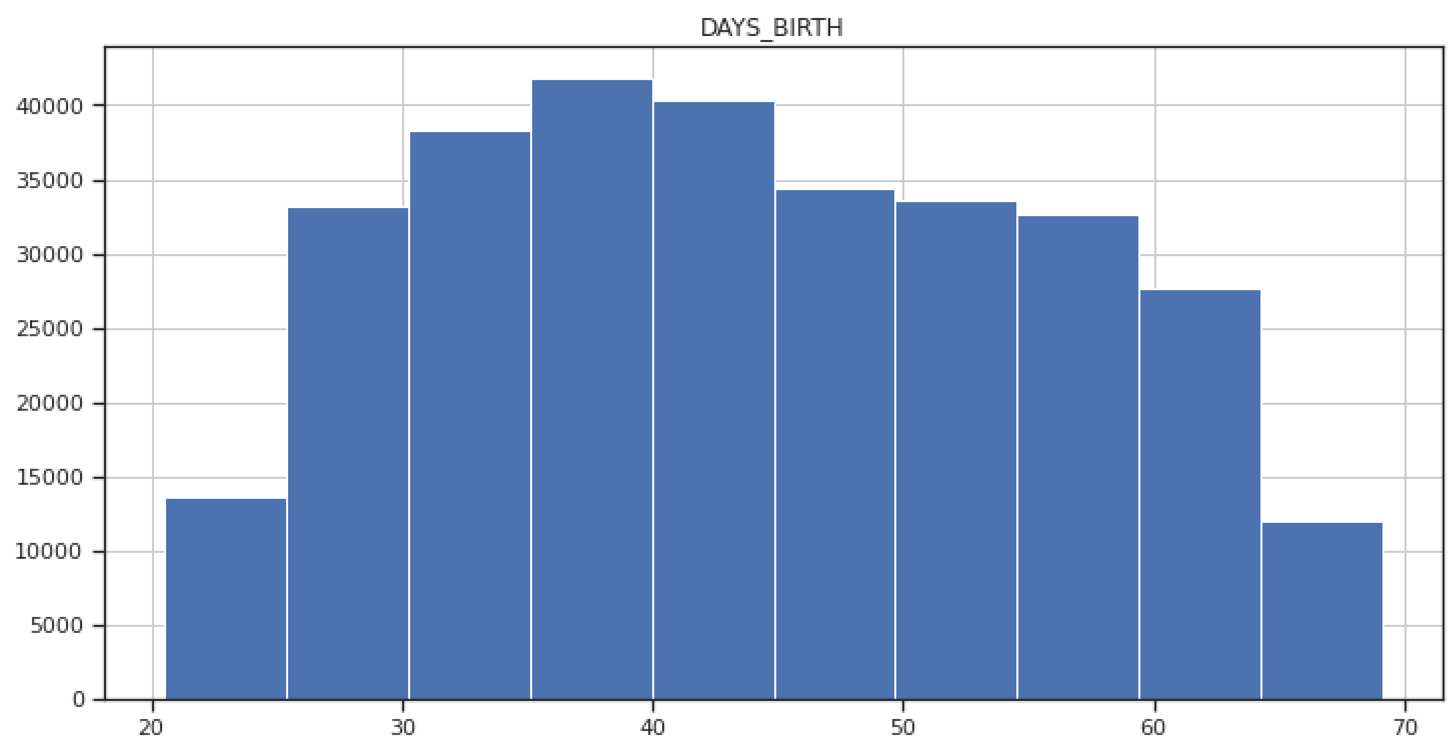
- For **`AMT_GOODS_PRICE`** between ~250000 and ~550000, there are more clients with Payment difficulties
- Otherwise there are spikes on and off but they don't show any conclusive observations



- For **DAYS_EMPLOYED** less than 2000, there are more clients with Payment difficulties
- Conversely, for **DAYS_EMPLOYED** > 2000 , there are more clients with On-Time Payments
- This means that those who are employed longer have better chances of repaying the loan

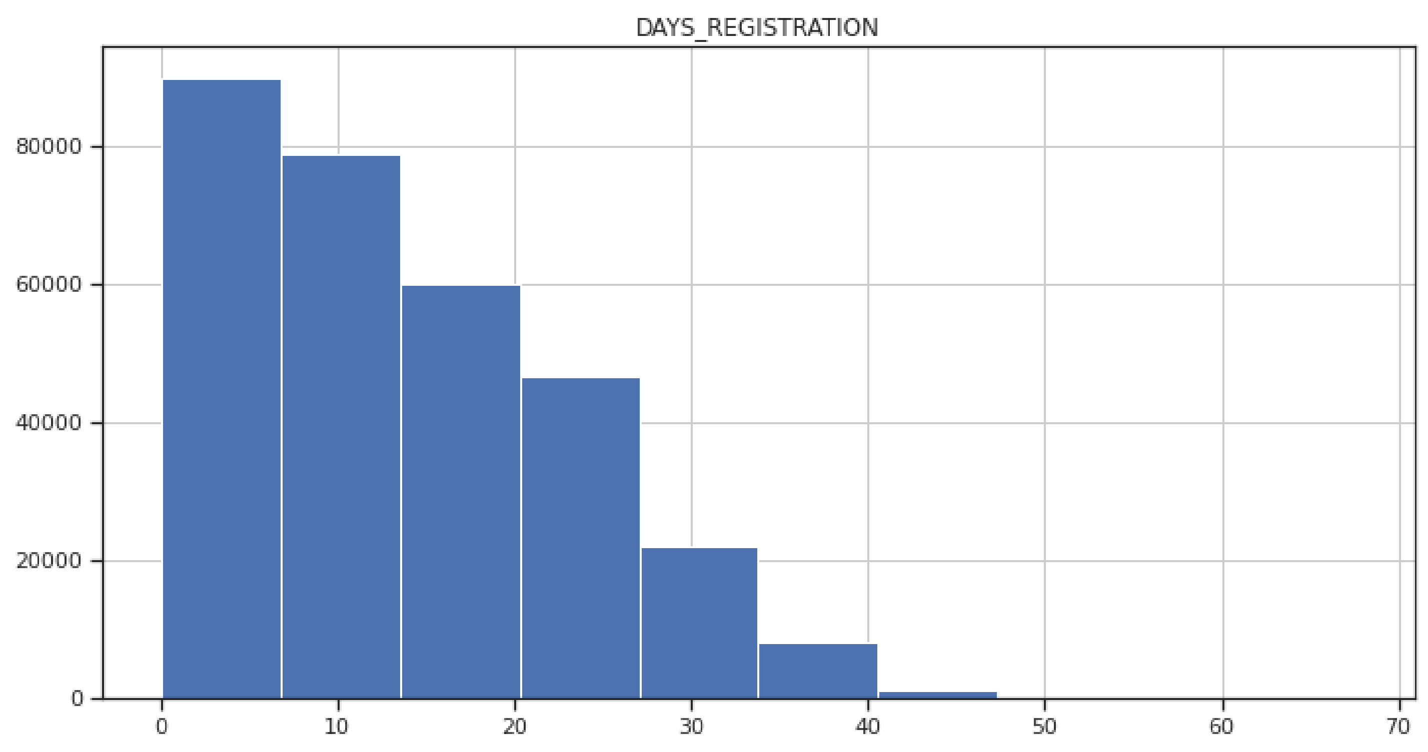


- Based on **AMT_INCOME_TOTAL**, for clients with Payment difficulties, the distribution resembles a normal distribution approximately
- But for clients with On-Time Payments, there are erratic spikes in the distribution which doesn't give any valid observations



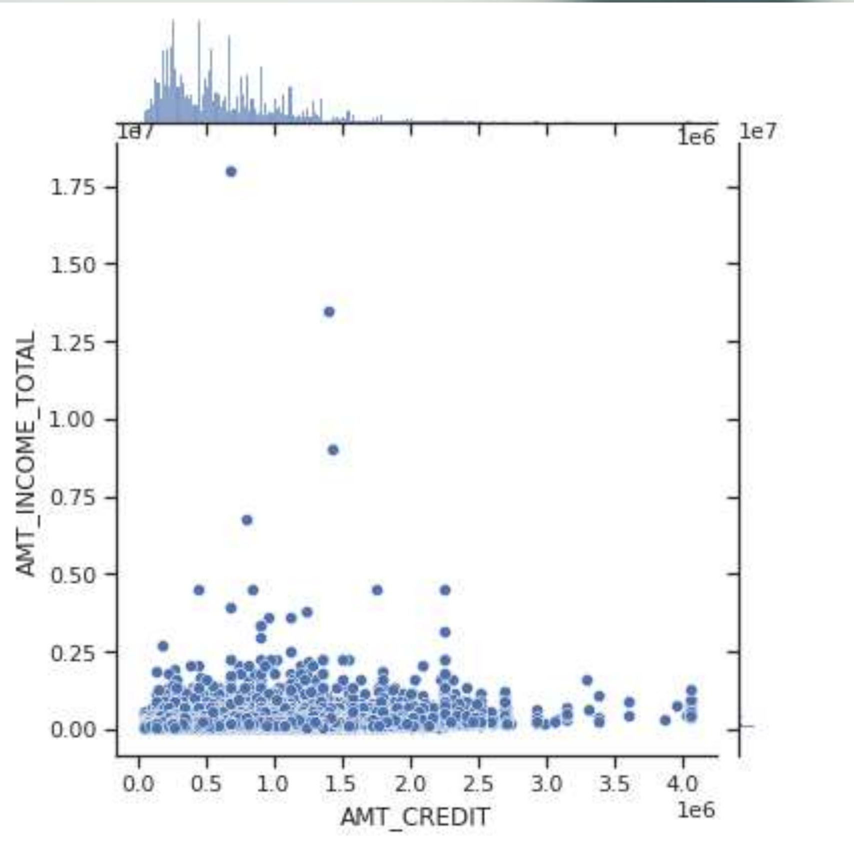
	DAYS_BIRTH
count	307511.000000
mean	43.936976
std	11.956135
min	20.520000
25%	34.010000
50%	43.150000
75%	53.920000
max	69.120000

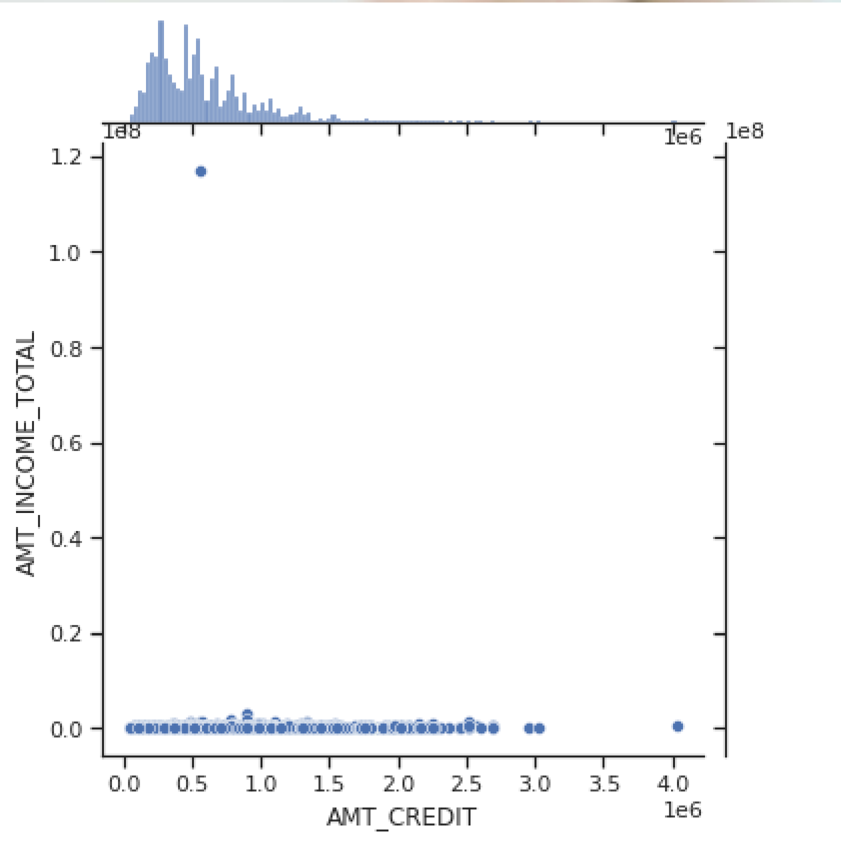
THE AVERAGE AGE of a CLIENT IS AROUND 44 YEARS.



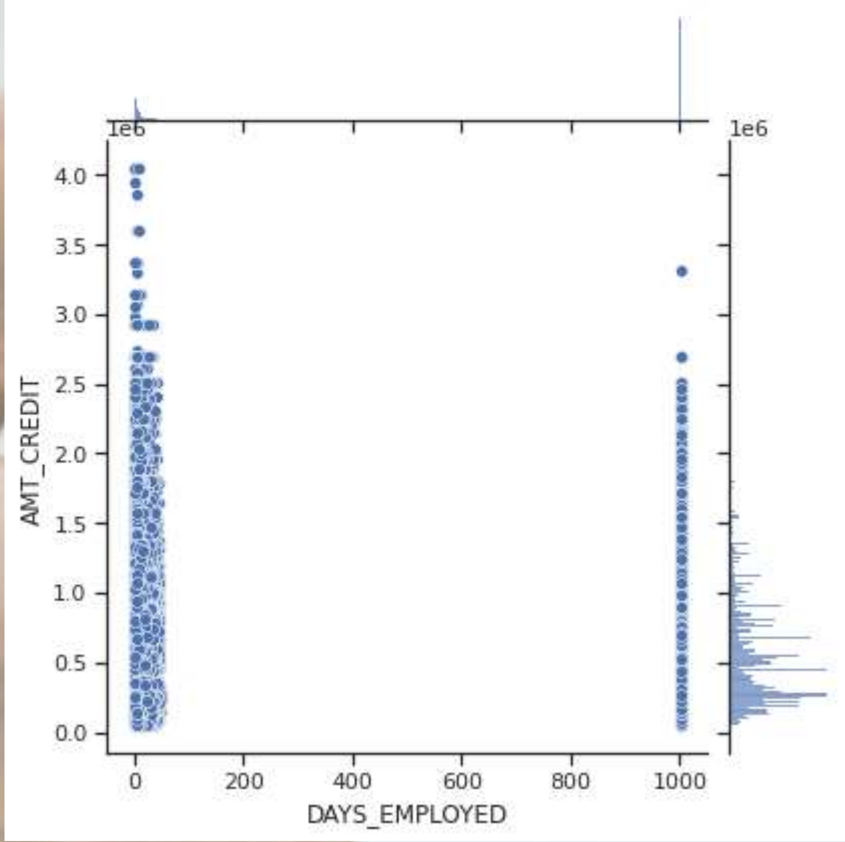
	DAYS_REGISTRATION
count	307511.000000
mean	13.660596
std	9.651742
min	0.000000
25%	5.510000
50%	12.340000
75%	20.490000
max	67.590000

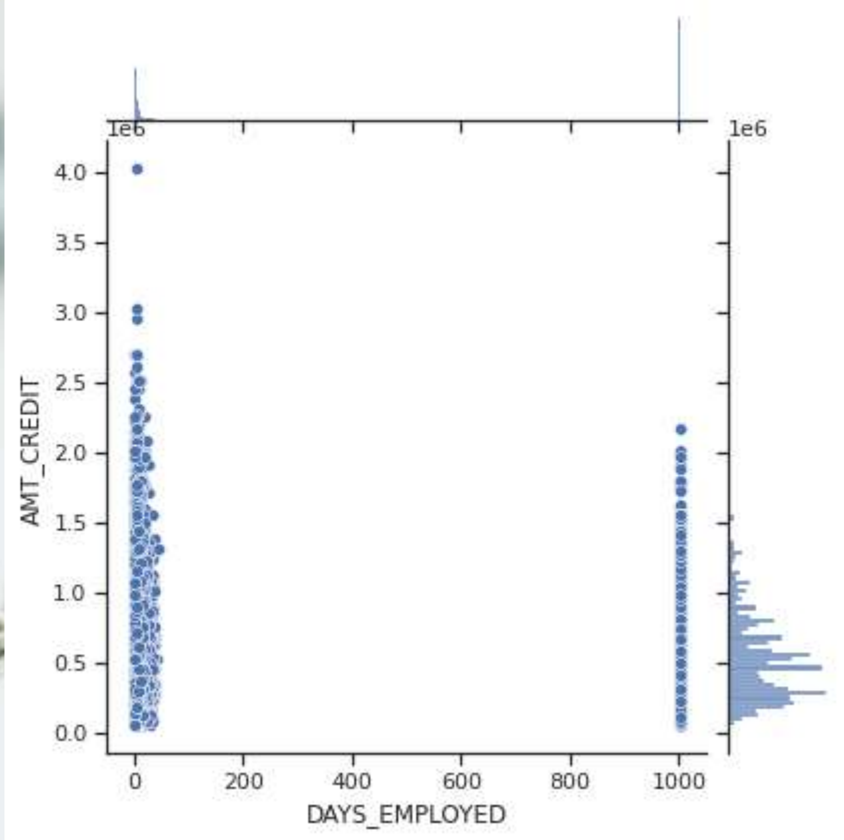
THE AVERAGE DAYS of a REGISTRATION IS AROUND 14 YEARS.



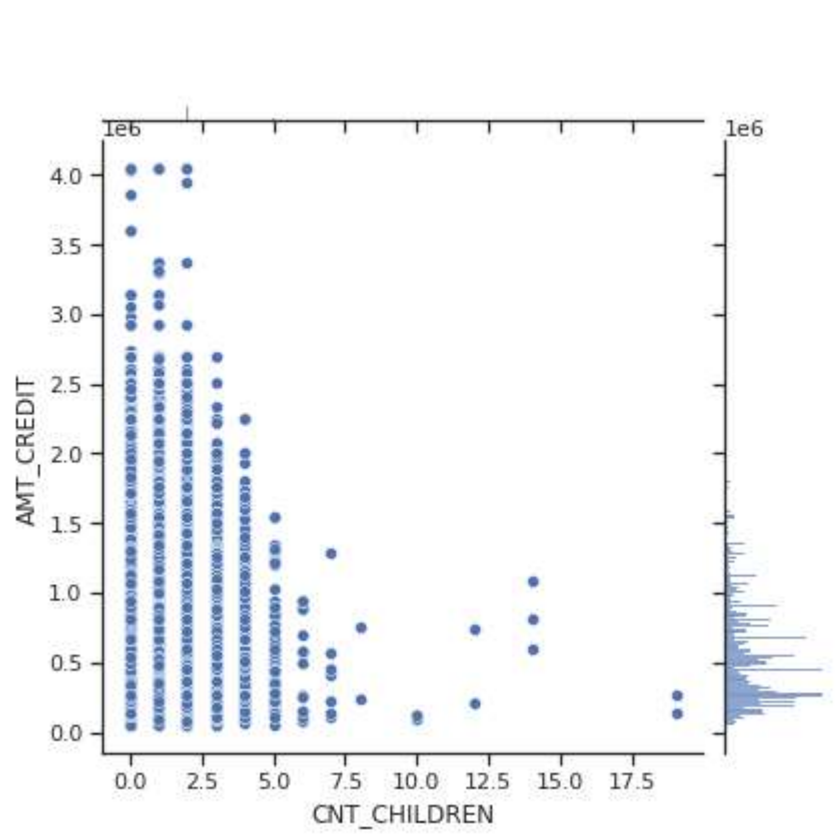


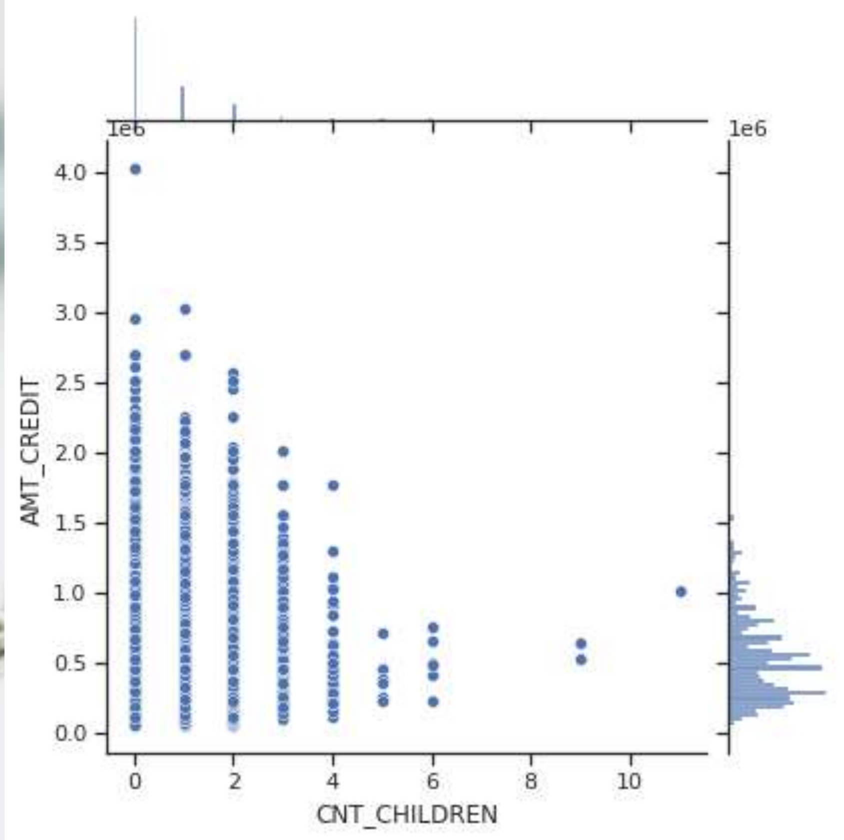
- Bivariate using IQR calculation.Values less than max value of both AMT_CREDIT and AMT_INCOME_TOTAL





- Bivariate using IQR calculation.Values less than max value of both AMT_CREDIT and DAYS_EMPLOYED



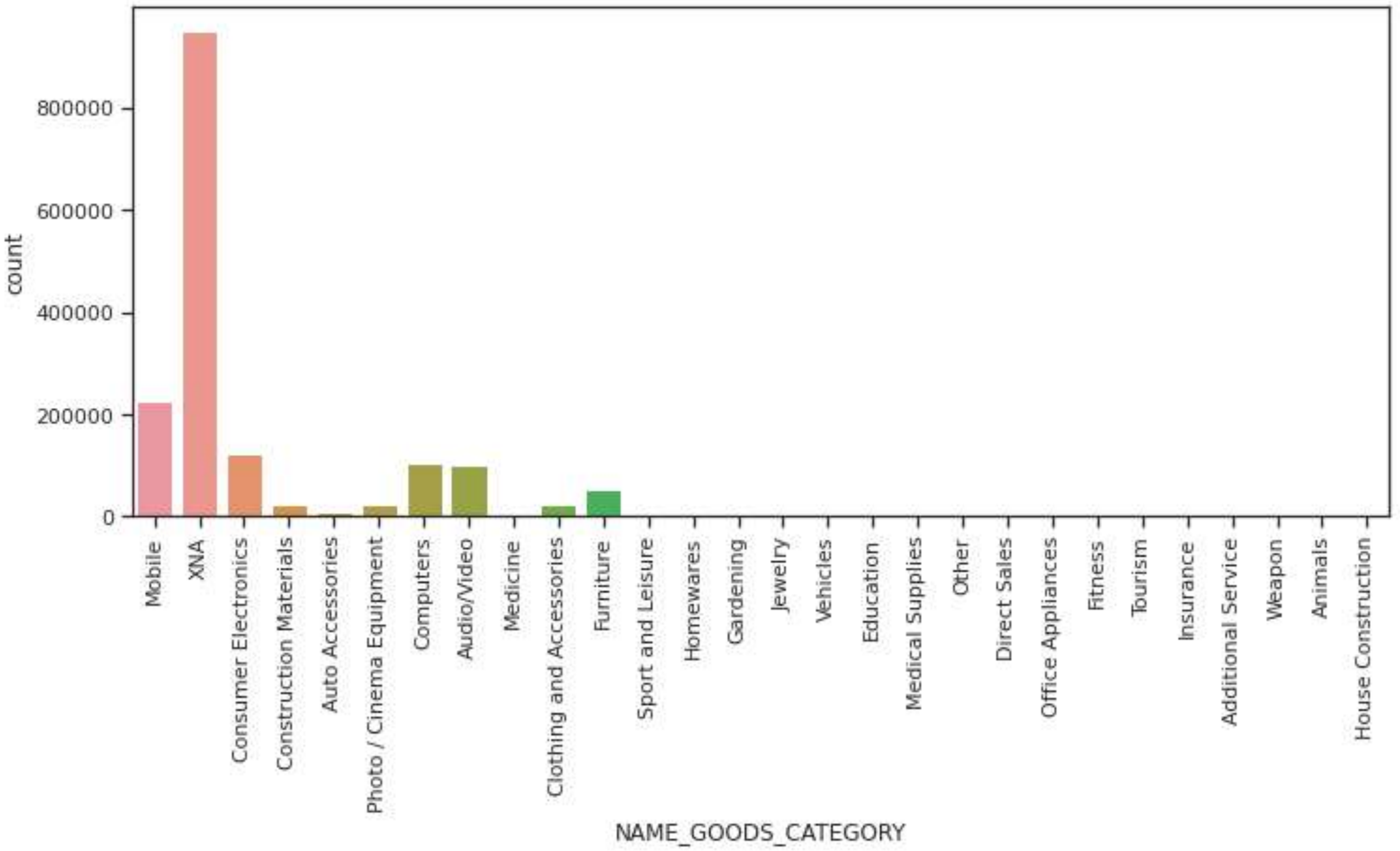


- Bivariate using IQR calculation.Values less than max value of both AMT_CREDIT and CNT_CHILDREN

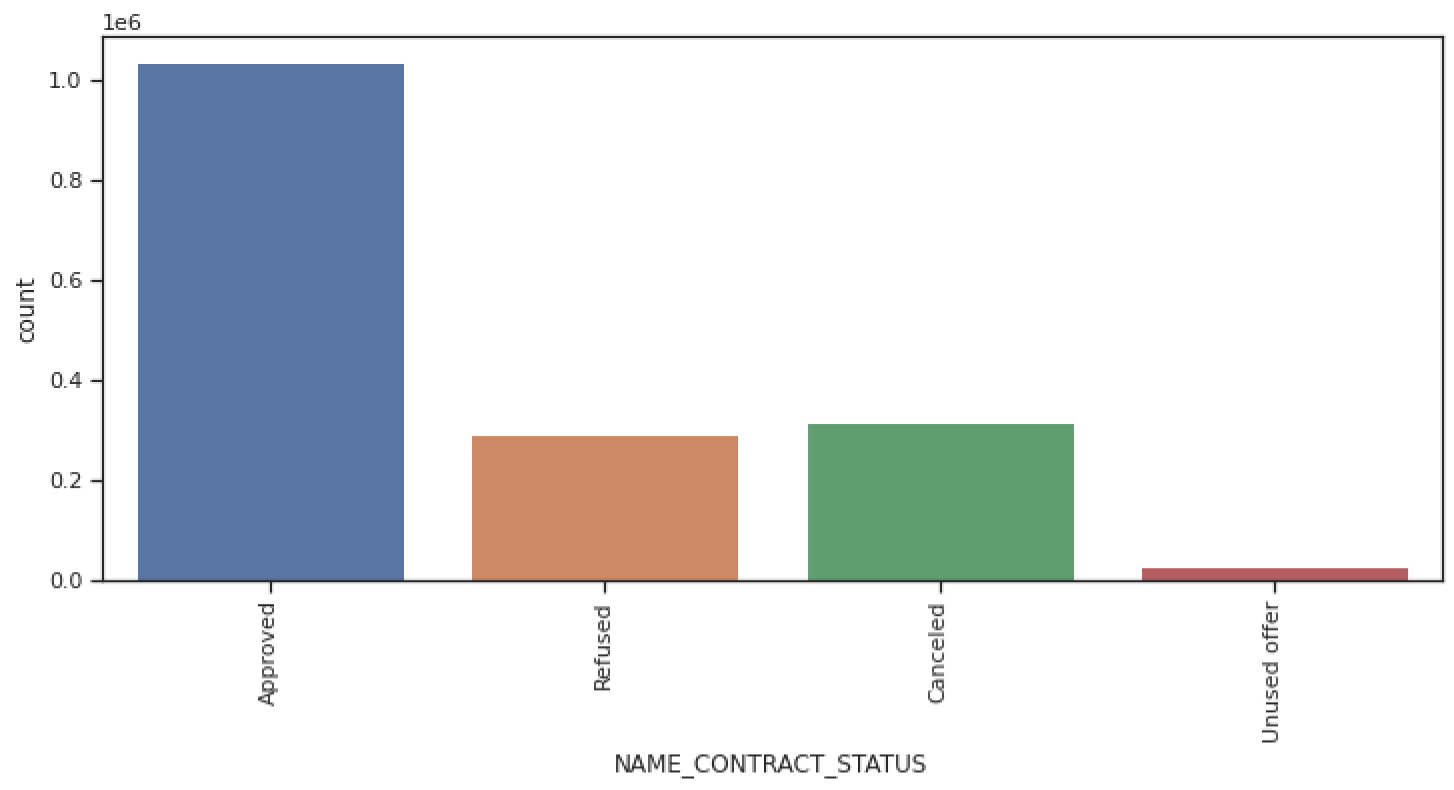
TO TEST CORRELATION BETWEEN CRITICAL QUANTITATIVE VALUES IN THE CURRENT APPLICATION SET FOR CLIENTS HAVING DIFFICULTY IN PAYMENT

TO TEST CORRELATION BETWEEN CRITICAL QUANTITATIVE VALUES IN THE CURRENT APPLICATION SET FOR CLIENTS WHO PAY ON TIME

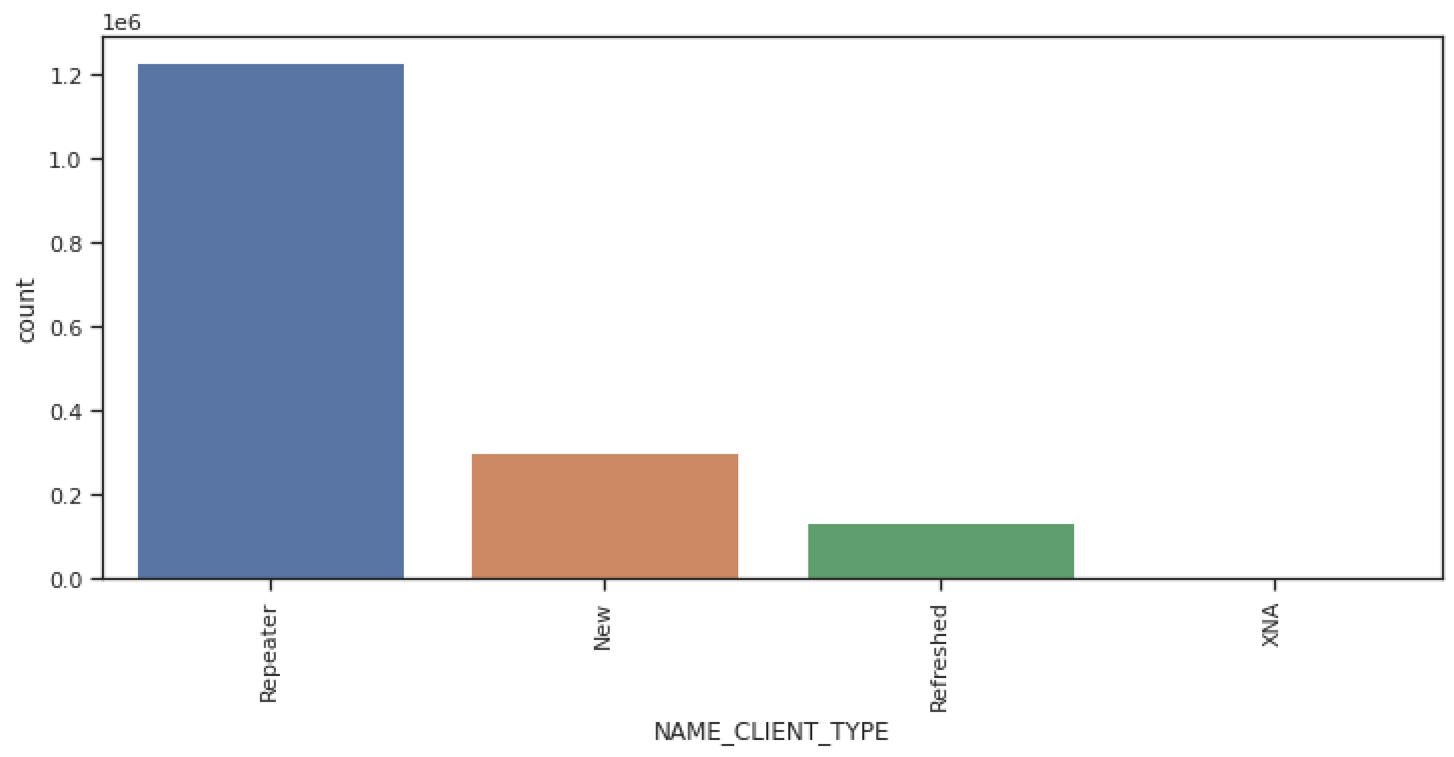
Previous Application Data Analysis



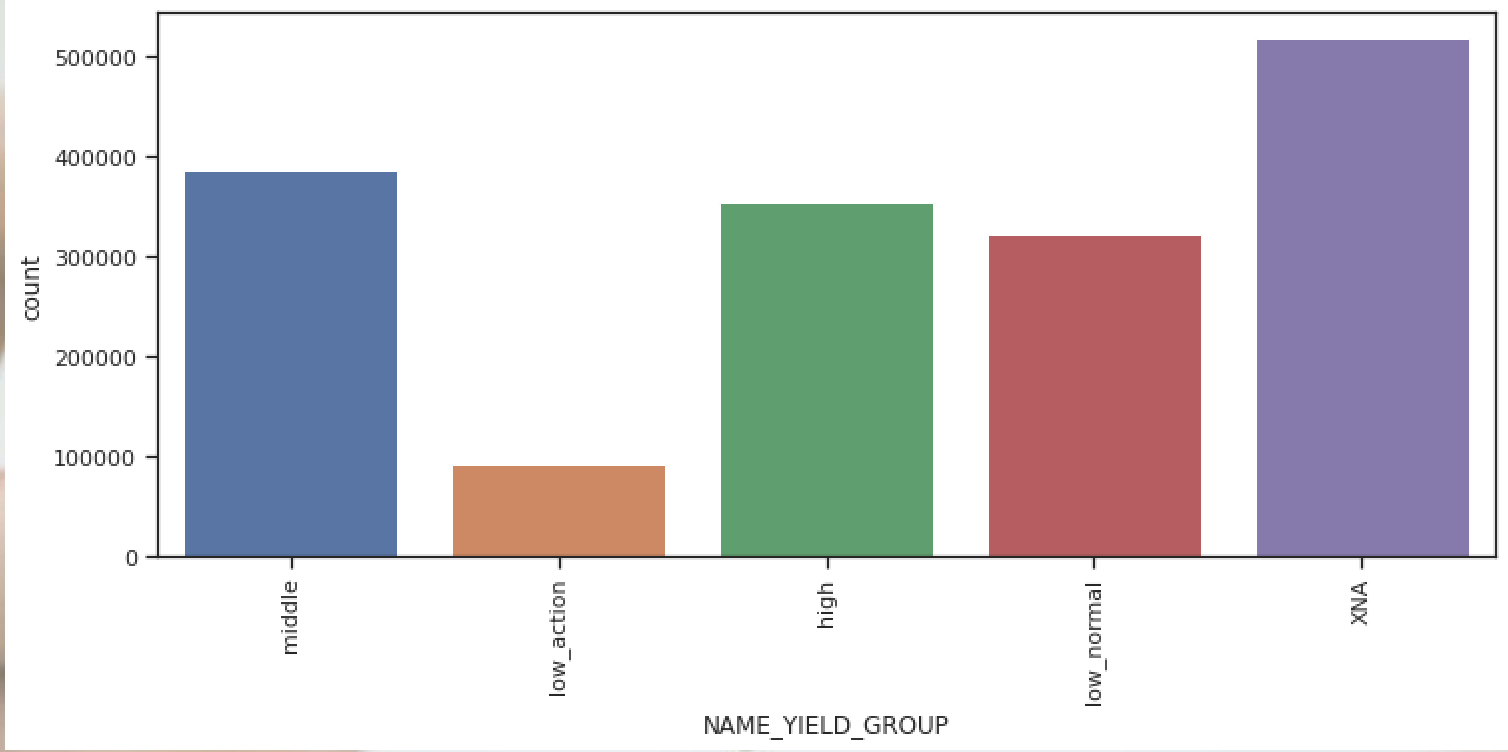
- XNA goods category is the highest among all loan applications
- mobile goods category is the second highest among all loan applications



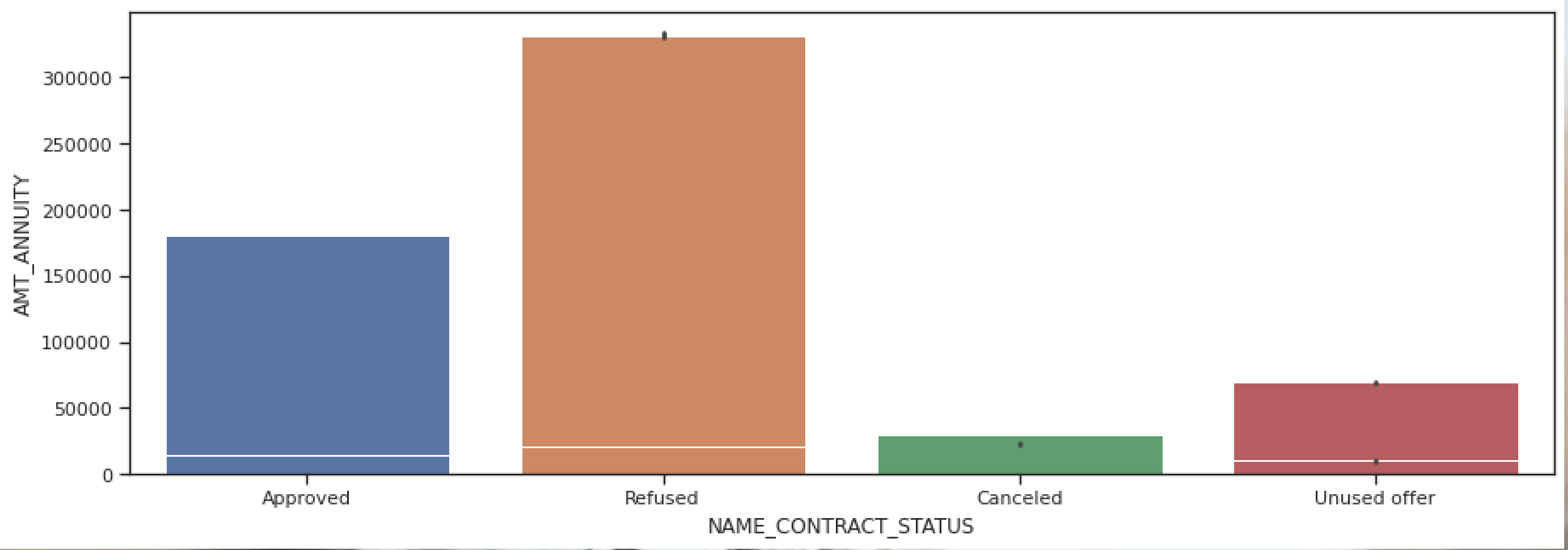
- Approved loan status is the highest among all loan applications
- Canceled loan status is the second highest among all loan applications



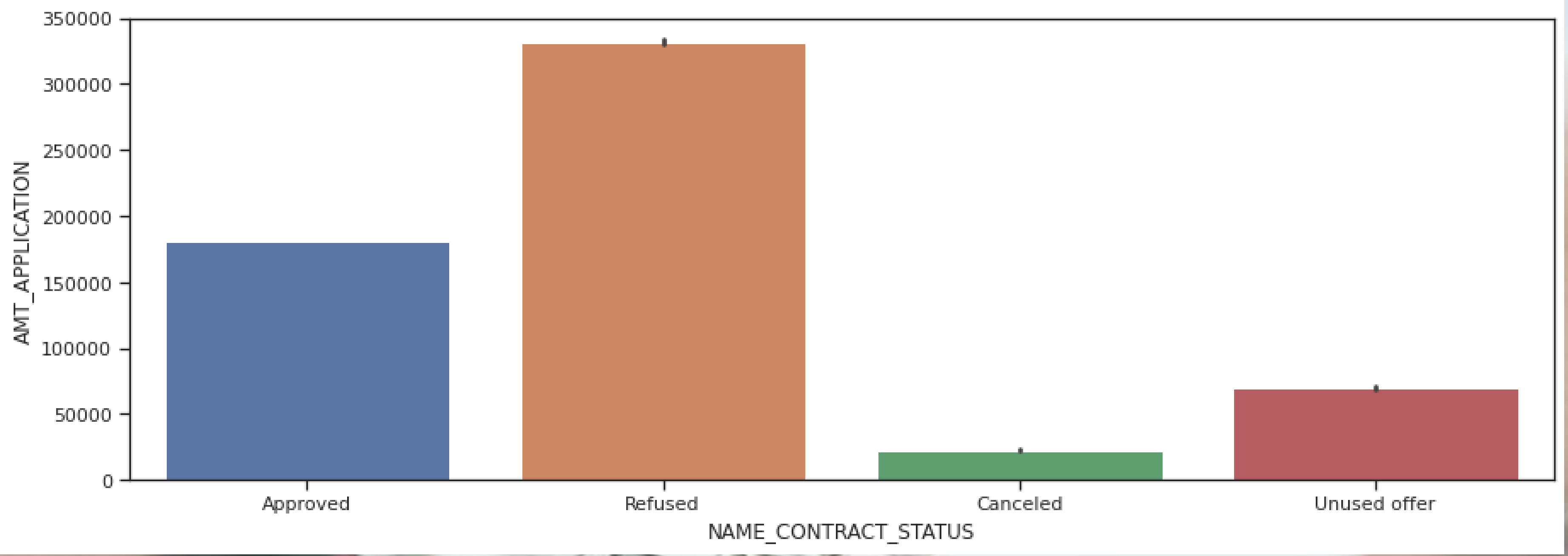
- Repeater client type is the highest among all loan applications
- New client type is the second highest among all loan applications



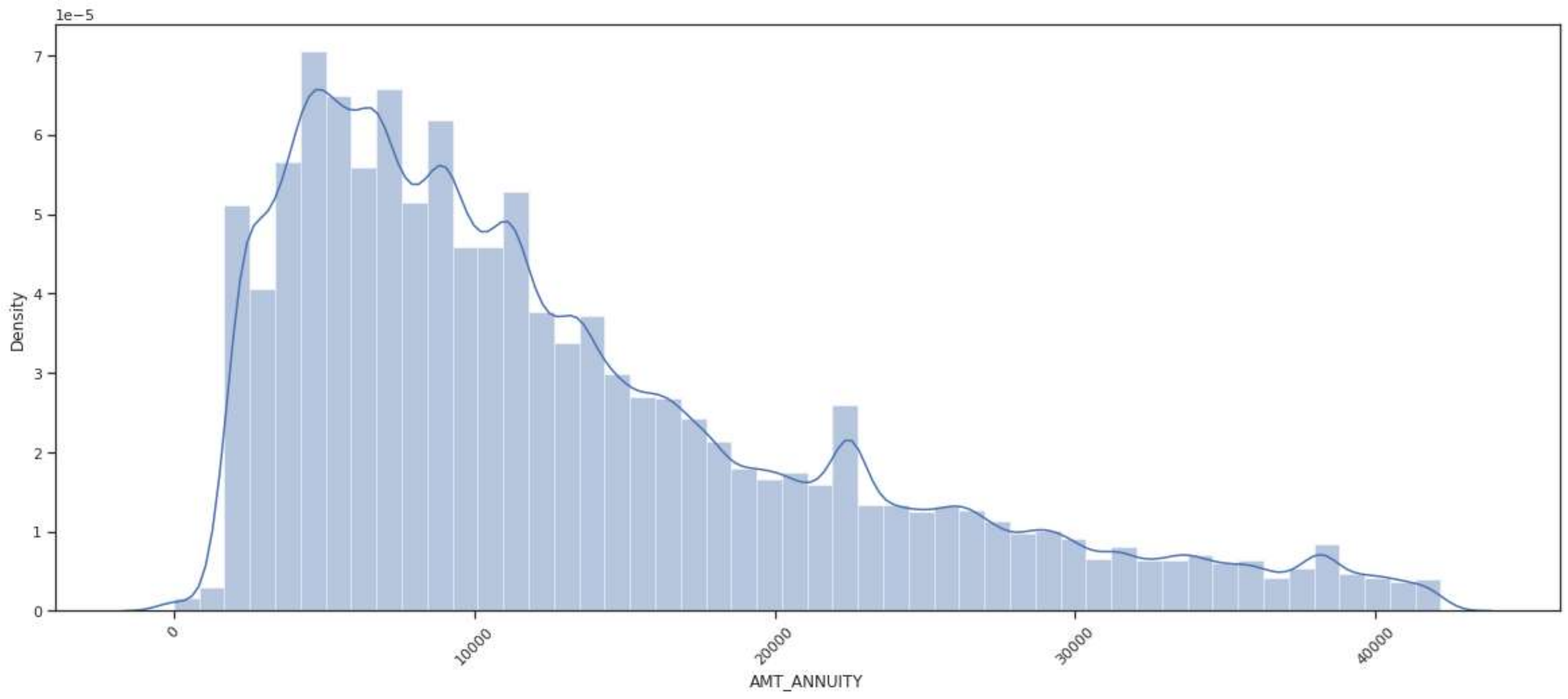
- **`XNA`** interest rate is the highest among all loan applications
- **`middle`** and **`high`** interest rates are the second and third highest among all loan applications



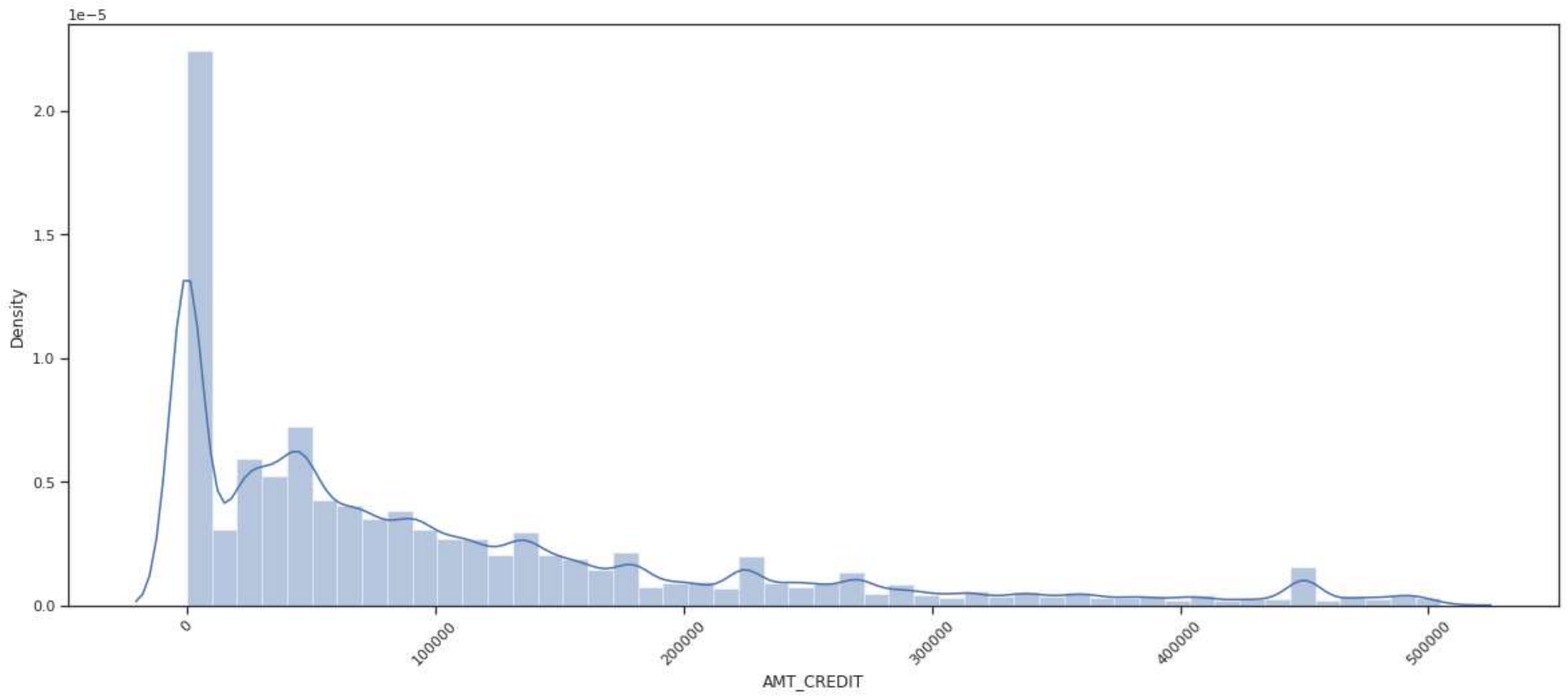
- Bivariate using IQR calculation.Values less than max value of both AMT_ANNUTY and NAME_CONTRACT_STATUS
- Refused status occurred most for AMT_ANNUITY



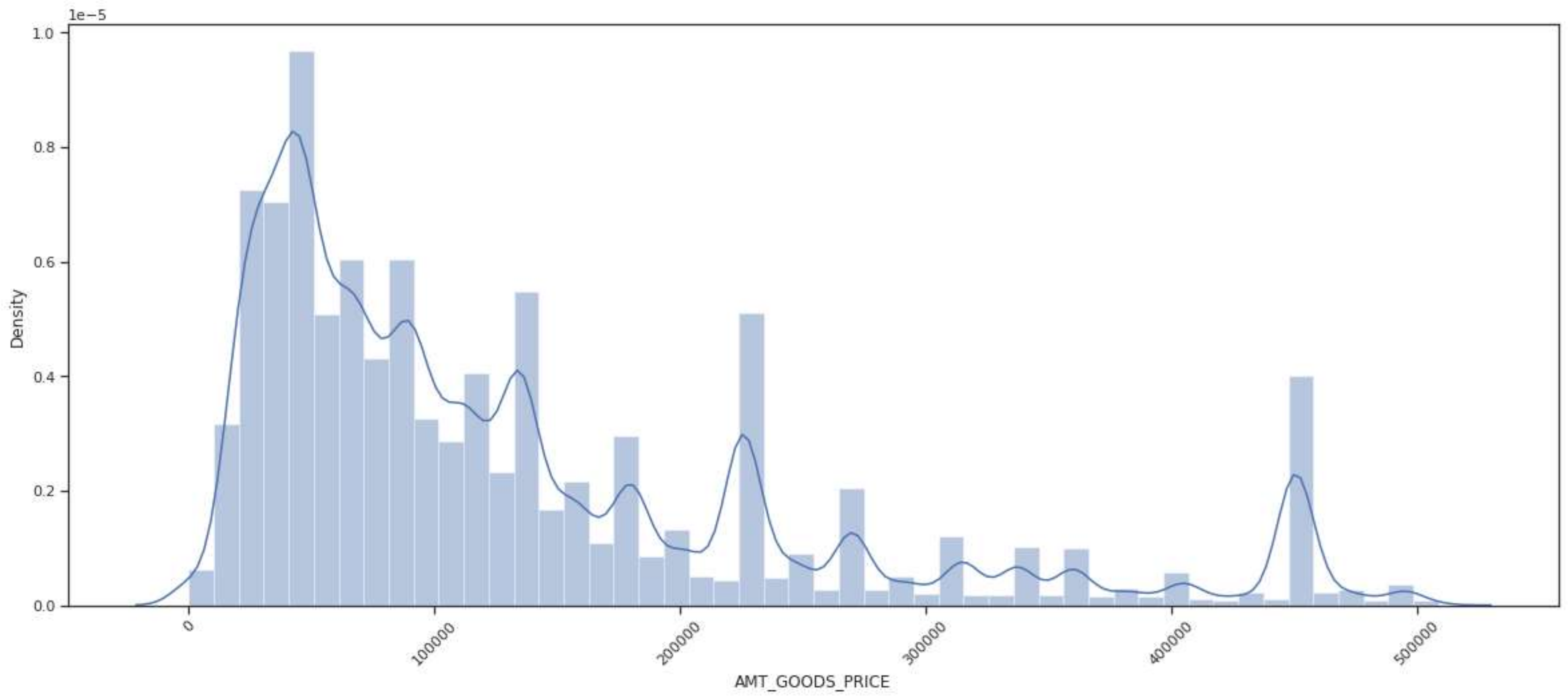
- Bivariate using IQR calculation.Values less than max value of both AMT_APPLICATION and NAME_CONTRACT_STATUS
- Refused status occurred most for AMT_APPLICATION



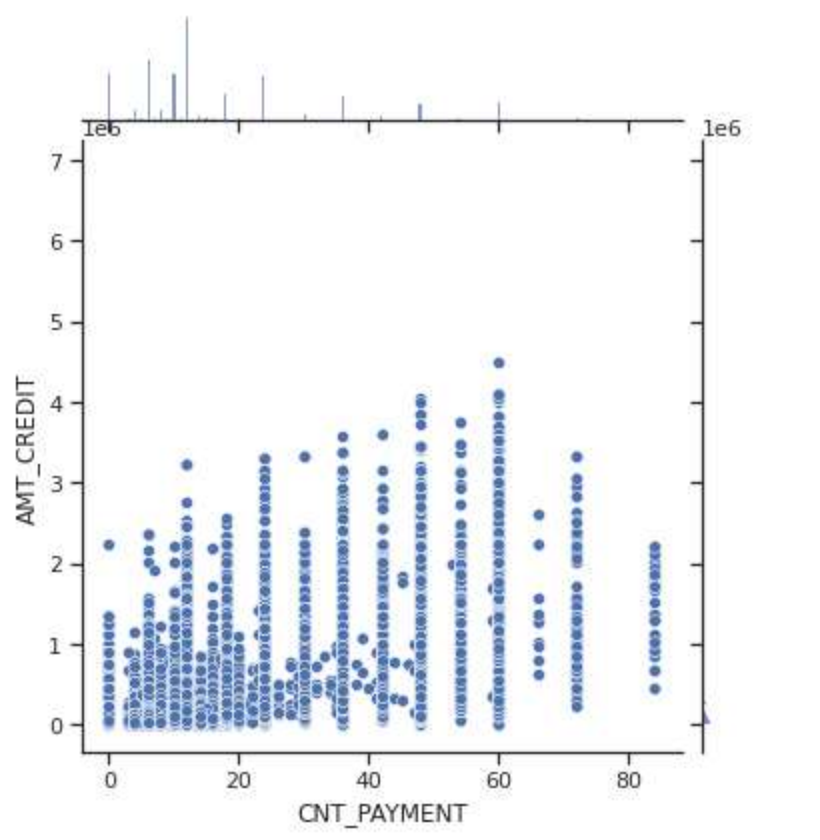
- Most of the previous loan's annuity from the clients is less than 10,000 as the distribution is high here
- As previous loan's annuity increases, the no. of clients decreases

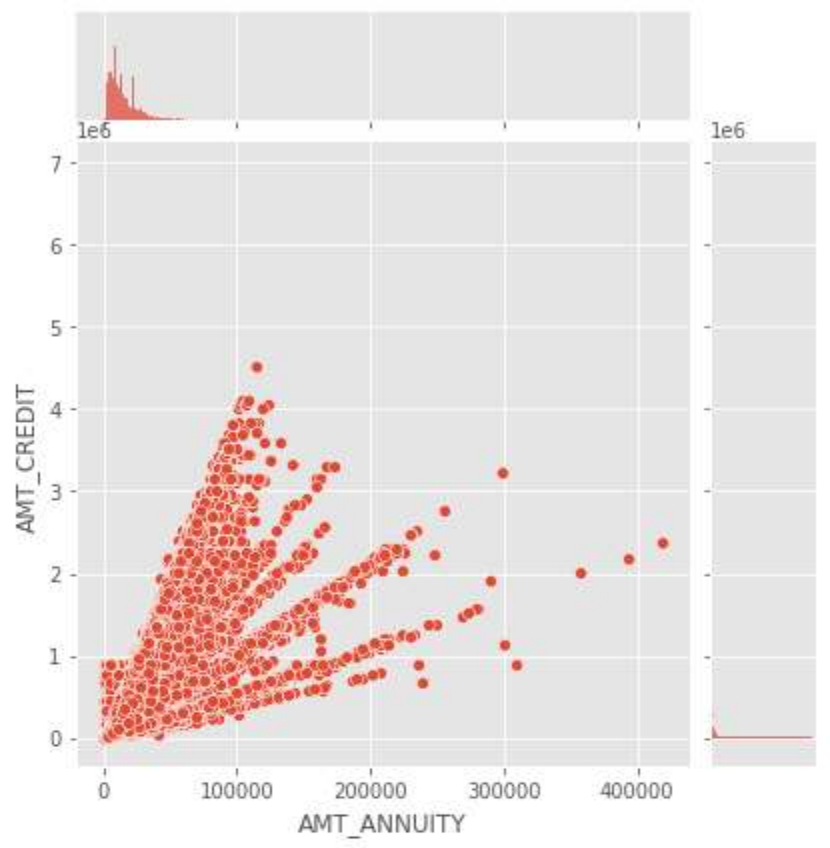


- This distribution very closely resembles that of AMT_APPLICATION. This means that most people received the loan amount that they applied for



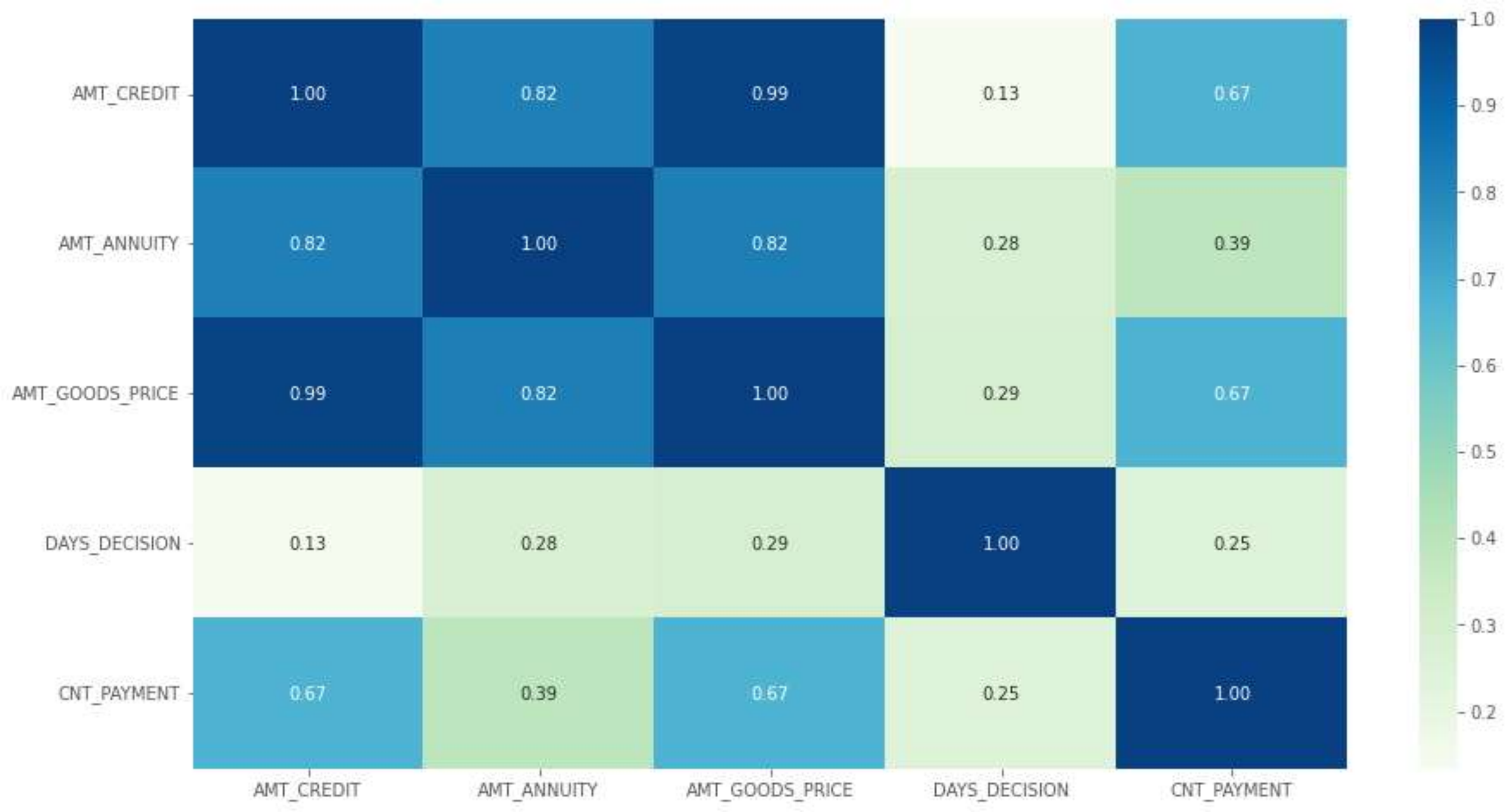
- Most of the goods price asked by clients in previous application is less than 100K



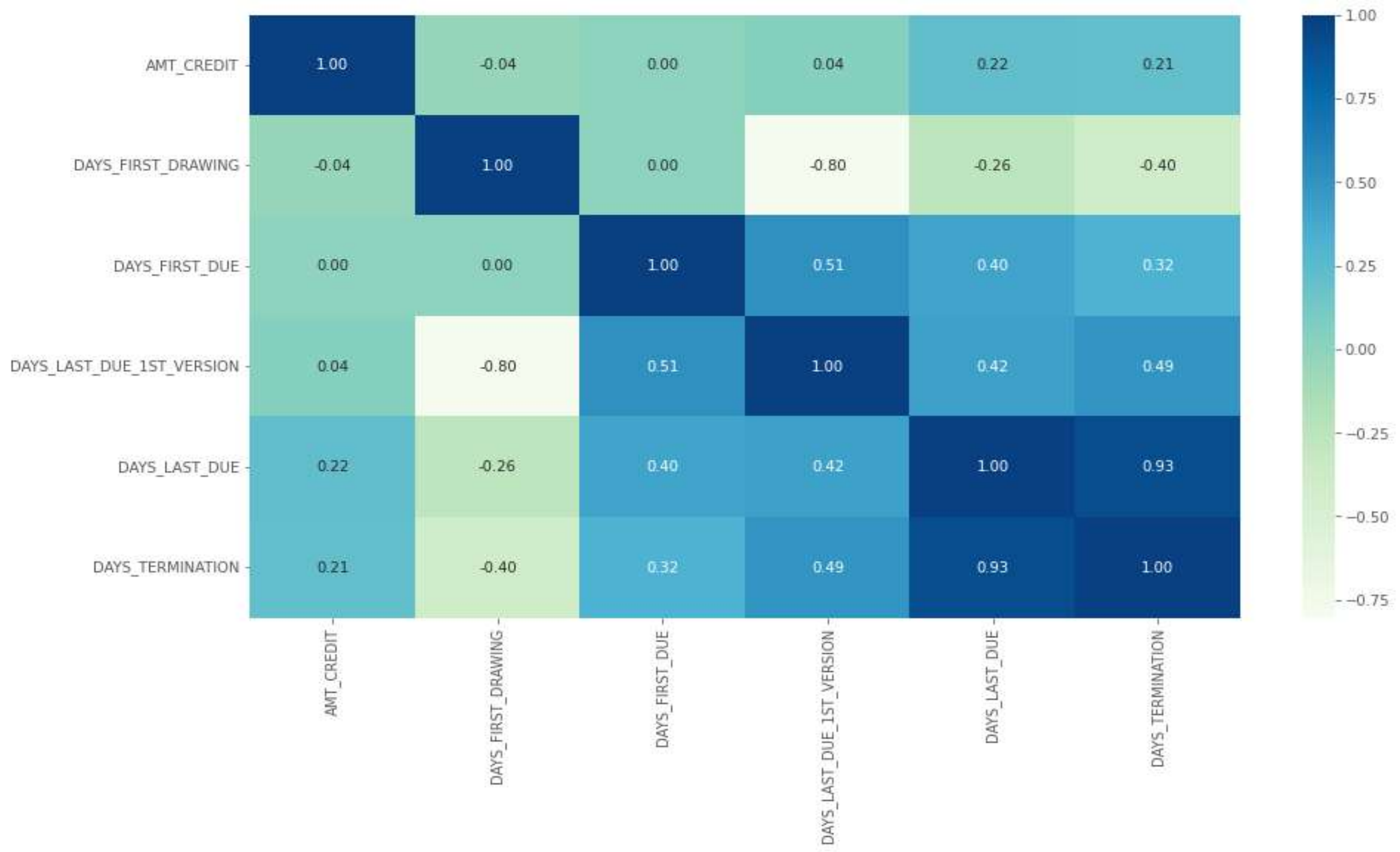


- Bivariate using IQR calculation.Values less than max value of both AMT_CREDIT and CNT_PAYMENT

- AMT_ANNUITY and AMT_CREDIT have strong positive correlation. This



The loan amount sanctioned has a strong correlation with the AMT_GOODS_PRICE and AMT_ANNUITY



The first drawing, First due, Last Due and Last termination has "NO" bearing to the saction loan amount

Observation

Top 10 correlation for payment difficulty after removing columns and filling incorrect/missing data

AMT_CREDIT	AMT_GOODS_PRICE	0.983103
REGION_RATING_CLIENT_W_CITY	REGION_RATING_CLIENT	0.956637
CNT_CHILDREN	CNT_FAM_MEMBERS	0.885484
AMT_ANNUITY	AMT_GOODS_PRICE	0.752699
AMT_CREDIT	AMT_ANNUITY	0.752195
DAYS_BIRTH	DAYS_EMPLOYED	0.582187
DAYS_REGISTRATION	DAYS_BIRTH	0.289111
DAYS_ID_PUBLISH	DAYS_BIRTH	0.252867
DAYS_EMPLOYED	DAYS_ID_PUBLISH	0.229094
DAYS_REGISTRATION	DAYS_EMPLOYED	0.192454