Data Import and Exploration

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

pd.options.display.max_columns = None
pd.options.display.max_rows = None
```

In [2]:

```
app = pd.read_csv("application_data.csv")
prev_app = pd.read_csv("previous_application.csv")
```

In [3]:

```
app.head()
```

Out[3]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL
0	100002	1	Cash loans	M	N	Y	0	202500.0
1	100003	0	Cash loans	F	N	N	0	270000.0
2	100004	0	Revolving loans	M	Y	Y	0	67500.0
3	100006	0	Cash loans	F	N	Y	0	135000.0
4	100007	0	Cash loans	M	N	Y	0	121500.0

Feature Selection

In [4]:

```
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]:
```
app.shape
```

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

In [6]:
```
msng_info = pd.DataFrame(app.isnull().sum().sort_values()).reset_index()
msng_info.rename(columns={'index':'col_name',0:'null_count'},inplace=True)
msng_info.head()
```

Out[6]:

	col_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

In [11]:
```
msng_info['msng_pct'] = msng_info['null_count']/app.shape[0]*100
msng_info.to_excel("missing_info.xlsx",index=False)
msng_info.head()
```

Out[11]:

	col_name	null_count	msng_pct
0	SK_ID_CURR	0	0.0
1	HOUR_APPR_PROCESS_START	0	0.0
2	REG_REGION_NOT_WORK_REGION	0	0.0

	col_name	null_count	msng_pct
3	LIVE_REGION_NOT_WORK_REGION	0	0.0
4	REG_CITY_NOT_LIVE_CITY	0	0.0

In [12]:

```
msng_col = msng_info[msng_info['msng_pct']>=40]['col_name'].to_list()
app_msng_rmvd = app.drop(labels=msng_col, axis=1)
app_msng_rmvd.shape
```

Out[12]: (307511, 73)

In [13]:

```
flag_col = []

for col in app_msng_rmvd.columns:
    if col.startswith("FLAG_"):
        flag_col.append(col)

len(flag_col)
```

Out[13]: 28

In [14]:

```
flag_tgt_col = app_msng_rmvd[flag_col+['TARGET']]
flag_tgt_col.head()
```

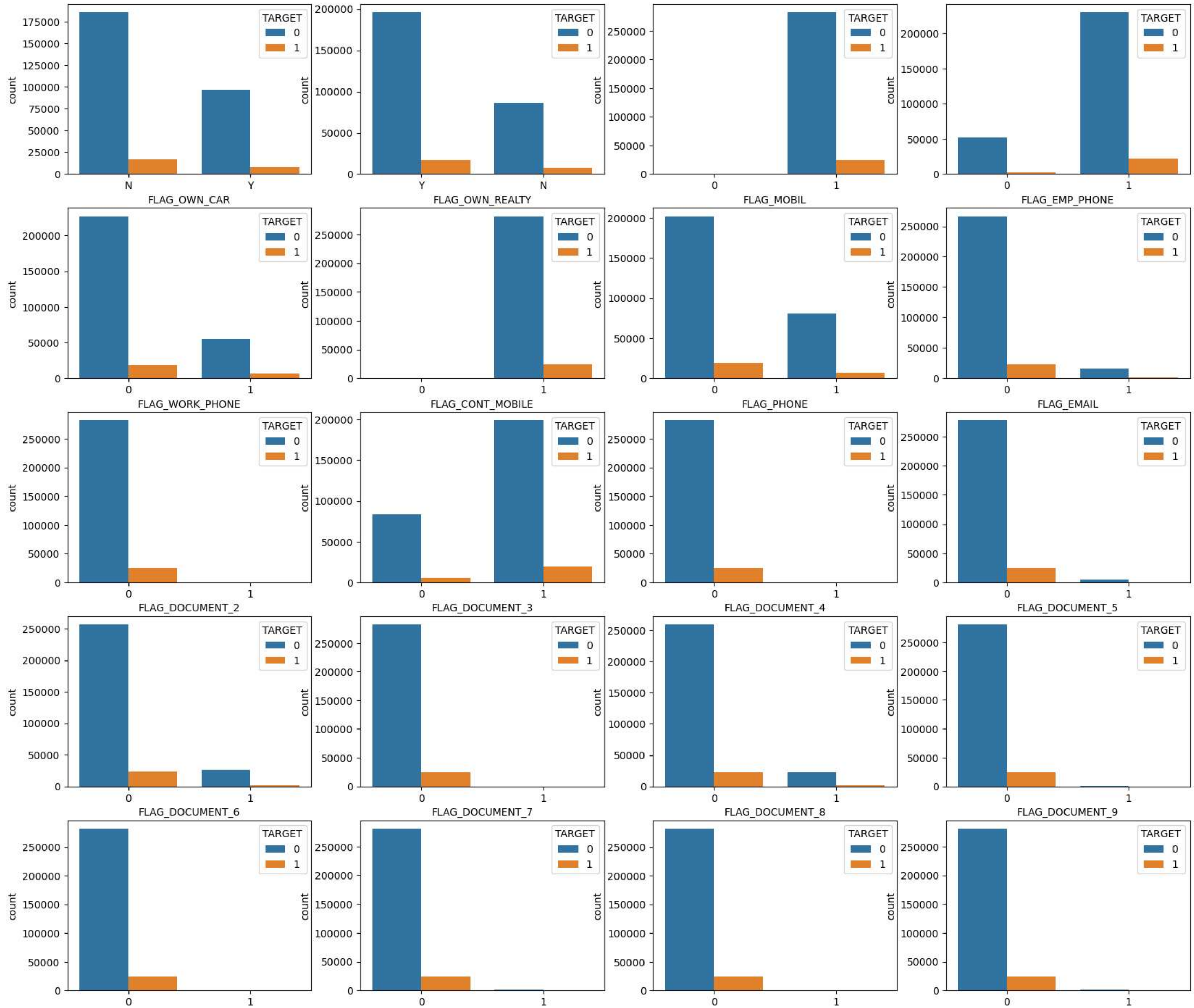
Out[14]:

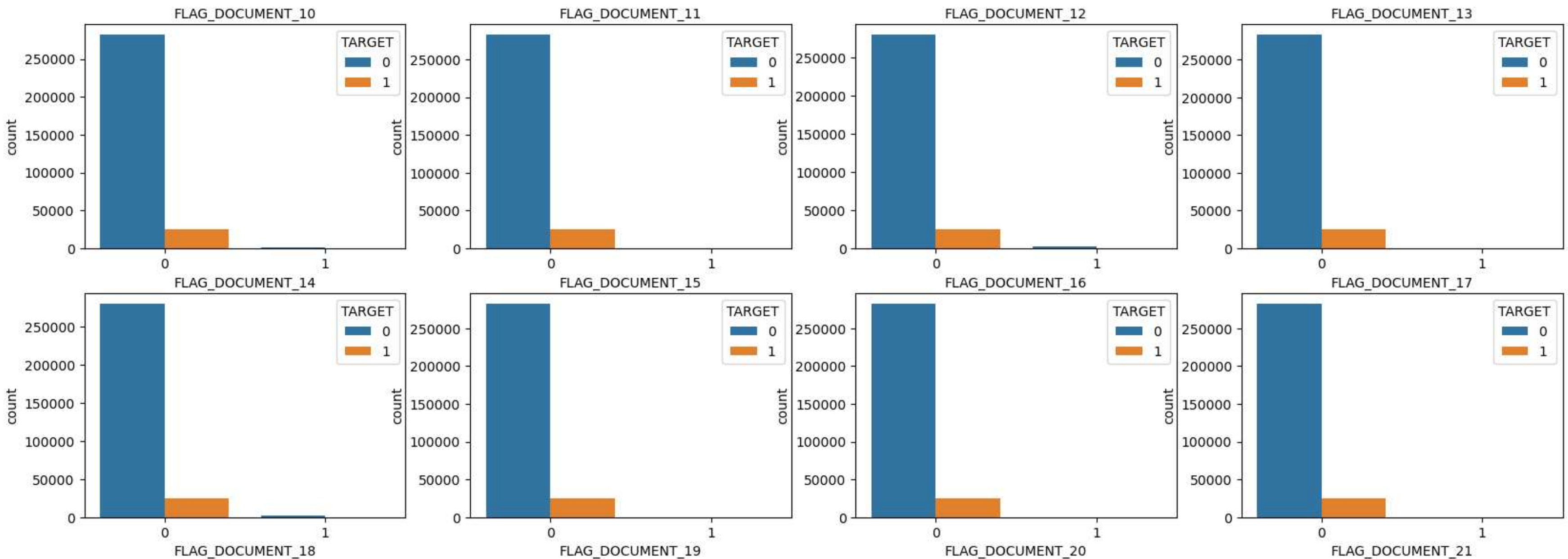
	FLAG_OWN_CAR	FLAG_OWN_REALTY	FLAG_MOBIL	FLAG_EMP_PHONE	FLAG_WORK_PHONE	FLAG_CONT_MOBILE	FLAG_PHONE	FLAG_EMAIL
0	N	Y	1	1	0	1	1	C
1	N	N	1	1	0	1	1	C
2	Y	Y	1	1	1	1	1	C
3	N	Y	1	1	0	1	0	C
4	N	Y	1	1	0	1	0	C

In [15]:

```
plt.figure(figsize=(20,25))
for i, col in enumerate(flag_col):
```

```
    plt.subplot(7,4,i+1)
    sns.countplot(data=flag_tgt_col,x= col, hue='TARGET')
```


TARGET
0
1
count
FLAG_OWN_CAR
FLAG_OWN_REALTY
FLAG_MOBIL
FLAG_EMP_PHONE
FLAG_WORK_PHONE
FLAG_CONT_MOBILE
FLAG_PHONE
FLAG_EMAIL
FLAG_DOCUMENT_2
FLAG_DOCUMENT_3
FLAG_DOCUMENT_4
FLAG_DOCUMENT_5
FLAG_DOCUMENT_6
FLAG_DOCUMENT_7
FLAG_DOCUMENT_8
FLAG_DOCUMENT_9



In [16]:

```
flg_corr =['FLAG_OWN_CAR','FLAG_OWN_REALTY','FLAG_MOBIL','FLAG_EMP_PHONE','FLAG_WORK_PHONE','FLAG_CONT_MOBILE','FLAG_PHON
flag_corr_df = app_msng_rmvd[flg_corr]
```

In [17]:

```
flag_corr_df.groupby(['FLAG_OWN_CAR']).size()
```

Out[17]:

```
FLAG_OWN_CAR
N    202924
Y    104587
dtype: int64
```

In [18]:

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

flag_corr_df.groupby(['FLAG_OWN_CAR']).size()
```

```
C:\Users\harsh\AppData\Local\Temp\ipykernel_17260\3198034239.py:1: 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#returning-a-v
iew-versus-a-copy
  flag_corr_df['FLAG_OWN_CAR'] = flag_corr_df['FLAG_OWN_CAR'].replace({'N':0, 'Y':1})
```

```
C:\Users\harsh\AppData\Local\Temp\ipykernel_17260\3198034239.py:2: 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#returning-a-v
iew-versus-a-copy
  flag_corr_df['FLAG_OWN_REALTY'] = flag_corr_df['FLAG_OWN_REALTY'].replace({'N':0, 'Y':1})
```

Out[18]:

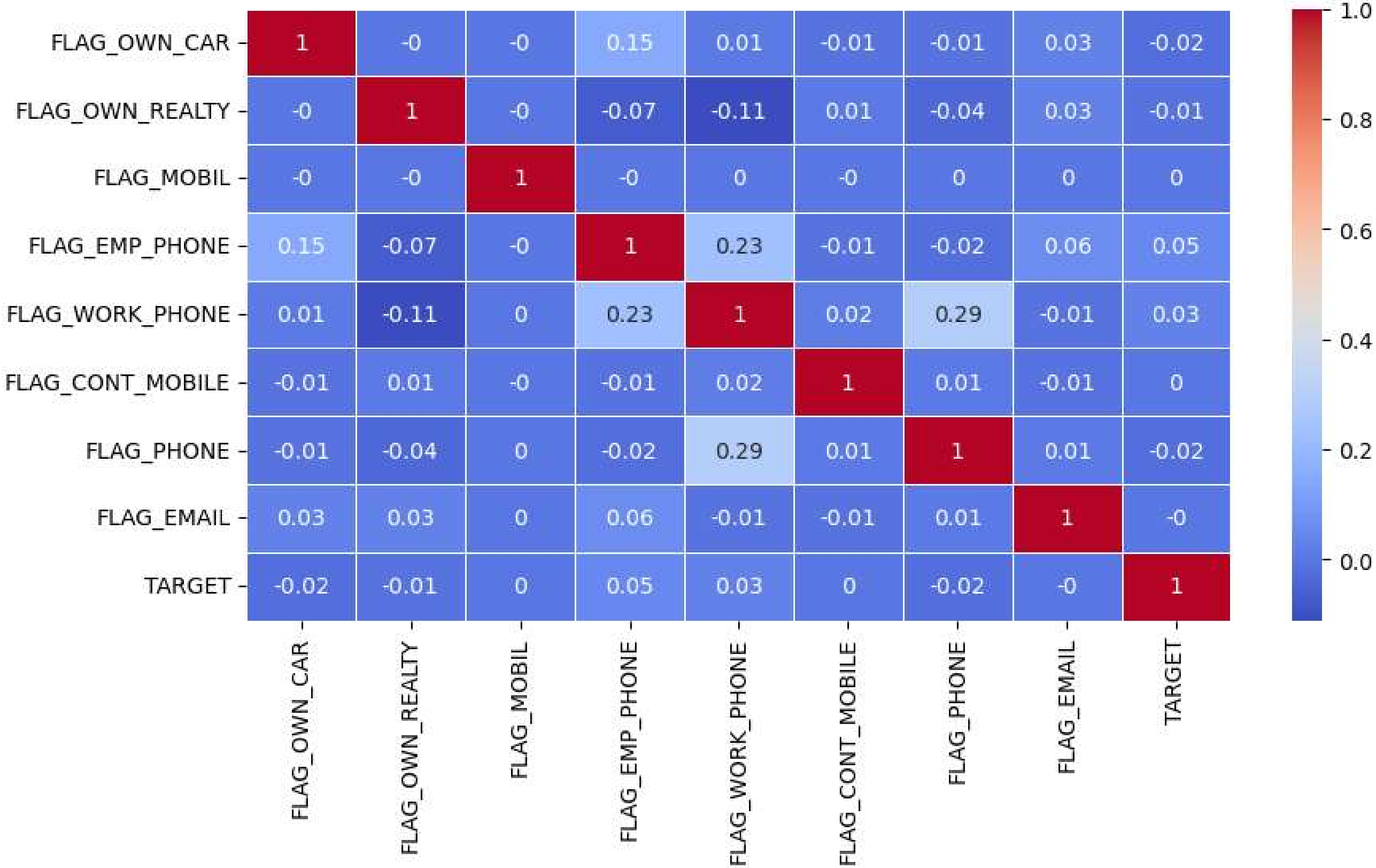
```
FLAG_OWN_CAR
0    202924
1    104587
dtype: int64
```

In [19]:

```
corr_df = round(flag_corr_df.corr(),2)

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

Out[19]:

```
<Axes: >
```



In [20]:

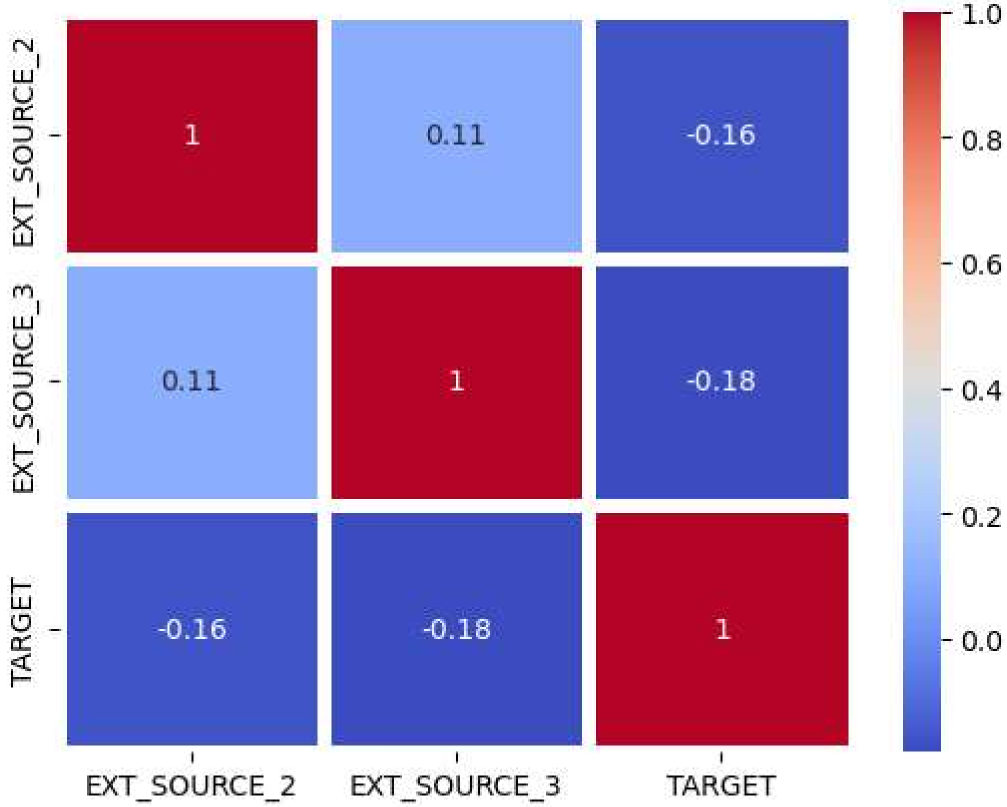
```
app_flag_rmvd = app_msng_rmvd.drop(labels = flag_col,axis=1)
app_flag_rmvd.shape
```

Out[20]: (307511, 45)

In [21]:

```
sns.heatmap(round(app_flag_rmvd[['EXT_SOURCE_2','EXT_SOURCE_3','TARGET']].corr(),2),cmap='coolwarm', linewidths=5,annot=T
```

Out[21]: <Axes: >



In [22]:

```
app_score_col_rmvd=app_flag_rmvd.drop(['EXT_SOURCE_2','EXT_SOURCE_3'], axis=1)
app_score_col_rmvd.shape
```

Out[22]: (307511, 43)

Feature Transformation

In [23]:

```
app_score_col_rmvd.isnull().sum().sort_values()/app_score_col_rmvd.shape[0]
```

Out[23]:

```
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 Values

In [24]:

```
#app_score_col_rmvd.groupby('CNT_FAM_MEMBERS').size()
app_score_col_rmvd['CNT_FAM_MEMBERS'] = app_score_col_rmvd['CNT_FAM_MEMBERS'].fillna((app_score_col_rmvd['CNT_FAM_MEMBERS
```

In [25]:
```
app_score_col_rmvd['CNT_FAM_MEMBERS'].isnull().sum()
```
Out[25]:
```
0
```
In [26]:
```
app_score_col_rmvd['OCCUPATION_TYPE'] = app_score_col_rmvd['OCCUPATION_TYPE'].fillna((app_score_col_rmvd['OCCUPATION_TYPE
#app_score_col_rmvd['OCCUPATION_TYPE'].mode()[0]
```
In [27]:
```
app_score_col_rmvd['OCCUPATION_TYPE'].isnull().sum()
```
Out[27]:
```
0
```
In [28]:
```
#app_score_col_rmvd.groupby(['NAME_TYPE_SUITE']).size()
app_score_col_rmvd['NAME_TYPE_SUITE']= app_score_col_rmvd['NAME_TYPE_SUITE'].fillna((app_score_col_rmvd['NAME_TYPE_SUITE'
```
In [29]:
```
app_score_col_rmvd['NAME_TYPE_SUITE'].isnull().sum()
```
Out[29]:
```
0
```
In [30]:
```
app_score_col_rmvd['AMT_ANNUITY']= app_score_col_rmvd['AMT_ANNUITY'].fillna((app_score_col_rmvd['AMT_ANNUITY'].mean()))
```
In [31]:
```
app_score_col_rmvd['AMT_ANNUITY'].isnull().sum()
```
Out[31]:
```
0
```
In [32]:
```
app_score_col_rmvd['AMT_REQ_CREDIT_BUREAU_HOUR'].describe()
```
Out[32]:
```
count    265992.000000
mean          0.006402
std           0.083849
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
```

```
max           4.000000
Name: AMT_REQ_CREDIT_BUREAU_HOUR, dtype: float64
```

In [33]:
```
amt_req_col = []

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

amt_req_col
#app_score_col_rmvd['AMT_REQ_CREDIT_BUREAU_HOUR'].head()
```

Out[33]:
```
['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 [34]:
```
for col in amt_req_col:
    app_score_col_rmvd[col] = app_score_col_rmvd[col].fillna((app_score_col_rmvd[col].median()))
```

In [35]:
```
app_score_col_rmvd[col].isnull().sum()
```

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

In [36]:
```
#app_score_col_rmvd.isnull().sum().sort_values()
```

In [37]:
```
app_score_col_rmvd['AMT_GOODS_PRICE'] = app_score_col_rmvd['AMT_GOODS_PRICE'].fillna((app_score_col_rmvd['AMT_GOODS_PRICE
```

In [38]:
```
app_score_col_rmvd['AMT_GOODS_PRICE'].isnull().sum()
```

Out[38]:
```
0
```

In [39]:
```
app_score_col_rmvd['AMT_GOODS_PRICE'].mean()
```

Out[39]: 538316.2943667056

In [40]:
```
#app_score_col_rmvd['AMT_CREDIT'].isnull().sum()
```

Values modification

In [41]:
```
days_col = []

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

days_col
```

Out[41]:
```
['DAYS_BIRTH',
 'DAYS_EMPLOYED',
 'DAYS_REGISTRATION',
 'DAYS_ID_PUBLISH',
 'DAYS_LAST_PHONE_CHANGE']
```

In [42]:
```
for col in days_col:
    app_score_col_rmvd[col] = abs(app_score_col_rmvd[col])
```

In [43]:
```
app_score_col_rmvd.head()
```

Out[43]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GO
0	100002	1	Cash loans	M	0	202500.0	406597.5	24700.5	
1	100003	0	Cash loans	F	0	270000.0	1293502.5	35698.5	
2	100004	0	Revolving loans	M	0	67500.0	135000.0	6750.0	
3	100006	0	Cash loans	F	0	135000.0	312682.5	29686.5	

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GO
4	100007	0	Cash loans	M	0	121500.0	513000.0	21865.5	

In [44]:
```
#app_score_col_rmvd.info()
```

In [45]:
```
app_score_col_rmvd.nunique().sort_values()
```

Out[45]:
```
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 [46]:

```
app_score_col_rmvd['OBS_30_CNT_SOCIAL_CIRCLE'].unique()
```

Out[46]:

```
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 [47]:

```
app_score_col_rmvd['AMT_GOODS_PRICE'].agg(['min','max','median'])
```

Out[47]:

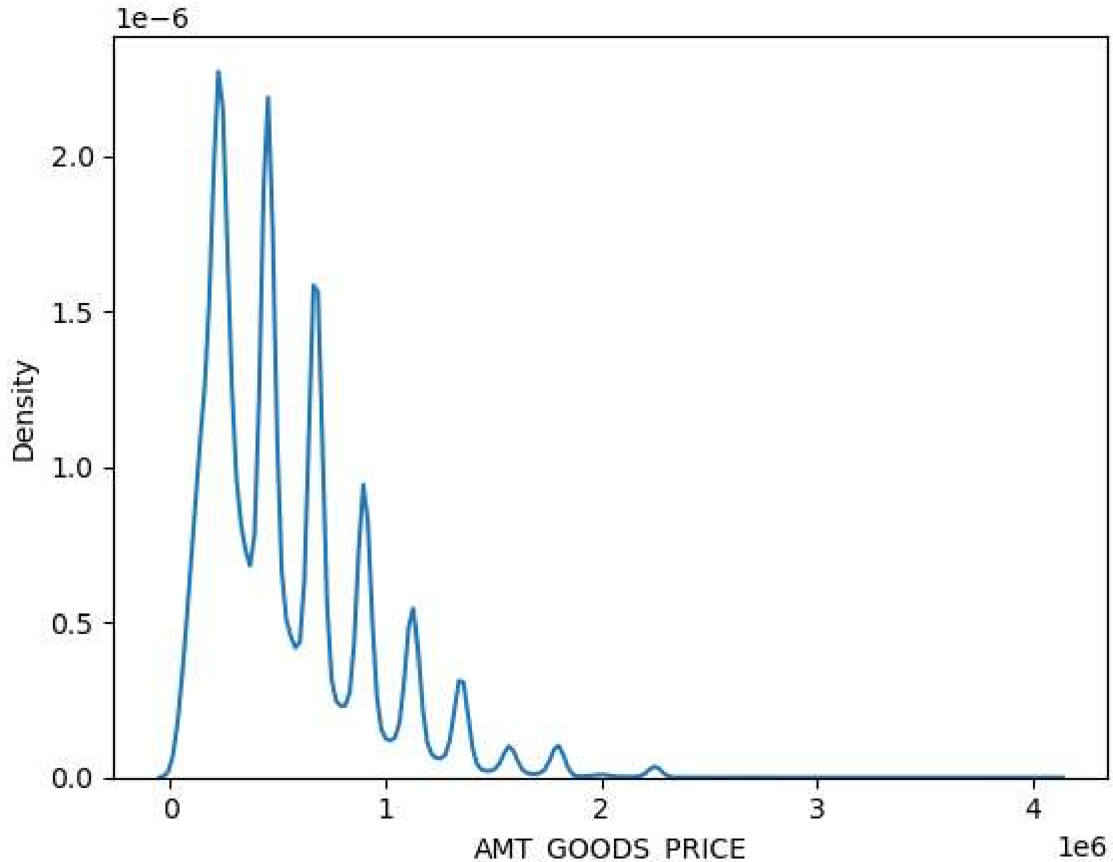
```
min         40500.0
max       4050000.0
median     450000.0
Name: AMT_GOODS_PRICE, dtype: float64
```

In [48]:

```
sns.kdeplot(data=app_score_col_rmvd,x='AMT_GOODS_PRICE')
```

Out[48]:

```
<Axes: xlabel='AMT_GOODS_PRICE', ylabel='Density'>
```

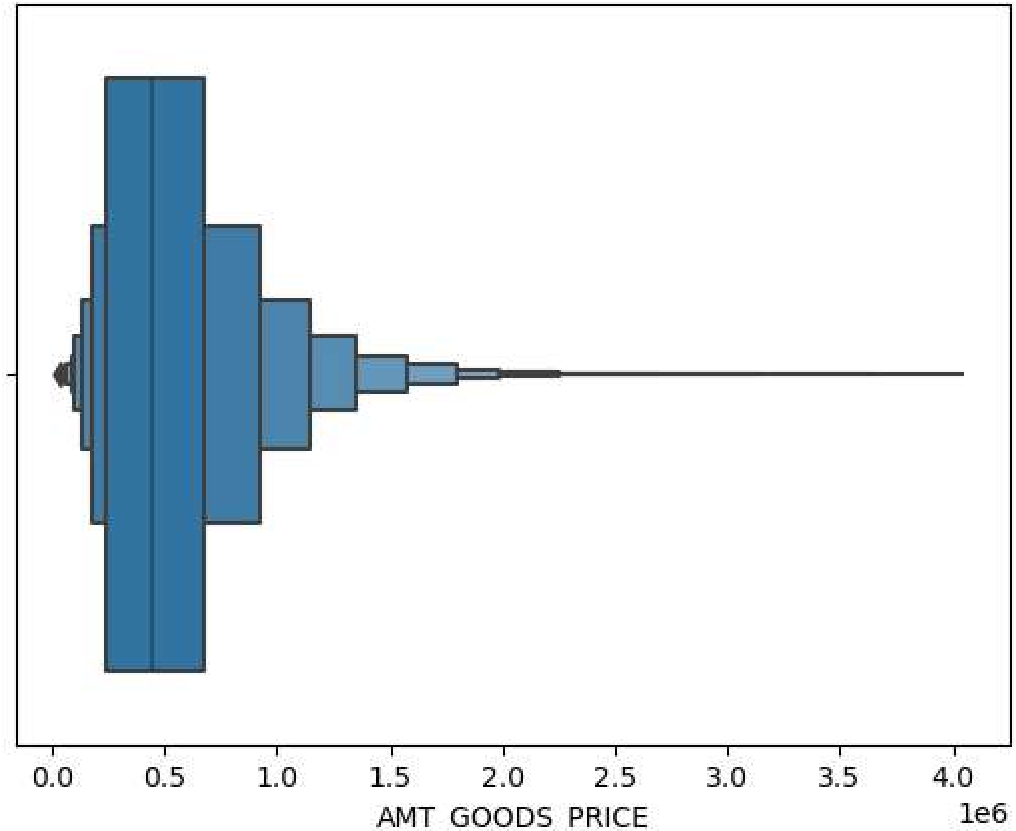


In [49]:

```
sns.boxenplot(data=app_score_col_rmvd,x='AMT_GOODS_PRICE')
```

Out[49]:

```
<Axes: xlabel='AMT_GOODS_PRICE'>
```



In [50]:
```
app_score_col_rmvd['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[50]:
```
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 [51]:
```
bins = [0,100000,200000,300000,400000,500000,600000,700000,800000,900000,4050000]
ranges = ['0-100K','100k-200k','200k-300k','300k-400k','400k-500k','500k-600k','600k-700k','700k-800k','800k-900k','Above
```

```
app_score_col_rmvd['AMT_GOODS_PRICE_RANGE']= pd.cut(app_score_col_rmvd['AMT_GOODS_PRICE'],bins, labels= ranges)
```

In [52]:
```
app_score_col_rmvd.groupby(['AMT_GOODS_PRICE_RANGE']).size()
```

Out[52]:
```
AMT_GOODS_PRICE_RANGE
0-100K          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 [53]:
```
app_score_col_rmvd['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[53]:
```
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 [54]:
```
bins = [0,100000,150000,200000,250000,300000,350000,400000,472500]
ranges = ['0-100K','100k-150k','150k-200k','200k-250k','250k-300k','300k-350k','350k-400k', 'Above 400k']

app_score_col_rmvd['AMT_INCOME_TOTAL_RANGE']= pd.cut(app_score_col_rmvd['AMT_INCOME_TOTAL'],bins, labels= ranges)
```

In [55]:
```
app_score_col_rmvd.groupby(['AMT_INCOME_TOTAL_RANGE']).size()
```

Out[55]:
```
AMT_INCOME_TOTAL_RANGE
0-100K          63698
100k-150k      91591
150k-200k      64307
```

```
200k-250k     48137
250k-300k     17039
300k-350k      8874
350k-400k      5802
Above 400k     5049
dtype: int64
```

In [56]:

```
app_score_col_rmvd['AMT_CREDIT'].quantile([0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,0.99])
```

Out[56]:

```
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 [57]:

```
bins = [0,200000,400000,600000,800000,900000,1000000,1854000]
ranges = ['0-200K','200k-400k','400k-600k','600k-800k','800k-900k','900k-1M', 'Above 1M']

app_score_col_rmvd['AMT_CREDIT_RANGE']= pd.cut(app_score_col_rmvd['AMT_CREDIT'],bins, labels= ranges)
```

In [58]:

```
app_score_col_rmvd.groupby(['AMT_CREDIT_RANGE']).size()
```

Out[58]:

```
AMT_CREDIT_RANGE
0-200K       36144
200k-400k    81151
400k-600k    66270
600k-800k    43242
800k-900k    21792
900k-1M       8927
Above 1M     46910
dtype: int64
```

In [59]:

```
app_score_col_rmvd['AMT_CREDIT'].isnull().sum()
```

Out[59]:

```
0
```

In [60]:
```python
app_score_col_rmvd['AMT_ANNUITY'].quantile([0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,0.99])
```

Out[60]:
```
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 [61]:
```python
app_score_col_rmvd['AMT_ANNUITY'].max()
```

Out[61]:
```
258025.5
```

In [62]:
```python
bins = [0,25000,50000,100000,150000,200000,258025]
ranges = ['0-25K','25k-50k','50k-100k','100k-150k','150k-200k', 'Above 200k']

app_score_col_rmvd['AMT_ANNUITY_RANGE']= pd.cut(app_score_col_rmvd['AMT_ANNUITY'],bins, labels= ranges)
```

In [63]:
```python
app_score_col_rmvd.groupby(['AMT_ANNUITY_RANGE']).size()
```

Out[63]:
```
AMT_ANNUITY_RANGE
0-25K          154867
25k-50k        131347
50k-100k        20792
100k-150k         437
150k-200k          32
Above 200k         35
dtype: int64
```

In [64]:
```python
app_score_col_rmvd['AMT_ANNUITY_RANGE'].isnull().sum()
```

Out[64]:
```
1
```

In [65]:
```python
app_score_col_rmvd['DAYS_EMPLOYED'].agg(['min','max','median'])
```

Out[65]:
```
min            0.0
max       365243.0
median      2219.0
Name: DAYS_EMPLOYED, dtype: float64
```

In [66]:
```
app_score_col_rmvd['DAYS_EMPLOYED'].quantile([0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.85,0.9,0.95,0.99])
```

Out[66]:
```
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
0.85    365243.0
0.90    365243.0
0.95    365243.0
0.99    365243.0
Name: DAYS_EMPLOYED, dtype: float64
```

In [67]:
```
app_score_col_rmvd[app_score_col_rmvd['DAYS_EMPLOYED']<app_score_col_rmvd['DAYS_EMPLOYED'].max()].max()['DAYS_EMPLOYED']
```

Out[67]:
```
17912
```

In [68]:
```
app_score_col_rmvd['DAYS_EMPLOYED'].max()
```

Out[68]:
```
365243
```

In [69]:
```
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', 'Above 50Y']

app_score_col_rmvd['DAYS_BIRTH_RANGE']= pd.cut(app_score_col_rmvd['DAYS_BIRTH'],bins, labels= ranges)
```

In [70]:
```
app_score_col_rmvd.groupby(['DAYS_BIRTH_RANGE']).size()
```

Out[70]:
```
DAYS_BIRTH_RANGE
0-5Y           0
5Y-10Y         0
10Y-15Y        0
```

```
15Y-20Y          0
20Y-25Y      12159
25Y-30Y      32862
30Y-35Y      39440
35Y-40Y      42868
40Y-45Y      41406
45Y-50Y      35135
Above 50Y   103641
dtype: int64
```

In [71]:

```
app_score_col_rmvd['DAYS_BIRTH_RANGE'].isnull().sum()
```

Out[71]:

```
0
```

DATA ANALYSIS

In [72]:

```
app_score_col_rmvd.dtypes.value_counts()
```

Out[72]:

```
float64     18
int64       15
object      10
category     1
category     1
category     1
category     1
category     1
dtype: int64
```

In [73]:

```
obj_var = app_score_col_rmvd.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]:

```
app_score_col_rmvd.groupby(['NAME_CONTRACT_TYPE']).size()
```

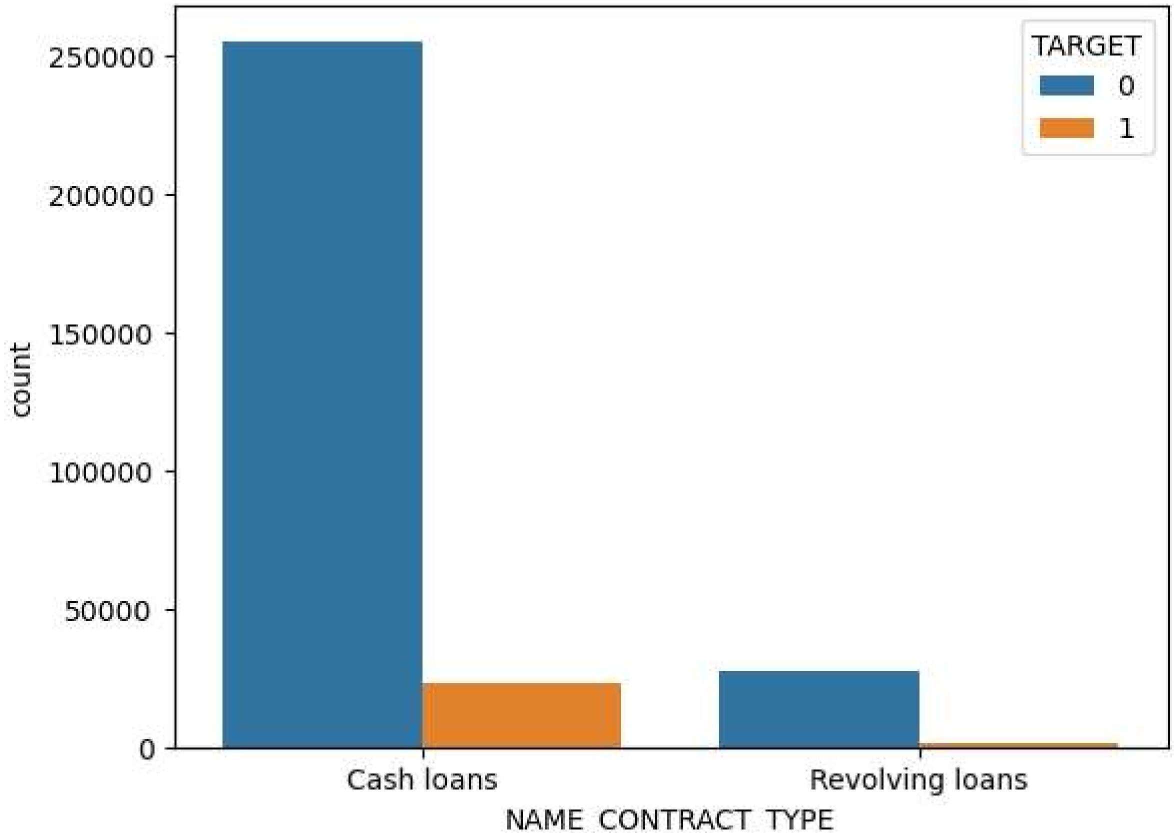
Out[74]:

```
NAME_CONTRACT_TYPE
Cash loans         278232
```

```
Revolving loans     29279
dtype: int64
```

In [75]:
```
sns.countplot(data=app_score_col_rmvd,x='NAME_CONTRACT_TYPE',hue='TARGET')
```

Out[75]:
```
<Axes: xlabel='NAME_CONTRACT_TYPE', ylabel='count'>
```



In [76]:
```
data_pct = app_score_col_rmvd[['NAME_CONTRACT_TYPE','TARGET']].groupby(['NAME_CONTRACT_TYPE'], as_index=False).mean().sor
```

In [77]:
```
data_pct['PCT']= data_pct['TARGET']*100
```

In [78]:
```
data_pct
```

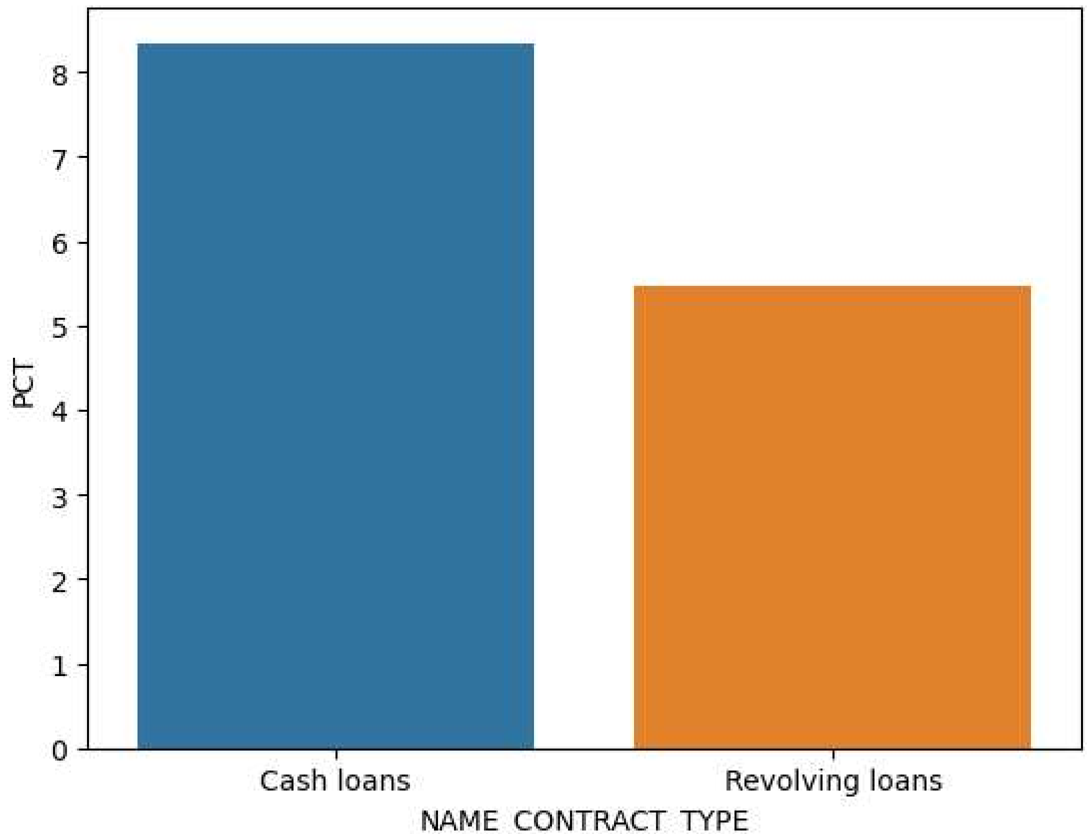
Out[78]:

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

In [79]:

```
sns.barplot(data=data_pct, x='NAME_CONTRACT_TYPE', y='PCT')
```

Out[79]: <Axes: xlabel='NAME_CONTRACT_TYPE', ylabel='PCT'>

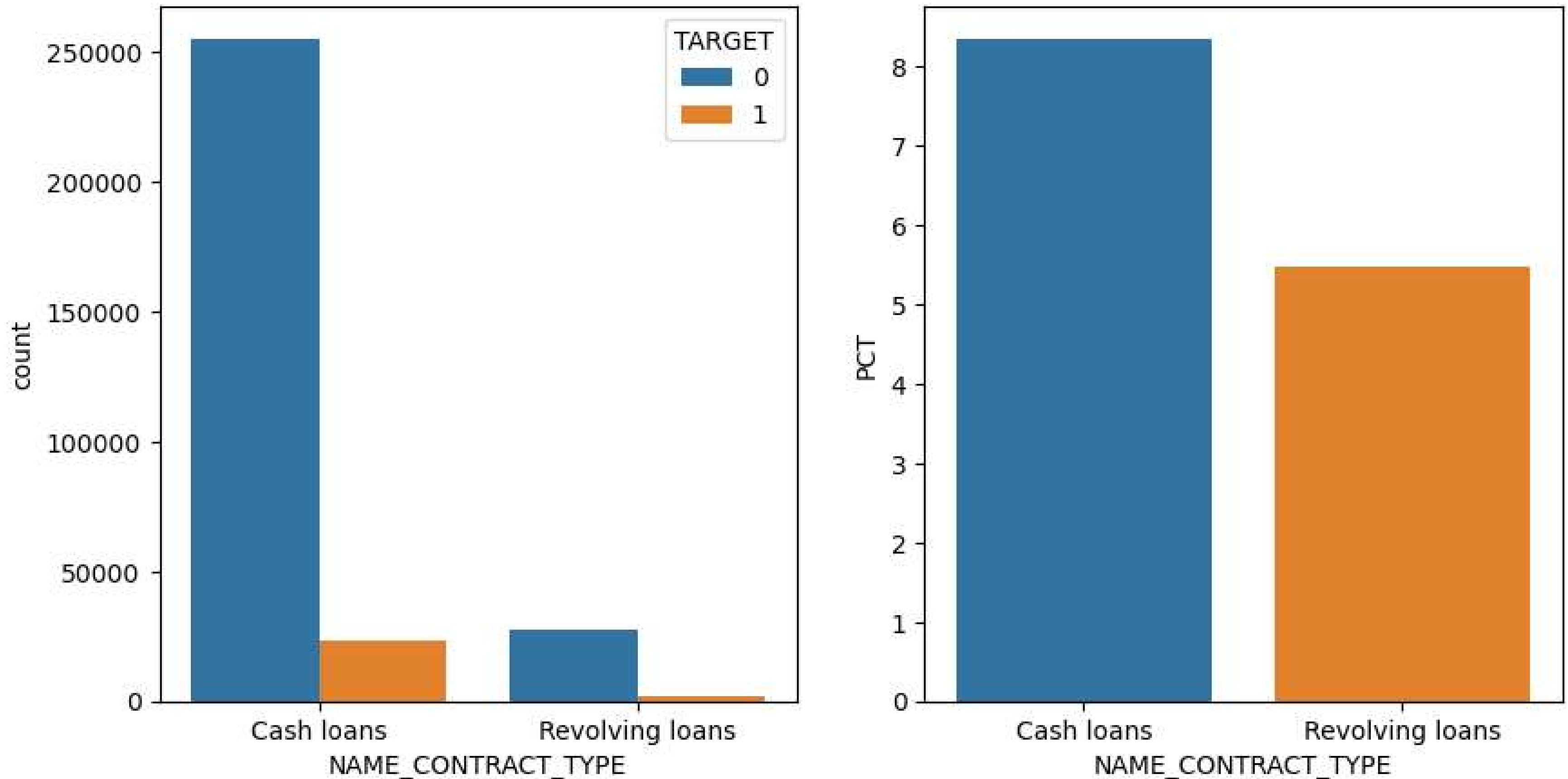


In [80]:

```
plt.figure(figsize=(10,5))

plt.subplot(1,2,1)

sns.countplot(data=app_score_col_rmvd, x='NAME_CONTRACT_TYPE', hue='TARGET')
```

```
plt.subplot(1,2,2)
sns.barplot(data=data_pct, x='NAME_CONTRACT_TYPE', y='PCT')
```

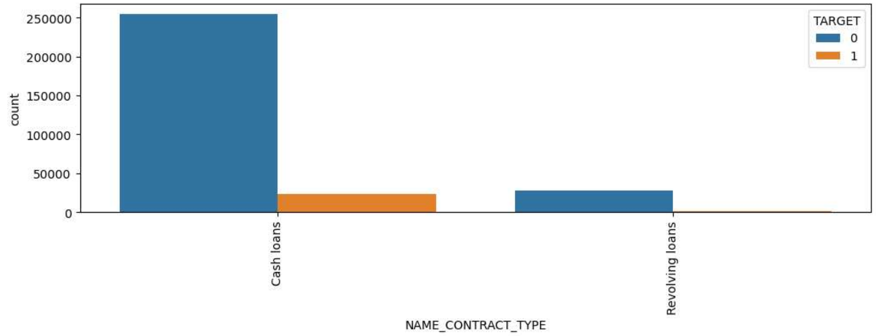
Out[80]: <Axes: xlabel='NAME_CONTRACT_TYPE', ylabel='PCT'>

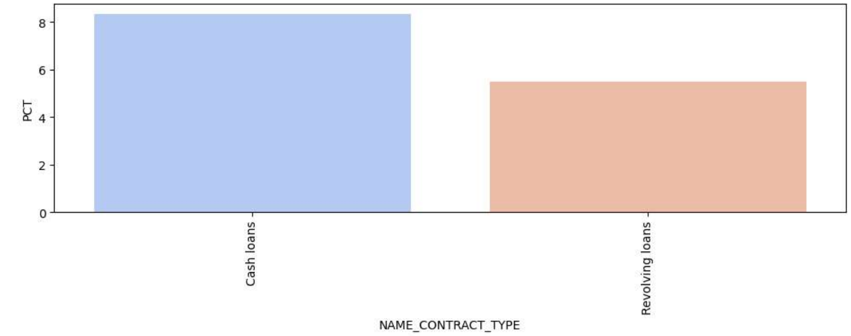


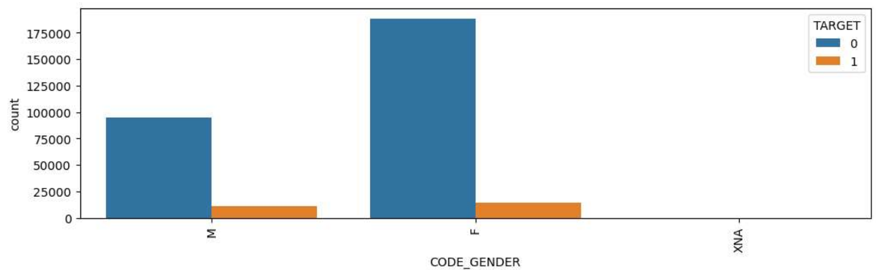
In [81]:

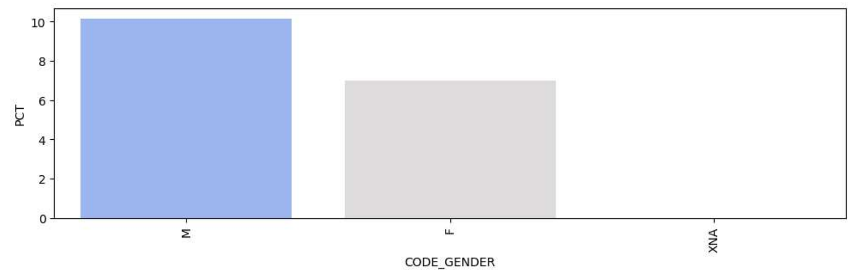
```
plt.figure(figsize=(25,60))

for i, var in enumerate (obj_var):

    data_pct = app_score_col_rmvd[[var,'TARGET']].groupby([var], as_index=False).mean().sort_values(by='TARGET',ascending
    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=app_score_col_rmvd,x=var, hue='TARGET')
```

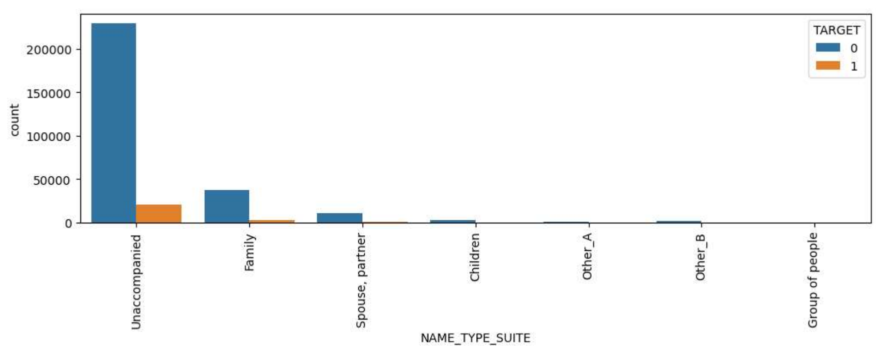
```
    plt.xticks(rotation=90)

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

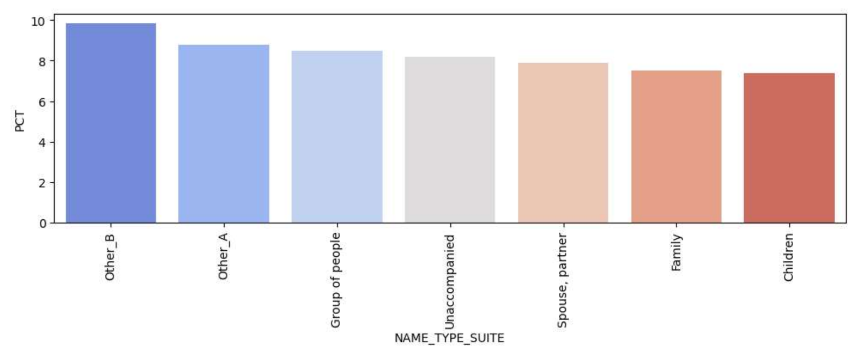

TARGET
0
1
count
250000
200000
150000
100000
50000
0
Cash loans
Revolving loans
NAME_CONTRACT_TYPE

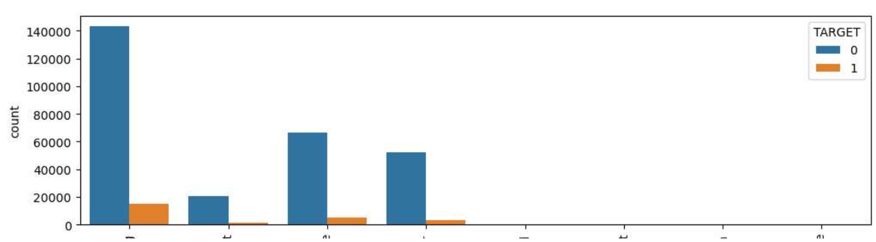

PCT
8
6
4
2
0
Cash loans
Revolving loans
NAME_CONTRACT_TYPE

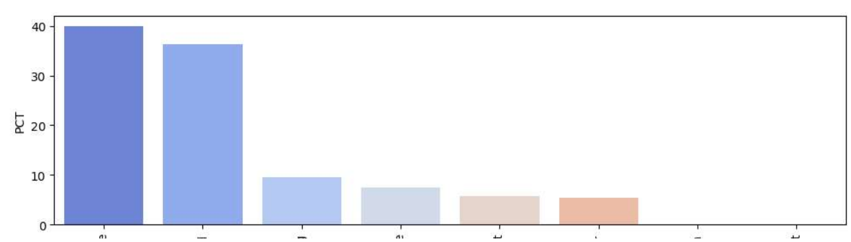

TARGET
0
1
count
175000
150000
125000
100000
75000
50000
25000
0
M
F
XNA
CODE_GENDER

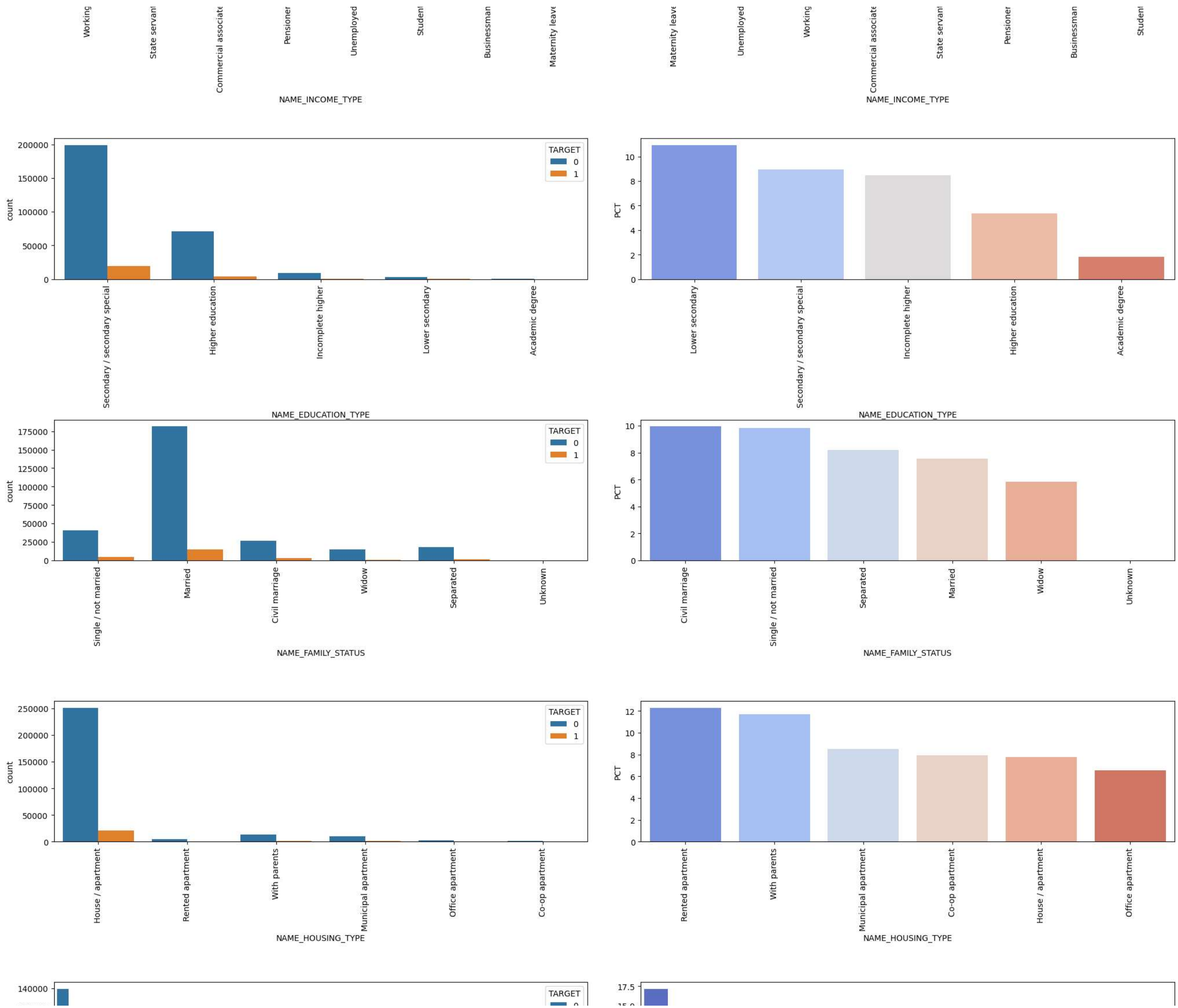

PCT
10
8
6
4
2
0
M
F
XNA
CODE_GENDER


TARGET
0
1
count
200000
150000
100000
50000
0
Unaccompanied
Family
Spouse, partner
Children
Other_A
Other_B
Group of people
NAME_TYPE_SUITE


PCT
10
8
6
4
2
0
Other_B
Other_A
Group of people
Unaccompanied
Spouse, partner
Family
Children
NAME_TYPE_SUITE


TARGET
0
1
count
140000
120000
100000
80000
60000
40000
20000
0


PCT
40
30
20
10
0


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

In [82]:

```
app_score_col_rmvd['NAME_EDUCATION_TYPE'].unique()
```

Out[82]:

```
array(['Secondary / secondary special', 'Higher education',
       'Incomplete higher', 'Lower secondary', 'Academic degree'],
      dtype=object)
```

In [83]:

```
app_score_col_rmvd.dtypes.value_counts()
```

Out[83]:

```
float64     18
int64       15
object      10
category     1
category     1
category     1
category     1
category     1
dtype: int64
```

In [84]:

```
num_var = app_score_col_rmvd.select_dtypes(include=['float64','int64']).columns
num_cat_var = app_score_col_rmvd.select_dtypes(include=['float64','int64','category']).columns
len(num_var)
```

Out[84]:

```
33
```

In [85]:

```
num_data = app_score_col_rmvd[num_var]
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[85]:

	SK_ID_CURR	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_PRICE	REGION_POPULATION_RELATIVE	DAYS_
1	100003	0	270000.0	1293502.5	35698.5	1129500.0	0.003541	
2	100004	0	67500.0	135000.0	6750.0	135000.0	0.010032	
3	100006	0	135000.0	312682.5	29686.5	297000.0	0.008019	
4	100007	0	121500.0	513000.0	21865.5	513000.0	0.028663	

	SK_ID_CURR	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_PRICE	REGION_POPULATION_RELATIVE	DAYS_
5	100008	0	99000.0	490495.5	27517.5	454500.0	0.035792	

In [86]:

```
defaulters[['SK_ID_CURR','CNT_CHILDREN','AMT_INCOME_TOTAL']].corr()
```

Out[86]:

	SK_ID_CURR	CNT_CHILDREN	AMT_INCOME_TOTAL
SK_ID_CURR	1.000000	-0.005144	-0.010165
CNT_CHILDREN	-0.005144	1.000000	0.004796
AMT_INCOME_TOTAL	-0.010165	0.004796	1.000000

In [87]:

```
defaulter_corr = defaulters.corr()
defaulter_corr_unstck = defaulter_corr.where(np.triu(np.ones(defaulter_corr.shape),k=1).astype(np.bool)).unstack().reset_

defaulter_corr_unstck['corr']= abs(defaulter_corr_unstck['corr'])
defaulter_corr_unstck.dropna(subset=['corr']).sort_values(by=['corr'],ascending=False).head(10)
```

```
C:\Users\harsh\AppData\Local\Temp\ipykernel_17260\3999762344.py:2: DeprecationWarning: `np.bool` is a deprecated alias fo
r the builtin `bool`. To silence this warning, use `bool` by itself. Doing this will not modify any behavior and is safe.
If you specifically wanted the numpy scalar type, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
  defaulter_corr_unstck = defaulter_corr.where(np.triu(np.ones(defaulter_corr.shape),k=1).astype(np.bool)).unstack().rese
t_index().rename(columns={'level_0':'var1','level_1':'var2',0:'corr'})
```

Out[87]:

	var1	var2	corr
757	OBS_60_CNT_SOCIAL_CIRCLE	OBS_30_CNT_SOCIAL_CIRCLE	0.998269
163	AMT_GOODS_PRICE	AMT_CREDIT	0.982783
428	REGION_RATING_CLIENT_W_CITY	REGION_RATING_CLIENT	0.956637
353	CNT_FAM_MEMBERS	CNT_CHILDREN	0.885484
790	DEF_60_CNT_SOCIAL_CIRCLE	DEF_30_CNT_SOCIAL_CIRCLE	0.868994
560	LIVE_REGION_NOT_WORK_REGION	REG_REGION_NOT_WORK_REGION	0.847885
659	LIVE_CITY_NOT_WORK_CITY	REG_CITY_NOT_WORK_CITY	0.778540

	var1	var2	corr
164	AMT_GOODS_PRICE	AMT_ANNUITY	0.752295
131	AMT_ANNUITY	AMT_CREDIT	0.752195
263	DAYS_EMPLOYED	DAYS_BIRTH	0.582185

In [88]:

```
repayers_corr = repayers.corr()
repayers_corr_unstck = repayers_corr.where(np.triu(np.ones(repayers_corr.shape),k=1).astype(np.bool)).unstack().reset_inc

repayers_corr_unstck['corr']= abs(repayers_corr_unstck['corr'])
repayers_corr_unstck.dropna(subset=['corr']).sort_values(by=['corr'],ascending=False).head(10)
```

```
C:\Users\harsh\AppData\Local\Temp\ipykernel_17260\1297184003.py:2: DeprecationWarning: `np.bool` is a deprecated alias fo
r the builtin `bool`. To silence this warning, use `bool` by itself. Doing this will not modify any behavior and is safe.
If you specifically wanted the numpy scalar type, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
  repayers_corr_unstck = repayers_corr.where(np.triu(np.ones(repayers_corr.shape),k=1).astype(np.bool)).unstack().reset_i
ndex().rename(columns={'level_0':'var1','level_1':'var2',0:'corr'})
```

Out[88]:

	var1	var2	corr
757	OBS_60_CNT_SOCIAL_CIRCLE	OBS_30_CNT_SOCIAL_CIRCLE	0.998508
163	AMT_GOODS_PRICE	AMT_CREDIT	0.987022
428	REGION_RATING_CLIENT_W_CITY	REGION_RATING_CLIENT	0.950149
353	CNT_FAM_MEMBERS	CNT_CHILDREN	0.878571
560	LIVE_REGION_NOT_WORK_REGION	REG_REGION_NOT_WORK_REGION	0.861861
790	DEF_60_CNT_SOCIAL_CIRCLE	DEF_30_CNT_SOCIAL_CIRCLE	0.859332
659	LIVE_CITY_NOT_WORK_CITY	REG_CITY_NOT_WORK_CITY	0.830381
164	AMT_GOODS_PRICE	AMT_ANNUITY	0.776421
131	AMT_ANNUITY	AMT_CREDIT	0.771297
263	DAYS_EMPLOYED	DAYS_BIRTH	0.626114

In [89]:

```
num_data.head()
```

Out[89]:

	SK_ID_CURR	TARGET	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_PRICE	REGION_POPULATION_RELATI
0	100002	1	0	202500.0	406597.5	24700.5	351000.0	0.0188
1	100003	0	0	270000.0	1293502.5	35698.5	1129500.0	0.0035
2	100004	0	0	67500.0	135000.0	6750.0	135000.0	0.0100
3	100006	0	0	135000.0	312682.5	29686.5	297000.0	0.0080
4	100007	0	0	121500.0	513000.0	21865.5	513000.0	0.0286

In [90]:

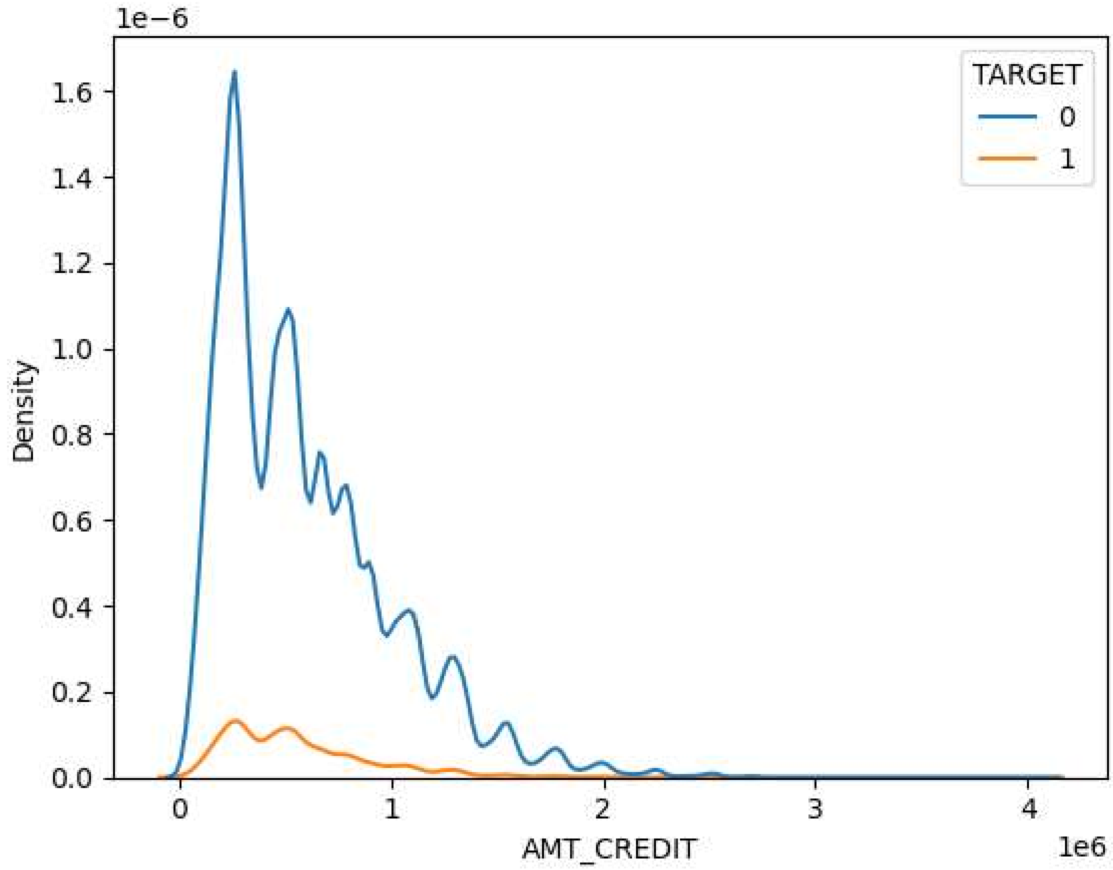
```
amt_var = ['AMT_INCOME_TOTAL', 'AMT_CREDIT','AMT_ANNUITY','AMT_GOODS_PRICE']
```

In [91]:

```
sns.kdeplot(data=num_data, x='AMT_CREDIT', hue='TARGET')
```

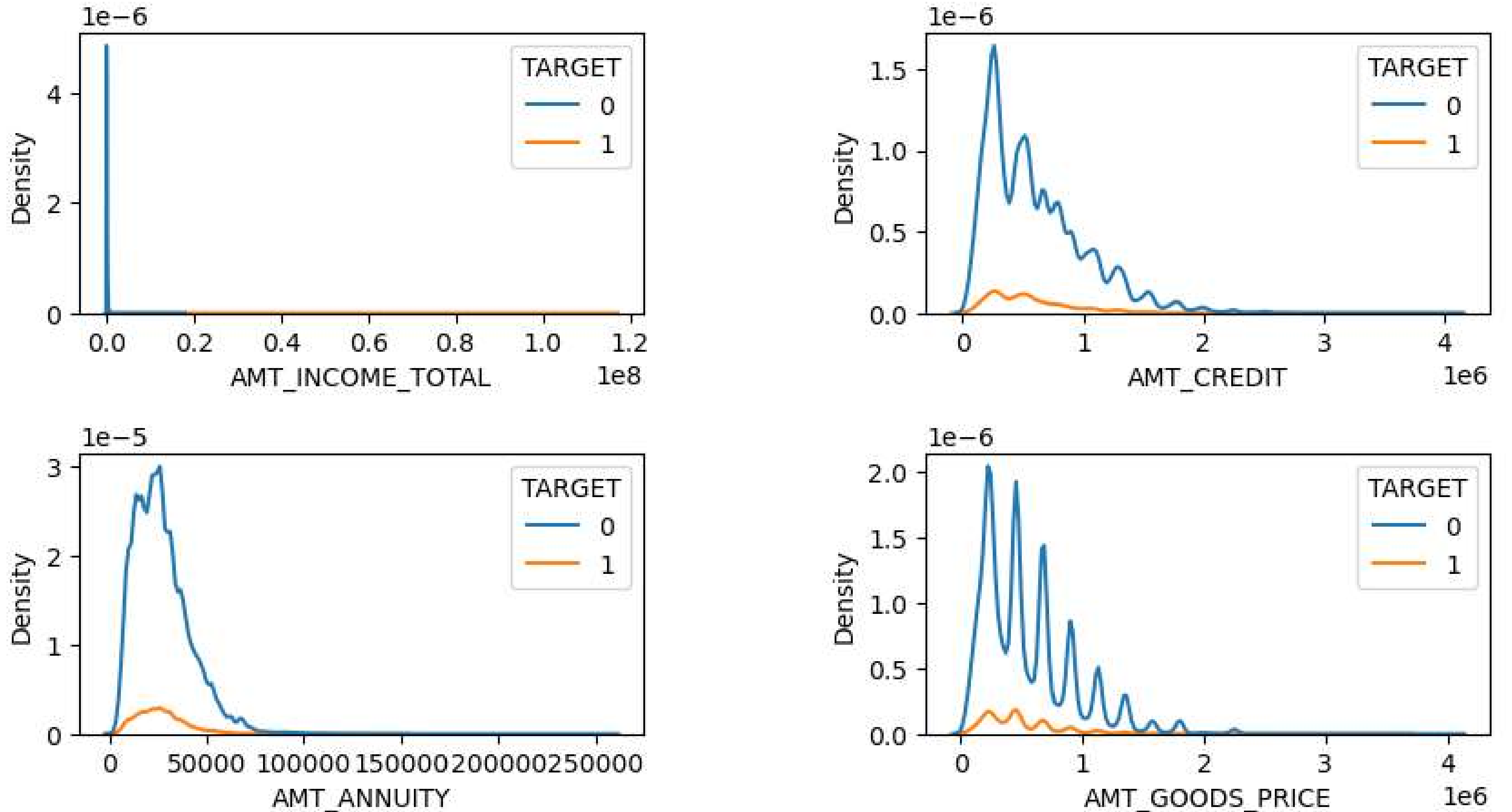
Out[91]:

```
<Axes: xlabel='AMT_CREDIT', ylabel='Density'>
```



In [92]:

```
plt.figure(figsize=(10,5))

for i, col in enumerate(amt_var):
    plt.subplot(2,2,i+1)
    sns.kdeplot(data=num_data,x=col,hue='TARGET')
    plt.subplots_adjust(wspace=0.5,hspace=0.5)
```



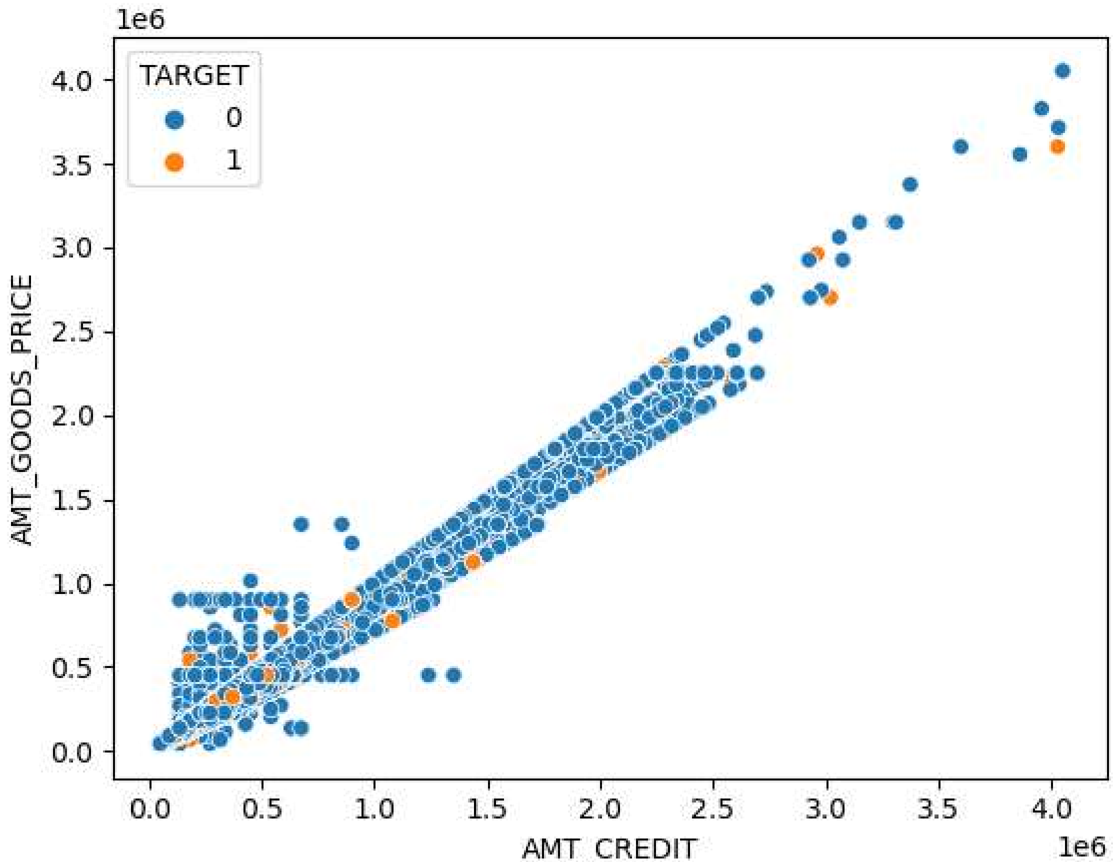
In [93]:
```
num_data.head()
```

Out[93]:

	SK_ID_CURR	TARGET	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_PRICE	REGION_POPULATION_RELATI
0	100002	1	0	202500.0	406597.5	24700.5	351000.0	0.0188
1	100003	0	0	270000.0	1293502.5	35698.5	1129500.0	0.0035
2	100004	0	0	67500.0	135000.0	6750.0	135000.0	0.0100
3	100006	0	0	135000.0	312682.5	29686.5	297000.0	0.0080
4	100007	0	0	121500.0	513000.0	21865.5	513000.0	0.0286

In [94]:
```
sns.scatterplot(data=num_data,x='AMT_CREDIT',y='AMT_GOODS_PRICE', hue='TARGET')
```

Out[94]: <Axes: xlabel='AMT_CREDIT', ylabel='AMT_GOODS_PRICE'>



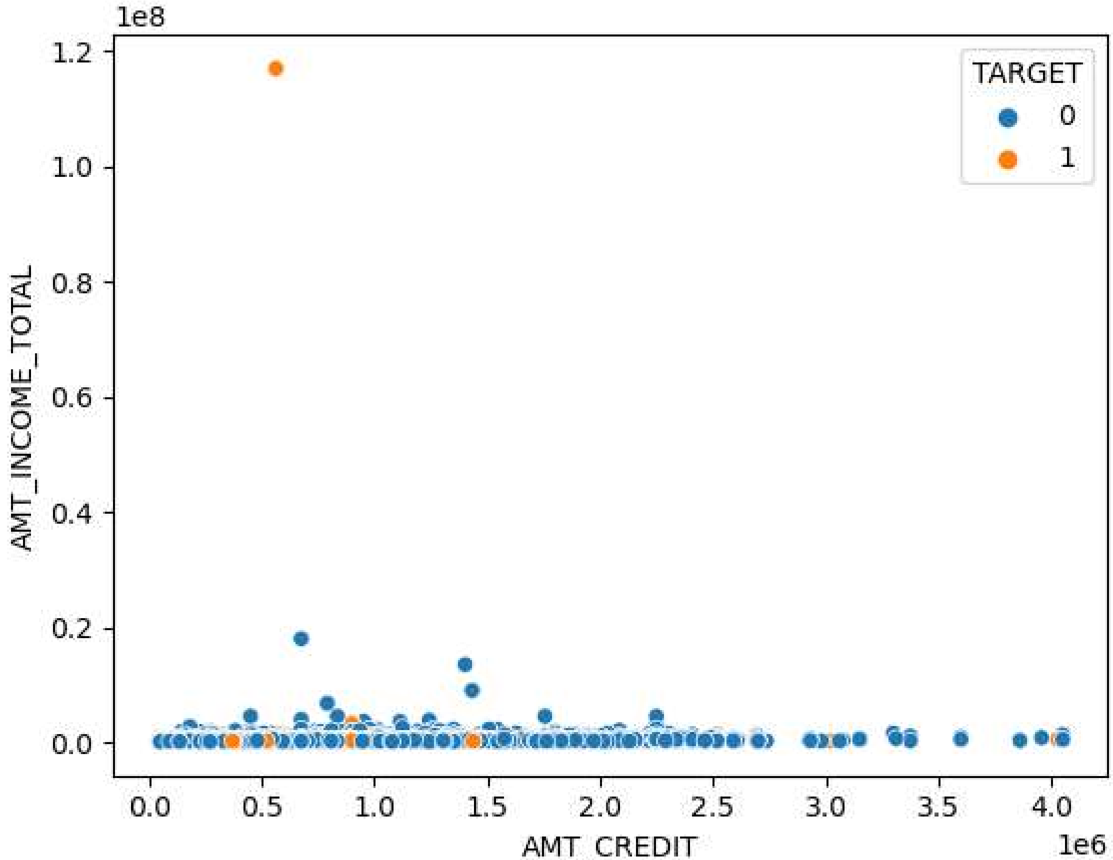
In [128…
```
# num_data[['AMT_CREDIT','AMT_INCOME_TOTAL']].corr()
```

Out[128…

	AMT_CREDIT	AMT_INCOME_TOTAL
AMT_CREDIT	1.00000	0.15687
AMT_INCOME_TOTAL	0.15687	1.00000

In [95]:
```
sns.scatterplot(data=num_data,x='AMT_CREDIT',y='AMT_INCOME_TOTAL', hue='TARGET')
```

Out[95]: <Axes: xlabel='AMT_CREDIT', ylabel='AMT_INCOME_TOTAL'>



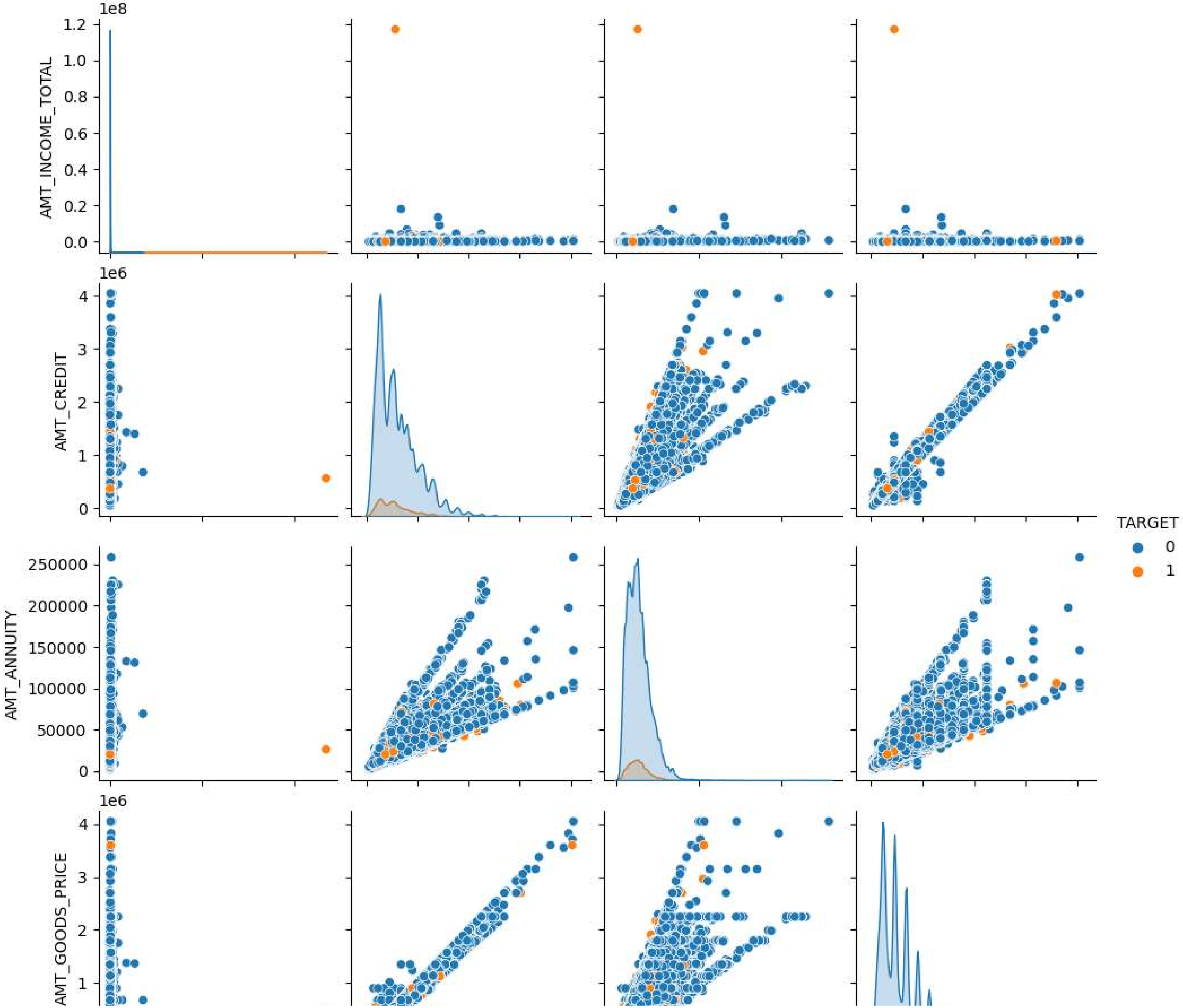
In [96]:

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

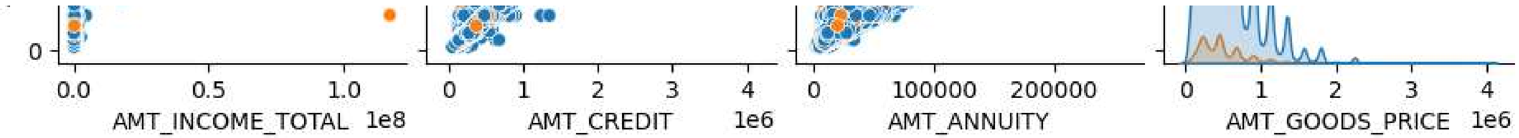
In [97]:

```
sns.pairplot(data=amt_var,hue='TARGET')
```

Out[97]: <seaborn.axisgrid.PairGrid at 0x1b6c206ff70>



1e8
1.2
1.0
0.8
0.6
0.4
0.2
0.0
AMT_INCOME_TOTAL
1e6
4
3
2
1
0
AMT_CREDIT
250000
200000
150000
100000
50000
0
AMT_ANNUITY
1e6
4
3
2
1
AMT_GOODS_PRICE
TARGET
0
1



In [98]:

```
null_count = pd.DataFrame(prev_app.isnull().sum().sort_values(ascending=False)/prev_app.shape[0]*100).reset_index().renam

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

Out[98]:

```
['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 [99]:

```
nva_cols=var_msng_ge_40+['WEEKDAY_APPR_PROCESS_START','HOUR_APPR_PROCESS_START','FLAG_LAST_APPL_PER_CONTRACT','NFLAG_LAST
len(nva_cols)
```

Out[99]:

```
15
```

In [100…

```
len(prev_app.columns)
```

Out[100…

```
37
```

In [101…

```
prev_app_nva_col_rmvd = prev_app.drop(labels=nva_cols, axis=1)


len(prev_app_nva_col_rmvd.columns)
```

Out[101…

```
22
```

In [102…

```
prev_app_nva_col_rmvd.columns
```

Out[102…

```
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 [103…

```
prev_app_nva_col_rmvd.head()
```

Out[103…

	SK_ID_PREV	SK_ID_CURR	NAME_CONTRACT_TYPE	AMT_ANNUITY	AMT_APPLICATION	AMT_CREDIT	AMT_GOODS_PRICE	NAME_CASH_LOAN
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 [104…

```
prev_app_nva_col_rmvd.isnull().sum().sort_values(ascending=False)/prev_app_nva_col_rmvd.shape[0]*100
```

Out[104…

```
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 [105…

```
prev_app_nva_col_rmvd['AMT_GOODS_PRICE'].agg(func=['mean','median'])
```

Out[105…

```
mean      227847.279283
median    112320.000000
Name: AMT_GOODS_PRICE, dtype: float64
```

In [106…

```
prev_app_nva_col_rmvd['AMT_GOODS_PRICE_MEDIAN']=prev_app_nva_col_rmvd['AMT_GOODS_PRICE'].fillna(prev_app_nva_col_rmvd['AM
```

In [107…

```
prev_app_nva_col_rmvd['AMT_GOODS_PRICE_MEAN']=prev_app_nva_col_rmvd['AMT_GOODS_PRICE'].fillna(prev_app_nva_col_rmvd['AMT_
```

In [108…

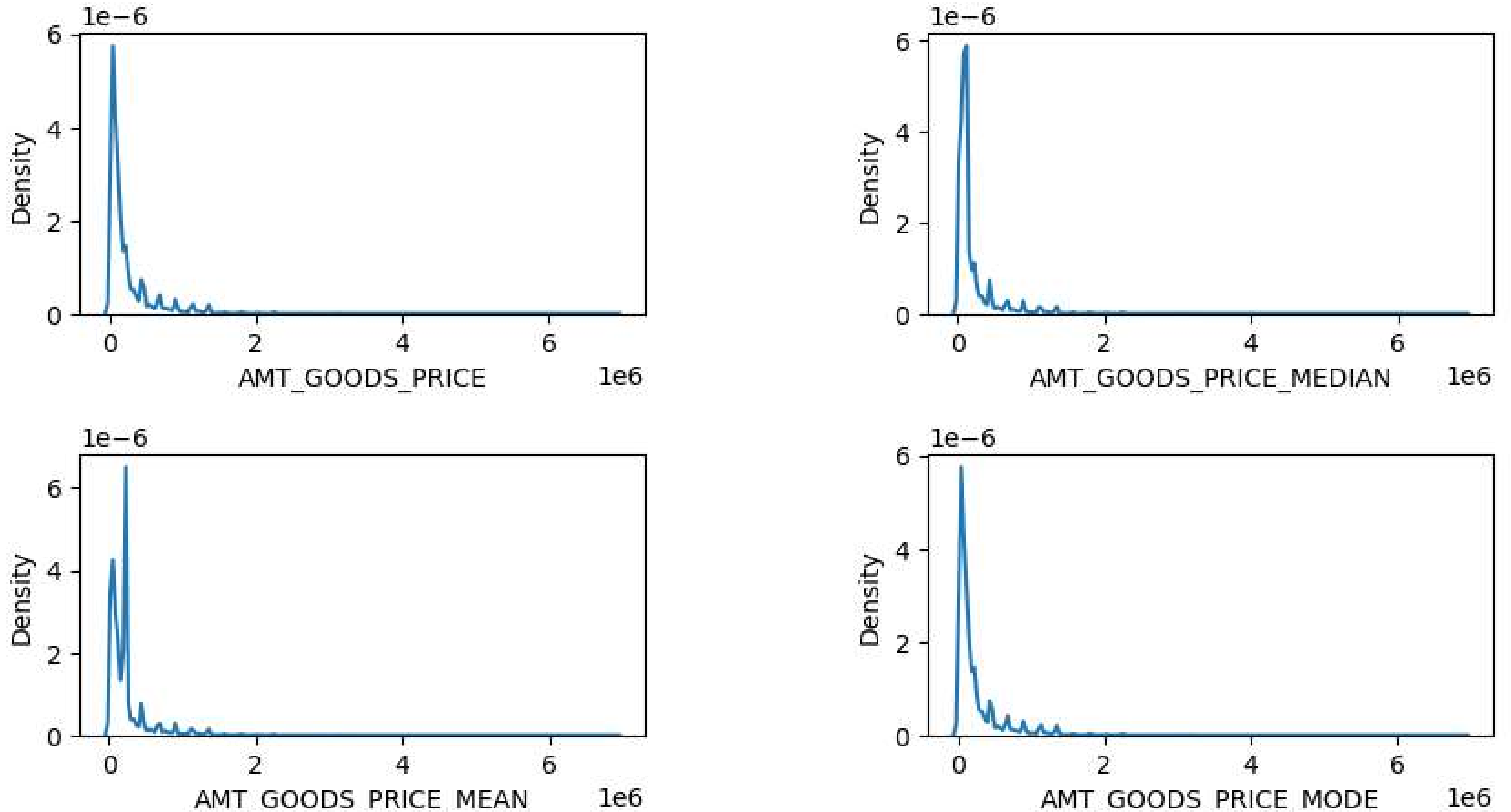
```
prev_app_nva_col_rmvd['AMT_GOODS_PRICE_MODE']=prev_app_nva_col_rmvd['AMT_GOODS_PRICE'].fillna(prev_app_nva_col_rmvd['AMT_
```

In [109…

```
gp_cols = ['AMT_GOODS_PRICE','AMT_GOODS_PRICE_MEDIAN','AMT_GOODS_PRICE_MEAN','AMT_GOODS_PRICE_MODE']
```

In [110…

```
plt.figure(figsize=(10,5))

for i, col in enumerate(gp_cols):
    plt.subplot(2,2,i+1)
    sns.kdeplot(data=prev_app_nva_col_rmvd,x=col)
    plt.subplots_adjust(wspace=0.5,hspace=0.5)
```



In [111…

```
prev_app_nva_col_rmvd['AMT_GOODS_PRICE']=prev_app_nva_col_rmvd['AMT_GOODS_PRICE'].fillna(prev_app_nva_col_rmvd['AMT_GOODS
```

In [112…

```
prev_app_nva_col_rmvd['AMT_GOODS_PRICE'].isnull().sum()
```

Out[112…

```
0
```

In [113…

```
prev_app_nva_col_rmvd['AMT_ANNUITY'].agg(func=['mean','median','max'])
```

Out[113…

```
mean       15955.120659
median     11250.000000
max       418058.145000
Name: AMT_ANNUITY, dtype: float64
```

In [114…

```
prev_app_nva_col_rmvd['AMT_ANNUITY']=prev_app_nva_col_rmvd['AMT_ANNUITY'].fillna(prev_app_nva_col_rmvd['AMT_ANNUITY'].mec
```

In [115…

```
#prev_app_nva_col_rmvd['PRODUCT_COMBINATION'].head()
prev_app_nva_col_rmvd['PRODUCT_COMBINATION'] = prev_app_nva_col_rmvd['PRODUCT_COMBINATION'].fillna(prev_app_nva_col_rmvd[
```

In [116…

```
prev_app_nva_col_rmvd['CNT_PAYMENT'].agg(func=['mean','median','max'])

prev_app_nva_col_rmvd[prev_app_nva_col_rmvd['CNT_PAYMENT'].isnull()].groupby(['NAME_CONTRACT_STATUS']).size().sort_values
```

Out[116…

```
NAME_CONTRACT_STATUS
Canceled        305805
Refused          40897
Unused offer     25524
Approved             4
dtype: int64
```

In [117…

```
prev_app_nva_col_rmvd['CNT_PAYMENT'] = prev_app_nva_col_rmvd['CNT_PAYMENT'].fillna(0)
```

In [118…

```
prev_app_nva_col_rmvd.isnull().sum().sort_values(ascending=False)
```

Out[118…

```
AMT_GOODS_PRICE_MODE      385515
AMT_CREDIT                     1
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
SK_ID_PREV                     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
NAME_CLIENT_TYPE             0
dtype: int64
```

In [119…

```
prev_app_nva_col_rmvd=prev_app_nva_col_rmvd.drop(labels=['AMT_GOODS_PRICE_MEDIAN','AMT_GOODS_PRICE_MEAN','AMT_GOODS_PRICE
```

In [120…

```
prev_app_nva_col_rmvd.isnull().sum().sort_values(ascending=False)
```

Out[120…

```
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 [121…

```
len(prev_app_nva_col_rmvd.columns)
```

Out[121…

```
22
```

In [122…

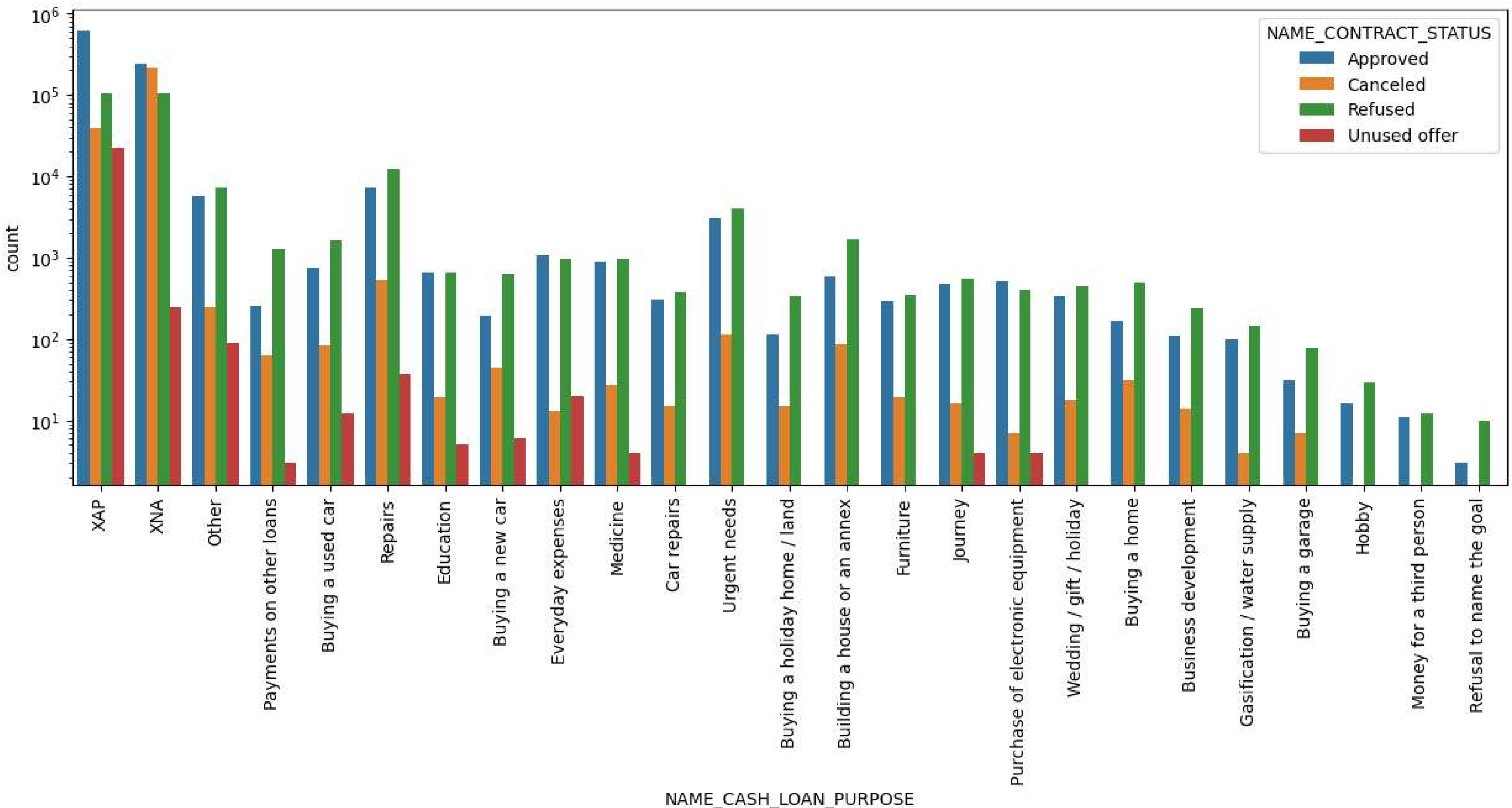
```
merged_df = pd.merge(app_score_col_rmvd,prev_app_nva_col_rmvd, how='inner', on='SK_ID_CURR')
merged_df.head()
```

Out[122...

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE_x	CODE_GENDER	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT_x	AMT_ANNUITY_x	AM
0	100002	1	Cash loans	M	0	202500.0	406597.5	24700.5	
1	100003	0	Cash loans	F	0	270000.0	1293502.5	35698.5	
2	100003	0	Cash loans	F	0	270000.0	1293502.5	35698.5	
3	100003	0	Cash loans	F	0	270000.0	1293502.5	35698.5	
4	100004	0	Revolving loans	M	0	67500.0	135000.0	6750.0	

In [123...

```
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')
```

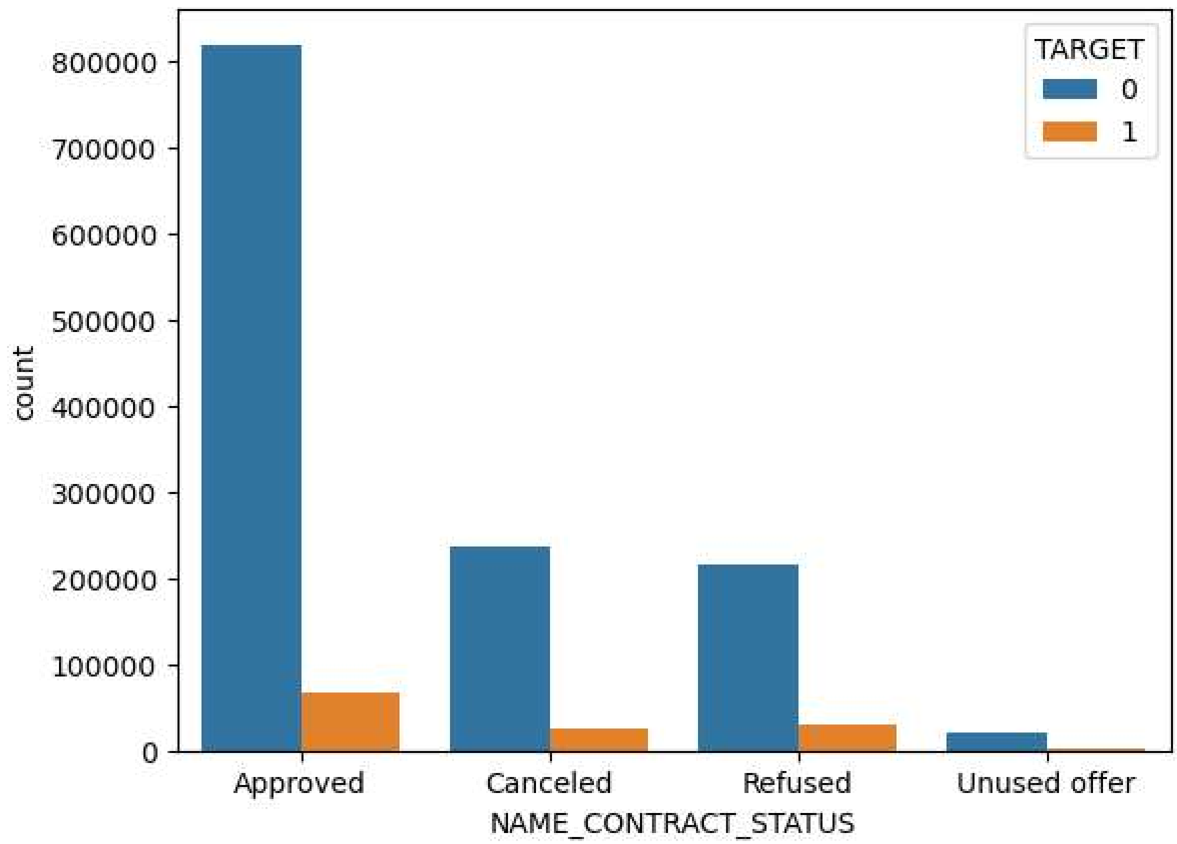


In [124...

```
sns.countplot(data=merged_df, x='NAME_CONTRACT_STATUS', hue='TARGET')
```

Out[124...

```
<Axes: xlabel='NAME_CONTRACT_STATUS', ylabel='count'>
```



In [125…

```
merged_agg = merged_df.groupby(['NAME_CONTRACT_STATUS','TARGET']).size().reset_index().rename(columns={0:'counts'})
sum_df = merged_agg.groupby(['NAME_CONTRACT_STATUS','TARGET'])['counts'].sum().reset_index()

merged_agg_2 = pd.merge(merged_agg,sum_df,how='left', on = 'NAME_CONTRACT_STATUS')
merged_agg_2['pct'] = round(merged_agg_2['counts_x']/merged_agg_2['counts_y']*100,2)
merged_agg_2
```

Out[125…

	NAME_CONTRACT_STATUS	TARGET_x	counts_x	TARGET_y	counts_y	pct
0	Approved	0	818856	0	818856	100.00
1	Approved	0	818856	1	67243	1217.76
2	Approved	1	67243	0	818856	8.21
3	Approved	1	67243	1	67243	100.00

	NAME_CONTRACT_STATUS	TARGET_x	counts_x	TARGET_y	counts_y	pct
4	Canceled	0	235641	0	235641	100.00
5	Canceled	0	235641	1	23800	990.09
6	Canceled	1	23800	0	235641	10.10
7	Canceled	1	23800	1	23800	100.00
8	Refused	0	215952	0	215952	100.00
9	Refused	0	215952	1	29438	733.58
10	Refused	1	29438	0	215952	13.63
11	Refused	1	29438	1	29438	100.00
12	Unused offer	0	20892	0	20892	100.00
13	Unused offer	0	20892	1	1879	1111.87
14	Unused offer	1	1879	0	20892	8.99
15	Unused offer	1	1879	1	1879	100.00

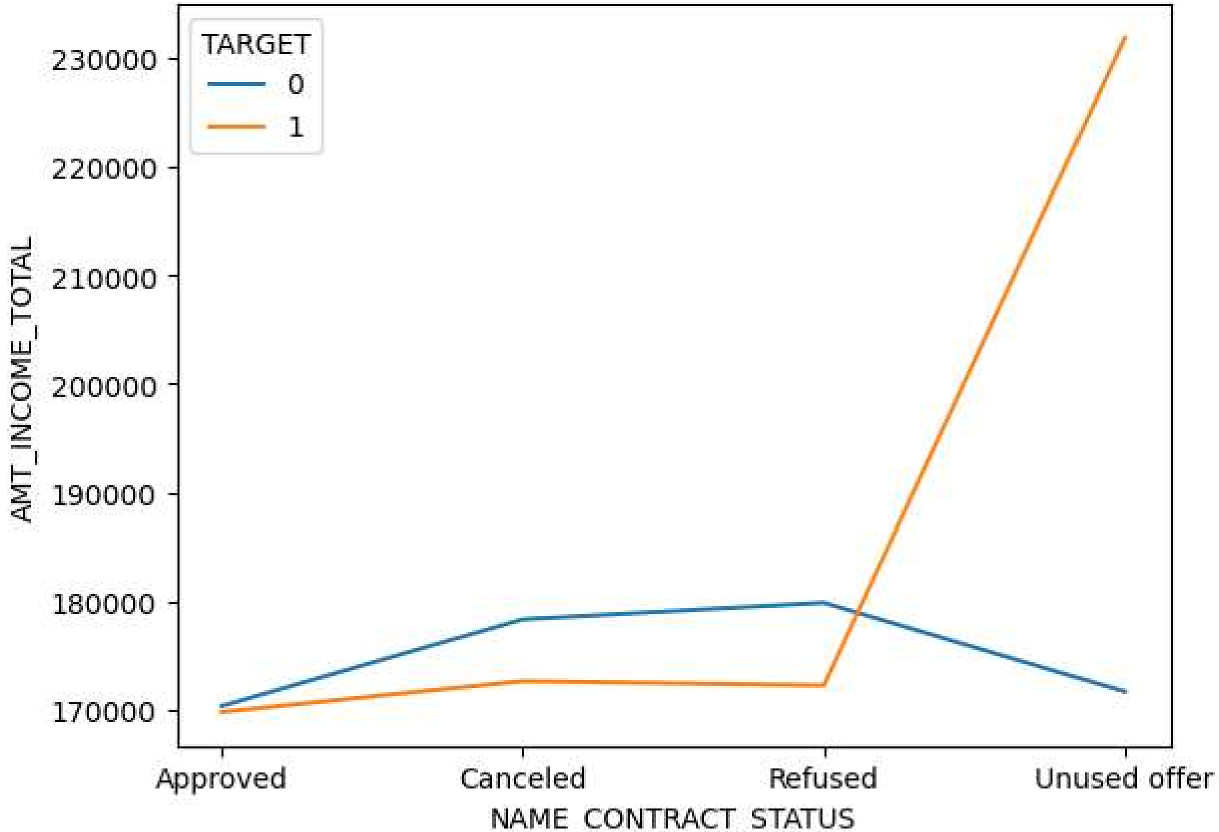
In [126…

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

```
C:\Users\harsh\AppData\Local\Temp\ipykernel_17260\4020938878.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[126…

```
<Axes: xlabel='NAME_CONTRACT_STATUS', ylabel='AMT_INCOME_TOTAL'>
```



In [127…

```
len(merged_df.columns)
```

Out[127…

```
69
```

Conclusion/Insights

Now we will be using different classifcation models to predict the loan default

In [ ]:

```

```