Loan Defaulter Risk Analysis(EDA)

Introduction

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

Import Python Libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# this will enable pandas to show all the items
pd.options.display.max_columns = None
pd.options.display.max_rows = None
```

Loading Datasets

In [2]:
```
df_app = pd.read_csv("Dataset/application_data.csv")
df_prev = pd.read_csv("Dataset/previous_application.csv")
```

In [3]:
```
df_app.head()
```

Out[3]:

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

Data Cleaning & Manipulation

Check the presence of missing values

In [4]:
```
df_app.columns
```

Out[4]:
```
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',
       ...
       'FLAG_DOCUMENT_18', 'FLAG_DOCUMENT_19', 'FLAG_DOCUMENT_20',
       'FLAG_DOCUMENT_21', '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'],
      dtype='object', length=122)
```

In [5]:
```
df_app.shape
```

Out[5]:
```
(307511, 122)
```

In [6]:
```
df_app.isnull().sum().sort_values()   # it shows How many have null values
```

Out[6]:
```
SK_ID_CURR                         0
HOUR_APPR_PROCESS_START            0
REG_REGION_NOT_WORK_REGION         0
LIVE_REGION_NOT_WORK_REGION        0
REG_CITY_NOT_LIVE_CITY             0
REG_CITY_NOT_WORK_CITY             0
LIVE_CITY_NOT_WORK_CITY            0
```

```
ORGANIZATION_TYPE                   0
FLAG_DOCUMENT_21                    0
FLAG_DOCUMENT_20                    0
FLAG_DOCUMENT_19                    0
FLAG_DOCUMENT_18                    0
FLAG_DOCUMENT_17                    0
FLAG_DOCUMENT_16                    0
FLAG_DOCUMENT_15                    0
FLAG_DOCUMENT_14                    0
FLAG_DOCUMENT_13                    0
FLAG_DOCUMENT_12                    0
FLAG_DOCUMENT_11                    0
FLAG_DOCUMENT_10                    0
FLAG_DOCUMENT_9                     0
FLAG_DOCUMENT_8                     0
FLAG_DOCUMENT_7                     0
FLAG_DOCUMENT_6                     0
FLAG_DOCUMENT_5                     0
FLAG_DOCUMENT_4                     0
FLAG_DOCUMENT_3                     0
FLAG_DOCUMENT_2                     0
WEEKDAY_APPR_PROCESS_START          0
REGION_RATING_CLIENT_W_CITY         0
REG_REGION_NOT_LIVE_REGION          0
NAME_HOUSING_TYPE                   0
CNT_CHILDREN                        0
NAME_INCOME_TYPE                    0
NAME_EDUCATION_TYPE                 0
NAME_FAMILY_STATUS                  0
REGION_RATING_CLIENT                0
REGION_POPULATION_RELATIVE          0
DAYS_BIRTH                          0
DAYS_EMPLOYED                       0
DAYS_REGISTRATION                   0
DAYS_ID_PUBLISH                     0
AMT_INCOME_TOTAL                    0
FLAG_OWN_REALTY                     0
CODE_GENDER                         0
NAME_CONTRACT_TYPE                  0
FLAG_MOBIL                          0
FLAG_EMP_PHONE                      0
FLAG_WORK_PHONE                     0
FLAG_CONT_MOBILE                    0
FLAG_PHONE                          0
TARGET                              0
FLAG_EMAIL                          0
FLAG_OWN_CAR                        0
AMT_CREDIT                          0
DAYS_LAST_PHONE_CHANGE              1
CNT_FAM_MEMBERS                     2
AMT_ANNUITY                        12
AMT_GOODS_PRICE                   278
EXT_SOURCE_2                      660
DEF_30_CNT_SOCIAL_CIRCLE         1021
DEF_60_CNT_SOCIAL_CIRCLE         1021
OBS_60_CNT_SOCIAL_CIRCLE         1021
OBS_30_CNT_SOCIAL_CIRCLE         1021
NAME_TYPE_SUITE                  1292
AMT_REQ_CREDIT_BUREAU_HOUR      41519
AMT_REQ_CREDIT_BUREAU_DAY       41519
AMT_REQ_CREDIT_BUREAU_MON       41519
AMT_REQ_CREDIT_BUREAU_WEEK      41519
AMT_REQ_CREDIT_BUREAU_YEAR      41519
AMT_REQ_CREDIT_BUREAU_QRT       41519
EXT_SOURCE_3                    60965
OCCUPATION_TYPE                 96391
EMERGENCYSTATE_MODE            145755
TOTALAREA_MODE                 148431
YEARS_BEGINEXPLUATATION_MODE   150007
YEARS_BEGINEXPLUATATION_AVG    150007
YEARS_BEGINEXPLUATATION_MEDI   150007
FLOORSMAX_AVG                  153020
FLOORSMAX_MEDI                 153020
FLOORSMAX_MODE                 153020
HOUSETYPE_MODE                 154297
LIVINGAREA_AVG                 154350
LIVINGAREA_MODE                154350
LIVINGAREA_MEDI                154350
ENTRANCES_AVG                  154828
ENTRANCES_MODE                 154828
ENTRANCES_MEDI                 154828
APARTMENTS_MEDI                156061
APARTMENTS_AVG                 156061
```

```
APARTMENTS_MODE                 156061
WALLSMATERIAL_MODE              156341
ELEVATORS_MEDI                  163891
ELEVATORS_AVG                   163891
ELEVATORS_MODE                  163891
NONLIVINGAREA_MODE              169682
NONLIVINGAREA_AVG               169682
NONLIVINGAREA_MEDI              169682
EXT_SOURCE_1                    173378
BASEMENTAREA_MODE               179943
BASEMENTAREA_AVG                179943
BASEMENTAREA_MEDI               179943
LANDAREA_MEDI                   182590
LANDAREA_AVG                    182590
LANDAREA_MODE                   182590
OWN_CAR_AGE                     202929
YEARS_BUILD_MODE                204488
YEARS_BUILD_AVG                 204488
YEARS_BUILD_MEDI                204488
FLOORSMIN_AVG                   208642
FLOORSMIN_MODE                  208642
FLOORSMIN_MEDI                  208642
LIVINGAPARTMENTS_AVG            210199
LIVINGAPARTMENTS_MODE           210199
LIVINGAPARTMENTS_MEDI           210199
FONDKAPREMONT_MODE              210295
NONLIVINGAPARTMENTS_AVG         213514
NONLIVINGAPARTMENTS_MEDI        213514
NONLIVINGAPARTMENTS_MODE        213514
COMMONAREA_MODE                 214865
COMMONAREA_AVG                  214865
COMMONAREA_MEDI                 214865
dtype: int64
```

In [7]:
```
df_app_info = pd.DataFrame(df_app.isnull().sum().sort_values()).reset_index()
df_app_info.rename(columns={"index":"Cols_name",0:'null_count'},inplace=True)
df_app_info
```

Out[7]:

	Cols_name	null_count
0	SK_ID_CURR	0
1	HOUR_APPR_PROCESS_START	0
2	REG_REGION_NOT_WORK_REGION	0
3	LIVE_REGION_NOT_WORK_REGION	0
4	REG_CITY_NOT_LIVE_CITY	0
5	REG_CITY_NOT_WORK_CITY	0
6	LIVE_CITY_NOT_WORK_CITY	0
7	ORGANIZATION_TYPE	0
8	FLAG_DOCUMENT_21	0
9	FLAG_DOCUMENT_20	0
10	FLAG_DOCUMENT_19	0
11	FLAG_DOCUMENT_18	0
12	FLAG_DOCUMENT_17	0
13	FLAG_DOCUMENT_16	0
14	FLAG_DOCUMENT_15	0
15	FLAG_DOCUMENT_14	0
16	FLAG_DOCUMENT_13	0
17	FLAG_DOCUMENT_12	0
18	FLAG_DOCUMENT_11	0
19	FLAG_DOCUMENT_10	0
20	FLAG_DOCUMENT_9	0
21	FLAG_DOCUMENT_8	0
22	FLAG_DOCUMENT_7	0
23	FLAG_DOCUMENT_6	0
24	FLAG_DOCUMENT_5	0
25	FLAG_DOCUMENT_4	0
26	FLAG_DOCUMENT_3	0

27	FLAG_DOCUMENT_2	0
28	WEEKDAY_APPR_PROCESS_START	0
29	REGION_RATING_CLIENT_W_CITY	0
30	REG_REGION_NOT_LIVE_REGION	0
31	NAME_HOUSING_TYPE	0
32	CNT_CHILDREN	0
33	NAME_INCOME_TYPE	0
34	NAME_EDUCATION_TYPE	0
35	NAME_FAMILY_STATUS	0
36	REGION_RATING_CLIENT	0
37	REGION_POPULATION_RELATIVE	0
38	DAYS_BIRTH	0
39	DAYS_EMPLOYED	0
40	DAYS_REGISTRATION	0
41	DAYS_ID_PUBLISH	0
42	AMT_INCOME_TOTAL	0
43	FLAG_OWN_REALTY	0
44	CODE_GENDER	0
45	NAME_CONTRACT_TYPE	0
46	FLAG_MOBIL	0
47	FLAG_EMP_PHONE	0
48	FLAG_WORK_PHONE	0
49	FLAG_CONT_MOBILE	0
50	FLAG_PHONE	0
51	TARGET	0
52	FLAG_EMAIL	0
53	FLAG_OWN_CAR	0
54	AMT_CREDIT	0
55	DAYS_LAST_PHONE_CHANGE	1
56	CNT_FAM_MEMBERS	2
57	AMT_ANNUITY	12
58	AMT_GOODS_PRICE	278
59	EXT_SOURCE_2	660
60	DEF_30_CNT_SOCIAL_CIRCLE	1021
61	DEF_60_CNT_SOCIAL_CIRCLE	1021
62	OBS_60_CNT_SOCIAL_CIRCLE	1021
63	OBS_30_CNT_SOCIAL_CIRCLE	1021
64	NAME_TYPE_SUITE	1292
65	AMT_REQ_CREDIT_BUREAU_HOUR	41519
66	AMT_REQ_CREDIT_BUREAU_DAY	41519
67	AMT_REQ_CREDIT_BUREAU_MON	41519
68	AMT_REQ_CREDIT_BUREAU_WEEK	41519
69	AMT_REQ_CREDIT_BUREAU_YEAR	41519
70	AMT_REQ_CREDIT_BUREAU_QRT	41519
71	EXT_SOURCE_3	60965
72	OCCUPATION_TYPE	96391
73	EMERGENCYSTATE_MODE	145755
74	TOTALAREA_MODE	148431
75	YEARS_BEGINEXPLUATATION_MODE	150007
76	YEARS_BEGINEXPLUATATION_AVG	150007
77	YEARS_BEGINEXPLUATATION_MEDI	150007

78	FLOORSMAX_AVG	153020
79	FLOORSMAX_MEDI	153020
80	FLOORSMAX_MODE	153020
81	HOUSETYPE_MODE	154297
82	LIVINGAREA_AVG	154350
83	LIVINGAREA_MODE	154350
84	LIVINGAREA_MEDI	154350
85	ENTRANCES_AVG	154828
86	ENTRANCES_MODE	154828
87	ENTRANCES_MEDI	154828
88	APARTMENTS_MEDI	156061
89	APARTMENTS_AVG	156061
90	APARTMENTS_MODE	156061
91	WALLSMATERIAL_MODE	156341
92	ELEVATORS_MEDI	163891
93	ELEVATORS_AVG	163891
94	ELEVATORS_MODE	163891
95	NONLIVINGAREA_MODE	169682
96	NONLIVINGAREA_AVG	169682
97	NONLIVINGAREA_MEDI	169682
98	EXT_SOURCE_1	173378
99	BASEMENTAREA_MODE	179943
100	BASEMENTAREA_AVG	179943
101	BASEMENTAREA_MEDI	179943
102	LANDAREA_MEDI	182590
103	LANDAREA_AVG	182590
104	LANDAREA_MODE	182590
105	OWN_CAR_AGE	202929
106	YEARS_BUILD_MODE	204488
107	YEARS_BUILD_AVG	204488
108	YEARS_BUILD_MEDI	204488
109	FLOORSMIN_AVG	208642
110	FLOORSMIN_MODE	208642
111	FLOORSMIN_MEDI	208642
112	LIVINGAPARTMENTS_AVG	210199
113	LIVINGAPARTMENTS_MODE	210199
114	LIVINGAPARTMENTS_MEDI	210199
115	FONDKAPREMONT_MODE	210295
116	NONLIVINGAPARTMENTS_AVG	213514
117	NONLIVINGAPARTMENTS_MEDI	213514
118	NONLIVINGAPARTMENTS_MODE	213514
119	COMMONAREA_MODE	214865
120	COMMONAREA_AVG	214865
121	COMMONAREA_MEDI	214865

In [8]:
```
df_app_info['percentage %'] =  df_app_info['null_count']/df_app.shape[0] * 100
```

In [9]:
```
df_app_info
```

Out[9]:

	Cols_name	null_count	percentage %
0	SK_ID_CURR	0	0.000000
1	HOUR_APPR_PROCESS_START	0	0.000000
2	REG_REGION_NOT_WORK_REGION	0	0.000000

2	REG_REGION_NOT_WORK_REGION	0	0.000000
3	LIVE_REGION_NOT_WORK_REGION	0	0.000000
4	REG_CITY_NOT_LIVE_CITY	0	0.000000
5	REG_CITY_NOT_WORK_CITY	0	0.000000
6	LIVE_CITY_NOT_WORK_CITY	0	0.000000
7	ORGANIZATION_TYPE	0	0.000000
8	FLAG_DOCUMENT_21	0	0.000000
9	FLAG_DOCUMENT_20	0	0.000000
10	FLAG_DOCUMENT_19	0	0.000000
11	FLAG_DOCUMENT_18	0	0.000000
12	FLAG_DOCUMENT_17	0	0.000000
13	FLAG_DOCUMENT_16	0	0.000000
14	FLAG_DOCUMENT_15	0	0.000000
15	FLAG_DOCUMENT_14	0	0.000000
16	FLAG_DOCUMENT_13	0	0.000000
17	FLAG_DOCUMENT_12	0	0.000000
18	FLAG_DOCUMENT_11	0	0.000000
19	FLAG_DOCUMENT_10	0	0.000000
20	FLAG_DOCUMENT_9	0	0.000000
21	FLAG_DOCUMENT_8	0	0.000000
22	FLAG_DOCUMENT_7	0	0.000000
23	FLAG_DOCUMENT_6	0	0.000000
24	FLAG_DOCUMENT_5	0	0.000000
25	FLAG_DOCUMENT_4	0	0.000000
26	FLAG_DOCUMENT_3	0	0.000000
27	FLAG_DOCUMENT_2	0	0.000000
28	WEEKDAY_APPR_PROCESS_START	0	0.000000
29	REGION_RATING_CLIENT_W_CITY	0	0.000000
30	REG_REGION_NOT_LIVE_REGION	0	0.000000
31	NAME_HOUSING_TYPE	0	0.000000
32	CNT_CHILDREN	0	0.000000
33	NAME_INCOME_TYPE	0	0.000000
34	NAME_EDUCATION_TYPE	0	0.000000
35	NAME_FAMILY_STATUS	0	0.000000
36	REGION_RATING_CLIENT	0	0.000000
37	REGION_POPULATION_RELATIVE	0	0.000000
38	DAYS_BIRTH	0	0.000000
39	DAYS_EMPLOYED	0	0.000000
40	DAYS_REGISTRATION	0	0.000000
41	DAYS_ID_PUBLISH	0	0.000000
42	AMT_INCOME_TOTAL	0	0.000000
43	FLAG_OWN_REALTY	0	0.000000
44	CODE_GENDER	0	0.000000
45	NAME_CONTRACT_TYPE	0	0.000000
46	FLAG_MOBIL	0	0.000000
47	FLAG_EMP_PHONE	0	0.000000
48	FLAG_WORK_PHONE	0	0.000000
49	FLAG_CONT_MOBILE	0	0.000000
50	FLAG_PHONE	0	0.000000
51	TARGET	0	0.000000
52	FLAG_EMAIL	0	0.000000

53	FLAG_OWN_CAR	0	0.000000
54	AMT_CREDIT	0	0.000000
55	DAYS_LAST_PHONE_CHANGE	1	0.000325
56	CNT_FAM_MEMBERS	2	0.000650
57	AMT_ANNUITY	12	0.003902
58	AMT_GOODS_PRICE	278	0.090403
59	EXT_SOURCE_2	660	0.214626
60	DEF_30_CNT_SOCIAL_CIRCLE	1021	0.332021
61	DEF_60_CNT_SOCIAL_CIRCLE	1021	0.332021
62	OBS_60_CNT_SOCIAL_CIRCLE	1021	0.332021
63	OBS_30_CNT_SOCIAL_CIRCLE	1021	0.332021
64	NAME_TYPE_SUITE	1292	0.420148
65	AMT_REQ_CREDIT_BUREAU_HOUR	41519	13.501631
66	AMT_REQ_CREDIT_BUREAU_DAY	41519	13.501631
67	AMT_REQ_CREDIT_BUREAU_MON	41519	13.501631
68	AMT_REQ_CREDIT_BUREAU_WEEK	41519	13.501631
69	AMT_REQ_CREDIT_BUREAU_YEAR	41519	13.501631
70	AMT_REQ_CREDIT_BUREAU_QRT	41519	13.501631
71	EXT_SOURCE_3	60965	19.825307
72	OCCUPATION_TYPE	96391	31.345545
73	EMERGENCYSTATE_MODE	145755	47.398304
74	TOTALAREA_MODE	148431	48.268517
75	YEARS_BEGINEXPLUATATION_MODE	150007	48.781019
76	YEARS_BEGINEXPLUATATION_AVG	150007	48.781019
77	YEARS_BEGINEXPLUATATION_MEDI	150007	48.781019
78	FLOORSMAX_AVG	153020	49.760822
79	FLOORSMAX_MEDI	153020	49.760822
80	FLOORSMAX_MODE	153020	49.760822
81	HOUSETYPE_MODE	154297	50.176091
82	LIVINGAREA_AVG	154350	50.193326
83	LIVINGAREA_MODE	154350	50.193326
84	LIVINGAREA_MEDI	154350	50.193326
85	ENTRANCES_AVG	154828	50.348768
86	ENTRANCES_MODE	154828	50.348768
87	ENTRANCES_MEDI	154828	50.348768
88	APARTMENTS_MEDI	156061	50.749729
89	APARTMENTS_AVG	156061	50.749729
90	APARTMENTS_MODE	156061	50.749729
91	WALLSMATERIAL_MODE	156341	50.840783
92	ELEVATORS_MEDI	163891	53.295980
93	ELEVATORS_AVG	163891	53.295980
94	ELEVATORS_MODE	163891	53.295980
95	NONLIVINGAREA_MODE	169682	55.179164
96	NONLIVINGAREA_AVG	169682	55.179164
97	NONLIVINGAREA_MEDI	169682	55.179164
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
111	FLOORSMIN_MEDI	208642	67.848630
112	LIVINGAPARTMENTS_AVG	210199	68.354953
113	LIVINGAPARTMENTS_MODE	210199	68.354953
114	LIVINGAPARTMENTS_MEDI	210199	68.354953
115	FONDKAPREMONT_MODE	210295	68.386172
116	NONLIVINGAPARTMENTS_AVG	213514	69.432963
117	NONLIVINGAPARTMENTS_MEDI	213514	69.432963
118	NONLIVINGAPARTMENTS_MODE	213514	69.432963
119	COMMONAREA_MODE	214865	69.872297
120	COMMONAREA_AVG	214865	69.872297
121	COMMONAREA_MEDI	214865	69.872297

In [10]:
```
missing_cols = df_app_info[df_app_info['percentage %']>=40]['Cols_name'].to_list()
df_app_msng_rmd = df_app.drop(labels=missing_cols,axis=1)
```

In [11]:
```
#After removing unwanted colnums
df_app_msng_rmd.shape
```

Out[11]: (307511, 73)

In [12]:
```
df_app_msng_rmd.head()
```

Out[12]:

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

In [13]:
```
flag_cols = []

for i in df_app_msng_rmd.columns:
    if i.startswith('FLAG_'):
        flag_cols.append(i)

len(flag_cols)
```

Out[13]: 28

In [14]:
```
flag_target_col = df_app_msng_rmd[flag_cols+["TARGET"]].head()
flag_target_col.head()
```

Out[14]:

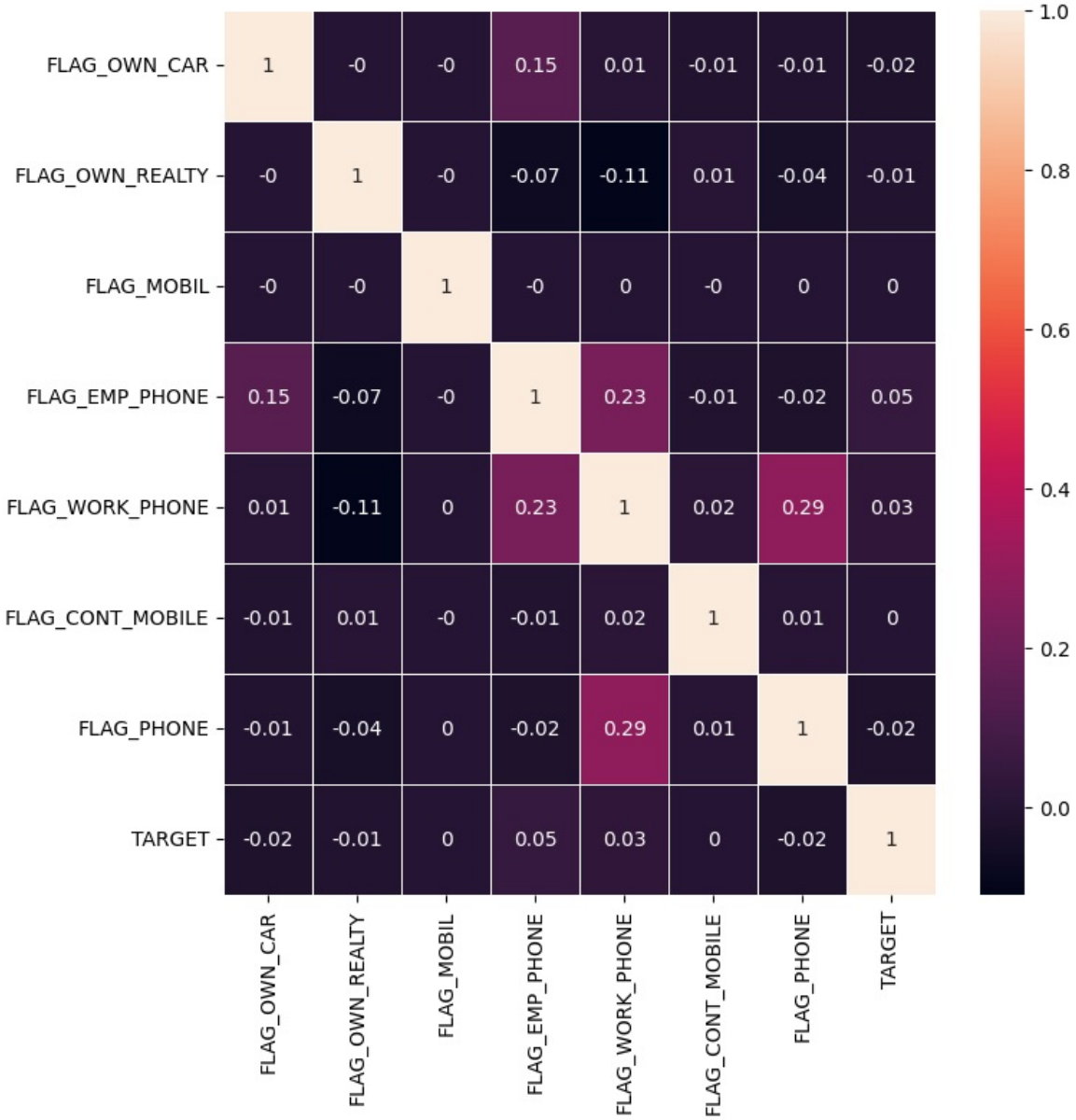
	FLAG_OWN_CAR	FLAG_OWN_REALTY	FLAG_MOBIL	FLAG_EMP_PHONE	FLAG_WORK_PHONE	FLAG_CONT_MOBILE	FLAG_
0	N	Y	1	1	0	1	
1	N	N	1	1	0	1	
2	Y	Y	1	1	1	1	
3	N	Y	1	1	0	1	
4	N	Y	1	1	0	1	

In [15]:
```
plt.figure(figsize=(20,25))
```

```
for i,col in enumerate(flag_cols):
    plt.subplot(7,4,i+1)

    sns.countplot(data=flag_target_col,x=col,hue="TARGET");
```

In [16]:
```
flag_corr = ['FLAG_OWN_CAR','FLAG_OWN_REALTY','FLAG_MOBIL','FLAG_EMP_PHONE','FLAG_WORK_PHONE','FLAG_CONT_MOBILE
flag_corr_df = df_app_msng_rmd[flag_corr]


flag_corr_df['FLAG_OWN_CAR'] =flag_corr_df['FLAG_OWN_CAR'].replace(['N','Y'],[0,1])
flag_corr_df['FLAG_OWN_REALTY'] =flag_corr_df['FLAG_OWN_REALTY'].replace(['N','Y'],[0,1])

plt.figure(figsize=(8,8))
sns.heatmap(round(flag_corr_df.corr(),2),linewidths=.5,annot=True);
```

```
C:\Users\Hariram\AppData\Local\Temp\ipykernel_2656\669372860.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#retu
rning-a-view-versus-a-copy
  flag_corr_df['FLAG_OWN_CAR'] =flag_corr_df['FLAG_OWN_CAR'].replace(['N','Y'],[0,1])
C:\Users\Hariram\AppData\Local\Temp\ipykernel_2656\669372860.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#retu
rning-a-view-versus-a-copy
  flag_corr_df['FLAG_OWN_REALTY'] =flag_corr_df['FLAG_OWN_REALTY'].replace(['N','Y'],[0,1])
```



In [17]:
```
df_app_flag_rmd = df_app_msng_rmd.drop(labels=flag_cols,axis=1)
df_app_flag_rmd.shape
```

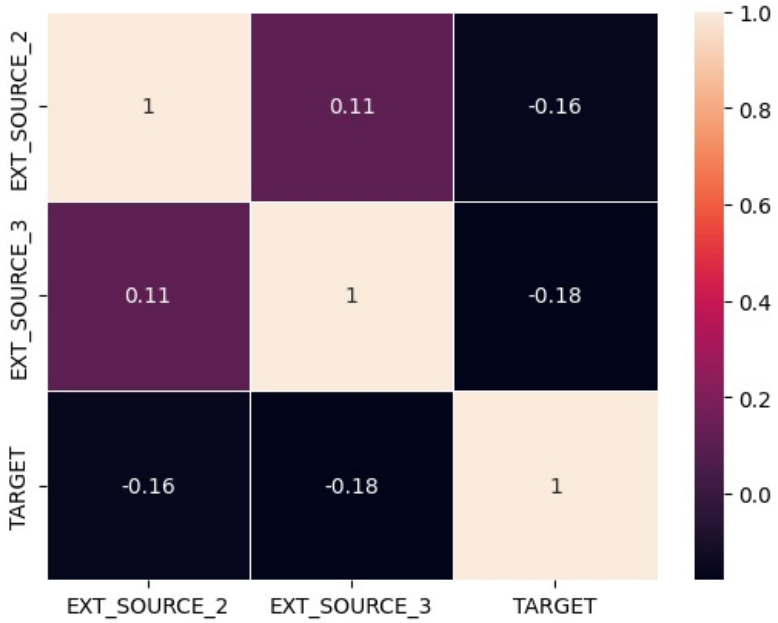
Out[17]: (307511, 45)

In [18]:
```
df_app_flag_rmd.head()
```

Out[18]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_A
0	100002	1	Cash loans	M	0	202500.0	406597.5	
1	100003	0	Cash loans	F	0	270000.0	1293502.5	
2	100004	0	Revolving loans	M	0	67500.0	135000.0	
3	100006	0	Cash loans	F	0	135000.0	312682.5	
4	100007	0	Cash loans	M	0	121500.0	513000.0	

In [19]:
```
sns.heatmap(data=round(df_app_flag_rmd[['EXT_SOURCE_2','EXT_SOURCE_3','TARGET']].corr(),2),linewidths=.5,annot='
```



In [20]:
```
df_app_score_rmd = df_app_flag_rmd.drop(['EXT_SOURCE_2','EXT_SOURCE_3'],axis=1)
df_app_score_rmd.shape
```

Out[20]: (307511, 43)

Feature Enginnering

In [21]:
```
df_app_score_rmd.isnull().sum().sort_values()/df_app_score_rmd.shape[0]
```

Out[21]:
```
SK_ID_CURR                    0.000000
ORGANIZATION_TYPE             0.000000
LIVE_CITY_NOT_WORK_CITY       0.000000
REG_CITY_NOT_WORK_CITY        0.000000
REG_CITY_NOT_LIVE_CITY        0.000000
LIVE_REGION_NOT_WORK_REGION   0.000000
REG_REGION_NOT_WORK_REGION    0.000000
REG_REGION_NOT_LIVE_REGION    0.000000
HOUR_APPR_PROCESS_START       0.000000
WEEKDAY_APPR_PROCESS_START    0.000000
REGION_RATING_CLIENT_W_CITY   0.000000
DAYS_ID_PUBLISH               0.000000
DAYS_REGISTRATION             0.000000
DAYS_EMPLOYED                 0.000000
DAYS_BIRTH                    0.000000
REGION_RATING_CLIENT          0.000000
NAME_HOUSING_TYPE             0.000000
TARGET                        0.000000
NAME_CONTRACT_TYPE            0.000000
REGION_POPULATION_RELATIVE    0.000000
CNT_CHILDREN                  0.000000
AMT_INCOME_TOTAL              0.000000
AMT_CREDIT                    0.000000
CODE_GENDER                   0.000000
NAME_INCOME_TYPE              0.000000
NAME_EDUCATION_TYPE           0.000000
NAME_FAMILY_STATUS            0.000000
DAYS_LAST_PHONE_CHANGE        0.000003
CNT_FAM_MEMBERS               0.000007
AMT_ANNUITY                   0.000039
AMT_GOODS_PRICE               0.000904
DEF_60_CNT_SOCIAL_CIRCLE      0.003320
OBS_60_CNT_SOCIAL_CIRCLE      0.003320
DEF_30_CNT_SOCIAL_CIRCLE      0.003320
OBS_30_CNT_SOCIAL_CIRCLE      0.003320
NAME_TYPE_SUITE               0.004201
AMT_REQ_CREDIT_BUREAU_QRT     0.135016
AMT_REQ_CREDIT_BUREAU_HOUR    0.135016
AMT_REQ_CREDIT_BUREAU_DAY     0.135016
AMT_REQ_CREDIT_BUREAU_WEEK    0.135016
AMT_REQ_CREDIT_BUREAU_MON     0.135016
AMT_REQ_CREDIT_BUREAU_YEAR    0.135016
OCCUPATION_TYPE               0.313455
dtype: float64
```

Missing imputation

In [22]:
```
df_app_score_rmd['CNT_FAM_MEMBERS'] = df_app_score_rmd['CNT_FAM_MEMBERS'].fillna((df_app_score_rmd['CNT_FAM_MEMI
```

In [23]:
```
df_app_score_rmd['CNT_FAM_MEMBERS'].isnull().sum()
```

Out[23]:
```
0
```

In [24]:
```
df_app_score_rmd["OCCUPATION_TYPE"] = df_app_score_rmd['OCCUPATION_TYPE'].fillna((df_app_score_rmd['OCCUPATION_T
```

In [25]:
```
df_app_score_rmd['OCCUPATION_TYPE'].isnull().sum()
```

Out[25]:
```
0
```

In [26]:
```
df_app_score_rmd["NAME_TYPE_SUITE"] = df_app_score_rmd['NAME_TYPE_SUITE'].fillna((df_app_score_rmd['NAME_TYPE_SU
```

In [27]:
```
df_app_score_rmd['NAME_TYPE_SUITE'].isnull().sum()
```

Out[27]:
```
0
```

In [28]:
```
df_app_score_rmd["AMT_ANNUITY"] = df_app_score_rmd['AMT_ANNUITY'].fillna((df_app_score_rmd['AMT_ANNUITY'].mean(
```

In [29]:
```
df_app_score_rmd['AMT_ANNUITY'].isnull().sum()
```

Out[29]:
```
0
```

In [30]:
```
amt_req_col = []

for col in df_app_score_rmd.columns:
    if col.startswith("AMT_REQ_CREDIT_BUREAU"):
       amt_req_col.append(col)

amt_req_col
```

Out[30]:
```
['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']
```

In [31]:
```
for col in amt_req_col:
    df_app_score_rmd[col] = df_app_score_rmd[col].fillna((df_app_score_rmd[col].median()))
```

In [32]:
```
df_app_score_rmd[col].isnull().sum()
```

Out[32]:
```
0
```

In [33]:
```
df_app_score_rmd.isnull().sum().sort_values()
```

Out[33]:
```
SK_ID_CURR                       0
AMT_REQ_CREDIT_BUREAU_QRT        0
REGION_RATING_CLIENT_W_CITY      0
WEEKDAY_APPR_PROCESS_START       0
HOUR_APPR_PROCESS_START          0
REG_REGION_NOT_LIVE_REGION       0
REG_REGION_NOT_WORK_REGION       0
LIVE_REGION_NOT_WORK_REGION      0
REG_CITY_NOT_LIVE_CITY           0
REG_CITY_NOT_WORK_CITY           0
LIVE_CITY_NOT_WORK_CITY          0
ORGANIZATION_TYPE                0
AMT_REQ_CREDIT_BUREAU_HOUR       0
AMT_REQ_CREDIT_BUREAU_DAY        0
AMT_REQ_CREDIT_BUREAU_WEEK       0
AMT_REQ_CREDIT_BUREAU_MON        0
CNT_FAM_MEMBERS                  0
OCCUPATION_TYPE                  0
REGION_RATING_CLIENT             0
DAYS_REGISTRATION                0
TARGET                           0
NAME_CONTRACT_TYPE               0
CODE_GENDER                      0
CNT_CHILDREN                     0
AMT_INCOME_TOTAL                 0
DAYS_ID_PUBLISH                  0
AMT_ANNUITY                      0
AMT_CREDIT                       0
NAME_INCOME_TYPE                 0
NAME_EDUCATION_TYPE              0
NAME_FAMILY_STATUS               0
NAME_HOUSING_TYPE                0
REGION_POPULATION_RELATIVE       0
DAYS_BIRTH                       0
DAYS_EMPLOYED                    0
NAME_TYPE_SUITE                  0
AMT_REQ_CREDIT_BUREAU_YEAR       0
DAYS_LAST_PHONE_CHANGE           1
AMT_GOODS_PRICE                278
OBS_30_CNT_SOCIAL_CIRCLE      1021
DEF_30_CNT_SOCIAL_CIRCLE      1021
OBS_60_CNT_SOCIAL_CIRCLE      1021
DEF_60_CNT_SOCIAL_CIRCLE      1021
dtype: int64
```

In [34]:
```
df_app_score_rmd["AMT_GOODS_PRICE"] = df_app_score_rmd['AMT_GOODS_PRICE'].fillna((df_app_score_rmd['AMT_GOODS_PI
```

In [35]:
```
df_app_score_rmd["AMT_GOODS_PRICE"].isnull().sum()
```

Out[35]:
```
0
```

Value modification

In [36]:
```
days_col = []

for col in df_app_score_rmd.columns:
    if col.startswith("DAYS"):
        days_col.append(col)

days_col
```

Out[36]:
```
['DAYS_BIRTH',
 'DAYS_EMPLOYED',
 'DAYS_REGISTRATION',
 'DAYS_ID_PUBLISH',
 'DAYS_LAST_PHONE_CHANGE']
```

In [37]:
```
for col in days_col:
    df_app_score_rmd[col] = abs(df_app_score_rmd[col])
```

In [38]:
```
df_app_score_rmd.head()
```

Out[38]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_A
0	100002	1	Cash loans	M	0	202500.0	406597.5	
1	100003	0	Cash loans	F	0	270000.0	1293502.5	
2	100004	0	Revolving loans	M	0	67500.0	135000.0	
3	100006	0	Cash loans	F	0	135000.0	312682.5	
4	100007	0	Cash loans	M	0	121500.0	513000.0	

In [39]:
```
df_app_score_rmd.nunique().sort_values()
```

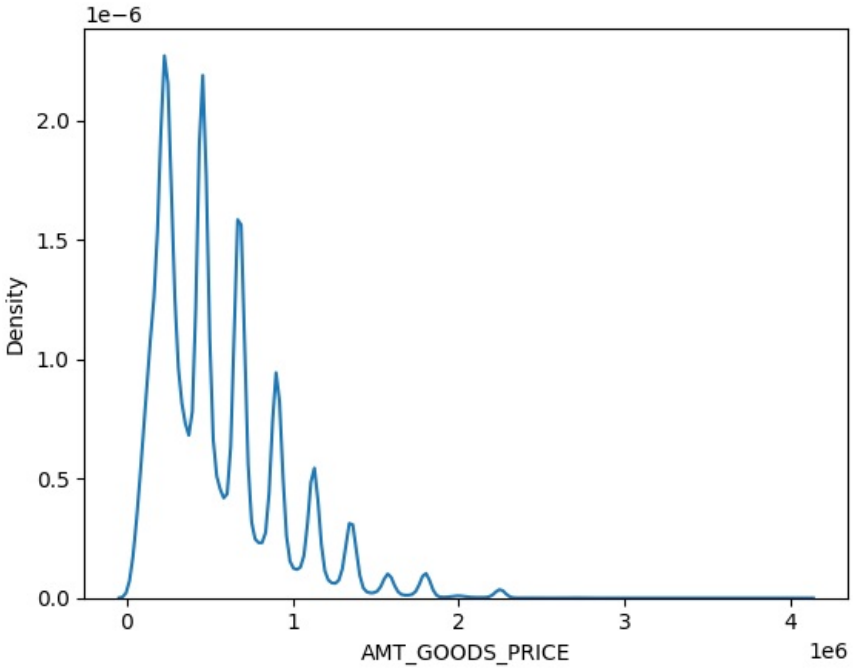
Out[39]:
```
LIVE_REGION_NOT_WORK_REGION         2
TARGET                              2
NAME_CONTRACT_TYPE                  2
REG_REGION_NOT_LIVE_REGION          2
REG_CITY_NOT_LIVE_CITY              2
REG_CITY_NOT_WORK_CITY              2
LIVE_CITY_NOT_WORK_CITY             2
REG_REGION_NOT_WORK_REGION          2
REGION_RATING_CLIENT_W_CITY         3
REGION_RATING_CLIENT                3
CODE_GENDER                         3
NAME_EDUCATION_TYPE                 5
AMT_REQ_CREDIT_BUREAU_HOUR          5
NAME_HOUSING_TYPE                   6
NAME_FAMILY_STATUS                  6
WEEKDAY_APPR_PROCESS_START          7
NAME_TYPE_SUITE                     7
NAME_INCOME_TYPE                    8
AMT_REQ_CREDIT_BUREAU_DAY           9
DEF_60_CNT_SOCIAL_CIRCLE            9
AMT_REQ_CREDIT_BUREAU_WEEK          9
DEF_30_CNT_SOCIAL_CIRCLE           10
AMT_REQ_CREDIT_BUREAU_QRT          11
CNT_CHILDREN                       15
CNT_FAM_MEMBERS                    17
OCCUPATION_TYPE                    18
HOUR_APPR_PROCESS_START            24
AMT_REQ_CREDIT_BUREAU_MON          24
AMT_REQ_CREDIT_BUREAU_YEAR         25
OBS_30_CNT_SOCIAL_CIRCLE           33
OBS_60_CNT_SOCIAL_CIRCLE           33
ORGANIZATION_TYPE                  58
REGION_POPULATION_RELATIVE         81
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
dtype: int64
```

In [40]:
```
df_app_score_rmd["OBS_30_CNT_SOCIAL_CIRCLE"].unique()
```

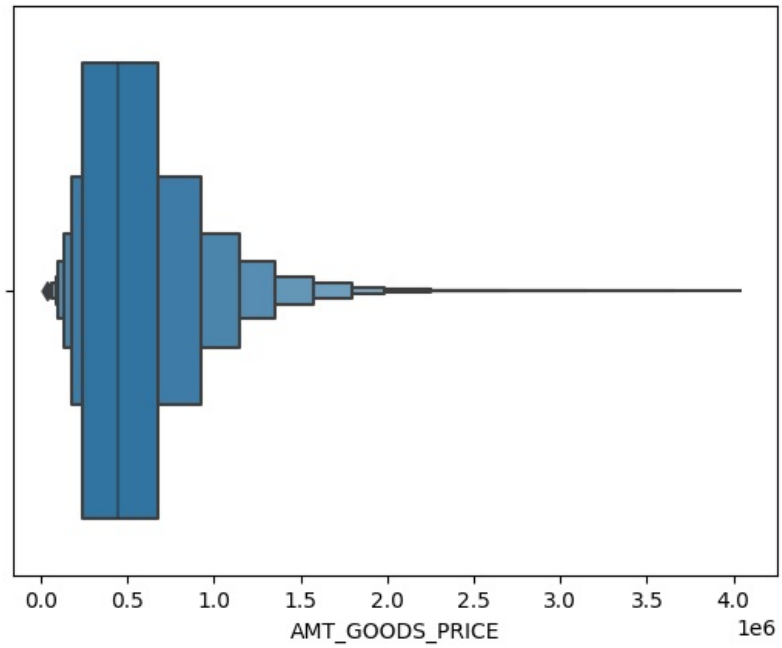
Out[40]:
```
array([  2.,   1.,   0.,   4.,   8.,  10.,  nan,   7.,   3.,   6.,   5.,
        12.,   9.,  13.,  11.,  14.,  22.,  16.,  15.,  17.,  20.,  25.,
        19.,  18.,  21.,  24.,  23.,  28.,  26.,  29.,  27.,  47., 348.,
        30.])
```

Outlier detection & treatment

In [41]:
```
sns.kdeplot(data=df_app_score_rmd,x="AMT_GOODS_PRICE");
```



In [42]:
```
sns.boxenplot(data=df_app_score_rmd,x="AMT_GOODS_PRICE");
```



In [43]:
```
df_app_score_rmd["AMT_GOODS_PRICE"].quantile([0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,0.99])
```

Out[43]:
```
0.10     180000.0
0.20     225000.0
0.30     270000.0
0.40     378000.0
0.50     450000.0
0.60     522000.0
0.70     675000.0
0.80     814500.0
0.90    1093500.0
0.99    1800000.0
Name: AMT_GOODS_PRICE, dtype: float64
```

In [44]:
```
bins = [0 , 100000 , 200000 , 300000 , 400000 , 500000 , 600000 , 700000 , 800000 , 900000 , 4050000]
ranges = ['0k-100k' , '100k-200k' , '200k-300k' , '300k-400k' , '400k-500k' , '500k-600k' , '600k-700k' , '700k
df_app_score_rmd['AMT_GOODS_PRICE_RANGE'] = pd.cut(df_app_score_rmd["AMT_GOODS_PRICE"],bins,labels=ranges)
```

In [45]:
```
df_app_score_rmd.groupby(['AMT_GOODS_PRICE_RANGE']).size()
```

Out[45]:
```
AMT_GOODS_PRICE_RANGE
0k-100k        8709
100k-200k     32956
200k-300k     62761
300k-400k     21219
400k-500k     57251
500k-600k     13117
600k-700k     40024
700k-800k      8110
800k-900k     21484
Above 900k    41880
dtype: int64
```

In [46]:
```
df_app_score_rmd["AMT_GOODS_PRICE_RANGE"].isnull().sum()
```

Out[46]:
```
0
```

In [47]:
```
df_app_score_rmd["AMT_INCOME_TOTAL"].quantile([0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,0.99])
```

Out[47]:
```
0.10     81000.0
0.20     99000.0
0.30    112500.0
0.40    135000.0
0.50    147150.0
0.60    162000.0
0.70    180000.0
0.80    225000.0
0.90    270000.0
0.99    472500.0
Name: AMT_INCOME_TOTAL, dtype: float64
```

In [48]:
```
df_app_score_rmd["AMT_INCOME_TOTAL"].max()
```

Out[48]:
```
117000000.0
```

In [49]:
```
bins = [0 , 100000 ,150000, 200000 ,250000, 300000 ,350000, 400000 ,117000000 ]
ranges = ['0k-100k' , '100k-150k' , '150k-200k','200k-250k' ,'250k-300k' , '300k-350k','350k-400k' , 'Above 400
df_app_score_rmd['AMT_INCOME_TOTAL_RANGE'] = pd.cut(df_app_score_rmd["AMT_INCOME_TOTAL"],bins,labels=ranges)
```

In [50]:
```
df_app_score_rmd.groupby(['AMT_INCOME_TOTAL_RANGE']).size()
```

Out[50]:
```
AMT_INCOME_TOTAL_RANGE
0k-100k       63698
100k-150k     91591
150k-200k     64307
200k-250k     48137
250k-300k     17039
300k-350k      8874
350k-400k      5802
Above 400k     8063
dtype: int64
```

In [51]:
```
df_app_score_rmd["AMT_INCOME_TOTAL_RANGE"].isnull().sum()
```

Out[51]:
```
0
```

In [52]:
```
df_app_score_rmd["AMT_CREDIT"].max()
```

Out[52]:
```
4050000.0
```

In [53]:
```
df_app_score_rmd["AMT_CREDIT"].quantile([0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,0.99])
```

Out[53]:
```
0.10     180000.0
0.20     254700.0
0.30     306306.0
0.40     432000.0
0.50     513531.0
0.60     604152.0
0.70     755190.0
0.80     900000.0
0.90    1133748.0
0.99    1854000.0
Name: AMT_CREDIT, dtype: float64
```

In [54]:
```
bins = [0 , 200000 , 400000 , 600000 , 800000 , 900000 , 1000000 ,4050000.0]
ranges = ['0k-200k' ,  '200k-400k' , '400k-600k' , '600k-800k' , '800k-900k' , '900k-1M' , 'Above 1M']
df_app_score_rmd['AMT_CREDIT_RANGE'] = pd.cut(df_app_score_rmd["AMT_CREDIT"],bins,labels=ranges)
```

In [55]:
```
df_app_score_rmd.groupby(['AMT_CREDIT_RANGE']).size()
```

Out[55]:
```
AMT_CREDIT_RANGE
0k-200k      36144
200k-400k    81151
400k-600k    66270
600k-800k    43242
800k-900k    21792
900k-1M       8927
Above 1M     49985
dtype: int64
```

In [56]:
```
df_app_score_rmd["AMT_CREDIT"].isnull().sum()
```

Out[56]:
```
0
```

In [57]:
```
df_app_score_rmd["AMT_ANNUITY"].quantile([0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,0.99])
```

Out[57]:
```
0.10    11074.5
0.20    14701.5
0.30    18189.0
0.40    21870.0
0.50    24903.0
0.60    28062.0
0.70    32004.0
0.80    37516.5
0.90    45954.0
0.99    70006.5
Name: AMT_ANNUITY, dtype: float64
```

In [58]:
```
df_app_score_rmd["AMT_ANNUITY"].max()
```

Out[58]:
```
258025.5
```

In [59]:
```
bins = [0 , 25000, 50000 , 100000 , 150000 , 200000 , 258025.5]
ranges = ['0k-25k','25k-50k' , '50k-100k' ,'100k-150k' , '150k-200k', 'Above 200k']
df_app_score_rmd['AMT_ANNUITY_RANGE'] = pd.cut(df_app_score_rmd['AMT_ANNUITY'],bins,labels=ranges)
```

In [60]:
```
df_app_score_rmd.groupby(['AMT_ANNUITY_RANGE']).size()
```

Out[60]:
```
AMT_ANNUITY_RANGE
0k-25k        154867
25k-50k       131347
50k-100k       20792
100k-150k        437
150k-200k         32
Above 200k        36
dtype: int64
```

In [61]:
```
df_app_score_rmd["AMT_ANNUITY_RANGE"].isnull().sum()
```

Out[61]:
```
0
```

In [62]:
```
df_app_score_rmd['DAYS_EMPLOYED'].agg(['min','max','median'])
```

Out[62]:
```
min            0.0
max       365243.0
median      2219.0
Name: DAYS_EMPLOYED, dtype: float64
```

In [63]:
```
df_app_score_rmd["DAYS_EMPLOYED"].quantile([0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,0.99])
```

Out[63]:
```
0.10       392.0
0.20       749.0
0.30      1132.0
0.40      1597.0
0.50      2219.0
0.60      3032.0
0.70      4435.0
0.80      9188.0
0.90    365243.0
0.99    365243.0
Name: DAYS_EMPLOYED, dtype: float64
```

In [64]:
```
df_app_score_rmd["DAYS_EMPLOYED"].max()
```

Out[64]:
```
365243
```

In [65]:
```
bins = [0 , 1825 , 3650 , 5475 , 7300 , 9125 , 10950 , 12775 , 14600 , 16425 , 18250 , 365243]
ranges = ['0-5Y','5Y-10Y' , '10Y-15Y' ,'15Y-20Y' , '20Y-25Y','25Y-30Y','30Y-35Y','35Y-40Y','40Y-45Y','45Y-50Y'
df_app_score_rmd['DAYS_EMPLOYED_RANGE'] = pd.cut(df_app_score_rmd['DAYS_EMPLOYED'],bins,labels=ranges)
```

In [66]:
```
df_app_score_rmd.groupby(['DAYS_EMPLOYED_RANGE']).size()
```

Out[66]:
```
DAYS_EMPLOYED_RANGE
0-5Y         136309
5Y-10Y        64872
10Y-15Y       27549
15Y-20Y       10849
20Y-25Y        6243
25Y-30Y        3308
30Y-35Y        1939
35Y-40Y         832
40Y-45Y         210
45Y-50Y          24
Above 50Y     55374
dtype: int64
```

In [67]:
```
df_app_score_rmd['DAYS_BIRTH'].min()
```

Out[67]:
```
7489
```

In [68]:
```
bins = [0 , 7300 , 10950 , 14600 , 18250 , 21900 , 25229]
ranges = ['20Y' , '20Y-30Y' , '30Y-40Y' , '40Y-50Y' ,'50Y-60Y' ,'Above 60Y']
df_app_score_rmd['DAYS_BIRTH_RANGE'] = pd.cut(df_app_score_rmd['DAYS_BIRTH'],bins,labels=ranges)
```

In [69]:
```
df_app_score_rmd.groupby(['DAYS_BIRTH_RANGE']).size()
```

Out[69]:
```
DAYS_BIRTH_RANGE
20Y              0
20Y-30Y      45021
30Y-40Y      82308
40Y-50Y      76541
50Y-60Y      68062
Above 60Y    35579
dtype: int64
```

In [70]:
```
df_app_score_rmd["DAYS_BIRTH_RANGE"].isnull().sum()
```

Out[70]:
```
0
```

Data Analysis

In [71]:
```
df_app_score_rmd.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Data columns (total 49 columns):
 #   Column                       Non-Null Count   Dtype
---  ------                       --------------   -----
 0   SK_ID_CURR                   307511 non-null  int64
 1   TARGET                       307511 non-null  int64
 2   NAME_CONTRACT_TYPE           307511 non-null  object
 3   CODE_GENDER                  307511 non-null  object
 4   CNT_CHILDREN                 307511 non-null  int64
 5   AMT_INCOME_TOTAL             307511 non-null  float64
 6   AMT_CREDIT                   307511 non-null  float64
 7   AMT_ANNUITY                  307511 non-null  float64
 8   AMT_GOODS_PRICE              307511 non-null  float64
 9   NAME_TYPE_SUITE              307511 non-null  object
 10  NAME_INCOME_TYPE             307511 non-null  object
 11  NAME_EDUCATION_TYPE          307511 non-null  object
 12  NAME_FAMILY_STATUS           307511 non-null  object
 13  NAME_HOUSING_TYPE            307511 non-null  object
 14  REGION_POPULATION_RELATIVE   307511 non-null  float64
 15  DAYS_BIRTH                   307511 non-null  int64
 16  DAYS_EMPLOYED                307511 non-null  int64
 17  DAYS_REGISTRATION            307511 non-null  float64
 18  DAYS_ID_PUBLISH              307511 non-null  int64
 19  OCCUPATION_TYPE              307511 non-null  object
 20  CNT_FAM_MEMBERS              307511 non-null  float64
 21  REGION_RATING_CLIENT         307511 non-null  int64
 22  REGION_RATING_CLIENT_W_CITY  307511 non-null  int64
 23  WEEKDAY_APPR_PROCESS_START   307511 non-null  object
 24  HOUR_APPR_PROCESS_START      307511 non-null  int64
 25  REG_REGION_NOT_LIVE_REGION   307511 non-null  int64
 26  REG_REGION_NOT_WORK_REGION   307511 non-null  int64
 27  LIVE_REGION_NOT_WORK_REGION  307511 non-null  int64
 28  REG_CITY_NOT_LIVE_CITY       307511 non-null  int64
 29  REG_CITY_NOT_WORK_CITY       307511 non-null  int64
 30  LIVE_CITY_NOT_WORK_CITY      307511 non-null  int64
 31  ORGANIZATION_TYPE            307511 non-null  object
 32  OBS_30_CNT_SOCIAL_CIRCLE     306490 non-null  float64
 33  DEF_30_CNT_SOCIAL_CIRCLE     306490 non-null  float64
 34  OBS_60_CNT_SOCIAL_CIRCLE     306490 non-null  float64
 35  DEF_60_CNT_SOCIAL_CIRCLE     306490 non-null  float64
 36  DAYS_LAST_PHONE_CHANGE       307510 non-null  float64
 37  AMT_REQ_CREDIT_BUREAU_HOUR   307511 non-null  float64
 38  AMT_REQ_CREDIT_BUREAU_DAY    307511 non-null  float64
 39  AMT_REQ_CREDIT_BUREAU_WEEK   307511 non-null  float64
 40  AMT_REQ_CREDIT_BUREAU_MON    307511 non-null  float64
 41  AMT_REQ_CREDIT_BUREAU_QRT    307511 non-null  float64
 42  AMT_REQ_CREDIT_BUREAU_YEAR   307511 non-null  float64
 43  AMT_GOODS_PRICE_RANGE        307511 non-null  category
 44  AMT_INCOME_TOTAL_RANGE       307511 non-null  category
 45  AMT_CREDIT_RANGE             307511 non-null  category
 46  AMT_ANNUITY_RANGE            307511 non-null  category
 47  DAYS_EMPLOYED_RANGE          307509 non-null  category
 48  DAYS_BIRTH_RANGE             307511 non-null  category
dtypes: category(6), float64(18), int64(15), object(10)
memory usage: 102.6+ MB
```

In [72]:
```
df_app_score_rmd.dtypes.value_counts()
```

Out[72]:
```
float64     18
int64       15
object      10
category     1
category     1
category     1
category     1
category     1
category     1
Name: count, dtype: int64
```

In [73]:
```
obj_var = df_app_score_rmd.select_dtypes(include=['object']).columns
obj_var
```

Out[73]:
```
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 [74]:
```
df_app_score_rmd.select_dtypes(include=['object']).head()
```

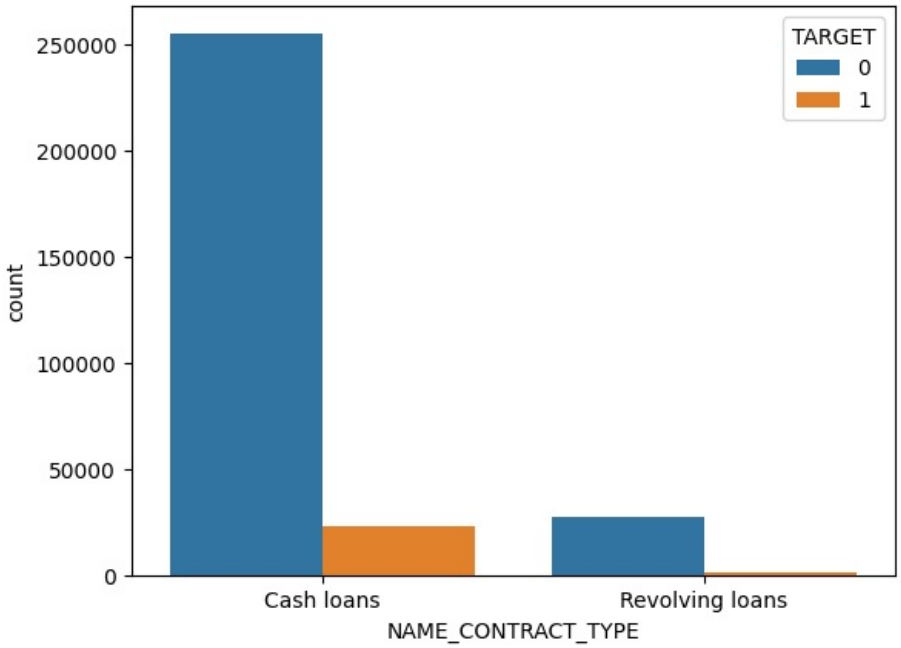
Out[74]:

	NAME_CONTRACT_TYPE	CODE_GENDER	NAME_TYPE_SUITE	NAME_INCOME_TYPE	NAME_EDUCATION_TYPE	NAME_FAMILY_
0	Cash loans	M	Unaccompanied	Working	Secondary / secondary special	Single / n
1	Cash loans	F	Family	State servant	Higher education	
2	Revolving loans	M	Unaccompanied	Working	Secondary / secondary special	Single / n
3	Cash loans	F	Unaccompanied	Working	Secondary / secondary special	Civil
4	Cash loans	M	Unaccompanied	Working	Secondary / secondary special	Single / n

In [75]:
```
df_app_score_rmd.groupby(['NAME_CONTRACT_TYPE']).size()
```

Out[75]:
```
NAME_CONTRACT_TYPE
Cash loans         278232
Revolving loans     29279
dtype: int64
```

In [76]:
```
sns.countplot(data=df_app_score_rmd,x='NAME_CONTRACT_TYPE',hue='TARGET');
```



In [77]:
```
data_pct = df_app_score_rmd[['NAME_CONTRACT_TYPE' , 'TARGET']].groupby(['NAME_CONTRACT_TYPE'],as_index=False).m
```

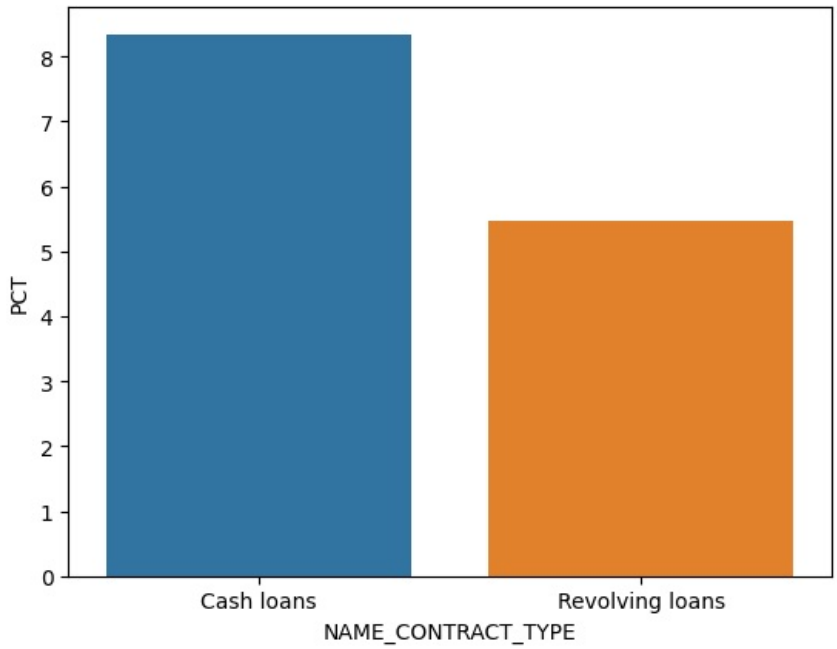
In [78]:
```
data_pct['PCT'] = data_pct['TARGET'] * 100
```

In [79]:
```
data_pct
```

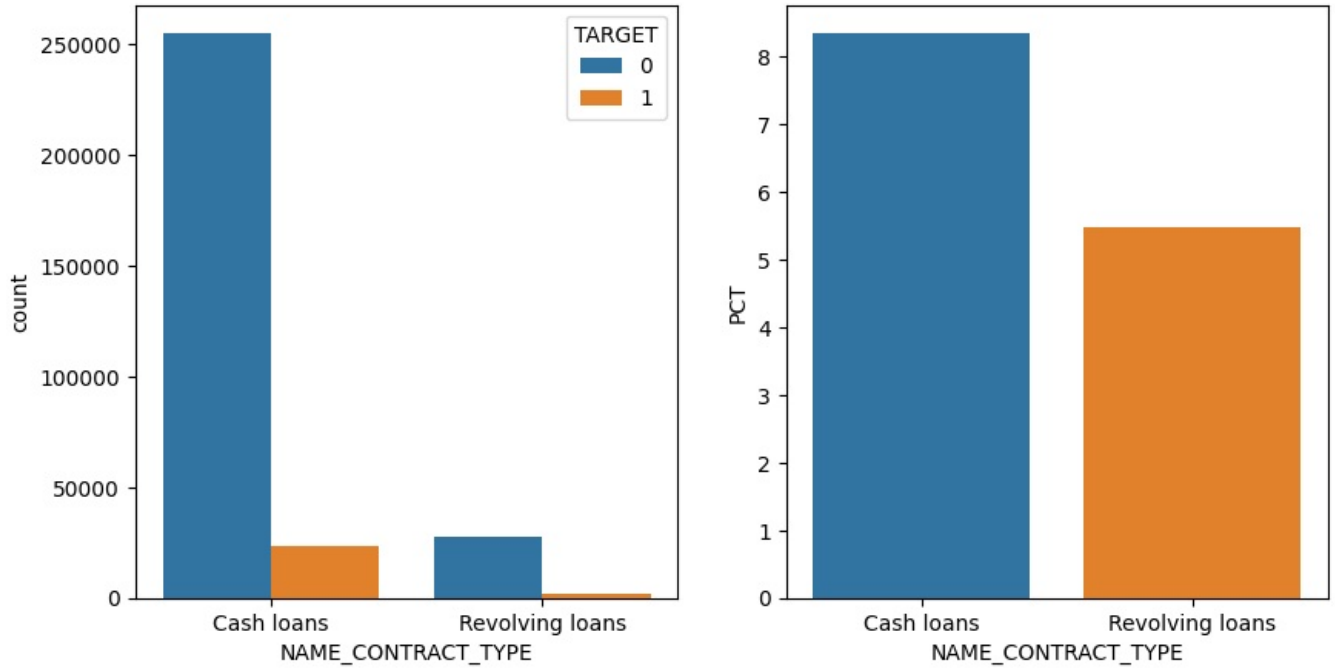
Out[79]:

	NAME_CONTRACT_TYPE	TARGET	PCT
0	Cash loans	0.083459	8.345913
1	Revolving loans	0.054783	5.478329

In [80]:
```
sns.barplot(data=data_pct,x='NAME_CONTRACT_TYPE',y='PCT');
```

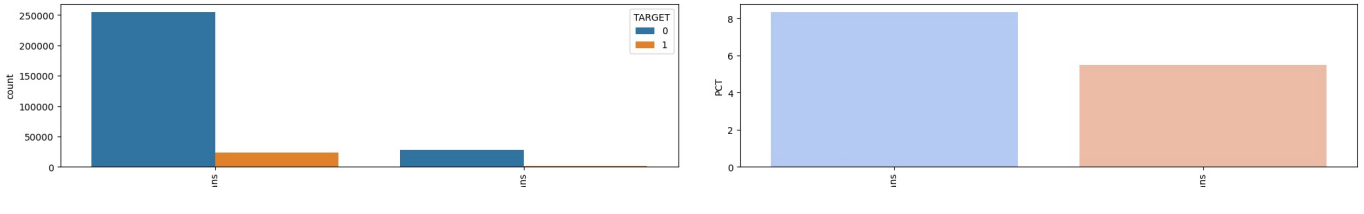


In [81]:
```
plt.figure(figsize=(10,5))
plt.subplot(1,2,1)
sns.countplot(data=df_app_score_rmd,x='NAME_CONTRACT_TYPE',hue='TARGET');
plt.subplot(1,2,2)
sns.barplot(data=data_pct,x='NAME_CONTRACT_TYPE',y='PCT');
```



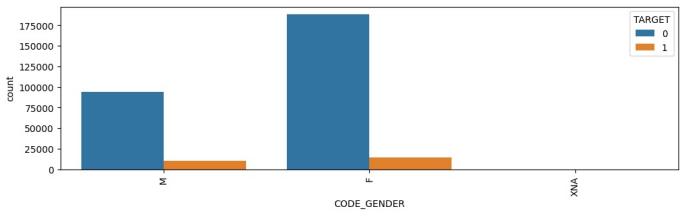
most of customers have taken cash loan.

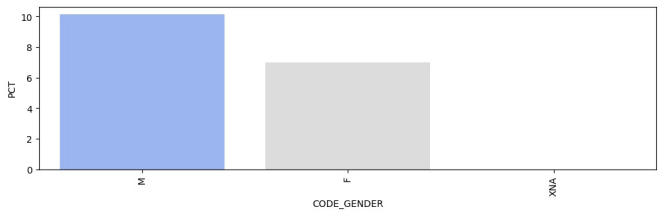
In [82]:
```
plt.figure(figsize=(25,60))

for i,var in enumerate(obj_var):
    data_pct = df_app_score_rmd[[var , 'TARGET']].groupby([var],as_index=False).mean().sort_values(by='TARGET',a
    data_pct['PCT'] = data_pct['TARGET'] * 100


    plt.subplot(10,2,i+i+1)
    plt.subplots_adjust(wspace=0.1,hspace=1)
    sns.countplot(data=df_app_score_rmd,x=var,hue='TARGET');
    plt.xticks(rotation=90)
    plt.subplot(10,2,i+i+2)
    sns.barplot(data=data_pct,x=var,y='PCT',palette='coolwarm');
    plt.xticks(rotation=90)
```

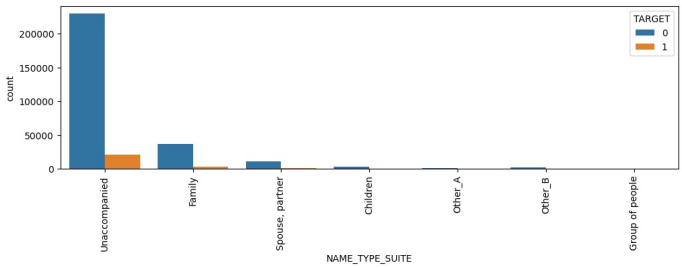


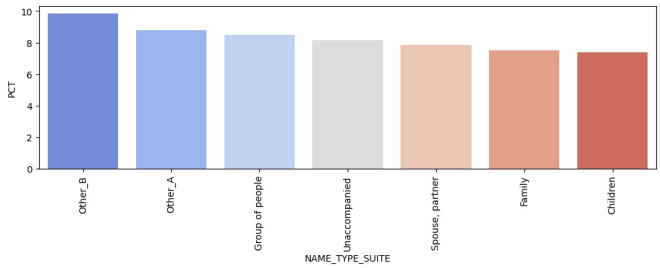

Cash loa
Revolving loa
NAME_CONTRACT_TYPE

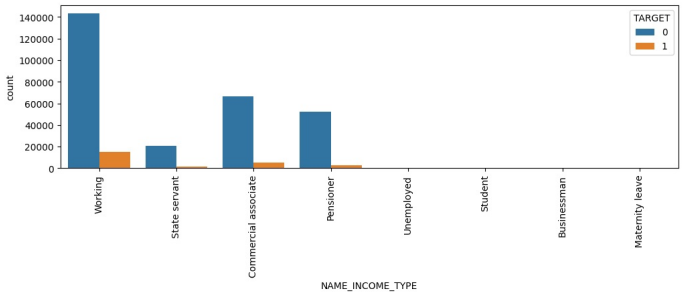

Cash loa
Revolving loa
NAME_CONTRACT_TYPE

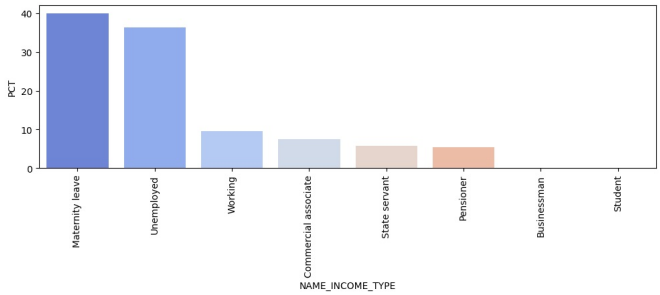

TARGET
0
1
count
M
F
XNA
CODE_GENDER

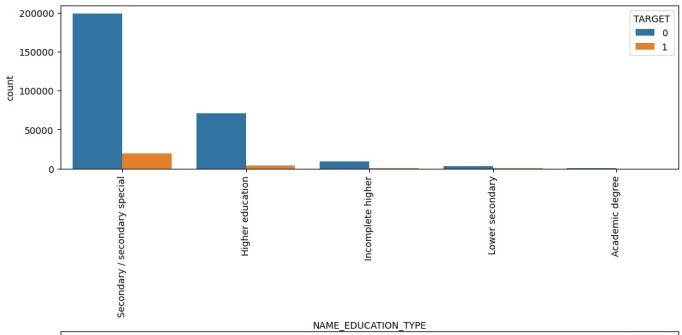

PCT
M
F
XNA
CODE_GENDER

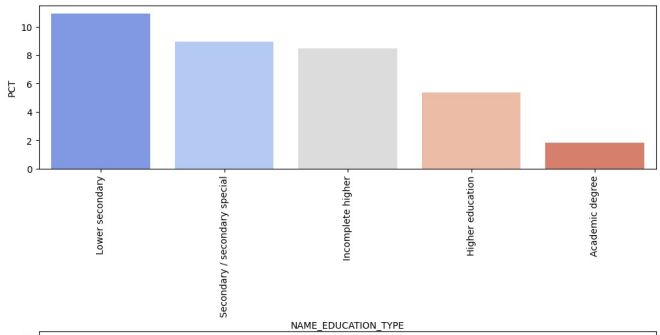

TARGET
0
1
count
Unaccompanied
Family
Spouse, partner
Children
Other_A
Other_B
Group of people
NAME_TYPE_SUITE

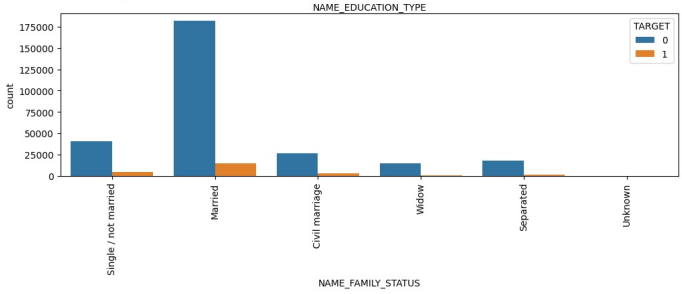

PCT
Other_B
Other_A
Group of people
Unaccompanied
Spouse, partner
Family
Children
NAME_TYPE_SUITE

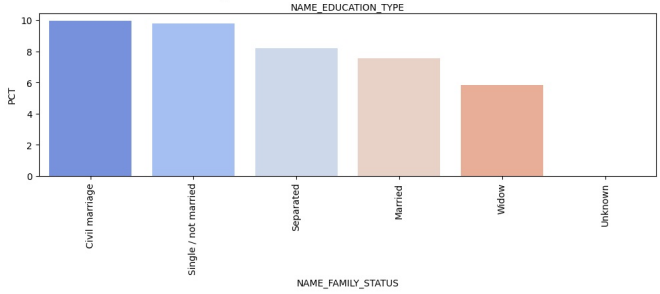

TARGET
0
1
count
Working
State servant
Commercial associate
Pensioner
Unemployed
Student
Businessman
Maternity leave
NAME_INCOME_TYPE

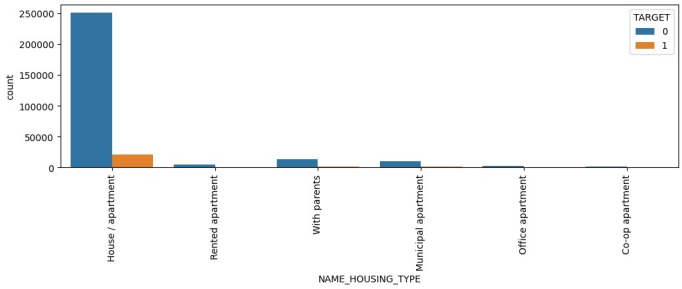

PCT
Maternity leave
Unemployed
Working
Commercial associate
State servant
Pensioner
Businessman
Student
NAME_INCOME_TYPE

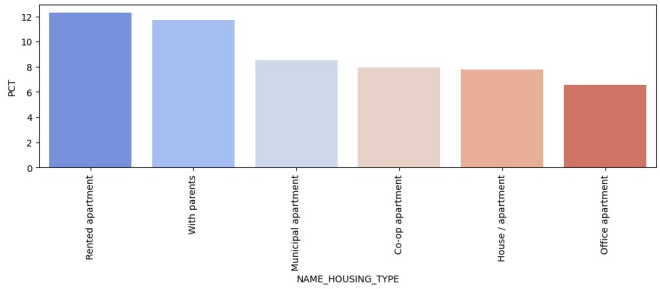

TARGET
0
1
count
Secondary / secondary special
Higher education
Incomplete higher
Lower secondary
Academic degree
NAME_EDUCATION_TYPE


PCT
Lower secondary
Secondary / secondary special
Incomplete higher
Higher education
Academic degree
NAME_EDUCATION_TYPE

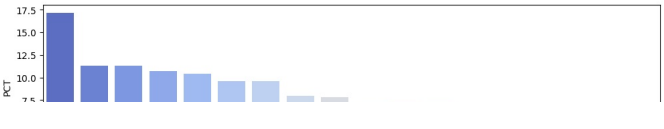

TARGET
0
1
count
Single / not married
Married
Civil marriage
Widow
Separated
Unknown
NAME_FAMILY_STATUS

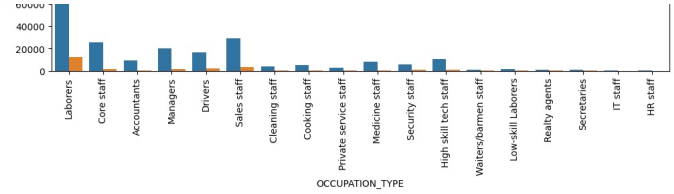

PCT
Civil marriage
Single / not married
Separated
Married
Widow
Unknown
NAME_FAMILY_STATUS

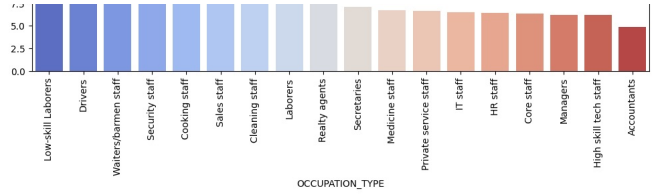

TARGET
0
1
count
House / apartment
Rented apartment
With parents
Municipal apartment
Office apartment
Co-op apartment
NAME_HOUSING_TYPE

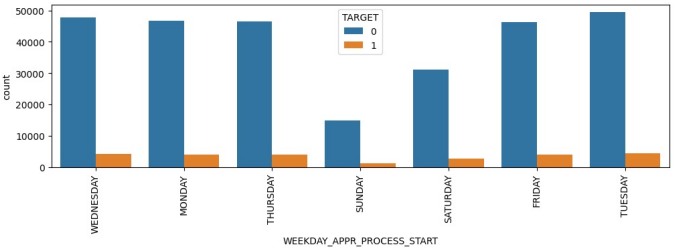

PCT
Rented apartment
With parents
Municipal apartment
Co-op apartment
House / apartment
Office apartment
NAME_HOUSING_TYPE

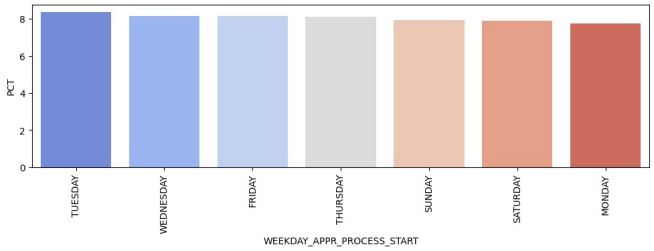

TARGET
0
1
count

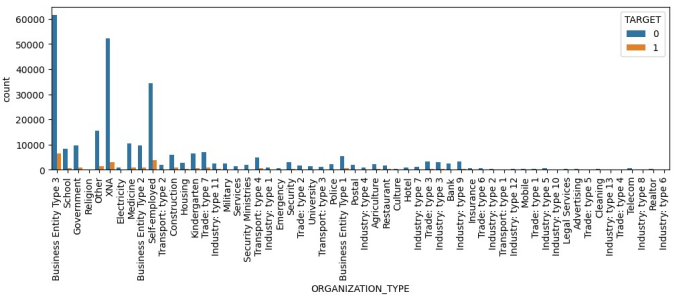

PCT

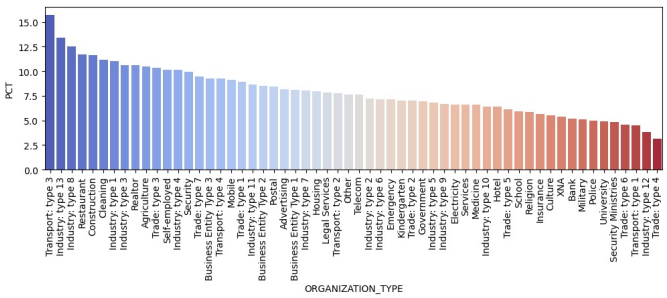












In [83]:
```
df_app_score_rmd.dtypes.value_counts()
```

Out[83]:
```
float64     18
int64       15
object      10
category     1
category     1
category     1
category     1
category     1
category     1
category     1
Name: count, dtype: int64
```

In [84]:
```
num_var = df_app_score_rmd.select_dtypes(include=['float64','int64']).columns
num_cat_var = df_app_score_rmd.select_dtypes(include=['float64','int64','category']).columns
num_var
```

Out[84]:
```
Index(['SK_ID_CURR', 'TARGET', 'CNT_CHILDREN', 'AMT_INCOME_TOTAL',
       'AMT_CREDIT', 'AMT_ANNUITY', 'AMT_GOODS_PRICE',
       'REGION_POPULATION_RELATIVE', 'DAYS_BIRTH', 'DAYS_EMPLOYED',
       'DAYS_REGISTRATION', 'DAYS_ID_PUBLISH', 'CNT_FAM_MEMBERS',
       'REGION_RATING_CLIENT', 'REGION_RATING_CLIENT_W_CITY',
       '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', 'OBS_30_CNT_SOCIAL_CIRCLE',
       'DEF_30_CNT_SOCIAL_CIRCLE', 'OBS_60_CNT_SOCIAL_CIRCLE',
       'DEF_60_CNT_SOCIAL_CIRCLE', 'DAYS_LAST_PHONE_CHANGE',
       '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'],
      dtype='object')
```

In [ ]:
```
```

In [85]:
```
num_data =  df_app_score_rmd[num_var]
```

In [86]:
```
defaulters = num_data[num_data['TARGET'] == 1].drop(['TARGET'],axis=1)
repayers = num_data[num_data['TARGET'] == 0].drop(['TARGET'],axis=1)
repayers.head()
```

Out[86]:

	SK_ID_CURR	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_PRICE	REGION_POPULATION
1	100003	0	270000.0	1293502.5	35698.5	1129500.0	
2	100004	0	67500.0	135000.0	6750.0	135000.0	
3	100006	0	135000.0	312682.5	29686.5	297000.0	
4	100007	0	121500.0	513000.0	21865.5	513000.0	
5	100008	0	99000.0	490495.5	27517.5	454500.0	

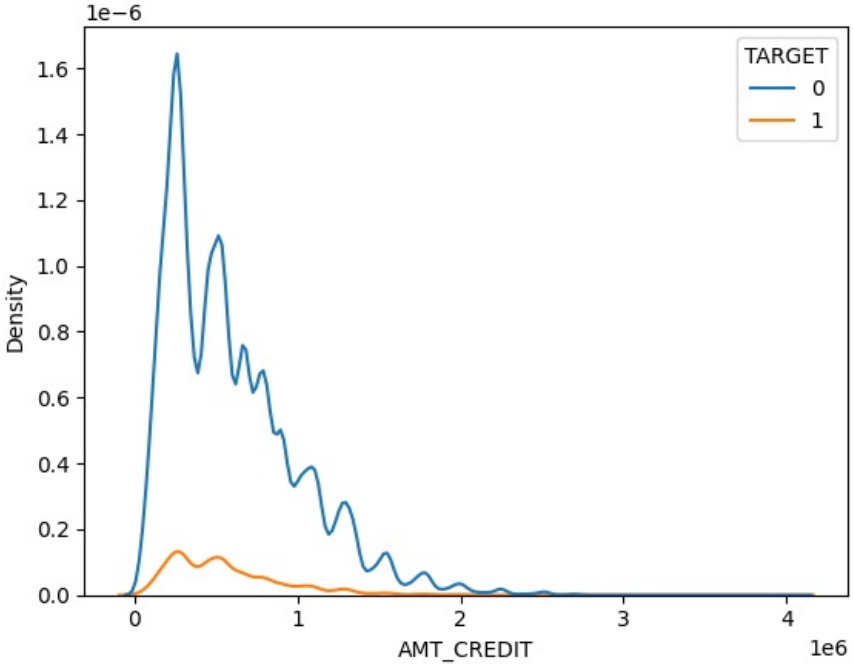
In [87]:
```
defaulters.corr()
```

Out[87]:

	SK_ID_CURR	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOO
SK_ID_CURR	1.000000	-0.005144	-0.010165	-0.001290	-0.007578	
CNT_CHILDREN	-0.005144	1.000000	0.004796	-0.001675	0.031257	
AMT_INCOME_TOTAL	-0.010165	0.004796	1.000000	0.038131	0.046421	
AMT_CREDIT	-0.001290	-0.001675	0.038131	1.000000	0.752195	
AMT_ANNUITY	-0.007578	0.031257	0.046421	0.752195	1.000000	
AMT_GOODS_PRICE	-0.001814	-0.008111	0.037591	0.982783	0.752295	
REGION_POPULATION_RELATIVE	0.006301	-0.031975	0.009135	0.069161	0.071690	
DAYS_BIRTH	0.001254	-0.259109	-0.003096	0.135316	0.014303	
DAYS_EMPLOYED	-0.005161	-0.192864	-0.014977	0.001930	-0.081207	
DAYS_REGISTRATION	-0.006342	-0.149154	-0.000158	0.025854	-0.034279	
DAYS_ID_PUBLISH	0.002539	0.032299	0.004215	0.052329	0.016767	
CNT_FAM_MEMBERS	-0.003816	0.885484	0.006654	0.051224	0.075711	
REGION_RATING_CLIENT	-0.005936	0.040680	-0.021486	-0.059193	-0.073784	
REGION_RATING_CLIENT_W_CITY	-0.004135	0.043185	-0.022808	-0.071377	-0.089291	
HOUR_APPR_PROCESS_START	0.005004	-0.023899	0.013775	0.031782	0.031236	
REG_REGION_NOT_LIVE_REGION	-0.004249	-0.024322	0.007577	0.019540	0.034807	
REG_REGION_NOT_WORK_REGION	0.004120	-0.020793	0.014531	0.033260	0.066565	
LIVE_REGION_NOT_WORK_REGION	0.004303	-0.012073	0.013409	0.033554	0.064109	
REG_CITY_NOT_LIVE_CITY	0.008328	-0.001174	-0.002223	-0.033034	-0.005745	
REG_CITY_NOT_WORK_CITY	0.000787	0.046115	-0.003019	-0.037720	0.001997	
LIVE_CITY_NOT_WORK_CITY	-0.002929	0.053515	-0.001353	-0.016509	0.009902	
OBS_30_CNT_SOCIAL_CIRCLE	-0.009395	0.025804	-0.004709	0.019098	0.004463	
DEF_30_CNT_SOCIAL_CIRCLE	-0.005549	0.001448	-0.005186	-0.025979	-0.022394	
OBS_60_CNT_SOCIAL_CIRCLE	-0.009058	0.025180	-0.004616	0.019487	0.005500	
DEF_60_CNT_SOCIAL_CIRCLE	-0.009428	-0.005106	-0.004866	-0.030880	-0.027495	
DAYS_LAST_PHONE_CHANGE	-0.002455	-0.011547	0.002429	0.110851	0.079870	
AMT_REQ_CREDIT_BUREAU_HOUR	-0.011106	0.000316	0.001079	-0.003771	0.012968	
AMT_REQ_CREDIT_BUREAU_DAY	-0.007388	-0.011255	0.000135	0.004346	0.000074	
AMT_REQ_CREDIT_BUREAU_WEEK	-0.003075	-0.009316	0.000941	0.010598	0.028784	
AMT_REQ_CREDIT_BUREAU_MON	0.005180	-0.008852	0.005718	0.056227	0.049000	
AMT_REQ_CREDIT_BUREAU_QRT	-0.001614	-0.013029	0.001037	-0.007201	-0.007261	
AMT_REQ_CREDIT_BUREAU_YEAR	0.006843	-0.027253	0.004516	-0.020698	-0.009819	

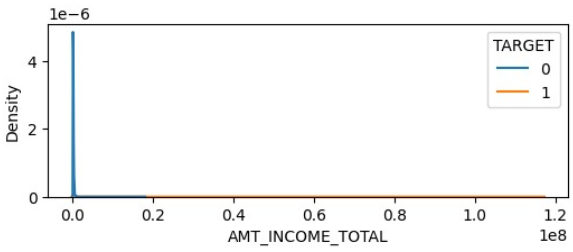
In [88]:
```
amt_var = ['AMT_INCOME_TOTAL','AMT_CREDIT','AMT_ANNUITY','AMT_GOODS_PRICE']
```

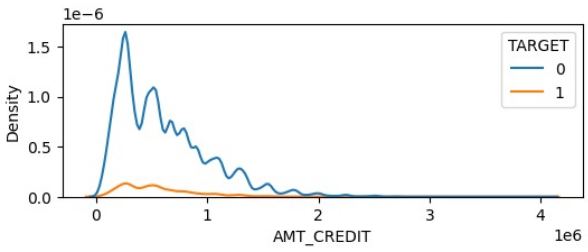
In [89]:
```
sns.kdeplot(data=num_data,x='AMT_CREDIT',hue='TARGET');
```

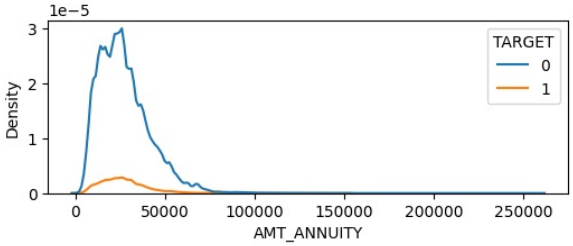


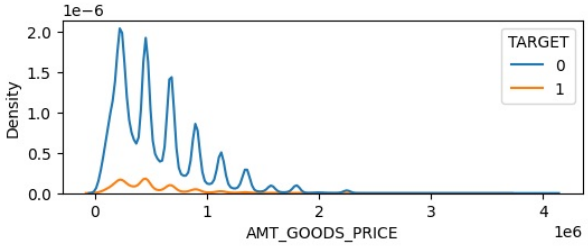
Univariate Numeric Analysis

In [90]:
```
plt.figure(figsize=(15,5))

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)
```









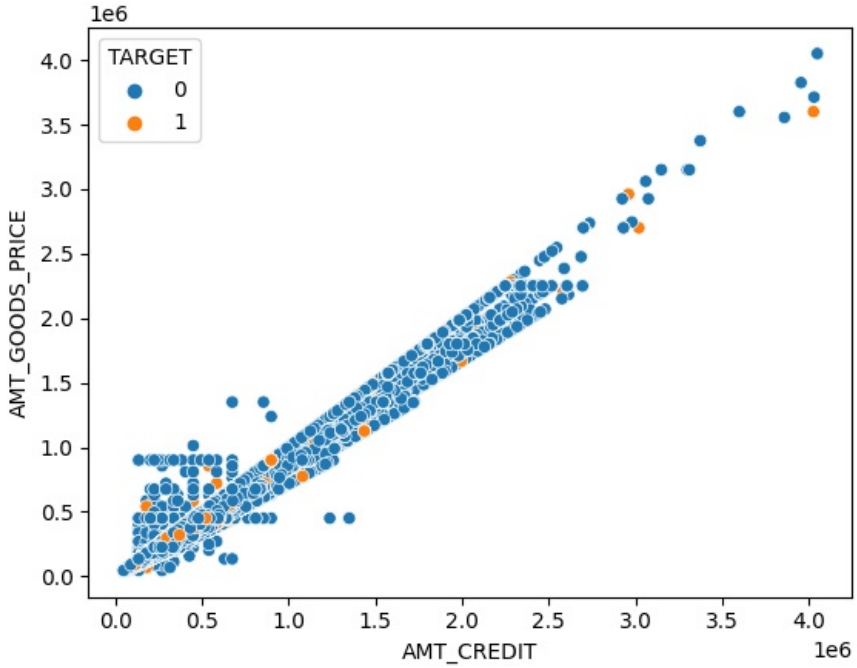
Univariate Numeric Analysis

1. most of the loans were given for the goods price ranging between 0 to 1 ml
2. most of the loans were given for the credit amount of 0 to 1ml
3. most of the customers are paying annuity of 0 to 50k
4. most of the customer have income between 0 to 1ml

Bivariate Numeric Analysis

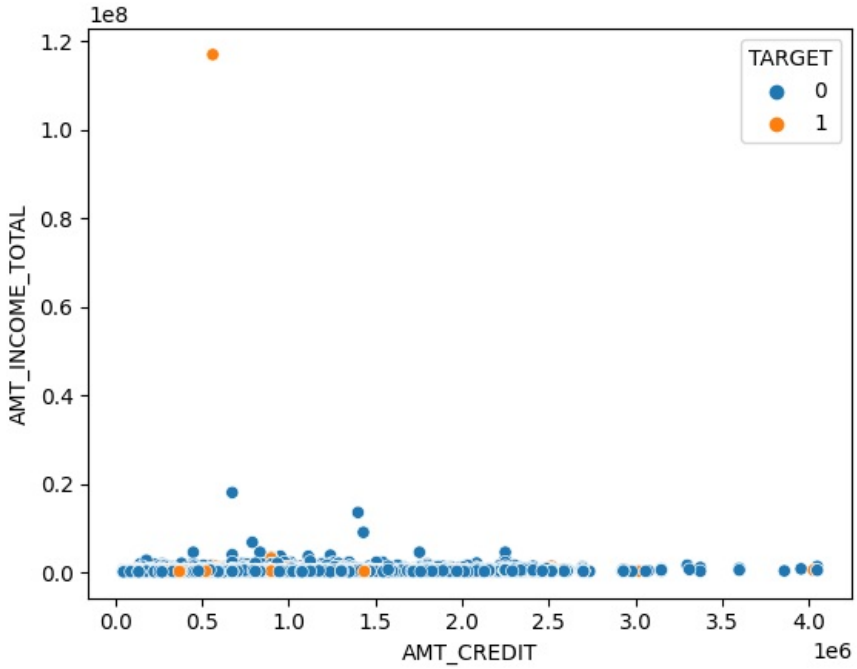
In [91]:
```
sns.scatterplot(data=num_data,x='AMT_CREDIT',y='AMT_GOODS_PRICE',hue='TARGET')
```

Out[91]:
```
<Axes: xlabel='AMT_CREDIT', ylabel='AMT_GOODS_PRICE'>
```



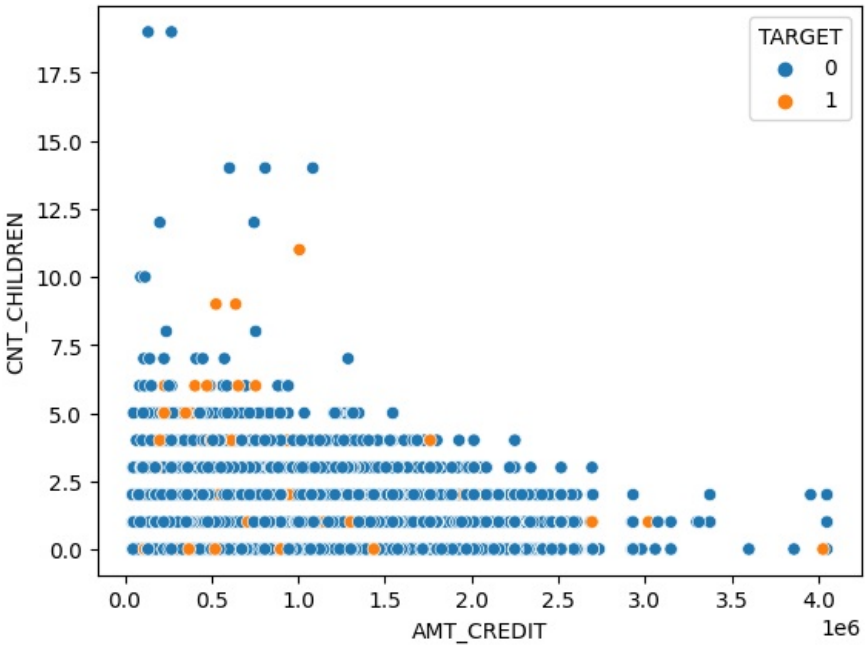
In [92]:
```
sns.scatterplot(data=num_data,x='AMT_CREDIT',y='AMT_INCOME_TOTAL',hue='TARGET')
```

Out[92]:
```
<Axes: xlabel='AMT_CREDIT', ylabel='AMT_INCOME_TOTAL'>
```



In [93]:
```
sns.scatterplot(data=num_data,x='AMT_CREDIT',y='CNT_CHILDREN',hue='TARGET')
```

Out[93]:
```
<Axes: xlabel='AMT_CREDIT', ylabel='CNT_CHILDREN'>
```

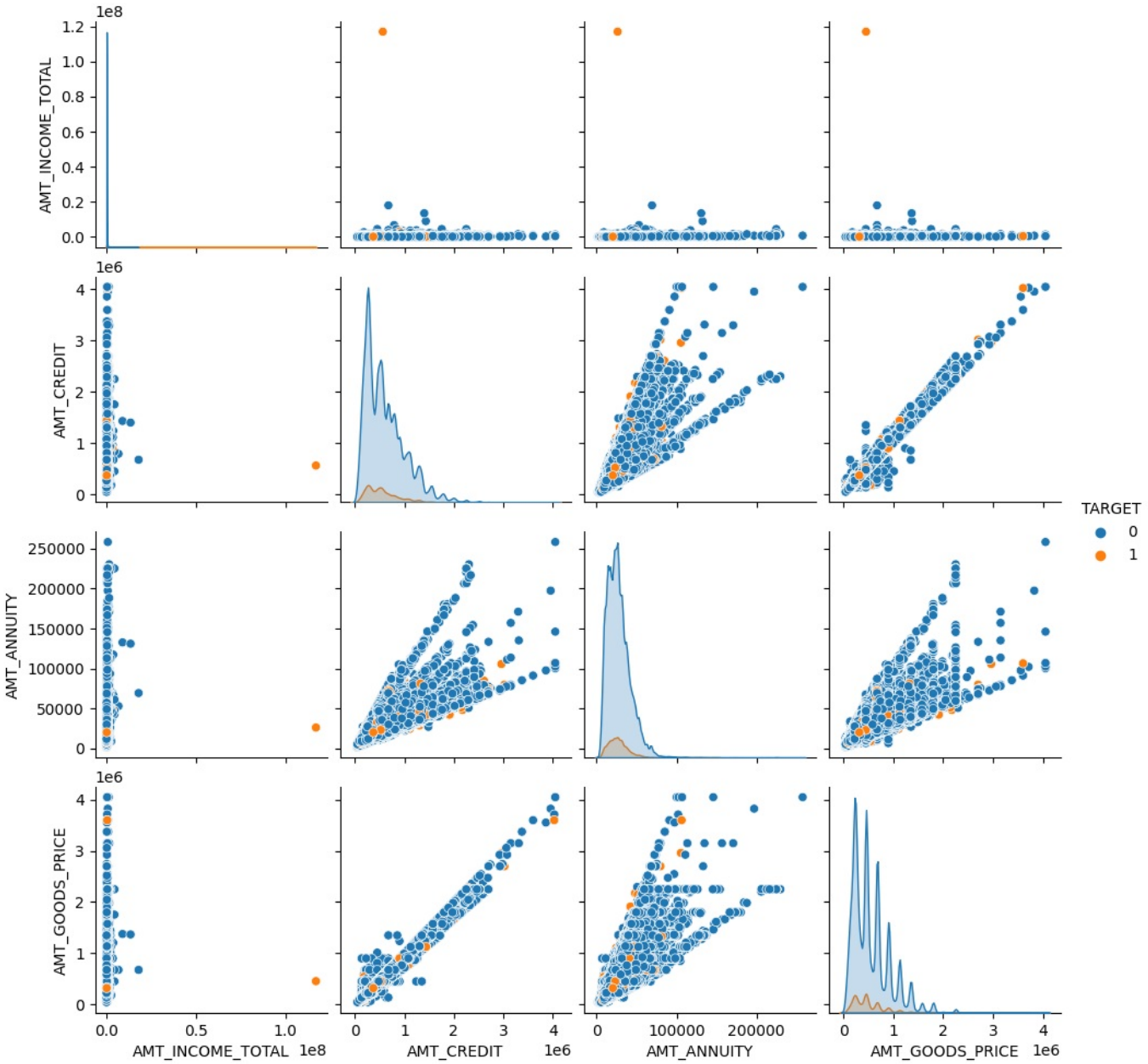


In [94]:
```
amt_var = num_data[['AMT_INCOME_TOTAL', 'AMT_CREDIT', 'AMT_ANNUITY', 'AMT_GOODS_PRICE' , 'TARGET']]
```

In [95]:
```
sns.pairplot(data=amt_var,hue='TARGET')
```

```
C:\Users\Hariram\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[95]:
```
<seaborn.axisgrid.PairGrid at 0x10505e90>
```



Bivariate Analysis

1. AMT_CREDIT and AMT_GOODS_PRICE are linearly corelated, if the AMT_CREDIT increases the defaulters are decreasing
2. people having income less than or equals to 1 ml, are more like to take loans out of which who are taking loan of less than 1.5 million, coudl turn out to be defaulters.
3. we can target income below 1 million and loan maount greater than 1.5 million
4. people having children 1 to less than 5 are safer to give the loan
5. People who can pay the annuity of 100K are more like to get the loan and that's upto less than 2ml (safer segment)

In [96]:
```
null_count = pd.DataFrame(df_prev.isnull().sum().sort_values(ascending=False)/df_prev.shape[0]*100).reset_index
null_count
var_msng_ge_40 = list(null_count[null_count['count_pct']>=40]['var'])
var_msng_ge_40
```

Out[96]:
```
['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 [97]:
```
nva_cols = var_msng_ge_40+['WEEKDAY_APPR_PROCESS_START','HOUR_APPR_PROCESS_START','FLAG_LAST_APPL_PER_CONTRACT'
len(nva_cols)
```

Out[97]: 15

In [98]:
```
len(df_prev.columns)
```

Out[98]: 37

In [99]:
```
df_prev_nva_cols = df_prev.drop(labels=nva_cols,axis=1)

len(df_prev_nva_cols.columns)
```

Out[99]: 22

In [100…
```
df_prev_nva_cols.columns
```

Out[100…
```
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', 'NAME_PORTFOLIO', 'NAME_PRODUCT_TYPE',
       'CHANNEL_TYPE', 'SELLERPLACE_AREA', 'NAME_SELLER_INDUSTRY',
       'CNT_PAYMENT', 'NAME_YIELD_GROUP', 'PRODUCT_COMBINATION'],
      dtype='object')
```

In [101…
```
df_prev_nva_cols.head()
```

Out[101…

	SK_ID_PREV	SK_ID_CURR	NAME_CONTRACT_TYPE	AMT_ANNUITY	AMT_APPLICATION	AMT_CREDIT	AMT_GOODS_PRICE	N
0	2030495	271877	Consumer loans	1730.430	17145.0	17145.0	17145.0	
1	2802425	108129	Cash loans	25188.615	607500.0	679671.0	607500.0	
2	2523466	122040	Cash loans	15060.735	112500.0	136444.5	112500.0	
3	2819243	176158	Cash loans	47041.335	450000.0	470790.0	450000.0	
4	1784265	202054	Cash loans	31924.395	337500.0	404055.0	337500.0	

In [102…
```
df_prev_nva_cols.isnull().sum().sort_values(ascending=False)/df_prev_nva_cols.shape[0]*100
```

Out[102…
```
AMT_GOODS_PRICE           23.081773
AMT_ANNUITY               22.286665
CNT_PAYMENT               22.286366
PRODUCT_COMBINATION        0.020716
AMT_CREDIT                 0.000060
NAME_GOODS_CATEGORY        0.000000
NAME_YIELD_GROUP           0.000000
NAME_SELLER_INDUSTRY       0.000000
SELLERPLACE_AREA           0.000000
CHANNEL_TYPE               0.000000
NAME_PRODUCT_TYPE          0.000000
NAME_PORTFOLIO             0.000000
SK_ID_PREV                 0.000000
NAME_CLIENT_TYPE           0.000000
SK_ID_CURR                 0.000000
NAME_PAYMENT_TYPE          0.000000
DAYS_DECISION              0.000000
NAME_CONTRACT_STATUS       0.000000
NAME_CASH_LOAN_PURPOSE     0.000000
AMT_APPLICATION            0.000000
NAME_CONTRACT_TYPE         0.000000
CODE_REJECT_REASON         0.000000
dtype: float64
```

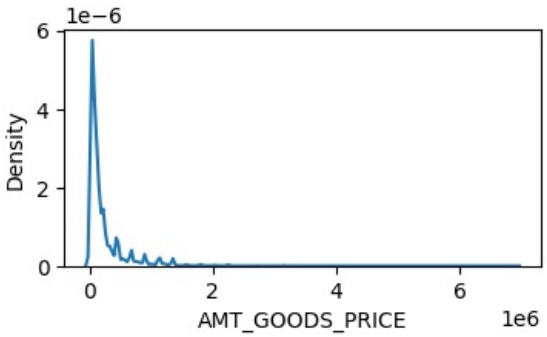
In [103…
```
df_prev_nva_cols['AMT_GOODS_PRICE'].agg(func=['mean','median'])
```

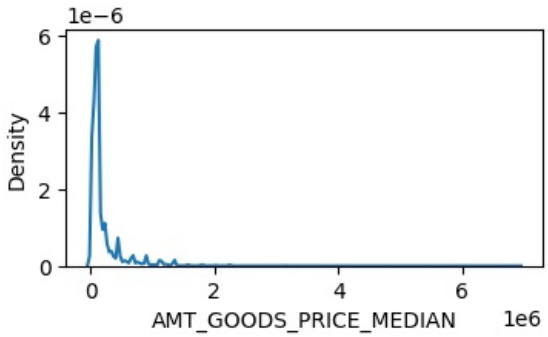
Out[103…
```
mean      227847.279283
median    112320.000000
Name: AMT_GOODS_PRICE, dtype: float64
```

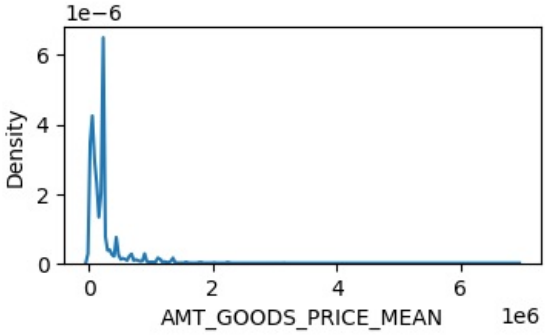
In [104…
```
df_prev_nva_cols['AMT_GOODS_PRICE_MEDIAN'] = df_prev_nva_cols['AMT_GOODS_PRICE'].fillna(df_prev_nva_cols['AMT_G
df_prev_nva_cols['AMT_GOODS_PRICE_MEAN'] = df_prev_nva_cols['AMT_GOODS_PRICE'].fillna(df_prev_nva_cols['AMT_GOO
df_prev_nva_cols['AMT_GOODS_PRICE_MODE'] = df_prev_nva_cols['AMT_GOODS_PRICE'].fillna(df_prev_nva_cols['AMT_GOO
```

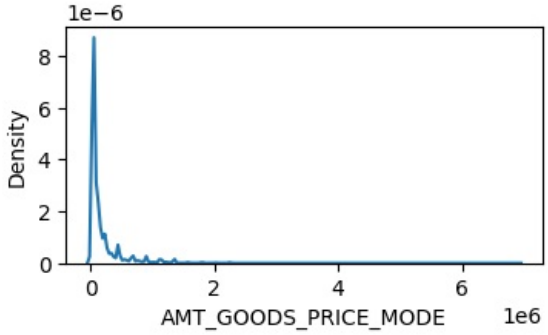
In [105…
```
gp_cols = ['AMT_GOODS_PRICE','AMT_GOODS_PRICE_MEDIAN','AMT_GOODS_PRICE_MEAN','AMT_GOODS_PRICE_MODE']
```

In [106…
```
plt.figure(figsize=(10,5))
for i,col in enumerate(gp_cols):
    plt.subplot(2,2,i+1)
    sns.kdeplot(data=df_prev_nva_cols,x=col)
    plt.subplots_adjust(wspace=0.5,hspace=0.5)
```









In [107…
```
df_prev_nva_cols['AMT_GOODS_PRICE'] = df_prev_nva_cols['AMT_GOODS_PRICE'].fillna(df_prev_nva_cols['AMT_GOODS_PR
```

In [108…
```
df_prev_nva_cols['AMT_GOODS_PRICE'].isnull().sum()
```

Out[108… 0

In [109…
```
df_prev_nva_cols['AMT_ANNUITY'].agg(func=['mean','median','max'])
```

Out[109…
```
mean       15955.120659
median     11250.000000
max       418058.145000
Name: AMT_ANNUITY, dtype: float64
```

In [110…
```
df_prev_nva_cols['AMT_ANNUITY'].isnull().sum()
```

Out[110… 372235

In [111…
```
df_prev_nva_cols['AMT_ANNUITY'] = df_prev_nva_cols['AMT_ANNUITY'].fillna(df_prev_nva_cols['AMT_GOODS_PRICE'].me
```

In [112…
```
df_prev_nva_cols['PRODUCT_COMBINATION'].head()
```

Out[112…
```
0    POS mobile with interest
1           Cash X-Sell: low
2          Cash X-Sell: high
3       Cash X-Sell: middle
4         Cash Street: high
Name: PRODUCT_COMBINATION, dtype: object
```

In [113…
```
df_prev_nva_cols['PRODUCT_COMBINATION'] = df_prev_nva_cols['PRODUCT_COMBINATION'].fillna(df_prev_nva_cols['PRODU
```

In [114…
```
df_prev_nva_cols['CNT_PAYMENT'].agg(func=['mean','median','max'])
```

Out[114…
```
mean      16.054082
median    12.000000
max       84.000000
Name: CNT_PAYMENT, dtype: float64
```

In [115…
```
df_prev_nva_cols[df_prev_nva_cols['CNT_PAYMENT'].isnull()].groupby(['NAME_CONTRACT_STATUS']).size().sort_values
```

Out[115…
```
NAME_CONTRACT_STATUS
Canceled        305805
Refused          40897
Unused offer     25524
Approved             4
dtype: int64
```

In [116…
```
df_prev_nva_cols['CNT_PAYMENT'] = df_prev_nva_cols['CNT_PAYMENT'].fillna(0)
```

In [117…
```
df_prev_nva_cols.isnull().sum().sort_values(ascending=False)
```

Out[117…
```
AMT_CREDIT                1
SK_ID_PREV                0
NAME_GOODS_CATEGORY       0
AMT_GOODS_PRICE_MEAN      0
AMT_GOODS_PRICE_MEDIAN    0
PRODUCT_COMBINATION       0
NAME_YIELD_GROUP          0
CNT_PAYMENT               0
NAME_SELLER_INDUSTRY      0
SELLERPLACE_AREA          0
CHANNEL_TYPE              0
NAME_PRODUCT_TYPE         0
NAME_PORTFOLIO            0
NAME_CLIENT_TYPE          0
SK_ID_CURR                0
CODE_REJECT_REASON        0
NAME_PAYMENT_TYPE         0
DAYS_DECISION             0
NAME_CONTRACT_STATUS      0
NAME_CASH_LOAN_PURPOSE    0
AMT_GOODS_PRICE           0
AMT_APPLICATION           0
AMT_ANNUITY               0
NAME_CONTRACT_TYPE        0
AMT_GOODS_PRICE_MODE      0
dtype: int64
```

In [118…
```
df_prev_nva_cols = df_prev_nva_cols.drop(labels=['AMT_GOODS_PRICE_MEDIAN','AMT_GOODS_PRICE_MEAN','AMT_GOODS_PRI
```

In [119…
```
df_prev_nva_cols.isnull().sum().sort_values(ascending=False)
```

Out[119…
```
AMT_CREDIT                1
SK_ID_PREV                0
NAME_CLIENT_TYPE          0
NAME_YIELD_GROUP          0
CNT_PAYMENT               0
NAME_SELLER_INDUSTRY      0
SELLERPLACE_AREA          0
CHANNEL_TYPE              0
NAME_PRODUCT_TYPE         0
NAME_PORTFOLIO            0
NAME_GOODS_CATEGORY       0
CODE_REJECT_REASON        0
SK_ID_CURR                0
NAME_PAYMENT_TYPE         0
DAYS_DECISION             0
NAME_CONTRACT_STATUS      0
NAME_CASH_LOAN_PURPOSE    0
AMT_GOODS_PRICE           0
AMT_APPLICATION           0
AMT_ANNUITY               0
NAME_CONTRACT_TYPE        0
PRODUCT_COMBINATION       0
dtype: int64
```

In [120…
```
len(df_prev_nva_cols.columns)
```

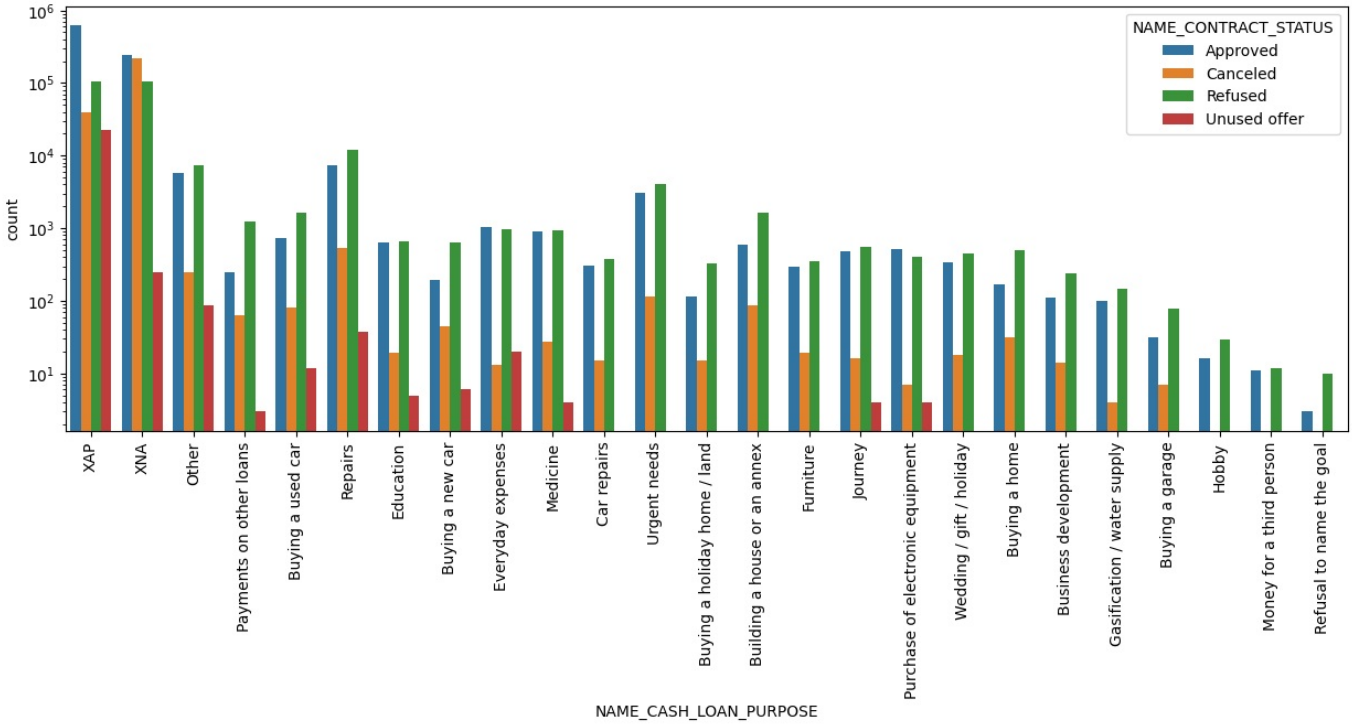
Out[120… 22

In [121…
```
merged_df = pd.merge(df_app_score_rmd,df_prev_nva_cols,how='inner',on='SK_ID_CURR')
merged_df.head()
```

Out[121…

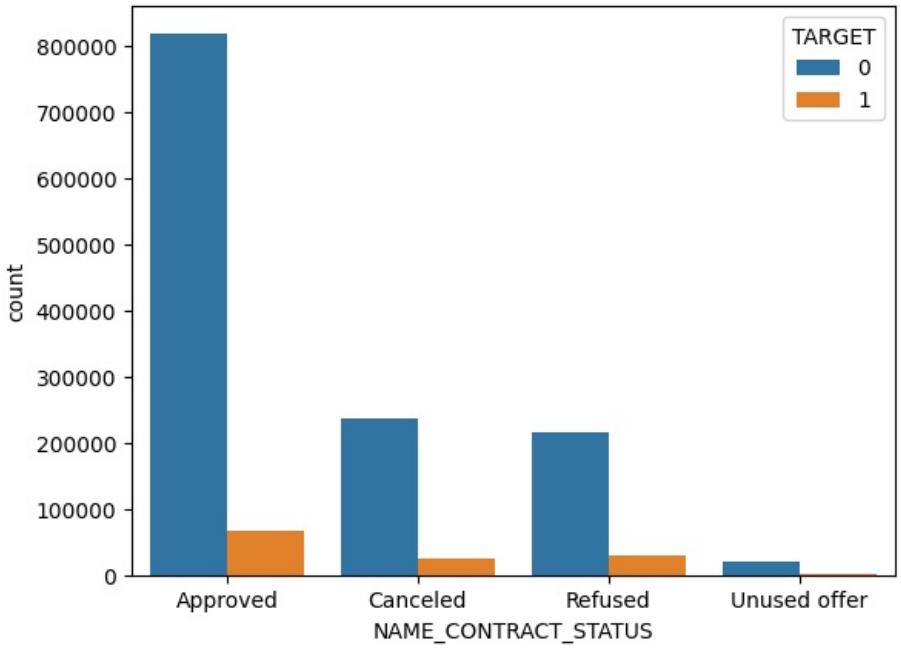
	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE_x	CODE_GENDER	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT_x	A
0	100002	1	Cash loans	M	0	202500.0	406597.5	
1	100003	0	Cash loans	F	0	270000.0	1293502.5	
2	100003	0	Cash loans	F	0	270000.0	1293502.5	
3	100003	0	Cash loans	F	0	270000.0	1293502.5	
4	100004	0	Revolving loans	M	0	67500.0	135000.0	

In [122…
```
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')
```



In [123…
```
sns.countplot(data=merged_df,x='NAME_CONTRACT_STATUS',hue='TARGET')
```

Out[123…
```
<Axes: xlabel='NAME_CONTRACT_STATUS', ylabel='count'>
```



In [124…
```
merged = merged_df.groupby(['NAME_CONTRACT_STATUS','TARGET']).size().reset_index().rename(columns={0:'counts'})
sum_agg = merged.groupby(['NAME_CONTRACT_STATUS'])['counts'].sum().reset_index()

merged_agg = pd.merge(merged,sum_agg,how='left',on='NAME_CONTRACT_STATUS')
merged_agg['pct'] = round(merged_agg['counts_x'] / merged_agg['counts_y']*100,2)
merged_agg
```

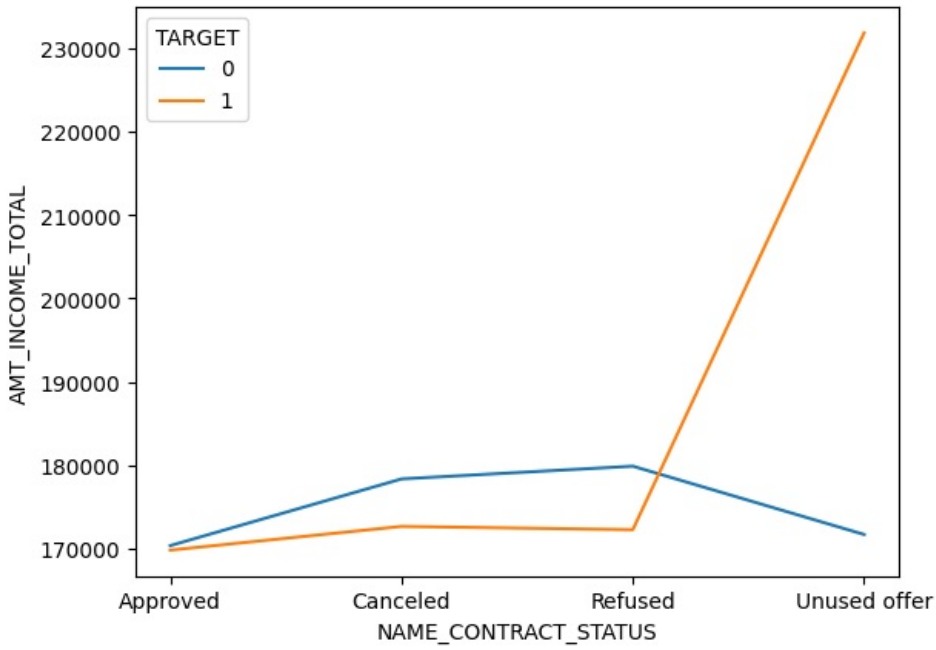
Out[124…

	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

In [125…
```
sns.lineplot(data=merged_df,x='NAME_CONTRACT_STATUS',y='AMT_INCOME_TOTAL',ci=None,hue='TARGET')
```

```
C:\Users\Hariram\AppData\Local\Temp\ipykernel_2656\563267390.py:1: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=merged_df,x='NAME_CONTRACT_STATUS',y='AMT_INCOME_TOTAL',ci=None,hue='TARGET')
```

Out[125… <Axes: xlabel='NAME_CONTRACT_STATUS', ylabel='AMT_INCOME_TOTAL'>



In [126…
```
len(merged_df.columns)
```

Out[126… 70

In [127…
```
merged_df.head()
```

Out[127…

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE_x	CODE_GENDER	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT_x	A
0	100002	1	Cash loans	M	0	202500.0	406597.5	
1	100003	0	Cash loans	F	0	270000.0	1293502.5	
2	100003	0	Cash loans	F	0	270000.0	1293502.5	
3	100003	0	Cash loans	F	0	270000.0	1293502.5	
4	100004	0	Revolving loans	M	0	67500.0	135000.0	

Decisive factor in whether an applicant will defaulter:

CODE_GENDER -

most of the loans have been taken by female
default rate for females are just ~7% which is safer and lesser than male

NAME_TYPE_SUITE -

unacompanied people had tanke most of the loans and the default rate is ~8.5% which is still okay

NAME_INCOME_TYPE -

the safest segments are working, commercial associates and pensioners

NAME_EDUCATION_TYPE -

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

NAME_FAMILY_STATUS -

Married people are safe to target, default rate is 8%

NAME_HOUSING_TYPE -

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

OCCUPATION_TYPE -

```
Low-Skill Laboreres 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%
```

ORGANIZATION_TYPE -

```
Transport type 3 highest defaulter
Others, Business Entity Type 3, Self Employed are good to go with default rate around 10 %
```

Final Conclusion

Bank should target the customers

1. Having low income i.e. below 1 ml
2. Working in Others, Business Entity Type 3, Self Employed org. type
3. Working as Accountants, Core staff, Managers and Laborers
4. Having house/appartment and are married and having children not more than 5
5. Highly educated
6. Preferably female
7. Unacompanied people can be safer - default rate is ~8.5%

Amount segment recommended

1. The credit amount should not be more than 1 ml
2. Annuity can be made of 50K (depending on the eligibility)
3. Income bracket could be below 1 ml
4. 80-90% of the customer who were prev. canceled/refused, are repayers. Bank can do the analysis and can consider to give loan to these segments