In [1]:
```
from google.colab import drive
drive.mount("/content/MyDrive/")
```

```
Mounted at /content/MyDrive/
```

DESCRIPTION

For safe and secure lending experience, it's important to analyze the past data. In this project, you have to build a deep learning model to predict the chance of default for future loans using the historical data. As you will see, this dataset is highly imbalanced and includes a lot of features that make this problem more challenging.

Objective:

Create a model that predicts whether or not an applicant will be able to repay a loan using historical data.

Analysis to be done:

Perform data preprocessing and build a deep learning prediction model.

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
pd.set_option("display.max_columns",None)
pd.set_option("display.max_rows",None)
```

Load the dataset that is given to you

In [3]:
```
df = pd.read_csv("/content/MyDrive/MyDrive/loan_data (1).csv")
```

In [4]:
```
df.head()
```

Out[4]:

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

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Columns: 122 entries, SK_ID_CURR to AMT_REQ_CREDIT_BUREAU_YEAR
dtypes: float64(65), int64(41), object(16)
memory usage: 286.2+ MB
```

In [6]:
```
df.shape
```

Out[6]:
```
(307511, 122)
```

In [7]:
```
df.isnull().sum()
```

Out[7]:
```
SK_ID_CURR                           0
TARGET                               0
NAME_CONTRACT_TYPE                   0
CODE_GENDER                          0
FLAG_OWN_CAR                         0
FLAG_OWN_REALTY                      0
CNT_CHILDREN                         0
AMT_INCOME_TOTAL                     0
AMT_CREDIT                           0
AMT_ANNUITY                         12
AMT_GOODS_PRICE                    278
NAME_TYPE_SUITE                   1292
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
DAYS_REGISTRATION                    0
DAYS_ID_PUBLISH                      0
OWN_CAR_AGE                     202929
FLAG_MOBIL                           0
FLAG_EMP_PHONE                       0
FLAG_WORK_PHONE                      0
FLAG_CONT_MOBILE                     0
FLAG_PHONE                           0
FLAG_EMAIL                           0
OCCUPATION_TYPE                  96391
CNT_FAM_MEMBERS                      2
REGION_RATING_CLIENT                 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
EXT_SOURCE_1                    173378
EXT_SOURCE_2                       660
EXT_SOURCE_3                     60965
APARTMENTS_AVG                  156061
BASEMENTAREA_AVG                179943
YEARS_BEGINEXPLUATATION_AVG     150007
YEARS_BUILD_AVG                 204488
COMMONAREA_AVG                  214865
ELEVATORS_AVG                   163891
ENTRANCES_AVG                   154828
FLOORSMAX_AVG                   153020
FLOORSMIN_AVG                   208642
LANDAREA_AVG                    182590
LIVINGAPARTMENTS_AVG            210199
LIVINGAREA_AVG                  154350
NONLIVINGAPARTMENTS_AVG         213514
NONLIVINGAREA_AVG               169682
APARTMENTS_MODE                 156061
BASEMENTAREA_MODE               179943
```

```
YEARS_BEGINEXPLUATATION_MODE    150007
YEARS_BUILD_MODE                204488
COMMONAREA_MODE                 214865
ELEVATORS_MODE                  163891
ENTRANCES_MODE                  154828
FLOORSMAX_MODE                  153020
FLOORSMIN_MODE                  208642
LANDAREA_MODE                   182590
LIVINGAPARTMENTS_MODE           210199
LIVINGAREA_MODE                 154350
NONLIVINGAPARTMENTS_MODE        213514
NONLIVINGAREA_MODE              169682
APARTMENTS_MEDI                 156061
BASEMENTAREA_MEDI               179943
YEARS_BEGINEXPLUATATION_MEDI    150007
YEARS_BUILD_MEDI                204488
COMMONAREA_MEDI                 214865
ELEVATORS_MEDI                  163891
ENTRANCES_MEDI                  154828
FLOORSMAX_MEDI                  153020
FLOORSMIN_MEDI                  208642
LANDAREA_MEDI                   182590
LIVINGAPARTMENTS_MEDI           210199
LIVINGAREA_MEDI                 154350
NONLIVINGAPARTMENTS_MEDI        213514
NONLIVINGAREA_MEDI              169682
FONDKAPREMONT_MODE              210295
HOUSETYPE_MODE                  154297
TOTALAREA_MODE                  148431
WALLSMATERIAL_MODE              156341
EMERGENCYSTATE_MODE             145755
OBS_30_CNT_SOCIAL_CIRCLE          1021
DEF_30_CNT_SOCIAL_CIRCLE          1021
OBS_60_CNT_SOCIAL_CIRCLE          1021
DEF_60_CNT_SOCIAL_CIRCLE          1021
DAYS_LAST_PHONE_CHANGE               1
FLAG_DOCUMENT_2                      0
FLAG_DOCUMENT_3                      0
FLAG_DOCUMENT_4                      0
FLAG_DOCUMENT_5                      0
FLAG_DOCUMENT_6                      0
FLAG_DOCUMENT_7                      0
FLAG_DOCUMENT_8                      0
FLAG_DOCUMENT_9                      0
FLAG_DOCUMENT_10                     0
FLAG_DOCUMENT_11                     0
FLAG_DOCUMENT_12                     0
FLAG_DOCUMENT_13                     0
FLAG_DOCUMENT_14                     0
FLAG_DOCUMENT_15                     0
FLAG_DOCUMENT_16                     0
FLAG_DOCUMENT_17                     0
FLAG_DOCUMENT_18                     0
FLAG_DOCUMENT_19                     0
FLAG_DOCUMENT_20                     0
FLAG_DOCUMENT_21                     0
AMT_REQ_CREDIT_BUREAU_HOUR       41519
AMT_REQ_CREDIT_BUREAU_DAY        41519
AMT_REQ_CREDIT_BUREAU_WEEK       41519
AMT_REQ_CREDIT_BUREAU_MON        41519
```

```
AMT_REQ_CREDIT_BUREAU_QRT        41519
AMT_REQ_CREDIT_BUREAU_YEAR       41519
dtype: int64
```

Check for null values in the dataset

In [8]:
```
df = df.drop(['AMT_ANNUITY','AMT_GOODS_PRICE','NAME_TYPE_SUITE','OWN_CAR_AGE','OCCUPATION_TYPE','CNT_FAM_MEMBERS','EXT_SOURCE_1','EXT_SOURCE_2',
         'EXT_SOURCE_3','APARTMENTS_AVG','BASEMENTAREA_AVG','YEARS_BEGINEXPLUATATION_AVG','YEARS_BUILD_AVG','COMMONAREA_AVG','ELEVATORS_AVG',
         'ENTRANCES_AVG','FLOORSMAX_AVG','FLOORSMIN_AVG','LANDAREA_AVG','LIVINGAPARTMENTS_AVG','LIVINGAREA_AVG','NONLIVINGAPARTMENTS_AVG',
         'NONLIVINGAREA_AVG','APARTMENTS_MODE','BASEMENTAREA_MODE','YEARS_BEGINEXPLUATATION_MODE','YEARS_BUILD_MODE','COMMONAREA_MODE','ELEVATORS_MODE',
         'ENTRANCES_MODE','FLOORSMAX_MODE','FLOORSMIN_MODE','LANDAREA_MODE','LIVINGAPARTMENTS_MODE','LIVINGAREA_MODE','NONLIVINGAPARTMENTS_MODE',
         'NONLIVINGAREA_MODE','APARTMENTS_MEDI','BASEMENTAREA_MEDI','YEARS_BEGINEXPLUATATION_MEDI','YEARS_BUILD_MEDI','COMMONAREA_MEDI','ELEVATORS_MEDI',
         'ENTRANCES_MEDI','FLOORSMAX_MEDI','FLOORSMIN_MEDI','LANDAREA_MEDI','LIVINGAPARTMENTS_MEDI','LIVINGAREA_MEDI','NONLIVINGAPARTMENTS_MEDI','NONLIVINGAREA_MEDI',
         'FONDKAPREMONT_MODE','HOUSETYPE_MODE','TOTALAREA_MODE','WALLSMATERIAL_MODE','EMERGENCYSTATE_MODE','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'],axis=1)
```

In [9]:
```
df.isnull().sum()
```

Out[9]:
```
SK_ID_CURR                         0
TARGET                             0
NAME_CONTRACT_TYPE                 0
CODE_GENDER                        0
FLAG_OWN_CAR                       0
FLAG_OWN_REALTY                    0
CNT_CHILDREN                       0
AMT_INCOME_TOTAL                   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
DAYS_REGISTRATION                  0
DAYS_ID_PUBLISH                    0
FLAG_MOBIL                         0
FLAG_EMP_PHONE                     0
FLAG_WORK_PHONE                    0
FLAG_CONT_MOBILE                   0
FLAG_PHONE                         0
FLAG_EMAIL                         0
REGION_RATING_CLIENT               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
FLAG_DOCUMENT_2                    0
FLAG_DOCUMENT_3                    0
FLAG_DOCUMENT_4                    0
FLAG_DOCUMENT_5                    0
FLAG_DOCUMENT_6                    0
FLAG_DOCUMENT_7                    0
FLAG_DOCUMENT_8                    0
FLAG_DOCUMENT_9                    0
FLAG_DOCUMENT_10                   0
FLAG_DOCUMENT_11                   0
FLAG_DOCUMENT_12                   0
FLAG_DOCUMENT_13                   0
FLAG_DOCUMENT_14                   0
FLAG_DOCUMENT_15                   0
FLAG_DOCUMENT_16                   0
FLAG_DOCUMENT_17                   0
FLAG_DOCUMENT_18                   0
FLAG_DOCUMENT_19                   0
FLAG_DOCUMENT_20                   0
FLAG_DOCUMENT_21                   0
AMT_REQ_CREDIT_BUREAU_YEAR     41519
dtype: int64
```

In [10]:
```
df = df.dropna(axis = 0, how ='any')
```

In [11]:
```
df.shape
```

Out[11]:
```
(265992, 56)
```

In [12]:
```
df.isnull().sum()
```

Out[12]:
```
SK_ID_CURR                     0
TARGET                         0
NAME_CONTRACT_TYPE             0
CODE_GENDER                    0
FLAG_OWN_CAR                   0
FLAG_OWN_REALTY                0
CNT_CHILDREN                   0
AMT_INCOME_TOTAL               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
DAYS_REGISTRATION              0
DAYS_ID_PUBLISH                0
FLAG_MOBIL                     0
FLAG_EMP_PHONE                 0
FLAG_WORK_PHONE                0
FLAG_CONT_MOBILE               0
FLAG_PHONE                     0
FLAG_EMAIL                     0
REGION_RATING_CLIENT           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
FLAG_DOCUMENT_2                0
FLAG_DOCUMENT_3                0
FLAG_DOCUMENT_4                0
FLAG_DOCUMENT_5                0
FLAG_DOCUMENT_6                0
FLAG_DOCUMENT_7                0
FLAG_DOCUMENT_8                0
FLAG_DOCUMENT_9                0
FLAG_DOCUMENT_10               0
FLAG_DOCUMENT_11               0
FLAG_DOCUMENT_12               0
FLAG_DOCUMENT_13               0
FLAG_DOCUMENT_14               0
FLAG_DOCUMENT_15               0
FLAG_DOCUMENT_16               0
FLAG_DOCUMENT_17               0
FLAG_DOCUMENT_18               0
FLAG_DOCUMENT_19               0
FLAG_DOCUMENT_20               0
FLAG_DOCUMENT_21               0
AMT_REQ_CREDIT_BUREAU_YEAR     0
dtype: int64
```

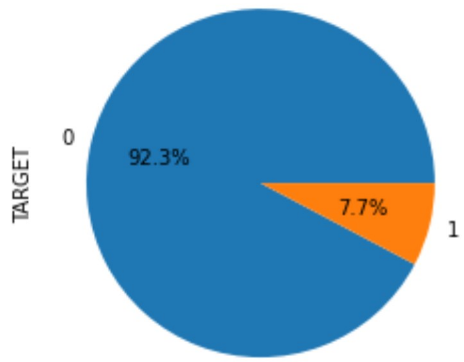
In [13]:
```
df['TARGET'].value_counts()
```

Out[13]:
```
0    245459
1     20533
Name: TARGET, dtype: int64
```

Print percentage of default to payer of the dataset for the TARGET column

In [14]:
```
df['TARGET'].value_counts().plot(kind='pie',autopct='%1.1f%%')
```

Out[14]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0a828a6e50>
```



Balance the dataset if the data is imbalanced

In [15]:
```
# Class count
count_class_0, count_class_1 = df['TARGET'].value_counts()
df_class_0 = df[df['TARGET']==0]
df_class_1 = df[df['TARGET']==1]
```

In [16]:
```
df_class_1_over = df_class_1.sample(count_class_0,replace=True)
```

In [17]:
```
df_over = pd.concat([df_class_0, df_class_1_over], axis=0)
```

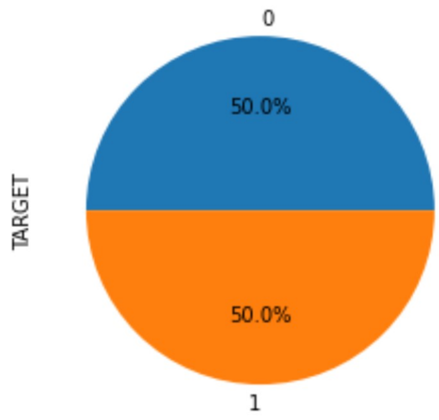
In [18]:
```
print('OverSampling:')
print(df_over['TARGET'].value_counts())
```

```
OverSampling:
0    245459
1    245459
Name: TARGET, dtype: int64
```

Plot the balanced data or imbalanced data

In [19]:
```
df_over['TARGET'].value_counts().plot(kind='pie',autopct='%1.1f%%')
```

Out[19]: <matplotlib.axes._subplots.AxesSubplot at 0x7f0a8296d110>

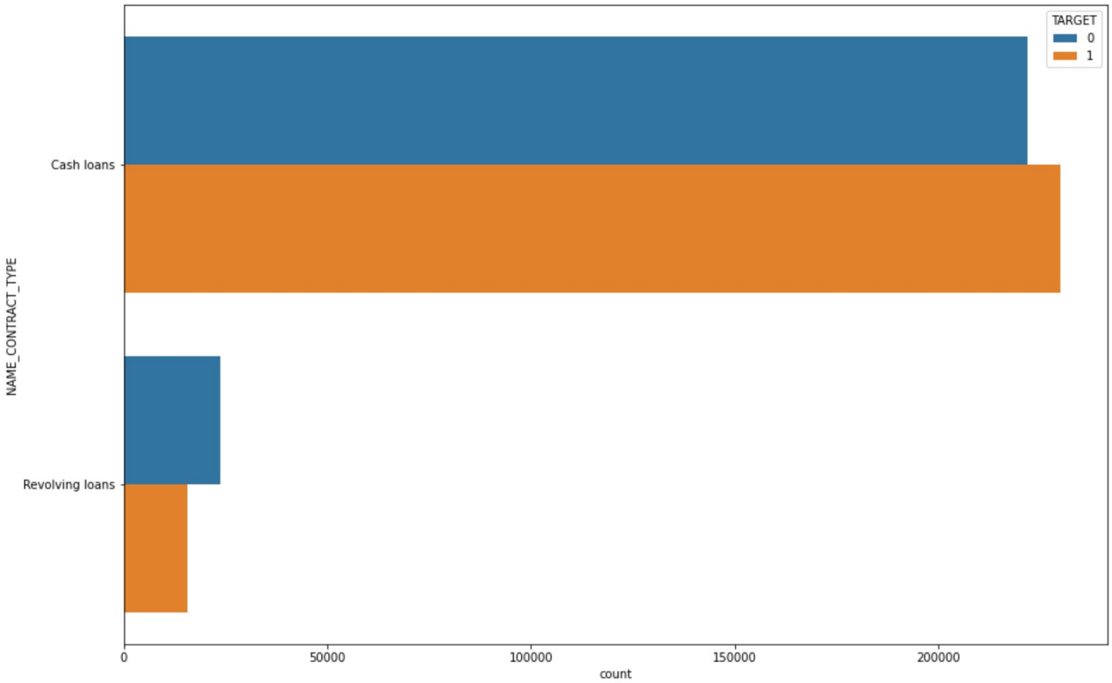


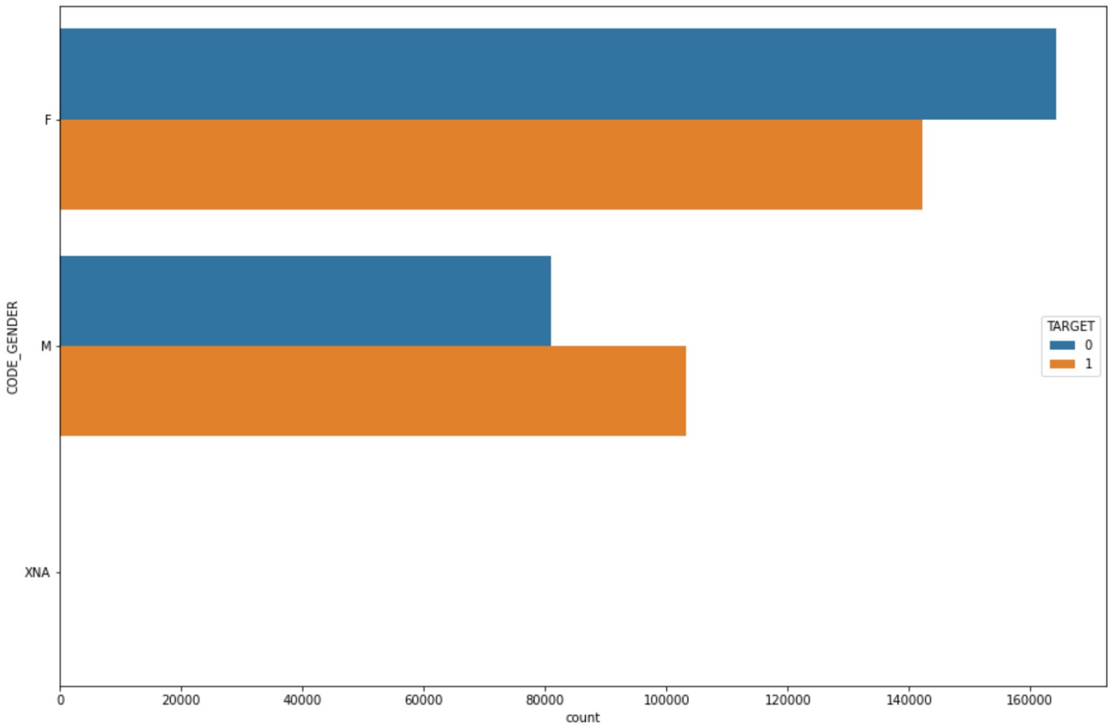
Encode the columns that is required for the model

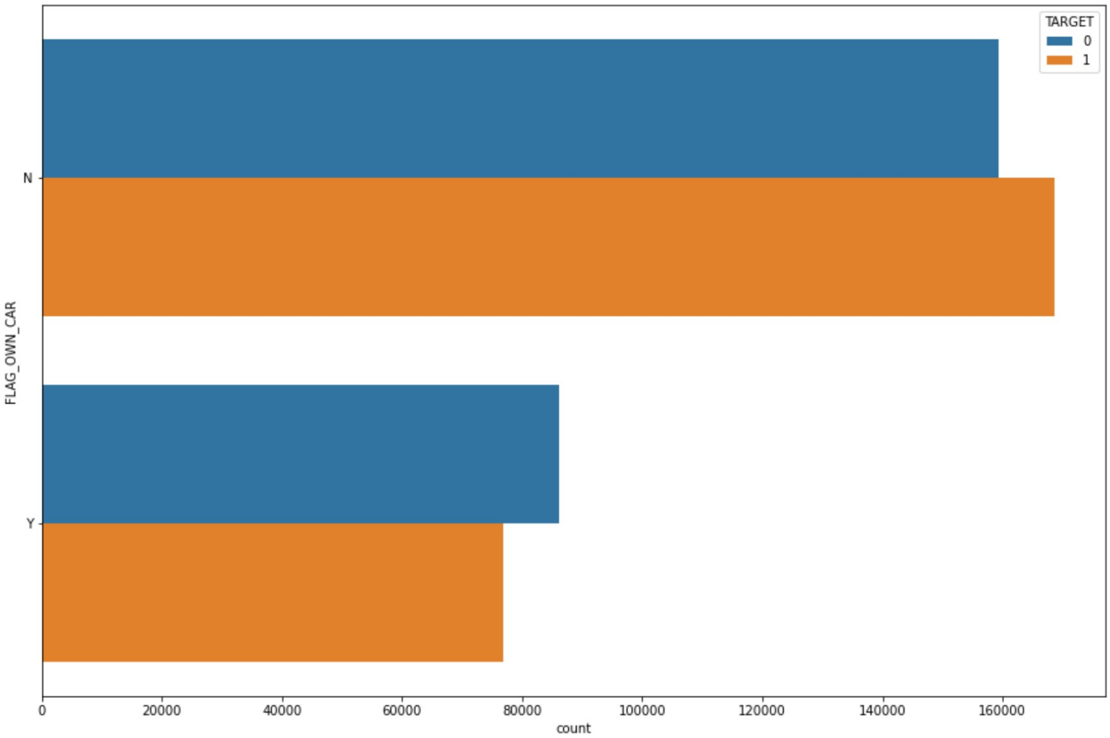
In [20]:
```
for column in df:
  if df[column].dtype == 'object':
    print(f'{column}:  {df[column].unique()}')
    print(f'{column}:  {df[column].nunique()}')
    print
('------------------------------------------------------------------')
```

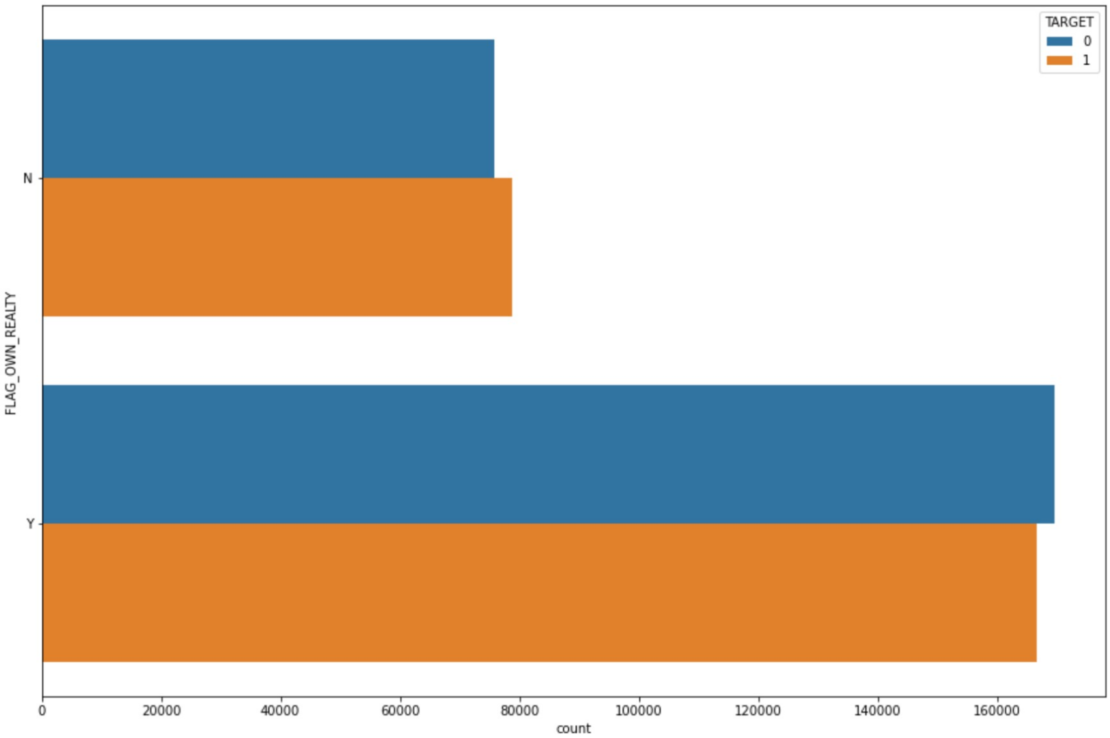
```
NAME_CONTRACT_TYPE:  ['Cash loans' 'Revolving loans']
NAME_CONTRACT_TYPE:  2
-----------------------------------------------------------------
CODE_GENDER:  ['M' 'F' 'XNA']
CODE_GENDER:  3
-----------------------------------------------------------------
FLAG_OWN_CAR:  ['N' 'Y']
FLAG_OWN_CAR:  2
-----------------------------------------------------------------
FLAG_OWN_REALTY:  ['Y' 'N']
FLAG_OWN_REALTY:  2
-----------------------------------------------------------------
NAME_INCOME_TYPE:  ['Working' 'State servant' 'Commercial associate' 'Pe
nsioner' 'Unemployed'
 'Student' 'Businessman' 'Maternity leave']
NAME_INCOME_TYPE:  8
-----------------------------------------------------------------
NAME_EDUCATION_TYPE:  ['Secondary / secondary special' 'Higher education
' 'Incomplete higher'
 'Lower secondary' 'Academic degree']
NAME_EDUCATION_TYPE:  5
-----------------------------------------------------------------
NAME_FAMILY_STATUS:  ['Single / not married' 'Married' 'Widow' 'Civil ma
rriage' 'Separated'
 'Unknown']
NAME_FAMILY_STATUS:  6
-----------------------------------------------------------------
NAME_HOUSING_TYPE:  ['House / apartment' 'Rented apartment' 'Municipal a
partment'
 'With parents' 'Office apartment' 'Co-op apartment']
NAME_HOUSING_TYPE:  6
-----------------------------------------------------------------
WEEKDAY_APPR_PROCESS_START:  ['WEDNESDAY' 'MONDAY' 'THURSDAY' 'SUNDAY' '
SATURDAY' 'FRIDAY' 'TUESDAY']
WEEKDAY_APPR_PROCESS_START:  7
-----------------------------------------------------------------
ORGANIZATION_TYPE:  ['Business Entity Type 3' 'School' 'Government' 'Rel
igion' 'Other' 'XNA'
 'Medicine' 'Business Entity Type 2' 'Self-employed' 'Housing'
 'Kindergarten' 'Trade: type 7' 'Industry: type 11' 'Military' 'Services
'
 'Transport: type 4' 'Industry: type 1' 'Emergency' 'Security'
 'Trade: type 2' 'University' 'Transport: type 3' 'Police' 'Construction
'
 'Business Entity Type 1' 'Postal' 'Industry: type 4' 'Agriculture'
 'Restaurant' 'Transport: type 2' 'Culture' 'Hotel' 'Industry: type 7'
 'Trade: type 3' 'Industry: type 3' 'Bank' 'Industry: type 9'
 'Trade: type 6' 'Industry: type 2' 'Transport: type 1' 'Electricity'
 'Industry: type 12' 'Insurance' 'Security Ministries' 'Mobile'
 'Trade: type 1' 'Industry: type 5' 'Industry: type 10' 'Legal Services'
 'Advertising' 'Trade: type 5' 'Cleaning' 'Industry: type 13'
 'Industry: type 8' 'Realtor' 'Telecom' 'Industry: type 6' 'Trade: type
4']
ORGANIZATION_TYPE:  58
-----------------------------------------------------------------
```

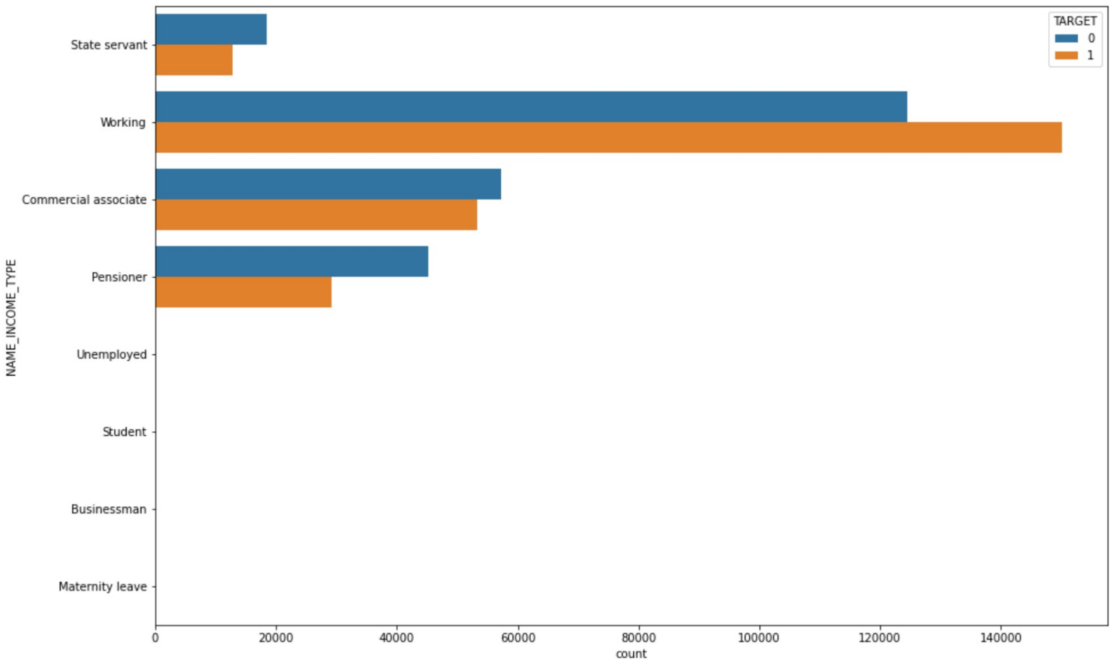
In [21]:
```
for column in df_over:
  if df_over[column].dtype == 'object':
    plt.figure(figsize=(15,10))
    plt.tight_layout
    sns.countplot(y=df_over[column],hue=df_over['TARGET'])
```

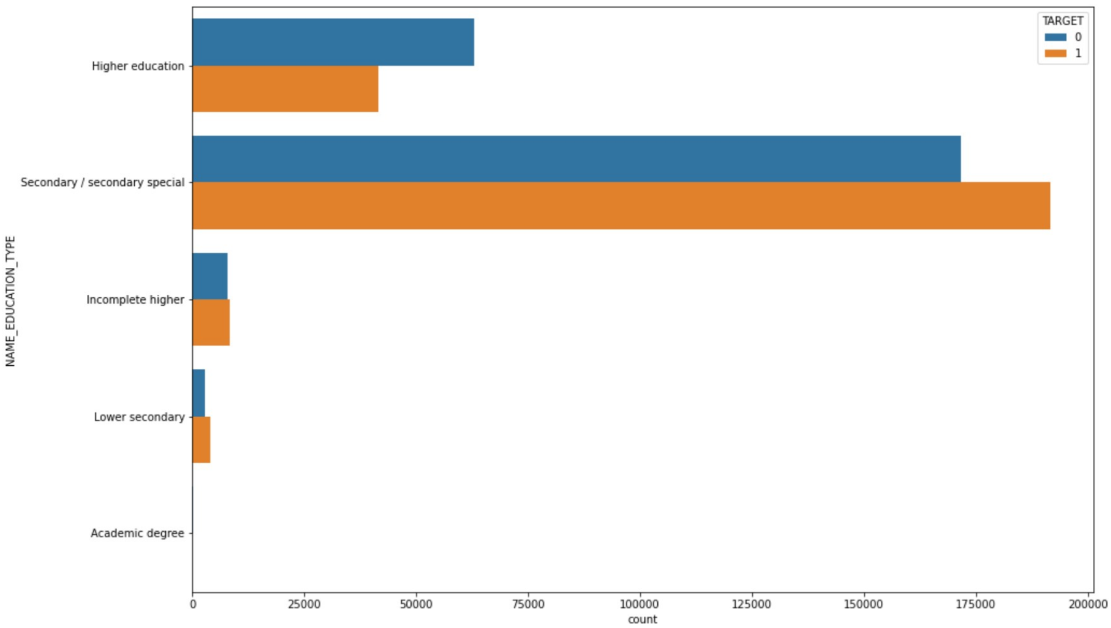

TARGET
0
1
Cash loans
Revolving loans
NAME_CONTRACT_TYPE
0
50000
100000
150000
200000
count

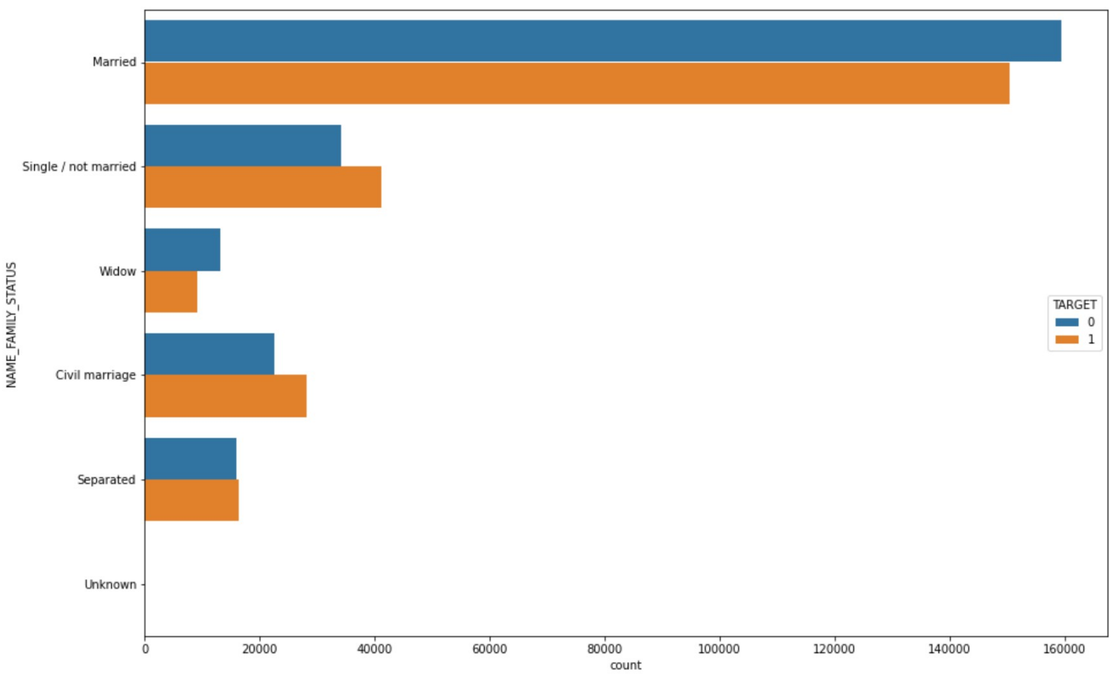

F
M
XNA
CODE_GENDER
TARGET
0
1
0
20000
40000
60000
80000
100000
120000
140000
160000
count

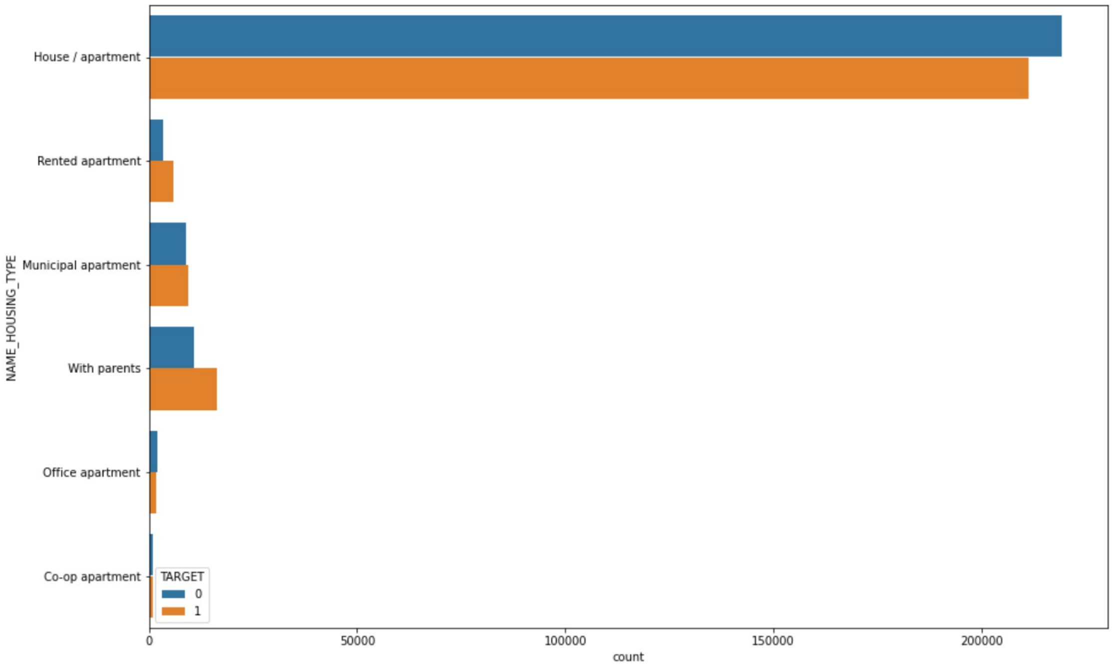

TARGET
0
1
N
Y
FLAG_OWN_CAR
0
20000
40000
60000
80000
100000
120000
140000
160000
count

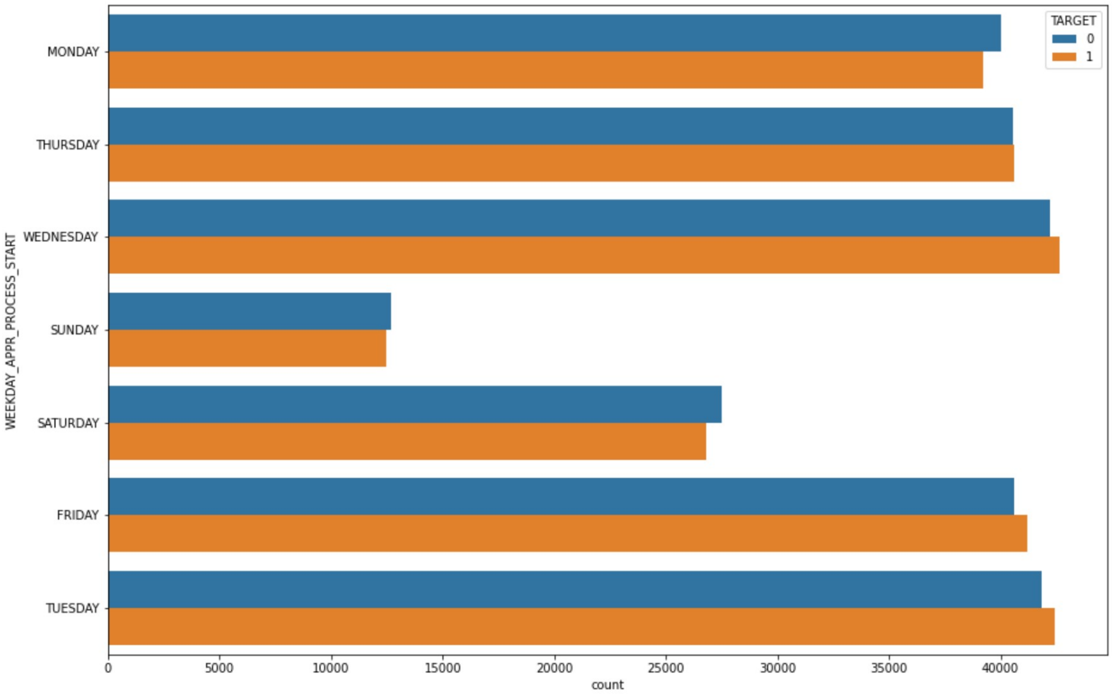

TARGET
0
1
N
Y
FLAG_OWN_REALTY
0
20000
40000
60000
80000
100000
120000
140000
160000
count

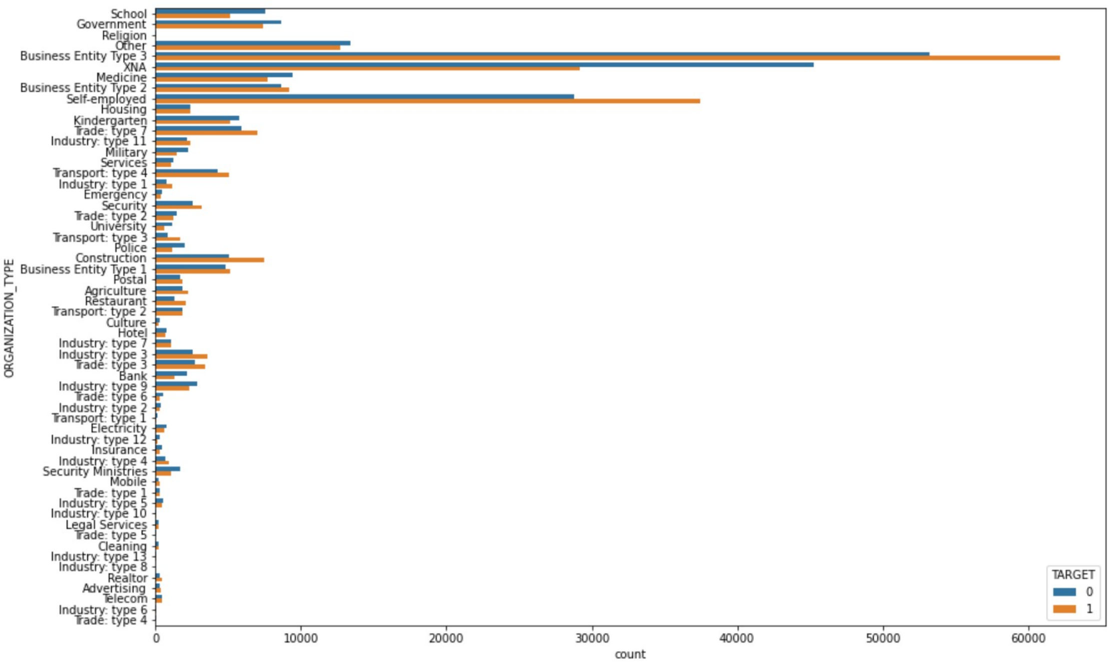

TARGET
0
1
State servant
Working
Commercial associate
Pensioner
Unemployed
Student
Businessman
Maternity leave
NAME_INCOME_TYPE
0
20000
40000
60000
80000
100000
120000
140000
count


TARGET
0
1
Higher education
Secondary / secondary special
Incomplete higher
Lower secondary
Academic degree
NAME_EDUCATION_TYPE
0
25000
50000
75000
100000
125000
150000
175000
200000
count


Married
Single / not married
Widow
Civil marriage
Separated
Unknown
NAME_FAMILY_STATUS
TARGET
0
1
0
20000
40000
60000
80000
100000
120000
140000
160000
count


House / apartment
Rented apartment
Municipal apartment
With parents
Office apartment
Co-op apartment
NAME_HOUSING_TYPE
TARGET
0
1
0
50000
100000
150000
200000
count


MONDAY
THURSDAY
WEDNESDAY
SUNDAY
SATURDAY
FRIDAY
TUESDAY
WEEKDAY_APPR_PROCESS_START
TARGET
0
1
0
5000
10000
15000
20000
25000
30000
35000
40000
count



In [22]:
```
df_over = pd.get_dummies(data=df_over,columns=['NAME_CONTRACT_TYPE'])
df_over = pd.get_dummies(data=df_over,columns=['CODE_GENDER'])
df_over = pd.get_dummies(data=df_over,columns=['NAME_INCOME_TYPE'])
df_over = pd.get_dummies(data=df_over,columns=['NAME_EDUCATION_TYPE'])
df_over = pd.get_dummies(data=df_over,columns=['NAME_FAMILY_STATUS'])
df_over = pd.get_dummies(data=df_over,columns=['NAME_HOUSING_TYPE'])
df_over = pd.get_dummies(data=df_over,columns=['WEEKDAY_APPR_PROCESS_STAR
T'],drop_first=True)
df_over = pd.get_dummies(data=df_over,columns=['ORGANIZATION_TYPE'])
```

In [23]:
```
df_over.head()
```

Out[23]:

	SK_ID_CURR	TARGET	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_I
1	100003	0	N	N	0	
2	100004	0	Y	Y	0	
4	100007	0	N	Y	0	
5	100008	0	N	Y	0	
6	100009	0	Y	Y	1	

In [24]:
```
df_over = df_over.drop(['CODE_GENDER_XNA','NAME_INCOME_TYPE_Businessman
','NAME_INCOME_TYPE_Maternity leave','NAME_INCOME_TYPE_Student','NAME_INC
OME_TYPE_Unemployed',
                    'NAME_EDUCATION_TYPE_Academic degree','NAME_FAMILY_S
TATUS_Unknown','NAME_HOUSING_TYPE_Co-op apartment','WEEKDAY_APPR_PROCESS_
START_SUNDAY',
                    'ORGANIZATION_TYPE_Religion','ORGANIZATION_TYPE_Tran
sport: type 1','ORGANIZATION_TYPE_Industry: type 10','ORGANIZATION_TYPE_T
rade: type 5',
                    'ORGANIZATION_TYPE_Industry: type 13','ORGANIZATION_
TYPE_Industry: type 8','ORGANIZATION_TYPE_Industry: type 6','ORGANIZATION
_TYPE_Trade: type 4'],axis=1)
```

In [25]:
```
df_over['FLAG_OWN_CAR'] = df_over['FLAG_OWN_CAR'].replace({'N':0,'Y':1})
df_over['FLAG_OWN_REALTY'] = df_over['FLAG_OWN_REALTY'].replace({'N':0,'Y
':1})
```

In [26]:
```
df_over.shape
```

Out[26]:
```
(490918, 125)
```

Calculate Sensitivity as a metrice

In [27]:
```
df_over.corr().transpose()
```

Out[27]:

	SK_ID_CURR	TARGET	FLAG_OWN_CA
SK_ID_CURR	1.000000	-0.006929	0.00456
TARGET	-0.006929	1.000000	-0.04005
FLAG_OWN_CAR	0.004561	-0.040055	1.00000
FLAG_OWN_REALTY	0.001679	-0.013011	0.00830
CNT_CHILDREN	-0.006622	0.033548	0.08369
AMT_INCOME_TOTAL	-0.008351	0.000074	0.02430
AMT_CREDIT	-0.002615	-0.057787	0.10524
REGION_POPULATION_RELATIVE	0.004884	-0.068839	0.04080
DAYS_BIRTH	-0.008527	0.145082	0.10770
DAYS_EMPLOYED	0.004063	-0.089290	-0.13500
DAYS_REGISTRATION	-0.002371	0.076090	0.07853
DAYS_ID_PUBLISH	-0.000449	0.092090	-0.00062
FLAG_MOBIL	NaN	NaN	Na
FLAG_EMP_PHONE	-0.003975	0.091224	0.13502
FLAG_WORK_PHONE	0.001916	0.049699	-0.00192
FLAG_CONT_MOBILE	-0.000519	0.001947	-0.00035
FLAG_PHONE	0.011656	-0.042848	-0.00492
FLAG_EMAIL	0.003278	-0.001094	0.03106
REGION_RATING_CLIENT	-0.003164	0.103494	-0.02052
REGION_RATING_CLIENT_W_CITY	-0.002342	0.106969	-0.01831
HOUR_APPR_PROCESS_START	0.003816	-0.040390	0.01409
REG_REGION_NOT_LIVE_REGION	-0.005319	0.010361	-0.00166
REG_REGION_NOT_WORK_REGION	0.000784	0.012150	0.03303
LIVE_REGION_NOT_WORK_REGION	0.003705	0.003924	0.03822
REG_CITY_NOT_LIVE_CITY	0.001785	0.073201	-0.00418
REG_CITY_NOT_WORK_CITY	-0.002714	0.086457	0.06151
LIVE_CITY_NOT_WORK_CITY	-0.003683	0.054802	0.07570
FLAG_DOCUMENT_2	NaN	NaN	Na
FLAG_DOCUMENT_3	-0.003611	0.087477	-0.07437
FLAG_DOCUMENT_4	-0.002909	-0.006992	-0.00059
FLAG_DOCUMENT_5	0.001697	0.000033	-0.01533
FLAG_DOCUMENT_6	0.002638	-0.056723	-0.09271
FLAG_DOCUMENT_7	-0.000690	-0.001745	-0.00030
FLAG_DOCUMENT_8	0.001941	-0.014772	0.21538
FLAG_DOCUMENT_9	-0.001969	-0.007030	-0.00896

	SK_ID_CURR	TARGET	FLAG_OWN_CA
FLAG_DOCUMENT_10	-0.000621	-0.003776	0.00077
FLAG_DOCUMENT_11	-0.006337	-0.010026	-0.00087
FLAG_DOCUMENT_12	-0.000815	-0.002018	0.00286
FLAG_DOCUMENT_13	0.000567	-0.025447	0.07195
FLAG_DOCUMENT_14	-0.000795	-0.017876	0.00234
FLAG_DOCUMENT_15	0.004713	-0.014881	0.03592
FLAG_DOCUMENT_16	0.000179	-0.021758	0.00332
FLAG_DOCUMENT_17	0.001274	-0.012361	-0.00136
FLAG_DOCUMENT_18	-0.003280	-0.013747	-0.00264
FLAG_DOCUMENT_19	-0.002362	-0.000989	-0.00465
FLAG_DOCUMENT_20	0.004789	0.003615	0.00307
FLAG_DOCUMENT_21	0.002574	0.008137	0.00569
AMT_REQ_CREDIT_BUREAU_YEAR	0.008857	0.035591	-0.02522
NAME_CONTRACT_TYPE_Cash loans	-0.001575	0.060715	-0.00388
NAME_CONTRACT_TYPE_Revolving loans	0.001575	-0.060715	0.00388
CODE_GENDER_F	0.003386	-0.092899	-0.33319
CODE_GENDER_M	-0.003374	0.092916	0.33319
NAME_INCOME_TYPE_Commercial associate	0.000200	-0.019764	0.05277
NAME_INCOME_TYPE_Pensioner	0.004080	-0.091340	-0.13528
NAME_INCOME_TYPE_State servant	-0.000947	-0.046005	0.00334
NAME_INCOME_TYPE_Working	-0.002650	0.105360	0.05165
NAME_EDUCATION_TYPE_Higher education	-0.002097	-0.107526	0.08603
NAME_EDUCATION_TYPE_Incomplete higher	0.006402	0.005797	0.01059
NAME_EDUCATION_TYPE_Lower secondary	-0.000989	0.019098	-0.01862
NAME_EDUCATION_TYPE_Secondary / secondary special	-0.000372	0.093414	-0.07974
NAME_FAMILY_STATUS_Civil marriage	-0.003012	0.037734	-0.02815
NAME_FAMILY_STATUS_Married	0.001367	-0.038236	0.14056
NAME_FAMILY_STATUS_Separated	0.002584	0.003160	-0.05880
NAME_FAMILY_STATUS_Single / not married	-0.002056	0.039883	-0.06258
NAME_FAMILY_STATUS_Widow	0.001721	-0.039516	-0.10608
NAME_HOUSING_TYPE_House / apartment	-0.001054	-0.049459	0.01777
NAME_HOUSING_TYPE_Municipal apartment	0.000163	0.006610	-0.03357
NAME_HOUSING_TYPE_Office apartment	-0.002380	-0.009485	0.00273
NAME_HOUSING_TYPE_Rented apartment	0.002550	0.033831	-0.01727
NAME_HOUSING_TYPE_With parents	0.000911	0.049912	0.00815
WEEKDAY_APPR_PROCESS_START_MONDAY	-0.006089	-0.004362	0.00511
WEEKDAY_APPR_PROCESS_START_SATURDAY	-0.001041	-0.004603	-0.01119

	SK_ID_CURR	TARGET	FLAG_OWN_CA
WEEKDAY_APPR_PROCESS_START_THURSDAY	-0.001259	0.000252	0.00956
WEEKDAY_APPR_PROCESS_START_TUESDAY	0.005717	0.003306	-0.00155
WEEKDAY_APPR_PROCESS_START_WEDNESDAY	0.003293	0.002306	-0.00117
ORGANIZATION_TYPE_Advertising	-0.000069	0.001237	-0.00080
ORGANIZATION_TYPE_Agriculture	-0.006980	0.010253	0.00461
ORGANIZATION_TYPE_Bank	-0.002046	-0.021090	0.00074
ORGANIZATION_TYPE_Business Entity Type 1	-0.000364	0.004729	0.00531
ORGANIZATION_TYPE_Business Entity Type 2	0.002809	0.005463	0.01972
ORGANIZATION_TYPE_Business Entity Type 3	-0.003142	0.043390	0.05889
ORGANIZATION_TYPE_Cleaning	0.001744	0.002239	0.00085
ORGANIZATION_TYPE_Construction	0.011787	0.031400	0.04082
ORGANIZATION_TYPE_Culture	-0.003312	-0.005246	-0.00115
ORGANIZATION_TYPE_Electricity	-0.000879	-0.006346	-0.00139
ORGANIZATION_TYPE_Emergency	0.000290	-0.002397	0.02099
ORGANIZATION_TYPE_Government	0.000664	-0.014473	-0.00639
ORGANIZATION_TYPE_Hotel	-0.004095	-0.003211	-0.01083
ORGANIZATION_TYPE_Housing	0.000310	0.000414	0.00630
ORGANIZATION_TYPE_Industry: type 1	-0.004812	0.010763	0.00853
ORGANIZATION_TYPE_Industry: type 11	0.002211	0.004739	-0.00033
ORGANIZATION_TYPE_Industry: type 12	0.000048	-0.011738	0.00104
ORGANIZATION_TYPE_Industry: type 2	-0.006606	-0.002544	0.00438
ORGANIZATION_TYPE_Industry: type 3	-0.000607	0.018688	-0.01207
ORGANIZATION_TYPE_Industry: type 4	-0.006330	0.007639	0.00407
ORGANIZATION_TYPE_Industry: type 5	0.003488	-0.003158	0.00492
ORGANIZATION_TYPE_Industry: type 7	0.000301	0.001654	-0.00303
ORGANIZATION_TYPE_Industry: type 9	-0.001458	-0.009858	0.02753
ORGANIZATION_TYPE_Insurance	-0.001458	-0.007959	-0.00031
ORGANIZATION_TYPE_Kindergarten	-0.001789	-0.007766	-0.04219
ORGANIZATION_TYPE_Legal Services	0.004128	-0.001628	0.00820
ORGANIZATION_TYPE_Medicine	-0.001444	-0.018810	-0.03508
ORGANIZATION_TYPE_Military	-0.001666	-0.018283	0.03109
ORGANIZATION_TYPE_Mobile	-0.000394	0.004102	0.00523
ORGANIZATION_TYPE_Other	-0.001700	-0.006399	0.00825
ORGANIZATION_TYPE_Police	0.000175	-0.021977	0.02357
ORGANIZATION_TYPE_Postal	-0.005226	0.003664	-0.02259
ORGANIZATION_TYPE_Realtor	0.006408	0.006255	0.00731
ORGANIZATION_TYPE_Restaurant	-0.003310	0.019341	-0.01887

	SK_ID_CURR	TARGET	FLAG_OWN_CAI
ORGANIZATION_TYPE_School	-0.000117	-0.030556	-0.02102(
ORGANIZATION_TYPE_Security	-0.006097	0.012961	0.01549!
ORGANIZATION_TYPE_Security Ministries	-0.008043	-0.017208	0.01711
ORGANIZATION_TYPE_Self-employed	0.001802	0.051653	0.02847!
ORGANIZATION_TYPE_Services	-0.003616	-0.006438	-0.00868
ORGANIZATION_TYPE_Telecom	-0.003299	-0.000519	0.00556
ORGANIZATION_TYPE_Trade: type 1	-0.001295	0.000719	-0.00923
ORGANIZATION_TYPE_Trade: type 2	0.005803	-0.006024	0.00485

In [28]:
```
df_over = df_over.drop(['FLAG_MOBIL','FLAG_DOCUMENT_2'],axis=1)
```

In [29]:
```
df_over.shape
```

Out[29]:
```
(490918, 123)
```

In [30]:
```
X = df_over.drop('TARGET',axis=1)
Y = df_over['TARGET'].values
```

In [31]:
```
Y
```

Out[31]:
```
array([0, 0, 0, ..., 1, 1, 1])
```

In [32]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.3,random
_state=101)
```

In [33]:
```
from sklearn.preprocessing import MinMaxScaler
Scaler = MinMaxScaler()
```

In [34]:
```
X_train = Scaler.fit_transform(X_train)
X_test = Scaler.transform(X_test)
```

In [35]:
```
X_train
```

Out[35]:
```
array([[0.37149722, 0.        , 1.        , ..., 0.        , 0.        ,
        0.        ],
       [0.93866718, 0.        , 1.        , ..., 0.        , 0.        ,
        0.        ],
       [0.29745714, 0.        , 1.        , ..., 0.        , 0.        ,
        0.        ],
       ...,
       [0.42947287, 0.        , 1.        , ..., 0.        , 0.        ,
        0.        ],
       [0.87450492, 0.        , 1.        , ..., 0.        , 0.        ,
        0.        ],
       [0.72147042, 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ]])
```

In [36]:
```
print(X_train.shape,X_test.shape,Y_train.shape,Y_test.shape)
```

```
(343642, 122) (147276, 122) (343642,) (147276,)
```

Modeling

```
After applying EDA and feature engineering, you are now ready to build the pre
dictive models

In this part, you will create a deep learning model using Keras with Tensorflo
w backend
```

In [37]:
```
import tensorflow as tf
from tensorflow import keras
```

In [38]:
```
model = keras.Sequential([
                          keras.layers.Dense(100,input_shape=(122,),activ
ation='relu'),
                          keras.layers.Dense(50,activation='relu'),
                          keras.layers.Dense(10,activation='relu'),
                          keras.layers.Dense(1,activation='sigmoid')
])

model.compile(
    optimizer='adam',
    loss='binary_crossentropy',
    metrics=['accuracy']
)

early_stop = keras.callbacks.EarlyStopping(
        monitor='val_loss',
        mode='min',
        verbose=1,
        patience=25
)
```

In [39]:
```
model.fit(X_train,Y_train,epochs=200,batch_size=256,validation_data=(X_te
st, Y_test),callbacks=[early_stop])
```

```
Epoch 1/200
1343/1343 [==============================] - 8s 5ms/step - loss: 0.6516
- accuracy: 0.6182 - val_loss: 0.6467 - val_accuracy: 0.6252
Epoch 2/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.6382
- accuracy: 0.6340 - val_loss: 0.6366 - val_accuracy: 0.6350
Epoch 3/200
1343/1343 [==============================] - 6s 4ms/step - loss: 0.6234
- accuracy: 0.6494 - val_loss: 0.6219 - val_accuracy: 0.6501
Epoch 4/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.6070
- accuracy: 0.6641 - val_loss: 0.6101 - val_accuracy: 0.6617
Epoch 5/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.5907
- accuracy: 0.6777 - val_loss: 0.5974 - val_accuracy: 0.6723
Epoch 6/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.5765
- accuracy: 0.6884 - val_loss: 0.5907 - val_accuracy: 0.6793
Epoch 7/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.5634
- accuracy: 0.6986 - val_loss: 0.5822 - val_accuracy: 0.6828
Epoch 8/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.5521
- accuracy: 0.7076 - val_loss: 0.5714 - val_accuracy: 0.6917
Epoch 9/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.5424
- accuracy: 0.7139 - val_loss: 0.5659 - val_accuracy: 0.6979
Epoch 10/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.5344
- accuracy: 0.7195 - val_loss: 0.5611 - val_accuracy: 0.6998
Epoch 11/200
1343/1343 [==============================] - 8s 6ms/step - loss: 0.5263
- accuracy: 0.7252 - val_loss: 0.5529 - val_accuracy: 0.7082
Epoch 12/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.5196
- accuracy: 0.7306 - val_loss: 0.5448 - val_accuracy: 0.7111
Epoch 13/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.5132
- accuracy: 0.7345 - val_loss: 0.5433 - val_accuracy: 0.7143
Epoch 14/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.5076
- accuracy: 0.7381 - val_loss: 0.5360 - val_accuracy: 0.7187
Epoch 15/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.5028
- accuracy: 0.7410 - val_loss: 0.5396 - val_accuracy: 0.7185
Epoch 16/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4985
- accuracy: 0.7441 - val_loss: 0.5371 - val_accuracy: 0.7170
Epoch 17/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.4936
- accuracy: 0.7473 - val_loss: 0.5303 - val_accuracy: 0.7250
Epoch 18/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4902
- accuracy: 0.7495 - val_loss: 0.5266 - val_accuracy: 0.7275
Epoch 19/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4861
- accuracy: 0.7525 - val_loss: 0.5236 - val_accuracy: 0.7312
Epoch 20/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4829
- accuracy: 0.7545 - val_loss: 0.5206 - val_accuracy: 0.7324
```

```
Epoch 21/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4796
- accuracy: 0.7562 - val_loss: 0.5193 - val_accuracy: 0.7326
Epoch 22/200
1343/1343 [==============================] - 8s 6ms/step - loss: 0.4763
- accuracy: 0.7588 - val_loss: 0.5189 - val_accuracy: 0.7314
Epoch 23/200
1343/1343 [==============================] - 8s 6ms/step - loss: 0.4730
- accuracy: 0.7607 - val_loss: 0.5137 - val_accuracy: 0.7379
Epoch 24/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4708
- accuracy: 0.7618 - val_loss: 0.5137 - val_accuracy: 0.7345
Epoch 25/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4680
- accuracy: 0.7638 - val_loss: 0.5122 - val_accuracy: 0.7411
Epoch 26/200
1343/1343 [==============================] - 6s 4ms/step - loss: 0.4656
- accuracy: 0.7654 - val_loss: 0.5050 - val_accuracy: 0.7416
Epoch 27/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4631
- accuracy: 0.7671 - val_loss: 0.5130 - val_accuracy: 0.7371
Epoch 28/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4613
- accuracy: 0.7682 - val_loss: 0.5032 - val_accuracy: 0.7453
Epoch 29/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4599
- accuracy: 0.7685 - val_loss: 0.5013 - val_accuracy: 0.7458
Epoch 30/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4567
- accuracy: 0.7711 - val_loss: 0.5059 - val_accuracy: 0.7446
Epoch 31/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4540
- accuracy: 0.7730 - val_loss: 0.4995 - val_accuracy: 0.7499
Epoch 32/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4529
- accuracy: 0.7735 - val_loss: 0.4968 - val_accuracy: 0.7509
Epoch 33/200
1343/1343 [==============================] - 8s 6ms/step - loss: 0.4511
- accuracy: 0.7740 - val_loss: 0.4979 - val_accuracy: 0.7447
Epoch 34/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4498
- accuracy: 0.7749 - val_loss: 0.4943 - val_accuracy: 0.7538
Epoch 35/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4476
- accuracy: 0.7769 - val_loss: 0.4931 - val_accuracy: 0.7548
Epoch 36/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4465
- accuracy: 0.7773 - val_loss: 0.5017 - val_accuracy: 0.7470
Epoch 37/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4446
- accuracy: 0.7787 - val_loss: 0.4887 - val_accuracy: 0.7566
Epoch 38/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4432
- accuracy: 0.7791 - val_loss: 0.4907 - val_accuracy: 0.7519
Epoch 39/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4425
- accuracy: 0.7802 - val_loss: 0.4863 - val_accuracy: 0.7579
Epoch 40/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4407
- accuracy: 0.7814 - val_loss: 0.4860 - val_accuracy: 0.7574
```

```
Epoch 41/200
1343/1343 [==============================] - 8s 6ms/step - loss: 0.4392
- accuracy: 0.7821 - val_loss: 0.4943 - val_accuracy: 0.7538
Epoch 42/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.4377
- accuracy: 0.7831 - val_loss: 0.4904 - val_accuracy: 0.7564
Epoch 43/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4367
- accuracy: 0.7835 - val_loss: 0.4901 - val_accuracy: 0.7565
Epoch 44/200
1343/1343 [==============================] - 9s 7ms/step - loss: 0.4354
- accuracy: 0.7842 - val_loss: 0.4892 - val_accuracy: 0.7534
Epoch 45/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4346
- accuracy: 0.7846 - val_loss: 0.4850 - val_accuracy: 0.7572
Epoch 46/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4340
- accuracy: 0.7848 - val_loss: 0.4833 - val_accuracy: 0.7586
Epoch 47/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4322
- accuracy: 0.7865 - val_loss: 0.4897 - val_accuracy: 0.7563
Epoch 48/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4325
- accuracy: 0.7864 - val_loss: 0.4867 - val_accuracy: 0.7570
Epoch 49/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4304
- accuracy: 0.7868 - val_loss: 0.4881 - val_accuracy: 0.7584
Epoch 50/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4307
- accuracy: 0.7866 - val_loss: 0.4815 - val_accuracy: 0.7621
Epoch 51/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4281
- accuracy: 0.7885 - val_loss: 0.4832 - val_accuracy: 0.7591
Epoch 52/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4276
- accuracy: 0.7887 - val_loss: 0.4789 - val_accuracy: 0.7617
Epoch 53/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4273
- accuracy: 0.7891 - val_loss: 0.4848 - val_accuracy: 0.7577
Epoch 54/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4257
- accuracy: 0.7900 - val_loss: 0.4800 - val_accuracy: 0.7630
Epoch 55/200
1343/1343 [==============================] - 8s 6ms/step - loss: 0.4253
- accuracy: 0.7905 - val_loss: 0.4847 - val_accuracy: 0.7616
Epoch 56/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4244
- accuracy: 0.7904 - val_loss: 0.4844 - val_accuracy: 0.7605
Epoch 57/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.4236
- accuracy: 0.7916 - val_loss: 0.4807 - val_accuracy: 0.7641
Epoch 58/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4234
- accuracy: 0.7914 - val_loss: 0.4792 - val_accuracy: 0.7653
Epoch 59/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4218
- accuracy: 0.7927 - val_loss: 0.4787 - val_accuracy: 0.7641
Epoch 60/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4205
- accuracy: 0.7931 - val_loss: 0.4742 - val_accuracy: 0.7656
```

```
Epoch 61/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4195
- accuracy: 0.7933 - val_loss: 0.4784 - val_accuracy: 0.7659
Epoch 62/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4187
- accuracy: 0.7944 - val_loss: 0.4767 - val_accuracy: 0.7657
Epoch 63/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4191
- accuracy: 0.7940 - val_loss: 0.4793 - val_accuracy: 0.7670
Epoch 64/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4179
- accuracy: 0.7946 - val_loss: 0.4794 - val_accuracy: 0.7666
Epoch 65/200
1343/1343 [==============================] - 8s 6ms/step - loss: 0.4165
- accuracy: 0.7960 - val_loss: 0.4751 - val_accuracy: 0.7659
Epoch 66/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4162
- accuracy: 0.7963 - val_loss: 0.4761 - val_accuracy: 0.7659
Epoch 67/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4163
- accuracy: 0.7958 - val_loss: 0.4768 - val_accuracy: 0.7684
Epoch 68/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4143
- accuracy: 0.7967 - val_loss: 0.4742 - val_accuracy: 0.7697
Epoch 69/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4138
- accuracy: 0.7972 - val_loss: 0.4769 - val_accuracy: 0.7689
Epoch 70/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4128
- accuracy: 0.7972 - val_loss: 0.4725 - val_accuracy: 0.7679
Epoch 71/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4124
- accuracy: 0.7973 - val_loss: 0.4778 - val_accuracy: 0.7689
Epoch 72/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4119
- accuracy: 0.7976 - val_loss: 0.4804 - val_accuracy: 0.7639
Epoch 73/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4118
- accuracy: 0.7979 - val_loss: 0.4755 - val_accuracy: 0.7704
Epoch 74/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4116
- accuracy: 0.7983 - val_loss: 0.4798 - val_accuracy: 0.7667
Epoch 75/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4100
- accuracy: 0.7990 - val_loss: 0.4728 - val_accuracy: 0.7702
Epoch 76/200
1343/1343 [==============================] - 8s 6ms/step - loss: 0.4088
- accuracy: 0.7996 - val_loss: 0.4723 - val_accuracy: 0.7702
Epoch 77/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4092
- accuracy: 0.7997 - val_loss: 0.4793 - val_accuracy: 0.7652
Epoch 78/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4093
- accuracy: 0.7997 - val_loss: 0.4772 - val_accuracy: 0.7681
Epoch 79/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.4084
- accuracy: 0.7998 - val_loss: 0.4775 - val_accuracy: 0.7685
Epoch 80/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4073
- accuracy: 0.8002 - val_loss: 0.4794 - val_accuracy: 0.7696
```

```
Epoch 81/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4062
- accuracy: 0.8009 - val_loss: 0.4737 - val_accuracy: 0.7722
Epoch 82/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.4055
- accuracy: 0.8019 - val_loss: 0.4755 - val_accuracy: 0.7696
Epoch 83/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4062
- accuracy: 0.8018 - val_loss: 0.4698 - val_accuracy: 0.7747
Epoch 84/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4054
- accuracy: 0.8019 - val_loss: 0.4748 - val_accuracy: 0.7672
Epoch 85/200
1343/1343 [==============================] - 8s 6ms/step - loss: 0.4045
- accuracy: 0.8023 - val_loss: 0.4714 - val_accuracy: 0.7732
Epoch 86/200
1343/1343 [==============================] - 9s 7ms/step - loss: 0.4036
- accuracy: 0.8029 - val_loss: 0.4801 - val_accuracy: 0.7693
Epoch 87/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4033
- accuracy: 0.8024 - val_loss: 0.4709 - val_accuracy: 0.7714
Epoch 88/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.4029
- accuracy: 0.8034 - val_loss: 0.4733 - val_accuracy: 0.7716
Epoch 89/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4029
- accuracy: 0.8029 - val_loss: 0.4746 - val_accuracy: 0.7702
Epoch 90/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.4016
- accuracy: 0.8039 - val_loss: 0.4711 - val_accuracy: 0.7709
Epoch 91/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4014
- accuracy: 0.8041 - val_loss: 0.4802 - val_accuracy: 0.7655
Epoch 92/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.4010
- accuracy: 0.8049 - val_loss: 0.4745 - val_accuracy: 0.7724
Epoch 93/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4002
- accuracy: 0.8044 - val_loss: 0.4776 - val_accuracy: 0.7680
Epoch 94/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4000
- accuracy: 0.8048 - val_loss: 0.4708 - val_accuracy: 0.7722
Epoch 95/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3998
- accuracy: 0.8052 - val_loss: 0.4753 - val_accuracy: 0.7695
Epoch 96/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.4005
- accuracy: 0.8047 - val_loss: 0.4726 - val_accuracy: 0.7745
Epoch 97/200
1343/1343 [==============================] - 9s 6ms/step - loss: 0.3982
- accuracy: 0.8066 - val_loss: 0.4715 - val_accuracy: 0.7736
Epoch 98/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.3979
- accuracy: 0.8062 - val_loss: 0.4684 - val_accuracy: 0.7769
Epoch 99/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3972
- accuracy: 0.8069 - val_loss: 0.4678 - val_accuracy: 0.7762
Epoch 100/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3978
- accuracy: 0.8063 - val_loss: 0.4737 - val_accuracy: 0.7751
```

```
Epoch 101/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.3977
- accuracy: 0.8062 - val_loss: 0.4718 - val_accuracy: 0.7762
Epoch 102/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3967
- accuracy: 0.8075 - val_loss: 0.4735 - val_accuracy: 0.7723
Epoch 103/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3960
- accuracy: 0.8078 - val_loss: 0.4764 - val_accuracy: 0.7763
Epoch 104/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.3966
- accuracy: 0.8074 - val_loss: 0.4687 - val_accuracy: 0.7776
Epoch 105/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3946
- accuracy: 0.8088 - val_loss: 0.4681 - val_accuracy: 0.7765
Epoch 106/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3945
- accuracy: 0.8078 - val_loss: 0.4678 - val_accuracy: 0.7785
Epoch 107/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3943
- accuracy: 0.8086 - val_loss: 0.4734 - val_accuracy: 0.7714
Epoch 108/200
1343/1343 [==============================] - 8s 6ms/step - loss: 0.3939
- accuracy: 0.8087 - val_loss: 0.4706 - val_accuracy: 0.7755
Epoch 109/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3937
- accuracy: 0.8086 - val_loss: 0.4748 - val_accuracy: 0.7755
Epoch 110/200
1343/1343 [==============================] - 9s 6ms/step - loss: 0.3932
- accuracy: 0.8090 - val_loss: 0.4698 - val_accuracy: 0.7750
Epoch 111/200
1343/1343 [==============================] - 9s 6ms/step - loss: 0.3926
- accuracy: 0.8100 - val_loss: 0.4737 - val_accuracy: 0.7750
Epoch 112/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3925
- accuracy: 0.8098 - val_loss: 0.4706 - val_accuracy: 0.7756
Epoch 113/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3915
- accuracy: 0.8099 - val_loss: 0.4680 - val_accuracy: 0.7793
Epoch 114/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3915
- accuracy: 0.8105 - val_loss: 0.4698 - val_accuracy: 0.7785
Epoch 115/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.3925
- accuracy: 0.8097 - val_loss: 0.4684 - val_accuracy: 0.7753
Epoch 116/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3911
- accuracy: 0.8103 - val_loss: 0.4660 - val_accuracy: 0.7777
Epoch 117/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3913
- accuracy: 0.8106 - val_loss: 0.4694 - val_accuracy: 0.7782
Epoch 118/200
1343/1343 [==============================] - 8s 6ms/step - loss: 0.3910
- accuracy: 0.8105 - val_loss: 0.4710 - val_accuracy: 0.7774
Epoch 119/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.3901
- accuracy: 0.8106 - val_loss: 0.4651 - val_accuracy: 0.7760
Epoch 120/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3909
- accuracy: 0.8107 - val_loss: 0.4623 - val_accuracy: 0.7791
```

```
Epoch 121/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3896
- accuracy: 0.8116 - val_loss: 0.4682 - val_accuracy: 0.7775
Epoch 122/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3896
- accuracy: 0.8108 - val_loss: 0.4699 - val_accuracy: 0.7789
Epoch 123/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3888
- accuracy: 0.8112 - val_loss: 0.4682 - val_accuracy: 0.7791
Epoch 124/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3893
- accuracy: 0.8110 - val_loss: 0.4662 - val_accuracy: 0.7792
Epoch 125/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3883
- accuracy: 0.8123 - val_loss: 0.4658 - val_accuracy: 0.7765
Epoch 126/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3884
- accuracy: 0.8119 - val_loss: 0.4726 - val_accuracy: 0.7760
Epoch 127/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.3877
- accuracy: 0.8119 - val_loss: 0.4662 - val_accuracy: 0.7784
Epoch 128/200
1343/1343 [==============================] - 8s 6ms/step - loss: 0.3867
- accuracy: 0.8127 - val_loss: 0.4679 - val_accuracy: 0.7803
Epoch 129/200
1343/1343 [==============================] - 9s 6ms/step - loss: 0.3867
- accuracy: 0.8124 - val_loss: 0.4629 - val_accuracy: 0.7809
Epoch 130/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3867
- accuracy: 0.8132 - val_loss: 0.4666 - val_accuracy: 0.7777
Epoch 131/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3857
- accuracy: 0.8134 - val_loss: 0.4627 - val_accuracy: 0.7837
Epoch 132/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.3862
- accuracy: 0.8134 - val_loss: 0.4704 - val_accuracy: 0.7771
Epoch 133/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3864
- accuracy: 0.8129 - val_loss: 0.4648 - val_accuracy: 0.7768
Epoch 134/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3856
- accuracy: 0.8132 - val_loss: 0.4615 - val_accuracy: 0.7830
Epoch 135/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3854
- accuracy: 0.8141 - val_loss: 0.4580 - val_accuracy: 0.7828
Epoch 136/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3850
- accuracy: 0.8140 - val_loss: 0.4625 - val_accuracy: 0.7801
Epoch 137/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3838
- accuracy: 0.8150 - val_loss: 0.4617 - val_accuracy: 0.7805
Epoch 138/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3846
- accuracy: 0.8143 - val_loss: 0.4610 - val_accuracy: 0.7825
Epoch 139/200
1343/1343 [==============================] - 8s 6ms/step - loss: 0.3839
- accuracy: 0.8145 - val_loss: 0.4640 - val_accuracy: 0.7805
Epoch 140/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3837
- accuracy: 0.8148 - val_loss: 0.4634 - val_accuracy: 0.7825
```

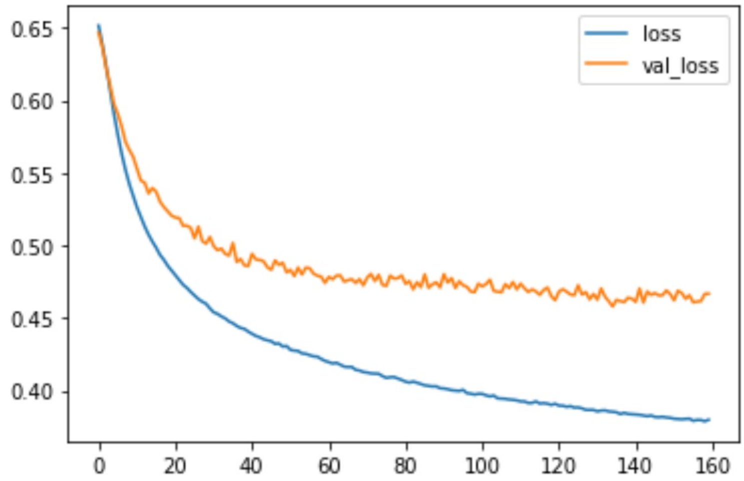
```
Epoch 141/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3834
- accuracy: 0.8149 - val_loss: 0.4612 - val_accuracy: 0.7796
Epoch 142/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.3833
- accuracy: 0.8152 - val_loss: 0.4703 - val_accuracy: 0.7792
Epoch 143/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.3828
- accuracy: 0.8155 - val_loss: 0.4607 - val_accuracy: 0.7824
Epoch 144/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.3824
- accuracy: 0.8151 - val_loss: 0.4683 - val_accuracy: 0.7801
Epoch 145/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3828
- accuracy: 0.8155 - val_loss: 0.4660 - val_accuracy: 0.7807
Epoch 146/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.3818
- accuracy: 0.8156 - val_loss: 0.4672 - val_accuracy: 0.7787
Epoch 147/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.3817
- accuracy: 0.8160 - val_loss: 0.4656 - val_accuracy: 0.7803
Epoch 148/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.3819
- accuracy: 0.8161 - val_loss: 0.4658 - val_accuracy: 0.7794
Epoch 149/200
1343/1343 [==============================] - 8s 6ms/step - loss: 0.3815
- accuracy: 0.8157 - val_loss: 0.4690 - val_accuracy: 0.7785
Epoch 150/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.3812
- accuracy: 0.8162 - val_loss: 0.4665 - val_accuracy: 0.7818
Epoch 151/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3806
- accuracy: 0.8162 - val_loss: 0.4622 - val_accuracy: 0.7791
Epoch 152/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.3805
- accuracy: 0.8169 - val_loss: 0.4688 - val_accuracy: 0.7825
Epoch 153/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.3802
- accuracy: 0.8167 - val_loss: 0.4672 - val_accuracy: 0.7847
Epoch 154/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.3804
- accuracy: 0.8166 - val_loss: 0.4633 - val_accuracy: 0.7779
Epoch 155/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.3807
- accuracy: 0.8163 - val_loss: 0.4657 - val_accuracy: 0.7846
Epoch 156/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.3792
- accuracy: 0.8173 - val_loss: 0.4608 - val_accuracy: 0.7832
Epoch 157/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.3798
- accuracy: 0.8168 - val_loss: 0.4614 - val_accuracy: 0.7816
Epoch 158/200
1343/1343 [==============================] - 6s 5ms/step - loss: 0.3797
- accuracy: 0.8169 - val_loss: 0.4617 - val_accuracy: 0.7848
Epoch 159/200
1343/1343 [==============================] - 7s 5ms/step - loss: 0.3788
- accuracy: 0.8173 - val_loss: 0.4666 - val_accuracy: 0.7816
Epoch 160/200
1343/1343 [==============================] - 8s 6ms/step - loss: 0.3799
```

Out[39]: <keras.callbacks.History at 0x7f0a25f26910>

Calculate area under receiver operating characteristics curve

In [40]:
```
pd.DataFrame(model.history.history)[['loss','val_loss']].plot()
```

Out[40]: <matplotlib.axes._subplots.AxesSubplot at 0x7f0a1d3896d0>



In [41]:
```
predictions =  np.round(model.predict(X_test).reshape(-1))
```

```
4603/4603 [==============================] - 8s 2ms/step
```

In [42]:
```
model.evaluate(X_test,Y_test)
```

```
4603/4603 [==============================] - 9s 2ms/step - loss: 0.4668
- accuracy: 0.7805
```

Out[42]: [0.46679458022117615, 0.7804734110832214]

In [43]:
```
from sklearn.metrics import confusion_matrix,classification_report
print(classification_report(Y_test,predictions))
print(confusion_matrix(Y_test,predictions))
```

```
              precision    recall  f1-score   support

           0       0.82      0.72      0.77     73829
           1       0.75      0.84      0.79     73447

    accuracy                           0.78    147276
   macro avg       0.78      0.78      0.78    147276
weighted avg       0.78      0.78      0.78    147276

[[53339 20490]
 [11841 61606]]
```

In [44]:
```
cm = tf.math.confusion_matrix(labels=Y_test,predictions=predictions)
sns.heatmap(cm, annot=True,fmt='d')
```

Out[44]: <matplotlib.axes._subplots.AxesSubplot at 0x7f0a21ada990>