*# Import the dataset*  
**import** pandas **as** pd  
**import** numpy **as** np  
**import** matplotlib.pyplot **as** plt  
**import** seaborn **as** sns; sns.set(color\_codes=True)

[2]

application\_train = pd.read\_csv('application\_data.csv')  
application\_train.head()

[3]

application\_train.shape

(307511, 122)

[4]

application\_train.columns

Index(['SK\_ID\_CURR', 'TARGET', 'NAME\_CONTRACT\_TYPE', 'CODE\_GENDER',  
 'FLAG\_OWN\_CAR', 'FLAG\_OWN\_REALTY', 'CNT\_CHILDREN', 'AMT\_INCOME\_TOTAL',  
 'AMT\_CREDIT', 'AMT\_ANNUITY',  
 ...  
 '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)

[5]

application\_train.describe()

[6]

application\_train.columns.values

array(['SK\_ID\_CURR', 'TARGET', 'NAME\_CONTRACT\_TYPE', 'CODE\_GENDER',  
 'FLAG\_OWN\_CAR', 'FLAG\_OWN\_REALTY', 'CNT\_CHILDREN',  
 'AMT\_INCOME\_TOTAL', 'AMT\_CREDIT', 'AMT\_ANNUITY', 'AMT\_GOODS\_PRICE',  
 'NAME\_TYPE\_SUITE', 'NAME\_INCOME\_TYPE', 'NAME\_EDUCATION\_TYPE',  
 'NAME\_FAMILY\_STATUS', 'NAME\_HOUSING\_TYPE',  
 'REGION\_POPULATION\_RELATIVE', 'DAYS\_BIRTH', 'DAYS\_EMPLOYED',  
 'DAYS\_REGISTRATION', 'DAYS\_ID\_PUBLISH', 'OWN\_CAR\_AGE',  
 'FLAG\_MOBIL', 'FLAG\_EMP\_PHONE', 'FLAG\_WORK\_PHONE',  
 'FLAG\_CONT\_MOBILE', 'FLAG\_PHONE', 'FLAG\_EMAIL', 'OCCUPATION\_TYPE',  
 'CNT\_FAM\_MEMBERS', 'REGION\_RATING\_CLIENT',  
 'REGION\_RATING\_CLIENT\_W\_CITY', 'WEEKDAY\_APPR\_PROCESS\_START',  
 'HOUR\_APPR\_PROCESS\_START', 'REG\_REGION\_NOT\_LIVE\_REGION',  
 'REG\_REGION\_NOT\_WORK\_REGION', 'LIVE\_REGION\_NOT\_WORK\_REGION',  
 'REG\_CITY\_NOT\_LIVE\_CITY', 'REG\_CITY\_NOT\_WORK\_CITY',  
 'LIVE\_CITY\_NOT\_WORK\_CITY', 'ORGANIZATION\_TYPE', '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', 'FLAG\_DOCUMENT\_2', 'FLAG\_DOCUMENT\_3',  
 'FLAG\_DOCUMENT\_4', 'FLAG\_DOCUMENT\_5', 'FLAG\_DOCUMENT\_6',  
 'FLAG\_DOCUMENT\_7', 'FLAG\_DOCUMENT\_8', 'FLAG\_DOCUMENT\_9',  
 'FLAG\_DOCUMENT\_10', 'FLAG\_DOCUMENT\_11', 'FLAG\_DOCUMENT\_12',  
 'FLAG\_DOCUMENT\_13', 'FLAG\_DOCUMENT\_14', 'FLAG\_DOCUMENT\_15',  
 'FLAG\_DOCUMENT\_16', 'FLAG\_DOCUMENT\_17', 'FLAG\_DOCUMENT\_18',  
 'FLAG\_DOCUMENT\_19', 'FLAG\_DOCUMENT\_20', 'FLAG\_DOCUMENT\_21',  
 '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)

[7]

application\_train.count()

SK\_ID\_CURR 307511  
TARGET 307511  
NAME\_CONTRACT\_TYPE 307511  
CODE\_GENDER 307511  
FLAG\_OWN\_CAR 307511  
 ...   
AMT\_REQ\_CREDIT\_BUREAU\_DAY 265992  
AMT\_REQ\_CREDIT\_BUREAU\_WEEK 265992  
AMT\_REQ\_CREDIT\_BUREAU\_MON 265992  
AMT\_REQ\_CREDIT\_BUREAU\_QRT 265992  
AMT\_REQ\_CREDIT\_BUREAU\_YEAR 265992  
Length: 122, dtype: int64

[8]

len(application\_train)

307511

[9]

application\_train.isna().sum()

SK\_ID\_CURR 0  
TARGET 0  
NAME\_CONTRACT\_TYPE 0  
CODE\_GENDER 0  
FLAG\_OWN\_CAR 0  
 ...   
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  
Length: 122, dtype: int64

[10]

application\_train.isna().sum().sort\_values(ascending=False).head(60)

COMMONAREA\_MEDI 214865  
COMMONAREA\_AVG 214865  
COMMONAREA\_MODE 214865  
NONLIVINGAPARTMENTS\_MODE 213514  
NONLIVINGAPARTMENTS\_MEDI 213514  
NONLIVINGAPARTMENTS\_AVG 213514  
FONDKAPREMONT\_MODE 210295  
LIVINGAPARTMENTS\_MEDI 210199  
LIVINGAPARTMENTS\_MODE 210199  
LIVINGAPARTMENTS\_AVG 210199  
FLOORSMIN\_MEDI 208642  
FLOORSMIN\_MODE 208642  
FLOORSMIN\_AVG 208642  
YEARS\_BUILD\_MEDI 204488  
YEARS\_BUILD\_AVG 204488  
YEARS\_BUILD\_MODE 204488  
OWN\_CAR\_AGE 202929  
LANDAREA\_MODE 182590  
LANDAREA\_AVG 182590  
LANDAREA\_MEDI 182590  
BASEMENTAREA\_MEDI 179943  
BASEMENTAREA\_AVG 179943  
BASEMENTAREA\_MODE 179943  
EXT\_SOURCE\_1 173378  
NONLIVINGAREA\_MEDI 169682  
NONLIVINGAREA\_AVG 169682  
NONLIVINGAREA\_MODE 169682  
ELEVATORS\_MODE 163891  
ELEVATORS\_AVG 163891  
ELEVATORS\_MEDI 163891  
WALLSMATERIAL\_MODE 156341  
APARTMENTS\_MODE 156061  
APARTMENTS\_AVG 156061  
APARTMENTS\_MEDI 156061  
ENTRANCES\_MEDI 154828  
ENTRANCES\_MODE 154828  
ENTRANCES\_AVG 154828  
LIVINGAREA\_MEDI 154350  
LIVINGAREA\_MODE 154350  
LIVINGAREA\_AVG 154350  
HOUSETYPE\_MODE 154297  
FLOORSMAX\_MODE 153020  
FLOORSMAX\_MEDI 153020  
FLOORSMAX\_AVG 153020  
YEARS\_BEGINEXPLUATATION\_MEDI 150007  
YEARS\_BEGINEXPLUATATION\_AVG 150007  
YEARS\_BEGINEXPLUATATION\_MODE 150007  
TOTALAREA\_MODE 148431  
EMERGENCYSTATE\_MODE 145755  
OCCUPATION\_TYPE 96391  
EXT\_SOURCE\_3 60965  
AMT\_REQ\_CREDIT\_BUREAU\_QRT 41519  
AMT\_REQ\_CREDIT\_BUREAU\_YEAR 41519  
AMT\_REQ\_CREDIT\_BUREAU\_WEEK 41519  
AMT\_REQ\_CREDIT\_BUREAU\_MON 41519  
AMT\_REQ\_CREDIT\_BUREAU\_DAY 41519  
AMT\_REQ\_CREDIT\_BUREAU\_HOUR 41519  
NAME\_TYPE\_SUITE 1292  
OBS\_30\_CNT\_SOCIAL\_CIRCLE 1021  
OBS\_60\_CNT\_SOCIAL\_CIRCLE 1021  
dtype: int64

[11]

application\_train.shape

(307511, 122)

[12]

x = len(application\_train)/2  
x

153755.5

[13]

type(application\_train)

pandas.core.frame.DataFrame

[14]

application\_train.columns[application\_train.isnull().sum() < 153755]

Index(['SK\_ID\_CURR', 'TARGET', 'NAME\_CONTRACT\_TYPE', 'CODE\_GENDER',  
 'FLAG\_OWN\_CAR', 'FLAG\_OWN\_REALTY', 'CNT\_CHILDREN', 'AMT\_INCOME\_TOTAL',  
 'AMT\_CREDIT', 'AMT\_ANNUITY', 'AMT\_GOODS\_PRICE', 'NAME\_TYPE\_SUITE',  
 'NAME\_INCOME\_TYPE', 'NAME\_EDUCATION\_TYPE', 'NAME\_FAMILY\_STATUS',  
 'NAME\_HOUSING\_TYPE', 'REGION\_POPULATION\_RELATIVE', 'DAYS\_BIRTH',  
 'DAYS\_EMPLOYED', 'DAYS\_REGISTRATION', 'DAYS\_ID\_PUBLISH', 'FLAG\_MOBIL',  
 'FLAG\_EMP\_PHONE', 'FLAG\_WORK\_PHONE', 'FLAG\_CONT\_MOBILE', 'FLAG\_PHONE',  
 'FLAG\_EMAIL', 'OCCUPATION\_TYPE', 'CNT\_FAM\_MEMBERS',  
 'REGION\_RATING\_CLIENT', 'REGION\_RATING\_CLIENT\_W\_CITY',  
 'WEEKDAY\_APPR\_PROCESS\_START', 'HOUR\_APPR\_PROCESS\_START',  
 'REG\_REGION\_NOT\_LIVE\_REGION', 'REG\_REGION\_NOT\_WORK\_REGION',  
 'LIVE\_REGION\_NOT\_WORK\_REGION', 'REG\_CITY\_NOT\_LIVE\_CITY',  
 'REG\_CITY\_NOT\_WORK\_CITY', 'LIVE\_CITY\_NOT\_WORK\_CITY',  
 'ORGANIZATION\_TYPE', 'EXT\_SOURCE\_2', 'EXT\_SOURCE\_3',  
 'YEARS\_BEGINEXPLUATATION\_AVG', 'FLOORSMAX\_AVG',  
 'YEARS\_BEGINEXPLUATATION\_MODE', 'FLOORSMAX\_MODE',  
 'YEARS\_BEGINEXPLUATATION\_MEDI', 'FLOORSMAX\_MEDI', 'TOTALAREA\_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', 'FLAG\_DOCUMENT\_2',  
 'FLAG\_DOCUMENT\_3', 'FLAG\_DOCUMENT\_4', 'FLAG\_DOCUMENT\_5',  
 'FLAG\_DOCUMENT\_6', 'FLAG\_DOCUMENT\_7', 'FLAG\_DOCUMENT\_8',  
 'FLAG\_DOCUMENT\_9', 'FLAG\_DOCUMENT\_10', 'FLAG\_DOCUMENT\_11',  
 'FLAG\_DOCUMENT\_12', 'FLAG\_DOCUMENT\_13', 'FLAG\_DOCUMENT\_14',  
 'FLAG\_DOCUMENT\_15', 'FLAG\_DOCUMENT\_16', 'FLAG\_DOCUMENT\_17',  
 'FLAG\_DOCUMENT\_18', 'FLAG\_DOCUMENT\_19', 'FLAG\_DOCUMENT\_20',  
 'FLAG\_DOCUMENT\_21', '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')

[15]

len(application\_train.columns[application\_train.isnull().sum() < 153755])

81

[16]

application\_train = application\_train.columns[application\_train.isnull().sum() < 153755]  
application\_train.shape

(307511, 81)

[17]

application\_train.head()

[18]

application\_train.isna.sum,sortvalues(asc=false)

FLOORSMAX\_AVG 153020  
FLOORSMAX\_MEDI 153020  
FLOORSMAX\_MODE 153020  
YEARS\_BEGINEXPLUATATION\_AVG 150007  
YEARS\_BEGINEXPLUATATION\_MEDI 150007  
YEARS\_BEGINEXPLUATATION\_MODE 150007  
TOTALAREA\_MODE 148431  
EMERGENCYSTATE\_MODE 145755  
OCCUPATION\_TYPE 96391  
EXT\_SOURCE\_3 60965  
AMT\_REQ\_CREDIT\_BUREAU\_YEAR 41519  
AMT\_REQ\_CREDIT\_BUREAU\_QRT 41519  
AMT\_REQ\_CREDIT\_BUREAU\_HOUR 41519  
AMT\_REQ\_CREDIT\_BUREAU\_DAY 41519  
AMT\_REQ\_CREDIT\_BUREAU\_WEEK 41519  
AMT\_REQ\_CREDIT\_BUREAU\_MON 41519  
NAME\_TYPE\_SUITE 1292  
DEF\_60\_CNT\_SOCIAL\_CIRCLE 1021  
OBS\_30\_CNT\_SOCIAL\_CIRCLE 1021  
DEF\_30\_CNT\_SOCIAL\_CIRCLE 1021  
OBS\_60\_CNT\_SOCIAL\_CIRCLE 1021  
EXT\_SOURCE\_2 660  
AMT\_GOODS\_PRICE 278  
AMT\_ANNUITY 12  
CNT\_FAM\_MEMBERS 2  
DAYS\_LAST\_PHONE\_CHANGE 1  
FLAG\_DOCUMENT\_5 0  
DAYS\_ID\_PUBLISH 0  
DAYS\_BIRTH 0  
DAYS\_EMPLOYED 0  
DAYS\_REGISTRATION 0  
FLAG\_MOBIL 0  
NAME\_HOUSING\_TYPE 0  
FLAG\_EMP\_PHONE 0  
FLAG\_WORK\_PHONE 0  
FLAG\_CONT\_MOBILE 0  
FLAG\_PHONE 0  
FLAG\_EMAIL 0  
REGION\_POPULATION\_RELATIVE 0  
NAME\_EDUCATION\_TYPE 0  
NAME\_FAMILY\_STATUS 0  
NAME\_INCOME\_TYPE 0  
FLAG\_DOCUMENT\_21 0  
AMT\_CREDIT 0  
AMT\_INCOME\_TOTAL 0  
CNT\_CHILDREN 0  
FLAG\_OWN\_REALTY 0  
FLAG\_OWN\_CAR 0  
CODE\_GENDER 0  
NAME\_CONTRACT\_TYPE 0  
TARGET 0  
FLAG\_DOCUMENT\_20 0  
REGION\_RATING\_CLIENT 0  
FLAG\_DOCUMENT\_4 0  
REGION\_RATING\_CLIENT\_W\_CITY 0  
FLAG\_DOCUMENT\_3 0  
FLAG\_DOCUMENT\_2 0  
FLAG\_DOCUMENT\_6 0  
FLAG\_DOCUMENT\_7 0  
FLAG\_DOCUMENT\_8 0  
dtype: int64

[19]

*# Categorical columns are:*  
  
selectdtypes

['NAME\_INCOME\_TYPE',  
 'ORGANIZATION\_TYPE',  
 'NAME\_CONTRACT\_TYPE',  
 'WEEKDAY\_APPR\_PROCESS\_START',  
 'NAME\_FAMILY\_STATUS',  
 'NAME\_EDUCATION\_TYPE',  
 'OCCUPATION\_TYPE',  
 'NAME\_TYPE\_SUITE',  
 'CODE\_GENDER',  
 'FLAG\_OWN\_REALTY',  
 'EMERGENCYSTATE\_MODE',  
 'FLAG\_OWN\_CAR',  
 'NAME\_HOUSING\_TYPE']

[20]

*# Numerical columns are:*

Index(['SK\_ID\_CURR', 'TARGET', 'CNT\_CHILDREN', 'AMT\_INCOME\_TOTAL',  
 'AMT\_CREDIT', 'AMT\_ANNUITY', 'AMT\_GOODS\_PRICE',  
 'REGION\_POPULATION\_RELATIVE', 'DAYS\_BIRTH', 'DAYS\_EMPLOYED',  
 'DAYS\_REGISTRATION', 'DAYS\_ID\_PUBLISH', 'FLAG\_MOBIL', 'FLAG\_EMP\_PHONE',  
 'FLAG\_WORK\_PHONE', 'FLAG\_CONT\_MOBILE', 'FLAG\_PHONE', 'FLAG\_EMAIL',  
 'CNT\_FAM\_MEMBERS', 'REGION\_RATING\_CLIENT',  
 'REGION\_RATING\_CLIENT\_W\_CITY', 'HOUR\_APPR\_PROCESS\_START',  
 'REG\_REGION\_NOT\_LIVE\_REGION', 'REG\_REGION\_NOT\_WORK\_REGION',  
 'LIVE\_REGION\_NOT\_WORK\_REGION', 'REG\_CITY\_NOT\_LIVE\_CITY',  
 'REG\_CITY\_NOT\_WORK\_CITY', 'LIVE\_CITY\_NOT\_WORK\_CITY', 'EXT\_SOURCE\_2',  
 'EXT\_SOURCE\_3', 'YEARS\_BEGINEXPLUATATION\_AVG', 'FLOORSMAX\_AVG',  
 'YEARS\_BEGINEXPLUATATION\_MODE', 'FLOORSMAX\_MODE',  
 'YEARS\_BEGINEXPLUATATION\_MEDI', 'FLOORSMAX\_MEDI', 'TOTALAREA\_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', 'FLAG\_DOCUMENT\_2', 'FLAG\_DOCUMENT\_3',  
 'FLAG\_DOCUMENT\_4', 'FLAG\_DOCUMENT\_5', 'FLAG\_DOCUMENT\_6',  
 'FLAG\_DOCUMENT\_7', 'FLAG\_DOCUMENT\_8', 'FLAG\_DOCUMENT\_9',  
 'FLAG\_DOCUMENT\_10', 'FLAG\_DOCUMENT\_11', 'FLAG\_DOCUMENT\_12',  
 'FLAG\_DOCUMENT\_13', 'FLAG\_DOCUMENT\_14', 'FLAG\_DOCUMENT\_15',  
 'FLAG\_DOCUMENT\_16', 'FLAG\_DOCUMENT\_17', 'FLAG\_DOCUMENT\_18',  
 'FLAG\_DOCUMENT\_19', 'FLAG\_DOCUMENT\_20', 'FLAG\_DOCUMENT\_21',  
 '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')

[21]

application\_train.shape

(307511, 81)

[22]

*# Not to be done*  
application\_train = application\_train.sample(25000)  
application\_train.shape

(25000, 81)

[23]

application\_train.head()

[24]

application\_train.columns

Index(['SK\_ID\_CURR', 'TARGET', 'NAME\_CONTRACT\_TYPE', 'CODE\_GENDER',  
 'FLAG\_OWN\_CAR', 'FLAG\_OWN\_REALTY', 'CNT\_CHILDREN', 'AMT\_INCOME\_TOTAL',  
 'AMT\_CREDIT', 'AMT\_ANNUITY', 'AMT\_GOODS\_PRICE', 'NAME\_TYPE\_SUITE',  
 'NAME\_INCOME\_TYPE', 'NAME\_EDUCATION\_TYPE', 'NAME\_FAMILY\_STATUS',  
 'NAME\_HOUSING\_TYPE', 'REGION\_POPULATION\_RELATIVE', 'DAYS\_BIRTH',  
 'DAYS\_EMPLOYED', 'DAYS\_REGISTRATION', 'DAYS\_ID\_PUBLISH', 'FLAG\_MOBIL',  
 'FLAG\_EMP\_PHONE', 'FLAG\_WORK\_PHONE', 'FLAG\_CONT\_MOBILE', 'FLAG\_PHONE',  
 'FLAG\_EMAIL', 'OCCUPATION\_TYPE', 'CNT\_FAM\_MEMBERS',  
 'REGION\_RATING\_CLIENT', 'REGION\_RATING\_CLIENT\_W\_CITY',  
 'WEEKDAY\_APPR\_PROCESS\_START', 'HOUR\_APPR\_PROCESS\_START',  
 'REG\_REGION\_NOT\_LIVE\_REGION', 'REG\_REGION\_NOT\_WORK\_REGION',  
 'LIVE\_REGION\_NOT\_WORK\_REGION', 'REG\_CITY\_NOT\_LIVE\_CITY',  
 'REG\_CITY\_NOT\_WORK\_CITY', 'LIVE\_CITY\_NOT\_WORK\_CITY',  
 'ORGANIZATION\_TYPE', 'EXT\_SOURCE\_2', 'EXT\_SOURCE\_3',  
 'YEARS\_BEGINEXPLUATATION\_AVG', 'FLOORSMAX\_AVG',  
 'YEARS\_BEGINEXPLUATATION\_MODE', 'FLOORSMAX\_MODE',  
 'YEARS\_BEGINEXPLUATATION\_MEDI', 'FLOORSMAX\_MEDI', 'TOTALAREA\_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', 'FLAG\_DOCUMENT\_2',  
 'FLAG\_DOCUMENT\_3', 'FLAG\_DOCUMENT\_4', 'FLAG\_DOCUMENT\_5',  
 'FLAG\_DOCUMENT\_6', 'FLAG\_DOCUMENT\_7', 'FLAG\_DOCUMENT\_8',  
 'FLAG\_DOCUMENT\_9', 'FLAG\_DOCUMENT\_10', 'FLAG\_DOCUMENT\_11',  
 'FLAG\_DOCUMENT\_12', 'FLAG\_DOCUMENT\_13', 'FLAG\_DOCUMENT\_14',  
 'FLAG\_DOCUMENT\_15', 'FLAG\_DOCUMENT\_16', 'FLAG\_DOCUMENT\_17',  
 'FLAG\_DOCUMENT\_18', 'FLAG\_DOCUMENT\_19', 'FLAG\_DOCUMENT\_20',  
 'FLAG\_DOCUMENT\_21', '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')

[25]

nulls = application\_train.isnull().sum()  
nulls[nulls > 0]

AMT\_GOODS\_PRICE 31  
NAME\_TYPE\_SUITE 124  
OCCUPATION\_TYPE 7791  
EXT\_SOURCE\_2 46  
EXT\_SOURCE\_3 4937  
YEARS\_BEGINEXPLUATATION\_AVG 12079  
FLOORSMAX\_AVG 12333  
YEARS\_BEGINEXPLUATATION\_MODE 12079  
FLOORSMAX\_MODE 12333  
YEARS\_BEGINEXPLUATATION\_MEDI 12079  
FLOORSMAX\_MEDI 12333  
TOTALAREA\_MODE 11949  
EMERGENCYSTATE\_MODE 11737  
OBS\_30\_CNT\_SOCIAL\_CIRCLE 89  
DEF\_30\_CNT\_SOCIAL\_CIRCLE 89  
OBS\_60\_CNT\_SOCIAL\_CIRCLE 89  
DEF\_60\_CNT\_SOCIAL\_CIRCLE 89  
AMT\_REQ\_CREDIT\_BUREAU\_HOUR 3334  
AMT\_REQ\_CREDIT\_BUREAU\_DAY 3334  
AMT\_REQ\_CREDIT\_BUREAU\_WEEK 3334  
AMT\_REQ\_CREDIT\_BUREAU\_MON 3334  
AMT\_REQ\_CREDIT\_BUREAU\_QRT 3334  
AMT\_REQ\_CREDIT\_BUREAU\_YEAR 3334  
dtype: int64

Age:  
35  
43  
12  
NaN  
45  
54  
NaN  
54  
33  
54  
  
count -> 8  
len -> 10  
  
(8/10)\*100 = 80% **is** the available value %age  
100 - 80% = 20% are the missing value %age

[26]

application\_train.count()

SK\_ID\_CURR 25000  
TARGET 25000  
NAME\_CONTRACT\_TYPE 25000  
CODE\_GENDER 25000  
FLAG\_OWN\_CAR 25000  
 ...   
AMT\_REQ\_CREDIT\_BUREAU\_DAY 21666  
AMT\_REQ\_CREDIT\_BUREAU\_WEEK 21666  
AMT\_REQ\_CREDIT\_BUREAU\_MON 21666  
AMT\_REQ\_CREDIT\_BUREAU\_QRT 21666  
AMT\_REQ\_CREDIT\_BUREAU\_YEAR 21666  
Length: 81, dtype: int64

[27]

len(application\_train)

25000

[28]

train\_missing = application\_train.count()/len(application\_train)  
train\_missing

SK\_ID\_CURR 1.00000  
TARGET 1.00000  
NAME\_CONTRACT\_TYPE 1.00000  
CODE\_GENDER 1.00000  
FLAG\_OWN\_CAR 1.00000  
 ...   
AMT\_REQ\_CREDIT\_BUREAU\_DAY 0.86664  
AMT\_REQ\_CREDIT\_BUREAU\_WEEK 0.86664  
AMT\_REQ\_CREDIT\_BUREAU\_MON 0.86664  
AMT\_REQ\_CREDIT\_BUREAU\_QRT 0.86664  
AMT\_REQ\_CREDIT\_BUREAU\_YEAR 0.86664  
Length: 81, dtype: float64

[29]

train\_missing = (1 - train\_missing)\*100  
train\_missing

SK\_ID\_CURR 0.000  
TARGET 0.000  
NAME\_CONTRACT\_TYPE 0.000  
CODE\_GENDER 0.000  
FLAG\_OWN\_CAR 0.000  
 ...   
AMT\_REQ\_CREDIT\_BUREAU\_DAY 13.336  
AMT\_REQ\_CREDIT\_BUREAU\_WEEK 13.336  
AMT\_REQ\_CREDIT\_BUREAU\_MON 13.336  
AMT\_REQ\_CREDIT\_BUREAU\_QRT 13.336  
AMT\_REQ\_CREDIT\_BUREAU\_YEAR 13.336  
Length: 81, dtype: float64

[30]

train\_missing.sort\_values(ascending=False).head(60)

FLOORSMAX\_AVG 49.332  
FLOORSMAX\_MEDI 49.332  
FLOORSMAX\_MODE 49.332  
YEARS\_BEGINEXPLUATATION\_AVG 48.316  
YEARS\_BEGINEXPLUATATION\_MEDI 48.316  
YEARS\_BEGINEXPLUATATION\_MODE 48.316  
TOTALAREA\_MODE 47.796  
EMERGENCYSTATE\_MODE 46.948  
OCCUPATION\_TYPE 31.164  
EXT\_SOURCE\_3 19.748  
AMT\_REQ\_CREDIT\_BUREAU\_YEAR 13.336  
AMT\_REQ\_CREDIT\_BUREAU\_QRT 13.336  
AMT\_REQ\_CREDIT\_BUREAU\_HOUR 13.336  
AMT\_REQ\_CREDIT\_BUREAU\_DAY 13.336  
AMT\_REQ\_CREDIT\_BUREAU\_WEEK 13.336  
AMT\_REQ\_CREDIT\_BUREAU\_MON 13.336  
NAME\_TYPE\_SUITE 0.496  
DEF\_60\_CNT\_SOCIAL\_CIRCLE 0.356  
OBS\_30\_CNT\_SOCIAL\_CIRCLE 0.356  
DEF\_30\_CNT\_SOCIAL\_CIRCLE 0.356  
OBS\_60\_CNT\_SOCIAL\_CIRCLE 0.356  
EXT\_SOURCE\_2 0.184  
AMT\_GOODS\_PRICE 0.124  
FLAG\_DOCUMENT\_5 0.000  
NAME\_FAMILY\_STATUS 0.000  
FLAG\_EMAIL 0.000  
FLAG\_PHONE 0.000  
FLAG\_CONT\_MOBILE 0.000  
FLAG\_WORK\_PHONE 0.000  
FLAG\_EMP\_PHONE 0.000  
FLAG\_MOBIL 0.000  
DAYS\_ID\_PUBLISH 0.000  
DAYS\_REGISTRATION 0.000  
DAYS\_EMPLOYED 0.000  
DAYS\_BIRTH 0.000  
REGION\_POPULATION\_RELATIVE 0.000  
NAME\_HOUSING\_TYPE 0.000  
NAME\_EDUCATION\_TYPE 0.000  
CNT\_FAM\_MEMBERS 0.000  
NAME\_INCOME\_TYPE 0.000  
FLAG\_DOCUMENT\_21 0.000  
AMT\_ANNUITY 0.000  
AMT\_CREDIT 0.000  
AMT\_INCOME\_TOTAL 0.000  
CNT\_CHILDREN 0.000  
FLAG\_OWN\_REALTY 0.000  
FLAG\_OWN\_CAR 0.000  
CODE\_GENDER 0.000  
NAME\_CONTRACT\_TYPE 0.000  
TARGET 0.000  
FLAG\_DOCUMENT\_20 0.000  
REGION\_RATING\_CLIENT 0.000  
FLAG\_DOCUMENT\_4 0.000  
FLAG\_DOCUMENT\_15 0.000  
FLAG\_DOCUMENT\_3 0.000  
FLAG\_DOCUMENT\_2 0.000  
DAYS\_LAST\_PHONE\_CHANGE 0.000  
FLAG\_DOCUMENT\_6 0.000  
FLAG\_DOCUMENT\_7 0.000  
FLAG\_DOCUMENT\_8 0.000  
dtype: float64

Missing value treatment  
  
As you can observe, there are lot of columns **with** missing values. There are some columns which has missing values around **or** more than 50%. Other columns has significantly less missing value. Also, the columns **for** which has missing values are around **or** more than 50% are mostly either mean, median **or** mode. So, there **is** no way one can replace these missing data. So, we will **not** consider these columns **for** analysis. We will consider other columns **for** analysis. Let's analyse the other columns.

[31]

application\_train['FLOORSMAX\_AVG'].head()

10864 0.1667  
250852 0.1667  
225261 NaN  
48543 NaN  
139310 NaN  
Name: FLOORSMAX\_AVG, dtype: float64

[32]

application\_train['FLOORSMAX\_AVG'].mean()

0.22571756532724152

[33]

application\_train['FLOORSMAX\_AVG'].mode()

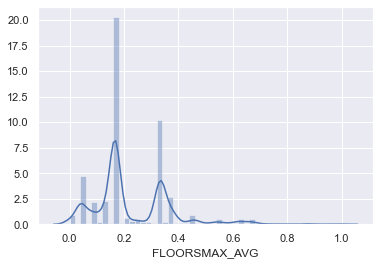
0 0.1667  
dtype: float64

[34]

application\_train['FLOORSMAX\_AVG'].describe()

count 12667.000000  
mean 0.225718  
std 0.144105  
min 0.000000  
25% 0.166700  
50% 0.166700  
75% 0.333300  
max 1.000000  
Name: FLOORSMAX\_AVG, dtype: float64

[35]



[36]

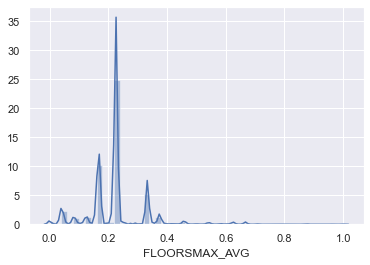
[37]

application\_train['FLOORSMAX\_AVG'].isna().sum()

0

[38]

sns.distplot(application\_train['FLOORSMAX\_AVG']);



[39]

train\_missing.sort\_values(ascending=False).head(60)

FLOORSMAX\_AVG 49.332  
FLOORSMAX\_MEDI 49.332  
FLOORSMAX\_MODE 49.332  
YEARS\_BEGINEXPLUATATION\_AVG 48.316  
YEARS\_BEGINEXPLUATATION\_MEDI 48.316  
YEARS\_BEGINEXPLUATATION\_MODE 48.316  
TOTALAREA\_MODE 47.796  
EMERGENCYSTATE\_MODE 46.948  
OCCUPATION\_TYPE 31.164  
EXT\_SOURCE\_3 19.748  
AMT\_REQ\_CREDIT\_BUREAU\_YEAR 13.336  
AMT\_REQ\_CREDIT\_BUREAU\_QRT 13.336  
AMT\_REQ\_CREDIT\_BUREAU\_HOUR 13.336  
AMT\_REQ\_CREDIT\_BUREAU\_DAY 13.336  
AMT\_REQ\_CREDIT\_BUREAU\_WEEK 13.336  
AMT\_REQ\_CREDIT\_BUREAU\_MON 13.336  
NAME\_TYPE\_SUITE 0.496  
DEF\_60\_CNT\_SOCIAL\_CIRCLE 0.356  
OBS\_30\_CNT\_SOCIAL\_CIRCLE 0.356  
DEF\_30\_CNT\_SOCIAL\_CIRCLE 0.356  
OBS\_60\_CNT\_SOCIAL\_CIRCLE 0.356  
EXT\_SOURCE\_2 0.184  
AMT\_GOODS\_PRICE 0.124  
FLAG\_DOCUMENT\_5 0.000  
NAME\_FAMILY\_STATUS 0.000  
FLAG\_EMAIL 0.000  
FLAG\_PHONE 0.000  
FLAG\_CONT\_MOBILE 0.000  
FLAG\_WORK\_PHONE 0.000  
FLAG\_EMP\_PHONE 0.000  
FLAG\_MOBIL 0.000  
DAYS\_ID\_PUBLISH 0.000  
DAYS\_REGISTRATION 0.000  
DAYS\_EMPLOYED 0.000  
DAYS\_BIRTH 0.000  
REGION\_POPULATION\_RELATIVE 0.000  
NAME\_HOUSING\_TYPE 0.000  
NAME\_EDUCATION\_TYPE 0.000  
CNT\_FAM\_MEMBERS 0.000  
NAME\_INCOME\_TYPE 0.000  
FLAG\_DOCUMENT\_21 0.000  
AMT\_ANNUITY 0.000  
AMT\_CREDIT 0.000  
AMT\_INCOME\_TOTAL 0.000  
CNT\_CHILDREN 0.000  
FLAG\_OWN\_REALTY 0.000  
FLAG\_OWN\_CAR 0.000  
CODE\_GENDER 0.000  
NAME\_CONTRACT\_TYPE 0.000  
TARGET 0.000  
FLAG\_DOCUMENT\_20 0.000  
REGION\_RATING\_CLIENT 0.000  
FLAG\_DOCUMENT\_4 0.000  
FLAG\_DOCUMENT\_15 0.000  
FLAG\_DOCUMENT\_3 0.000  
FLAG\_DOCUMENT\_2 0.000  
DAYS\_LAST\_PHONE\_CHANGE 0.000  
FLAG\_DOCUMENT\_6 0.000  
FLAG\_DOCUMENT\_7 0.000  
FLAG\_DOCUMENT\_8 0.000  
dtype: float64

Column: OCCUPATION\_TYPE  
As you can see, OCCUPATION\_TYPE column has 31% missing data, which **is** also a huge number. So, it would be approprite to remove this column, but **if** you go through this column, this seems to look important. So, we will **not** analyse this column.

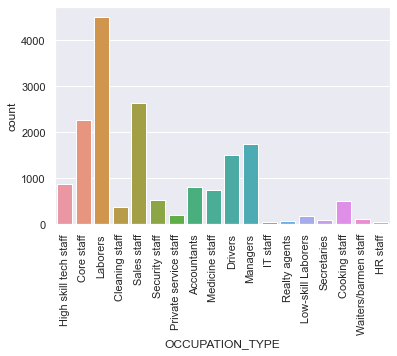
[40]

application\_train['OCCUPATION\_TYPE'].head()

10864 High skill tech staff  
250852 Core staff  
225261 Core staff  
48543 Laborers  
139310 NaN  
Name: OCCUPATION\_TYPE, dtype: object

[41]

*#application\_train['OCCUPATION\_TYPE']*



[42]

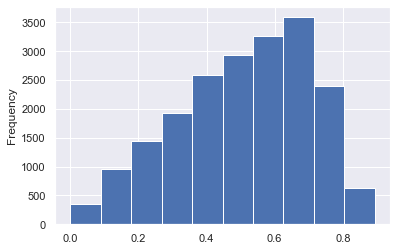
application\_train['EXT\_SOURCE\_3'].head(10)

10864 0.171468  
250852 0.256706  
225261 0.726711  
48543 0.176653  
139310 0.540654  
253366 0.108924  
149247 0.600658  
124813 0.282248  
69716 0.177704  
270030 NaN  
Name: EXT\_SOURCE\_3, dtype: float64

[43]

*# Plotting the distribution*

<matplotlib.axes.\_subplots.AxesSubplot at 0x25d627bb160>



[44]

application\_train['EXT\_SOURCE\_3'].mean()

0.5089481510656194

[45]

application\_train['EXT\_SOURCE\_3'].mode()

0 0.713631  
dtype: float64

This data **is** a skewed normal distribution

Column: AMT\_REQ\_CREDIT\_BUREAU\_QRT

[46]

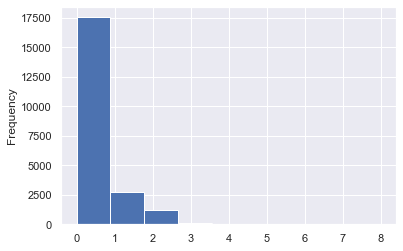
application\_train['AMT\_REQ\_CREDIT\_BUREAU\_QRT'].head()

10864 0.0  
250852 0.0  
225261 0.0  
48543 0.0  
139310 0.0  
Name: AMT\_REQ\_CREDIT\_BUREAU\_QRT, dtype: float64

[47]

*#application\_train['AMT\_REQ\_CREDIT\_BUREAU\_QRT'].plot.hist()*

<matplotlib.axes.\_subplots.AxesSubplot at 0x25d62836940>



[48]

application\_train.AMT\_REQ\_CREDIT\_BUREAU\_QRT.value\_counts()

0.0 17537  
1.0 2744  
2.0 1195  
3.0 140  
4.0 41  
5.0 5  
8.0 2  
6.0 2  
Name: AMT\_REQ\_CREDIT\_BUREAU\_QRT, dtype: int64

[49]

application\_train['AMT\_REQ\_CREDIT\_BUREAU\_QRT'].mean()

0.26636204190898183

[50]

application\_train['AMT\_REQ\_CREDIT\_BUREAU\_QRT'].mode()

0 0.0  
dtype: float64

Since the column only takes discrete values, 1 **or** 0, so we cannot replace it by mean value. Here, we will replace it by mode.

[51]

application\_train['AMT\_REQ\_CREDIT\_BUREAU\_QRT']=  
                application\_train['AMT\_REQ\_CREDIT\_BUREAU\_QRT'].  
                    fillna(application\_train['AMT\_REQ\_CREDIT\_BUREAU\_QRT'].mode().iloc[0])

[52]

application\_train['AMT\_REQ\_CREDIT\_BUREAU\_QRT'].isna().sum()

0

[53]

train\_missing.sort\_values(ascending=False).head(60)

FLOORSMAX\_AVG 49.332  
FLOORSMAX\_MEDI 49.332  
FLOORSMAX\_MODE 49.332  
YEARS\_BEGINEXPLUATATION\_AVG 48.316  
YEARS\_BEGINEXPLUATATION\_MEDI 48.316  
YEARS\_BEGINEXPLUATATION\_MODE 48.316  
TOTALAREA\_MODE 47.796  
EMERGENCYSTATE\_MODE 46.948  
OCCUPATION\_TYPE 31.164  
EXT\_SOURCE\_3 19.748  
AMT\_REQ\_CREDIT\_BUREAU\_YEAR 13.336  
AMT\_REQ\_CREDIT\_BUREAU\_QRT 13.336  
AMT\_REQ\_CREDIT\_BUREAU\_HOUR 13.336  
AMT\_REQ\_CREDIT\_BUREAU\_DAY 13.336  
AMT\_REQ\_CREDIT\_BUREAU\_WEEK 13.336  
AMT\_REQ\_CREDIT\_BUREAU\_MON 13.336  
NAME\_TYPE\_SUITE 0.496  
DEF\_60\_CNT\_SOCIAL\_CIRCLE 0.356  
OBS\_30\_CNT\_SOCIAL\_CIRCLE 0.356  
DEF\_30\_CNT\_SOCIAL\_CIRCLE 0.356  
OBS\_60\_CNT\_SOCIAL\_CIRCLE 0.356  
EXT\_SOURCE\_2 0.184  
AMT\_GOODS\_PRICE 0.124  
FLAG\_DOCUMENT\_5 0.000  
NAME\_FAMILY\_STATUS 0.000  
FLAG\_EMAIL 0.000  
FLAG\_PHONE 0.000  
FLAG\_CONT\_MOBILE 0.000  
FLAG\_WORK\_PHONE 0.000  
FLAG\_EMP\_PHONE 0.000  
FLAG\_MOBIL 0.000  
DAYS\_ID\_PUBLISH 0.000  
DAYS\_REGISTRATION 0.000  
DAYS\_EMPLOYED 0.000  
DAYS\_BIRTH 0.000  
REGION\_POPULATION\_RELATIVE 0.000  
NAME\_HOUSING\_TYPE 0.000  
NAME\_EDUCATION\_TYPE 0.000  
CNT\_FAM\_MEMBERS 0.000  
NAME\_INCOME\_TYPE 0.000  
FLAG\_DOCUMENT\_21 0.000  
AMT\_ANNUITY 0.000  
AMT\_CREDIT 0.000  
AMT\_INCOME\_TOTAL 0.000  
CNT\_CHILDREN 0.000  
FLAG\_OWN\_REALTY 0.000  
FLAG\_OWN\_CAR 0.000  
CODE\_GENDER 0.000  
NAME\_CONTRACT\_TYPE 0.000  
TARGET 0.000  
FLAG\_DOCUMENT\_20 0.000  
REGION\_RATING\_CLIENT 0.000  
FLAG\_DOCUMENT\_4 0.000  
FLAG\_DOCUMENT\_15 0.000  
FLAG\_DOCUMENT\_3 0.000  
FLAG\_DOCUMENT\_2 0.000  
DAYS\_LAST\_PHONE\_CHANGE 0.000  
FLAG\_DOCUMENT\_6 0.000  
FLAG\_DOCUMENT\_7 0.000  
FLAG\_DOCUMENT\_8 0.000  
dtype: float64

Column: AMT\_REQ\_CREDIT\_BUREAU\_WEEK

[54]

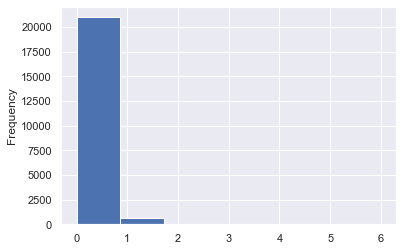
application\_train['AMT\_REQ\_CREDIT\_BUREAU\_WEEK'].head(10)

10864 0.0  
250852 0.0  
225261 0.0  
48543 0.0  
139310 0.0  
253366 0.0  
149247 0.0  
124813 0.0  
69716 0.0  
270030 NaN  
Name: AMT\_REQ\_CREDIT\_BUREAU\_WEEK, dtype: float64

[55]

num\_unique\_values =  len(application\_train.AMT\_REQ\_CREDIT\_BUREAU\_WEEK.unique())  
application\_train['AMT\_REQ\_CREDIT\_BUREAU\_WEEK'].plot.hist(bins = num\_unique\_values)

<matplotlib.axes.\_subplots.AxesSubplot at 0x25d62827100>



As you can see, mostly the values are 0's. So, this column is mostly acting as a constant and has no variation. So, we can ignore this column for analysis.

[56]

application\_train['DEF\_60\_CNT\_SOCIAL\_CIRCLE'].sample(5)

165976 0.0  
142356 1.0  
160193 0.0  
297324 0.0  
40759 0.0  
Name: DEF\_60\_CNT\_SOCIAL\_CIRCLE, dtype: float64

[57]

application\_train['DEF\_60\_CNT\_SOCIAL\_CIRCLE'].describe()

count 24911.000000  
mean 0.098029  
std 0.360350  
min 0.000000  
25% 0.000000  
50% 0.000000  
75% 0.000000  
max 6.000000  
Name: DEF\_60\_CNT\_SOCIAL\_CIRCLE, dtype: float64

[58]

application\_train['DEF\_60\_CNT\_SOCIAL\_CIRCLE'].isna().sum()

89

[59]

application\_train['DEF\_60\_CNT\_SOCIAL\_CIRCLE'].unique()

array([ 0., 1., nan, 2., 3., 5., 4., 6.])

[60]

application\_train['DEF\_60\_CNT\_SOCIAL\_CIRCLE'].mean()

0.09802898318012124

[61]

application\_train['DEF\_60\_CNT\_SOCIAL\_CIRCLE'].median()

0.0

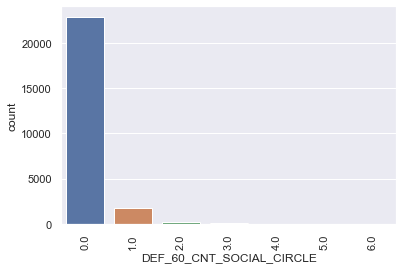
[62]

application\_train['DEF\_60\_CNT\_SOCIAL\_CIRCLE'].mode()

0 0.0  
dtype: float64

[63]

*#application\_train['DEF\_60\_CNT\_SOCIAL\_CIRCLE'].count.plot.hist()*



[64]

application\_train['AMT\_REQ\_CREDIT\_BUREAU\_QRT']=application\_train['DEF\_60\_CNT\_SOCIAL\_CIRCLE'].fillna(  
                application\_train['DEF\_60\_CNT\_SOCIAL\_CIRCLE'].mode().iloc[0])

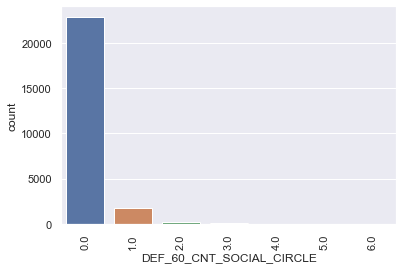
[65]

application\_train['AMT\_REQ\_CREDIT\_BUREAU\_QRT'].isna().sum()

0

[66]

*#application\_train['DEF\_60\_CNT\_SOCIAL\_CIRCLE'].count.plot.hist()*



[67]

*# Categorical columns are:*  
  
list(set(application\_train.columns) - set(application\_train.describe().columns))

['NAME\_INCOME\_TYPE',  
 'ORGANIZATION\_TYPE',  
 'NAME\_CONTRACT\_TYPE',  
 'WEEKDAY\_APPR\_PROCESS\_START',  
 'NAME\_FAMILY\_STATUS',  
 'NAME\_EDUCATION\_TYPE',  
 'OCCUPATION\_TYPE',  
 'NAME\_TYPE\_SUITE',  
 'CODE\_GENDER',  
 'FLAG\_OWN\_REALTY',  
 'EMERGENCYSTATE\_MODE',  
 'FLAG\_OWN\_CAR',  
 'NAME\_HOUSING\_TYPE']

[68]

*# Numerical columns are:*  
application\_train.describe().columns

Index(['SK\_ID\_CURR', 'TARGET', 'CNT\_CHILDREN', 'AMT\_INCOME\_TOTAL',  
 'AMT\_CREDIT', 'AMT\_ANNUITY', 'AMT\_GOODS\_PRICE',  
 'REGION\_POPULATION\_RELATIVE', 'DAYS\_BIRTH', 'DAYS\_EMPLOYED',  
 'DAYS\_REGISTRATION', 'DAYS\_ID\_PUBLISH', 'FLAG\_MOBIL', 'FLAG\_EMP\_PHONE',  
 'FLAG\_WORK\_PHONE', 'FLAG\_CONT\_MOBILE', 'FLAG\_PHONE', 'FLAG\_EMAIL',  
 'CNT\_FAM\_MEMBERS', 'REGION\_RATING\_CLIENT',  
 'REGION\_RATING\_CLIENT\_W\_CITY', 'HOUR\_APPR\_PROCESS\_START',  
 'REG\_REGION\_NOT\_LIVE\_REGION', 'REG\_REGION\_NOT\_WORK\_REGION',  
 'LIVE\_REGION\_NOT\_WORK\_REGION', 'REG\_CITY\_NOT\_LIVE\_CITY',  
 'REG\_CITY\_NOT\_WORK\_CITY', 'LIVE\_CITY\_NOT\_WORK\_CITY', 'EXT\_SOURCE\_2',  
 'EXT\_SOURCE\_3', 'YEARS\_BEGINEXPLUATATION\_AVG', 'FLOORSMAX\_AVG',  
 'YEARS\_BEGINEXPLUATATION\_MODE', 'FLOORSMAX\_MODE',  
 'YEARS\_BEGINEXPLUATATION\_MEDI', 'FLOORSMAX\_MEDI', 'TOTALAREA\_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', 'FLAG\_DOCUMENT\_2', 'FLAG\_DOCUMENT\_3',  
 'FLAG\_DOCUMENT\_4', 'FLAG\_DOCUMENT\_5', 'FLAG\_DOCUMENT\_6',  
 'FLAG\_DOCUMENT\_7', 'FLAG\_DOCUMENT\_8', 'FLAG\_DOCUMENT\_9',  
 'FLAG\_DOCUMENT\_10', 'FLAG\_DOCUMENT\_11', 'FLAG\_DOCUMENT\_12',  
 'FLAG\_DOCUMENT\_13', 'FLAG\_DOCUMENT\_14', 'FLAG\_DOCUMENT\_15',  
 'FLAG\_DOCUMENT\_16', 'FLAG\_DOCUMENT\_17', 'FLAG\_DOCUMENT\_18',  
 'FLAG\_DOCUMENT\_19', 'FLAG\_DOCUMENT\_20', 'FLAG\_DOCUMENT\_21',  
 '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')

[69]

*# Pandas Fillna of Multiple Columns with Mode of Each: Categorical*  
cols = list(set(application\_train.columns) - set(application\_train.describe().columns))   
  
application\_train[cols]=application\_train[cols].fillna(application\_train.mode().iloc[0])

[70]

application\_train[cols].isna().sum()

NAME\_INCOME\_TYPE 0  
ORGANIZATION\_TYPE 0  
NAME\_CONTRACT\_TYPE 0  
WEEKDAY\_APPR\_PROCESS\_START 0  
NAME\_FAMILY\_STATUS 0  
NAME\_EDUCATION\_TYPE 0  
OCCUPATION\_TYPE 0  
NAME\_TYPE\_SUITE 0  
CODE\_GENDER 0  
FLAG\_OWN\_REALTY 0  
EMERGENCYSTATE\_MODE 0  
FLAG\_OWN\_CAR 0  
NAME\_HOUSING\_TYPE 0  
dtype: int64

[71]

application\_train.describe()

[72]

application\_train[application\_train.describe().columns].isna().sum().tolist()

[0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 31,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 46,  
 4937,  
 12079,  
 0,  
 12079,  
 12333,  
 12079,  
 12333,  
 11949,  
 89,  
 89,  
 89,  
 89,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 3334,  
 3334,  
 3334,  
 3334,  
 0,  
 3334]

[73]

nulls = application\_train.isnull().sum()  
nulls[nulls > 0]

AMT\_GOODS\_PRICE 31  
EXT\_SOURCE\_2 46  
EXT\_SOURCE\_3 4937  
YEARS\_BEGINEXPLUATATION\_AVG 12079  
YEARS\_BEGINEXPLUATATION\_MODE 12079  
FLOORSMAX\_MODE 12333  
YEARS\_BEGINEXPLUATATION\_MEDI 12079  
FLOORSMAX\_MEDI 12333  
TOTALAREA\_MODE 11949  
OBS\_30\_CNT\_SOCIAL\_CIRCLE 89  
DEF\_30\_CNT\_SOCIAL\_CIRCLE 89  
OBS\_60\_CNT\_SOCIAL\_CIRCLE 89  
DEF\_60\_CNT\_SOCIAL\_CIRCLE 89  
AMT\_REQ\_CREDIT\_BUREAU\_HOUR 3334  
AMT\_REQ\_CREDIT\_BUREAU\_DAY 3334  
AMT\_REQ\_CREDIT\_BUREAU\_WEEK 3334  
AMT\_REQ\_CREDIT\_BUREAU\_MON 3334  
AMT\_REQ\_CREDIT\_BUREAU\_YEAR 3334  
dtype: int64

[74]

application\_train.mean()

SK\_ID\_CURR 277473.755360  
TARGET 0.080960  
CNT\_CHILDREN 0.416880  
AMT\_INCOME\_TOTAL 168748.506630  
AMT\_CREDIT 600757.606440  
 ...   
AMT\_REQ\_CREDIT\_BUREAU\_DAY 0.007477  
AMT\_REQ\_CREDIT\_BUREAU\_WEEK 0.034016  
AMT\_REQ\_CREDIT\_BUREAU\_MON 0.270931  
AMT\_REQ\_CREDIT\_BUREAU\_QRT 0.097680  
AMT\_REQ\_CREDIT\_BUREAU\_YEAR 1.892551  
Length: 68, dtype: float64

[75]

[76]

application\_train.isnull().sum()

SK\_ID\_CURR 0  
TARGET 0  
NAME\_CONTRACT\_TYPE 0  
CODE\_GENDER 0  
FLAG\_OWN\_CAR 0  
 ..  
AMT\_REQ\_CREDIT\_BUREAU\_DAY 0  
AMT\_REQ\_CREDIT\_BUREAU\_WEEK 0  
AMT\_REQ\_CREDIT\_BUREAU\_MON 0  
AMT\_REQ\_CREDIT\_BUREAU\_QRT 0  
AMT\_REQ\_CREDIT\_BUREAU\_YEAR 0  
Length: 81, dtype: int64

[77]

nulls = application\_train.isnull().sum()  
nulls[nulls > 0]

Series([], dtype: int64)

Data **is** Clean.

**Target**

[78]

application\_train['TARGET'].head()

10864 0  
250852 0  
225261 0  
48543 0  
139310 0  
Name: TARGET, dtype: int64

[79]

application\_train['TARGET'].isna().sum()

0

[80]

application\_train['TARGET'].unique()

array([0, 1], dtype=int64)

[81]

*# Client with payment difficulties*  
  
(application\_train['TARGET']==1).sum()

2024

[82]

*# Other clients*  
  
==0

22976

Check **for** imbalance

[83]

22976

[84]

2024

[85]

*# Ratio*  
  
  
(==0 / ==1).sum()

11.351778656126482

Now, we will analyse numerical **and** categorical data. Let's start with categorical data.

Analysing categorical data  
  
Since due to data imbalance, we will separate out the train **with** a target equal to 0 **and** train **with** a target equal to 1. And we will analyse them individually **and** **try** to find any relationship **if** it exists.

[86]

train\_0 = application\_train.loc[application\_train['TARGET'] == 0]  
train\_1 = application\_train.loc[application\_train['TARGET'] == 1]

[87]

*# Categorical columns are:*  
cat\_col = list(set(application\_train.columns) - set(application\_train.describe().columns))   
cat\_col

['NAME\_INCOME\_TYPE',  
 'ORGANIZATION\_TYPE',  
 'NAME\_CONTRACT\_TYPE',  
 'WEEKDAY\_APPR\_PROCESS\_START',  
 'NAME\_FAMILY\_STATUS',  
 'NAME\_EDUCATION\_TYPE',  
 'OCCUPATION\_TYPE',  
 'NAME\_TYPE\_SUITE',  
 'CODE\_GENDER',  
 'FLAG\_OWN\_REALTY',  
 'EMERGENCYSTATE\_MODE',  
 'FLAG\_OWN\_CAR',  
 'NAME\_HOUSING\_TYPE']

[88]

*# Numerical columns are:*  
num\_col = application\_train.describe().columns  
num\_col

Index(['SK\_ID\_CURR', 'TARGET', 'CNT\_CHILDREN', 'AMT\_INCOME\_TOTAL',  
 'AMT\_CREDIT', 'AMT\_ANNUITY', 'AMT\_GOODS\_PRICE',  
 'REGION\_POPULATION\_RELATIVE', 'DAYS\_BIRTH', 'DAYS\_EMPLOYED',  
 'DAYS\_REGISTRATION', 'DAYS\_ID\_PUBLISH', 'FLAG\_MOBIL', 'FLAG\_EMP\_PHONE',  
 'FLAG\_WORK\_PHONE', 'FLAG\_CONT\_MOBILE', 'FLAG\_PHONE', 'FLAG\_EMAIL',  
 'CNT\_FAM\_MEMBERS', 'REGION\_RATING\_CLIENT',  
 'REGION\_RATING\_CLIENT\_W\_CITY', 'HOUR\_APPR\_PROCESS\_START',  
 'REG\_REGION\_NOT\_LIVE\_REGION', 'REG\_REGION\_NOT\_WORK\_REGION',  
 'LIVE\_REGION\_NOT\_WORK\_REGION', 'REG\_CITY\_NOT\_LIVE\_CITY',  
 'REG\_CITY\_NOT\_WORK\_CITY', 'LIVE\_CITY\_NOT\_WORK\_CITY', 'EXT\_SOURCE\_2',  
 'EXT\_SOURCE\_3', 'YEARS\_BEGINEXPLUATATION\_AVG', 'FLOORSMAX\_AVG',  
 'YEARS\_BEGINEXPLUATATION\_MODE', 'FLOORSMAX\_MODE',  
 'YEARS\_BEGINEXPLUATATION\_MEDI', 'FLOORSMAX\_MEDI', 'TOTALAREA\_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', 'FLAG\_DOCUMENT\_2', 'FLAG\_DOCUMENT\_3',  
 'FLAG\_DOCUMENT\_4', 'FLAG\_DOCUMENT\_5', 'FLAG\_DOCUMENT\_6',  
 'FLAG\_DOCUMENT\_7', 'FLAG\_DOCUMENT\_8', 'FLAG\_DOCUMENT\_9',  
 'FLAG\_DOCUMENT\_10', 'FLAG\_DOCUMENT\_11', 'FLAG\_DOCUMENT\_12',  
 'FLAG\_DOCUMENT\_13', 'FLAG\_DOCUMENT\_14', 'FLAG\_DOCUMENT\_15',  
 'FLAG\_DOCUMENT\_16', 'FLAG\_DOCUMENT\_17', 'FLAG\_DOCUMENT\_18',  
 'FLAG\_DOCUMENT\_19', 'FLAG\_DOCUMENT\_20', 'FLAG\_DOCUMENT\_21',  
 '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')

[89]

application\_train.shape

(25000, 81)

Here, we have used 3 different plots for analysis

* Pie plot: For plotting the all the values present in a column in terms of percentage. So, the sum of those data types will be 100.
* Countplot 1: Here, plotted the count of the different categories. So, Target=0 will have higher count than Target=1.
* Countplot 2: To plot this dataset, we have first divided the dataset into 2 subsets, Target=0 and Target=1. Then again divided the individual Target=0 and Target=1 into different categories. Then, plotted these categories in terms of percentage. So, you can find that the values for Target=0 and Target=1 are mostly equal. Please go through the code of this plot to understand for further doubts.

[90]

**def** plotting(train, train0, train1, column):  
      
    train = train  
    train\_0 = train0  
    train\_1 = train1  
    col = column  
      
    fig = plt.figure(figsize=(13,10))  
      
    ax1 = plt.subplot(221)  
    train[col].value\_counts().plot.pie(autopct = "%1.0f%%", ax=ax1)  
    plt.title('Plotting data for the column: '+ column)  
      
    ax2 = plt.subplot(222)  
    sns.countplot(x= column, hue = 'TARGET', data = train, ax = ax2)  
    plt.xticks(rotation=90)  
    plt.title('Plotting data for target in terms of total count')  
  
  
    ax3 = plt.subplot(223)  
    df = pd.DataFrame()  
    df['0']= ((train\_0[col].value\_counts())/len(train\_0))  
    df['1']= ((train\_1[col].value\_counts())/len(train\_1))  
    df.plot.bar(ax=ax3)  
    plt.title('Plotting data for target in terms of percentage')  
  
  
    fig.tight\_layout() *# Or equivalently,  "plt.tight\_layout()"*  
  
    plt.show()

[91]

train\_categorical = application\_train.select\_dtypes(include=['object']).columns  
train\_categorical

Index(['NAME\_CONTRACT\_TYPE', 'CODE\_GENDER', 'FLAG\_OWN\_CAR', 'FLAG\_OWN\_REALTY',  
 'NAME\_TYPE\_SUITE', 'NAME\_INCOME\_TYPE', 'NAME\_EDUCATION\_TYPE',  
 'NAME\_FAMILY\_STATUS', 'NAME\_HOUSING\_TYPE', 'OCCUPATION\_TYPE',  
 'WEEKDAY\_APPR\_PROCESS\_START', 'ORGANIZATION\_TYPE',  
 'EMERGENCYSTATE\_MODE'],  
 dtype='object')

**Univariate Analysis of the Categorical data**

**Plotting the data.**

Ex: For column "NAME\_CONTRACT\_TYPE", in the first plot, 90% 'cash\_loans' are there and 10% 'Revolving\_loans' are there. In the second plot, the numnber of 'cash\_loans' is ~250,000 and ~2500 for 'cash\_loans' for Target=0. Similarly for Target=1. In the last plot, since we have plotted the dataset in terms of percentage, so, sum of 'blue' colour for 'cash\_loans' and 'Revolving\_loans' for Target=0 is 1. Similarly, sum of orange' colour for 'cash\_loans' and 'Revolving\_loans' for Target=1 is 1.

[92]

**for** column **in** train\_categorical:  
    print("Plotting ", column)  
    plotting(application\_train, train\_0, train\_1, column)  
    print('----------------------------------------------------------------------------------------------')

Plotting NAME\_CONTRACT\_TYPE

----------------------------------------------------------------------------------------------  
Plotting CODE\_GENDER

----------------------------------------------------------------------------------------------  
Plotting FLAG\_OWN\_CAR

----------------------------------------------------------------------------------------------  
Plotting FLAG\_OWN\_REALTY

----------------------------------------------------------------------------------------------  
Plotting NAME\_TYPE\_SUITE

----------------------------------------------------------------------------------------------  
Plotting NAME\_INCOME\_TYPE

----------------------------------------------------------------------------------------------  
Plotting NAME\_EDUCATION\_TYPE

----------------------------------------------------------------------------------------------  
Plotting NAME\_FAMILY\_STATUS

----------------------------------------------------------------------------------------------  
Plotting NAME\_HOUSING\_TYPE

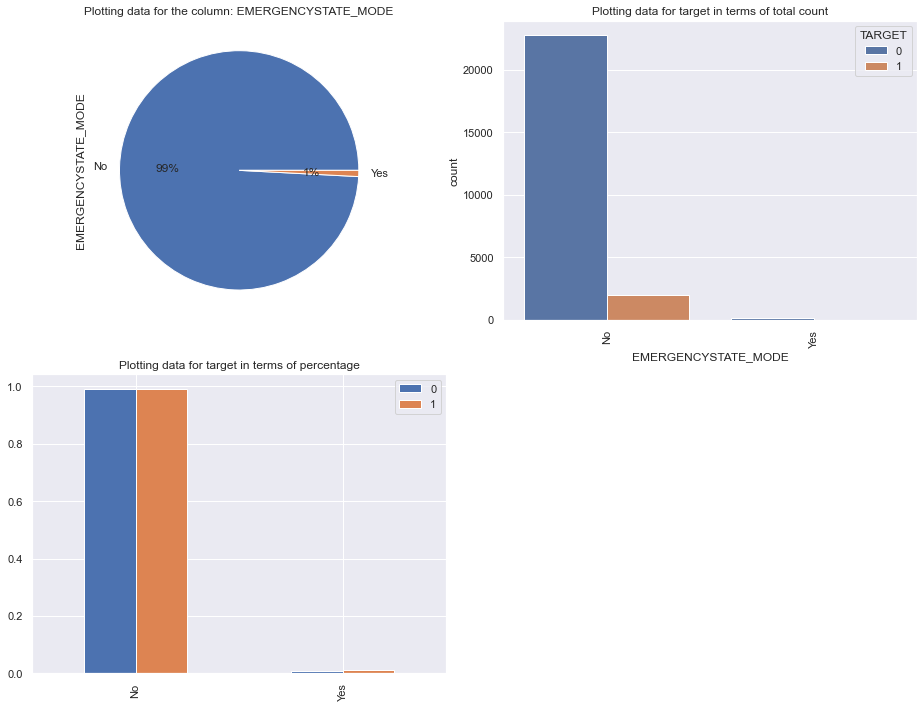
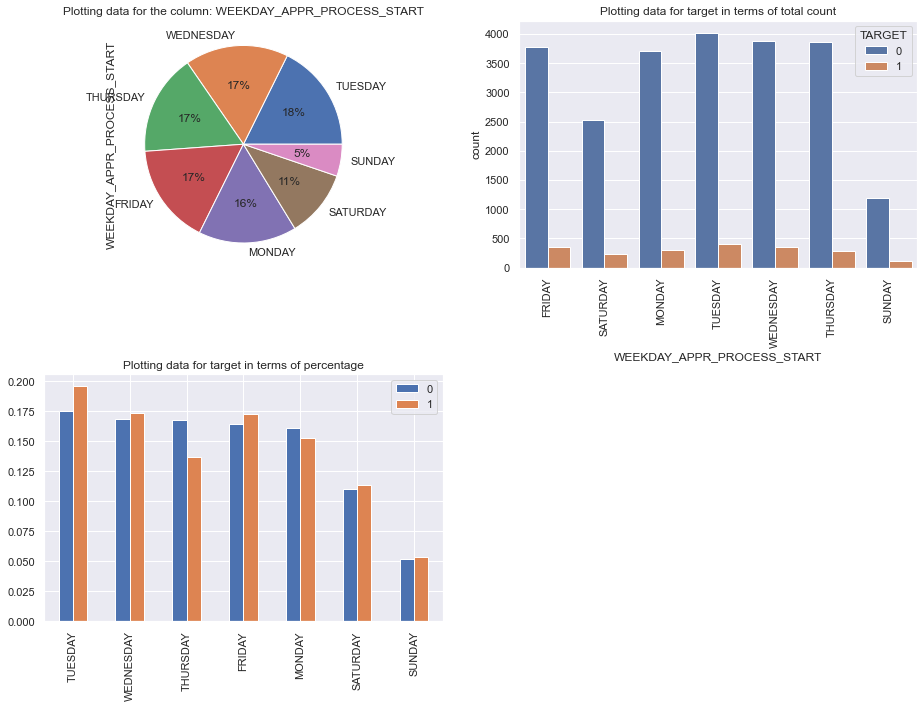
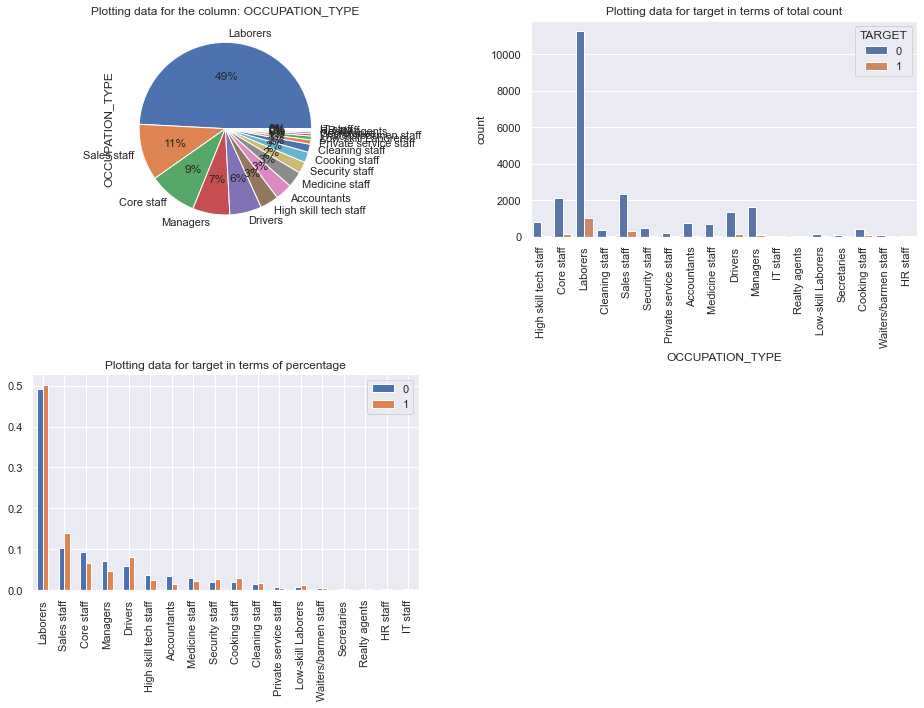
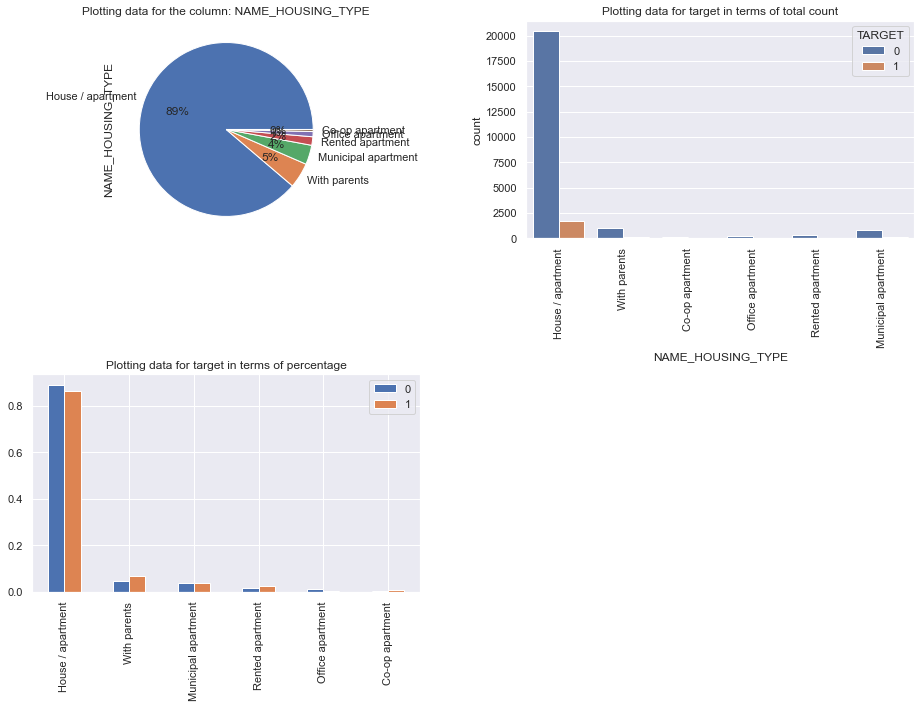
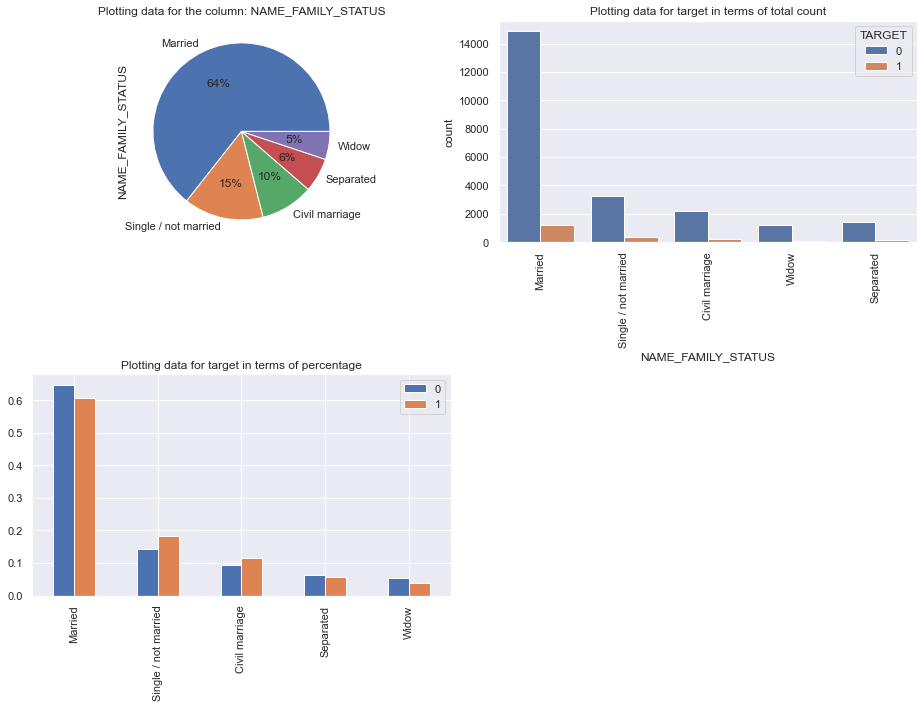
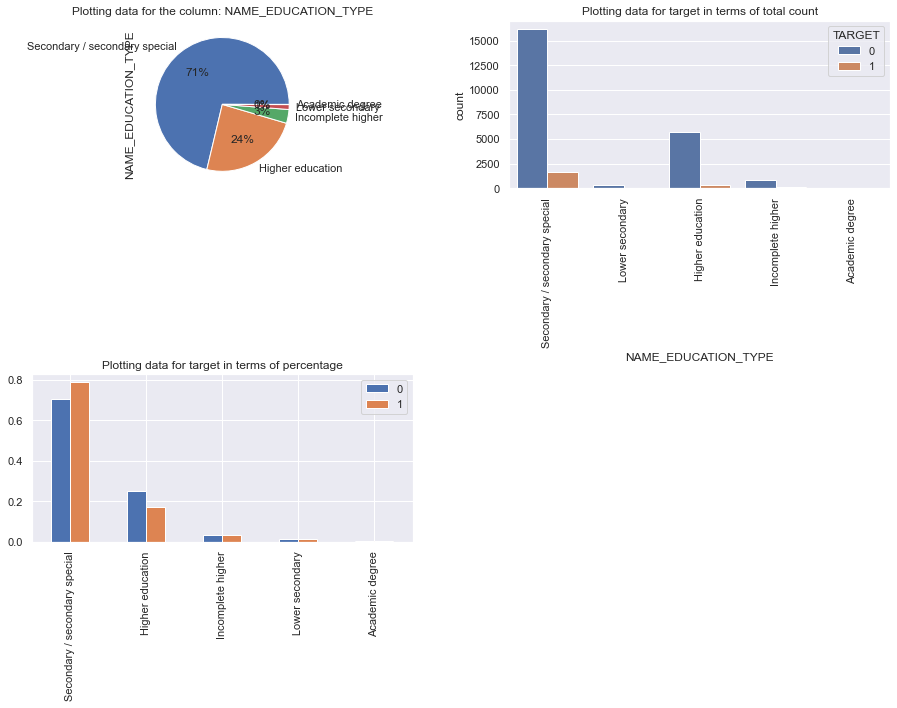
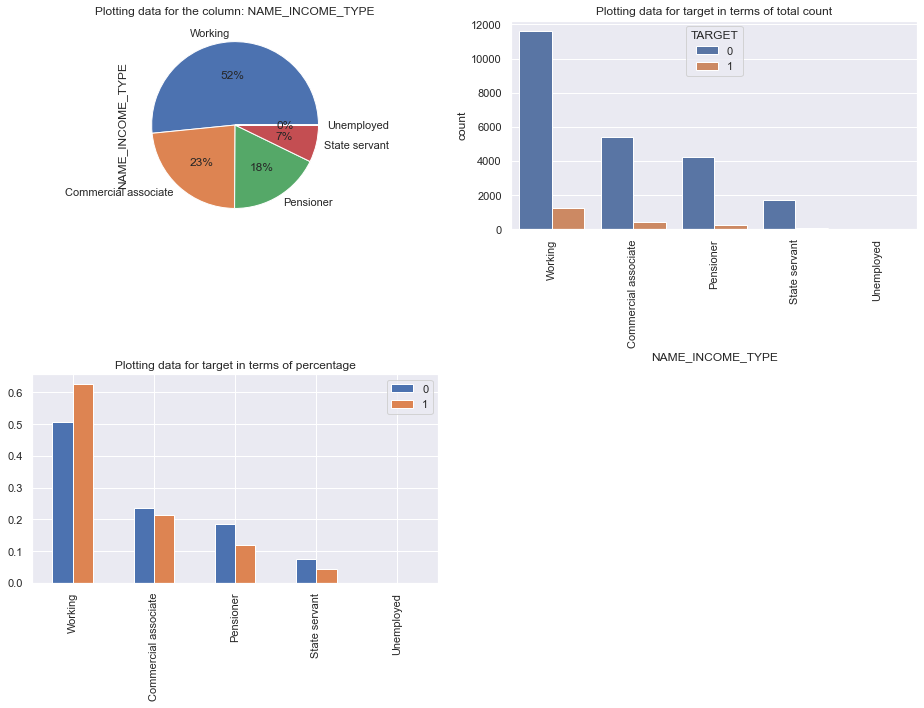
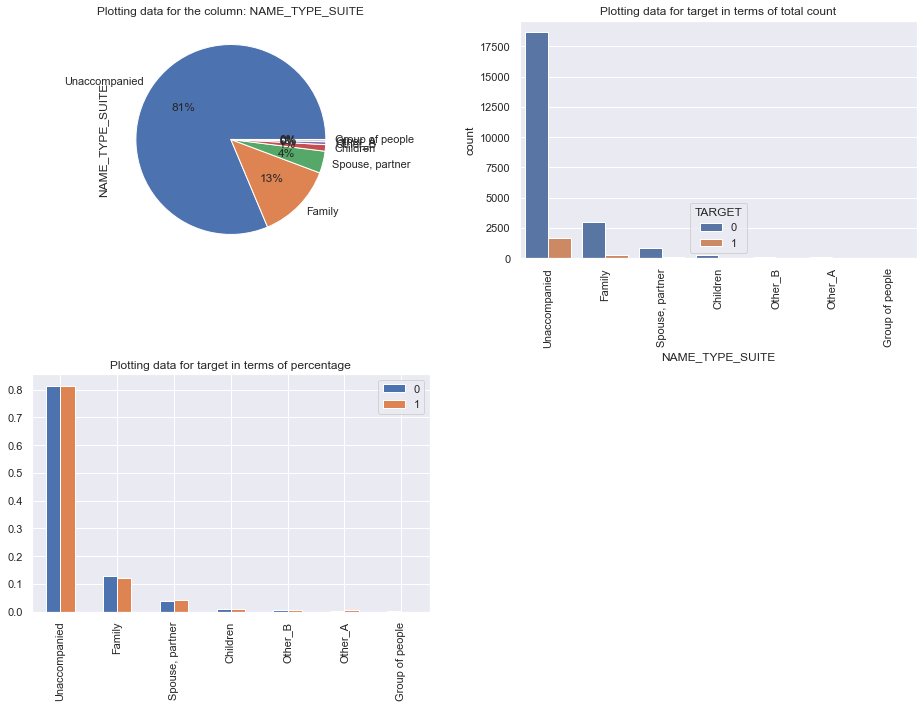
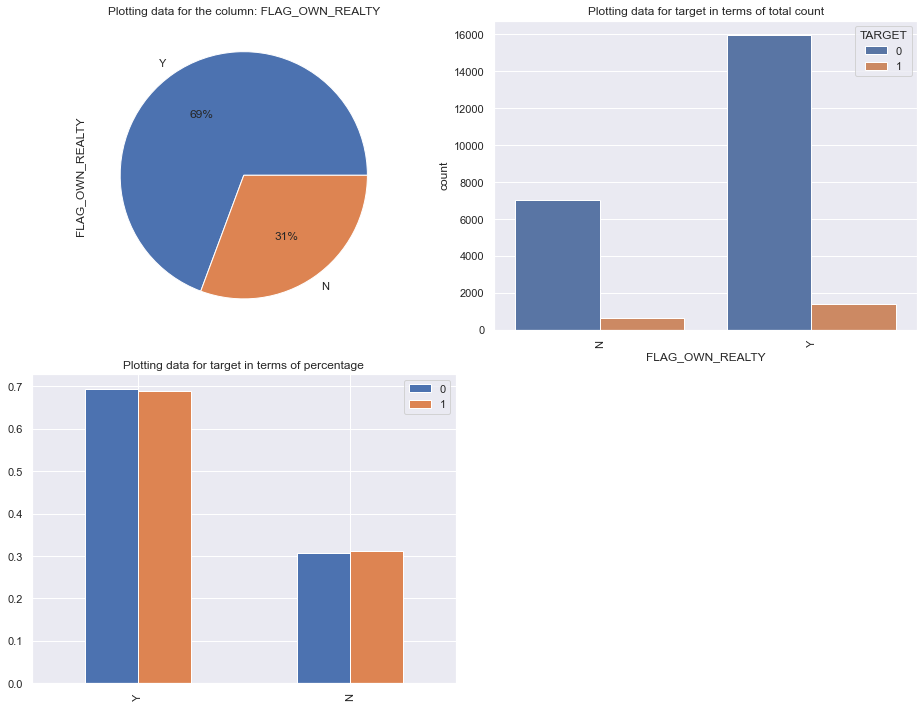
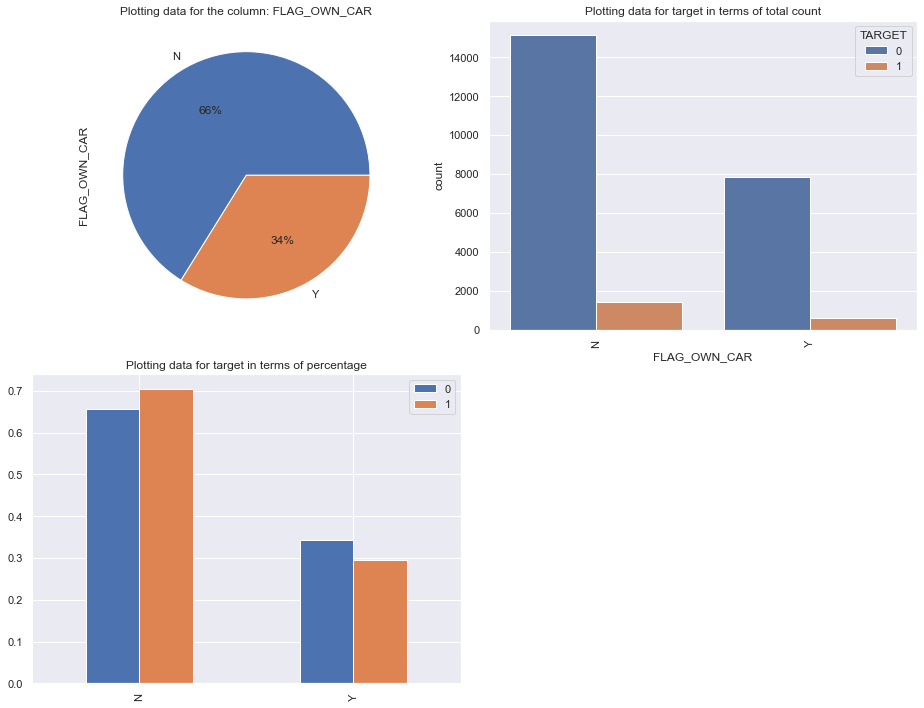
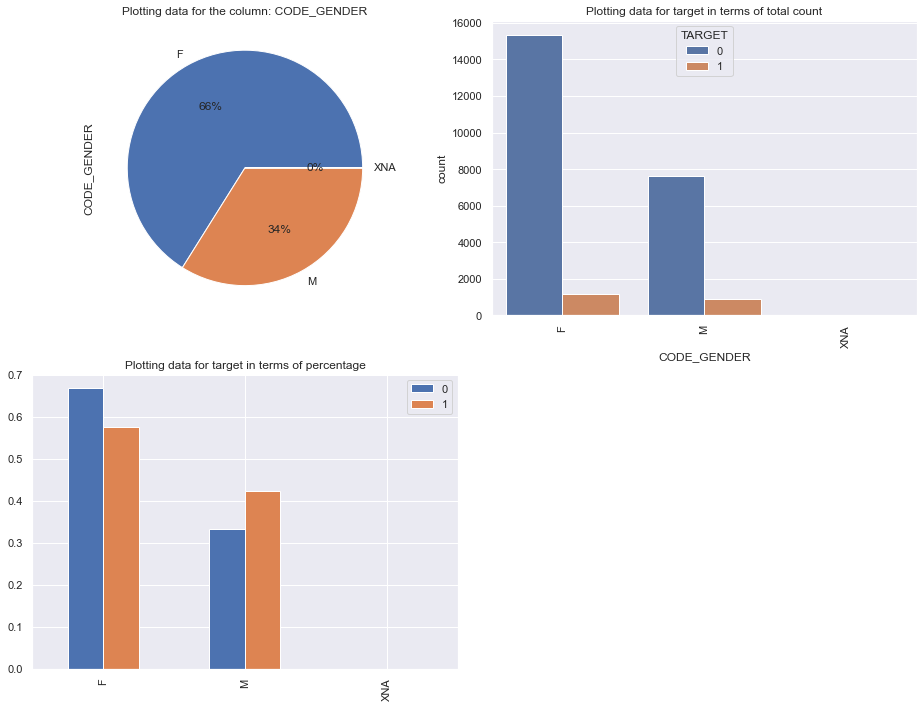
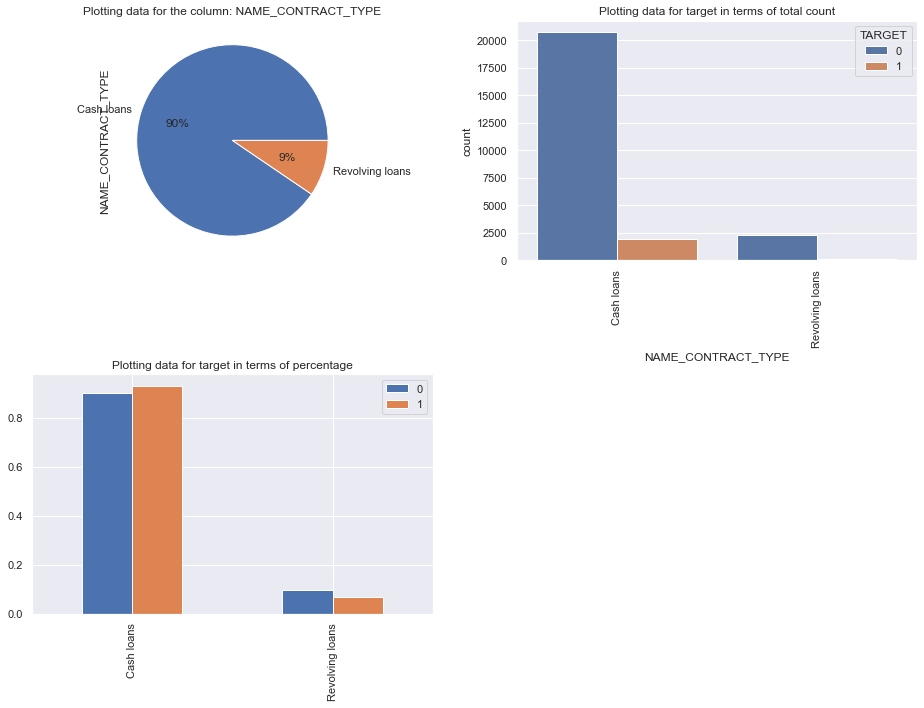
----------------------------------------------------------------------------------------------  
Plotting OCCUPATION\_TYPE

----------------------------------------------------------------------------------------------  
Plotting WEEKDAY\_APPR\_PROCESS\_START

----------------------------------------------------------------------------------------------  
Plotting ORGANIZATION\_TYPE

----------------------------------------------------------------------------------------------  
Plotting EMERGENCYSTATE\_MODE

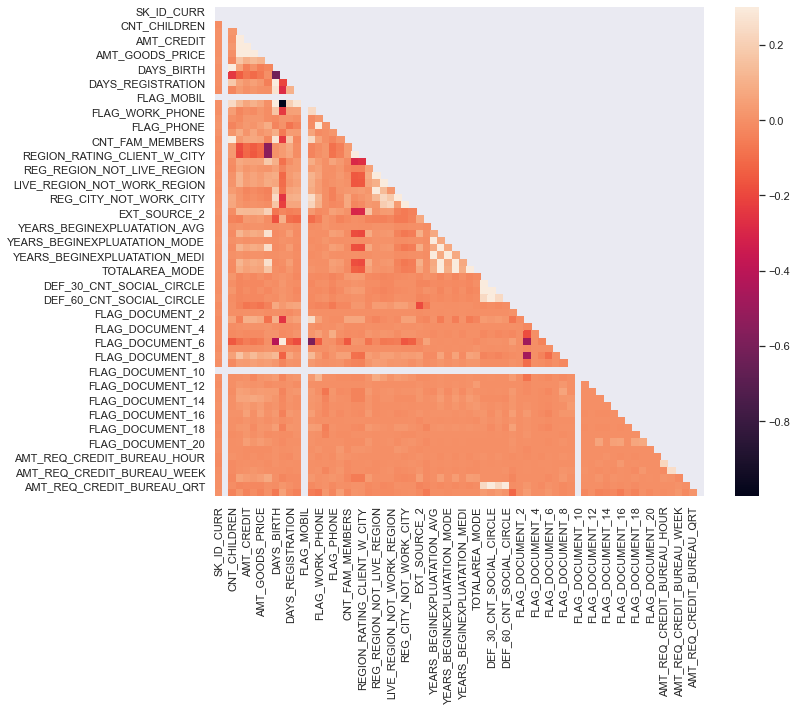
----------------------------------------------------------------------------------------------



**Numerical columns**

[93]

**import** numpy **as** np  
corr = train\_0.corr()  
mask = np.zeros\_like(corr)  
mask[np.triu\_indices\_from(mask)] = True  
f, ax = plt.subplots(figsize=(11, 9))  
**with** sns.axes\_style("white"):  
    ax = sns.heatmap(corr, mask=mask, vmax=.3, square=True)



1  
-1  
0.9  
-0.9  
0.85  
0.6  
0.5  
-0.3  
  
Top 2 correlations:   
Apply abs() on every value **and** then take top2

**Finding the top 10 correlation**

[94]

train\_0.corr()

[95]

train\_0.corr().abs()

[96]

train\_0.corr().abs().unstack()

SK\_ID\_CURR SK\_ID\_CURR 1.000000  
 TARGET NaN  
 CNT\_CHILDREN 0.005076  
 AMT\_INCOME\_TOTAL 0.001454  
 AMT\_CREDIT 0.002913  
 ...   
AMT\_REQ\_CREDIT\_BUREAU\_YEAR AMT\_REQ\_CREDIT\_BUREAU\_DAY 0.002707  
 AMT\_REQ\_CREDIT\_BUREAU\_WEEK 0.019208  
 AMT\_REQ\_CREDIT\_BUREAU\_MON 0.018483  
 AMT\_REQ\_CREDIT\_BUREAU\_QRT 0.011085  
 AMT\_REQ\_CREDIT\_BUREAU\_YEAR 1.000000  
Length: 4624, dtype: float64

[97]

train\_0.corr().abs().unstack().sort\_values()

FLAG\_DOCUMENT\_21 YEARS\_BEGINEXPLUATATION\_MODE 0.000005  
YEARS\_BEGINEXPLUATATION\_MODE FLAG\_DOCUMENT\_21 0.000005  
FLAG\_DOCUMENT\_2 AMT\_REQ\_CREDIT\_BUREAU\_WEEK 0.000006  
AMT\_REQ\_CREDIT\_BUREAU\_WEEK FLAG\_DOCUMENT\_2 0.000006  
FLAG\_DOCUMENT\_21 AMT\_REQ\_CREDIT\_BUREAU\_WEEK 0.000006  
 ...   
AMT\_REQ\_CREDIT\_BUREAU\_QRT FLAG\_MOBIL NaN  
 FLAG\_DOCUMENT\_10 NaN  
AMT\_REQ\_CREDIT\_BUREAU\_YEAR TARGET NaN  
 FLAG\_MOBIL NaN  
 FLAG\_DOCUMENT\_10 NaN  
Length: 4624, dtype: float64

[98]

correlation\_0 = train\_0.corr().abs().unstack().sort\_values().dropna()  
correlation\_0

FLAG\_DOCUMENT\_21 YEARS\_BEGINEXPLUATATION\_MODE 0.000005  
YEARS\_BEGINEXPLUATATION\_MODE FLAG\_DOCUMENT\_21 0.000005  
FLAG\_DOCUMENT\_2 AMT\_REQ\_CREDIT\_BUREAU\_WEEK 0.000006  
AMT\_REQ\_CREDIT\_BUREAU\_WEEK FLAG\_DOCUMENT\_2 0.000006  
FLAG\_DOCUMENT\_21 AMT\_REQ\_CREDIT\_BUREAU\_WEEK 0.000006  
 ...   
DEF\_30\_CNT\_SOCIAL\_CIRCLE DEF\_30\_CNT\_SOCIAL\_CIRCLE 1.000000  
OBS\_30\_CNT\_SOCIAL\_CIRCLE OBS\_30\_CNT\_SOCIAL\_CIRCLE 1.000000  
TOTALAREA\_MODE TOTALAREA\_MODE 1.000000  
FLAG\_DOCUMENT\_9 FLAG\_DOCUMENT\_9 1.000000  
AMT\_REQ\_CREDIT\_BUREAU\_YEAR AMT\_REQ\_CREDIT\_BUREAU\_YEAR 1.000000  
Length: 4225, dtype: float64

[99]

correlation\_0 = correlation\_0[ correlation\_0 != 1.0 ]  
  
print(correlation\_0)

FLAG\_DOCUMENT\_21 YEARS\_BEGINEXPLUATATION\_MODE 0.000005  
YEARS\_BEGINEXPLUATATION\_MODE FLAG\_DOCUMENT\_21 0.000005  
FLAG\_DOCUMENT\_2 AMT\_REQ\_CREDIT\_BUREAU\_WEEK 0.000006  
AMT\_REQ\_CREDIT\_BUREAU\_WEEK FLAG\_DOCUMENT\_2 0.000006  
FLAG\_DOCUMENT\_21 AMT\_REQ\_CREDIT\_BUREAU\_WEEK 0.000006  
 ...   
OBS\_30\_CNT\_SOCIAL\_CIRCLE OBS\_60\_CNT\_SOCIAL\_CIRCLE 0.998270  
DEF\_60\_CNT\_SOCIAL\_CIRCLE AMT\_REQ\_CREDIT\_BUREAU\_QRT 0.999850  
AMT\_REQ\_CREDIT\_BUREAU\_QRT DEF\_60\_CNT\_SOCIAL\_CIRCLE 0.999850  
DAYS\_EMPLOYED FLAG\_EMP\_PHONE 0.999890  
FLAG\_EMP\_PHONE DAYS\_EMPLOYED 0.999890  
Length: 4160, dtype: float64

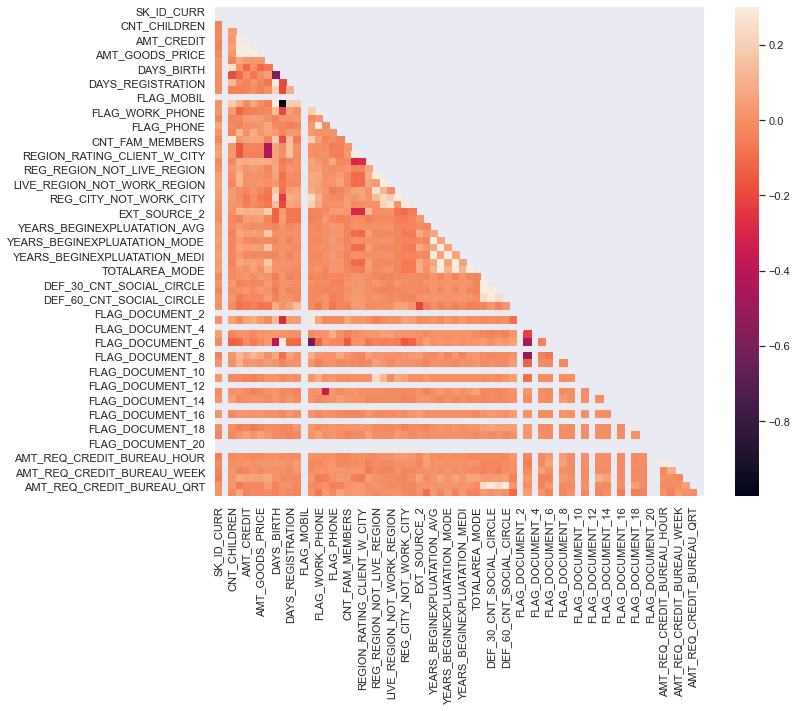
*#### Top correlations*  
'''  
EXT\_SOURCE\_1                 DAYS\_BIRTH                     0.601210  
  
DAYS\_EMPLOYED                DAYS\_BIRTH                     0.618048  
  
AMT\_CREDIT                   AMT\_ANNUITY                    0.771309  
  
AMT\_GOODS\_PRICE              AMT\_ANNUITY                    0.776686  
  
LIVE\_CITY\_NOT\_WORK\_CITY      REG\_CITY\_NOT\_WORK\_CITY         0.830381  
  
LIVE\_REGION\_NOT\_WORK\_REGION  REG\_REGION\_NOT\_WORK\_REGION     0.861861  
  
CNT\_FAM\_MEMBERS              CNT\_CHILDREN                   0.878571  
  
REGION\_RATING\_CLIENT\_W\_CITY  REGION\_RATING\_CLIENT           0.950149  
  
AMT\_CREDIT                   AMT\_GOODS\_PRICE                0.987250  
  
DAYS\_EMPLOYED                FLAG\_EMP\_PHONE                 0.999758  
'''

*### Top correlations for the defaulter*  
'''  
EXT\_SOURCE\_1                 DAYS\_BIRTH                     0.570054  
  
DAYS\_EMPLOYED                DAYS\_BIRTH                     0.575097  
  
FLAG\_EMP\_PHONE               DAYS\_BIRTH                     0.578519  
  
AMT\_CREDIT                   AMT\_ANNUITY                    0.752195  
  
AMT\_GOODS\_PRICE              AMT\_ANNUITY                    0.752699  
  
REG\_CITY\_NOT\_WORK\_CITY       LIVE\_CITY\_NOT\_WORK\_CITY        0.778540  
  
REG\_REGION\_NOT\_WORK\_REGION   LIVE\_REGION\_NOT\_WORK\_REGION    0.847885  
  
CNT\_CHILDREN                 CNT\_FAM\_MEMBERS                0.885484  
  
REGION\_RATING\_CLIENT\_W\_CITY  REGION\_RATING\_CLIENT           0.956637  
  
AMT\_CREDIT                   AMT\_GOODS\_PRICE                0.983103  
  
DAYS\_EMPLOYED                FLAG\_EMP\_PHONE                 0.999702  
'''

**Defaulter correlation**

[100]

corr = train\_1.corr()  
mask = np.zeros\_like(corr)  
mask[np.triu\_indices\_from(mask)] = True  
f, ax = plt.subplots(figsize=(11, 9))  
**with** sns.axes\_style("white"):  
    ax = sns.heatmap(corr, mask=mask, vmax=.3, square=True)



[101]

correlation\_1 = train\_1.corr().abs()  
correlation\_1 = correlation\_1.unstack().sort\_values(kind="quicksort")  
correlation\_1 = correlation\_1.dropna()  
correlation\_1 = correlation\_1[correlation\_1 != 1.0]  
  
print(correlation\_1)

LIVE\_CITY\_NOT\_WORK\_CITY OBS\_60\_CNT\_SOCIAL\_CIRCLE 0.000009  
OBS\_60\_CNT\_SOCIAL\_CIRCLE LIVE\_CITY\_NOT\_WORK\_CITY 0.000009  
TOTALAREA\_MODE AMT\_CREDIT 0.000100  
AMT\_CREDIT TOTALAREA\_MODE 0.000100  
AMT\_REQ\_CREDIT\_BUREAU\_HOUR FLAG\_WORK\_PHONE 0.000223  
 ...   
FLOORSMAX\_AVG FLOORSMAX\_MEDI 0.998744  
YEARS\_BEGINEXPLUATATION\_MEDI YEARS\_BEGINEXPLUATATION\_MODE 0.999742  
YEARS\_BEGINEXPLUATATION\_MODE YEARS\_BEGINEXPLUATATION\_MEDI 0.999742  
DAYS\_EMPLOYED FLAG\_EMP\_PHONE 0.999896  
FLAG\_EMP\_PHONE DAYS\_EMPLOYED 0.999896  
Length: 3190, dtype: float64

**Analysing through box plot**

[102]

train\_categorical = application\_train.select\_dtypes(include=['int64', 'float64']).columns

[103]

train\_categorical

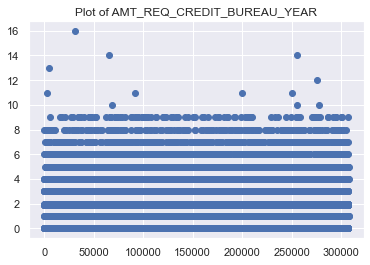
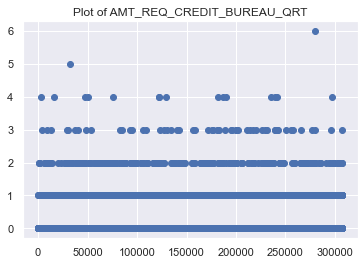
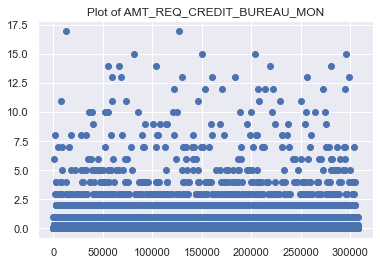
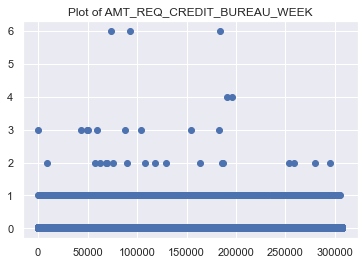
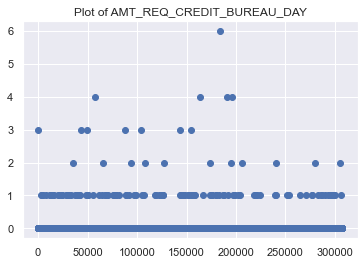
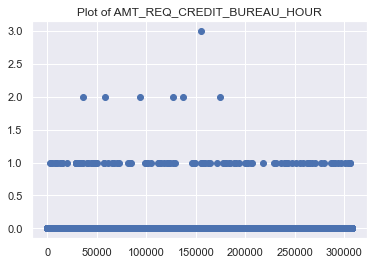
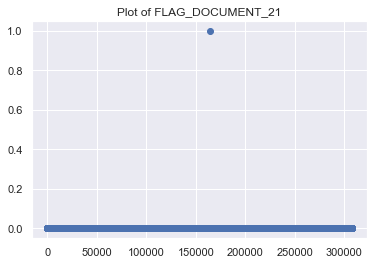
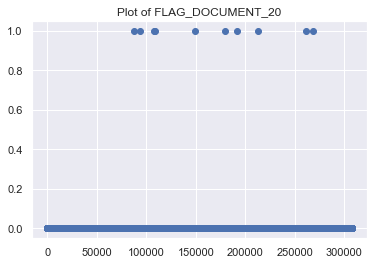
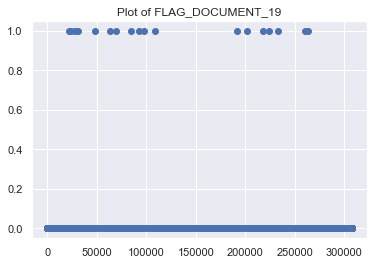
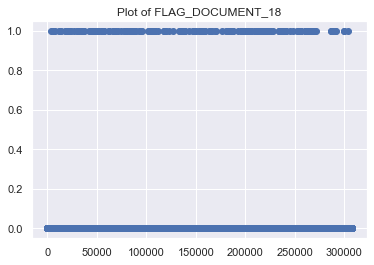
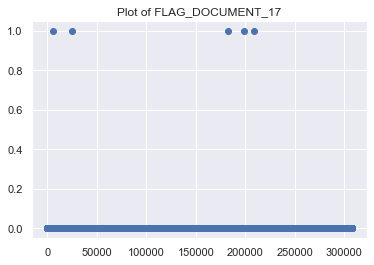
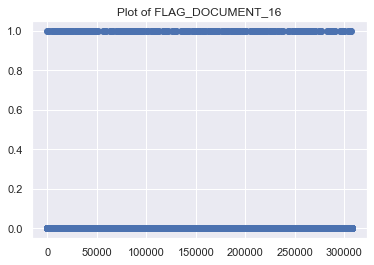
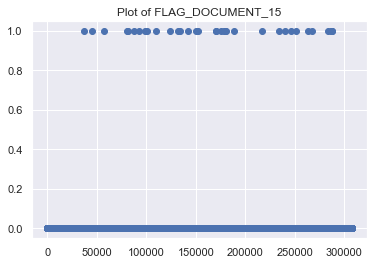
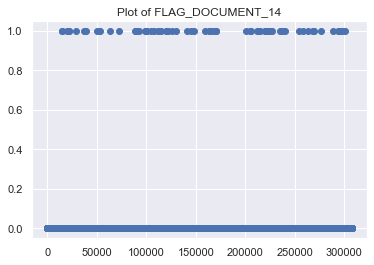
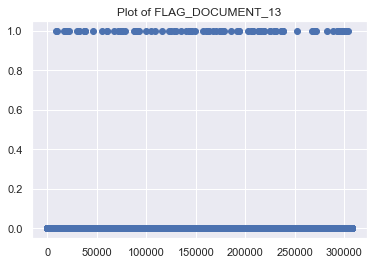
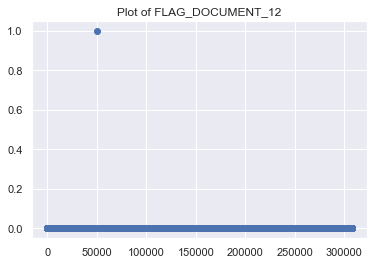
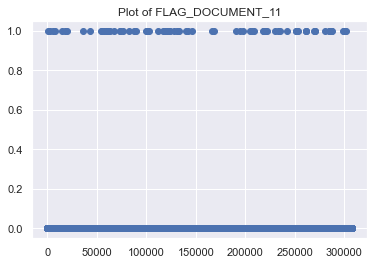
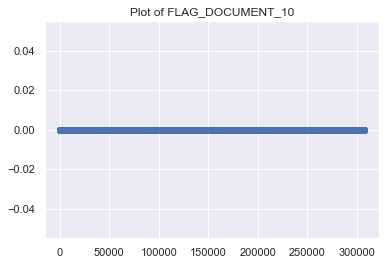
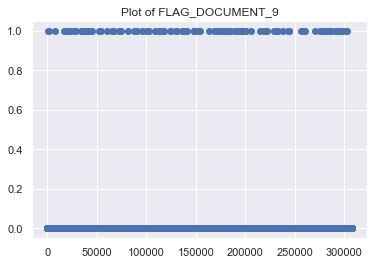
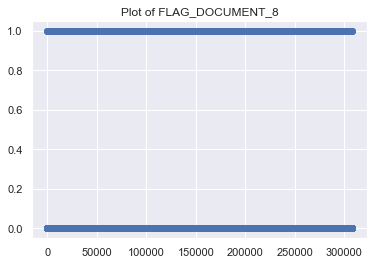
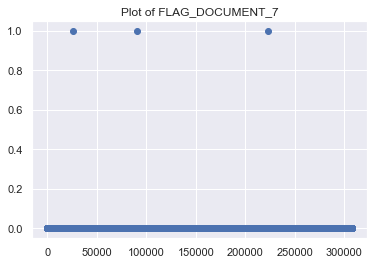
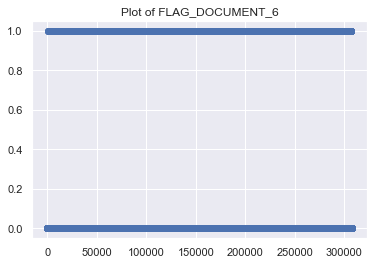
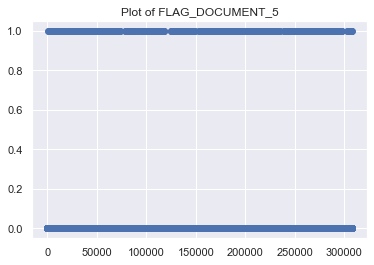
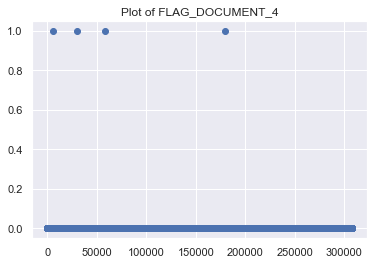
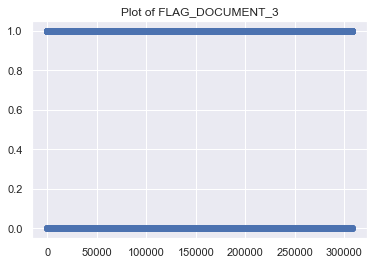
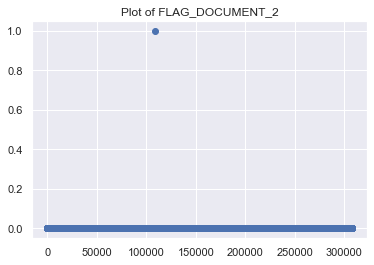
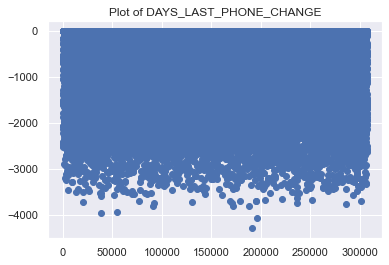
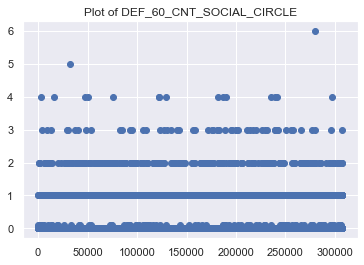
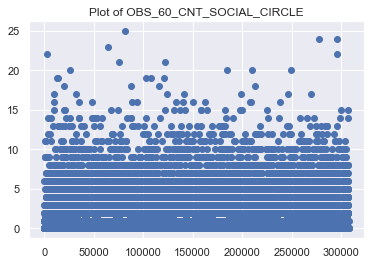
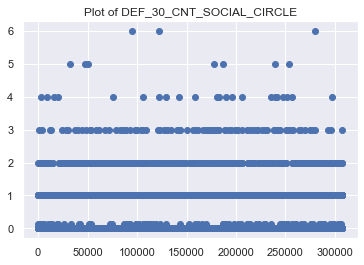
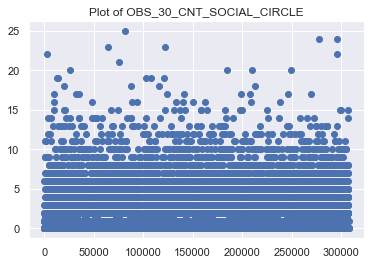
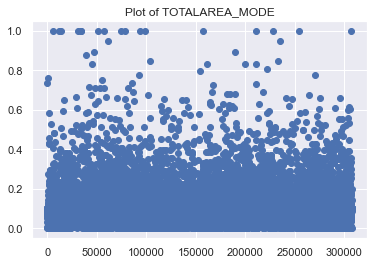
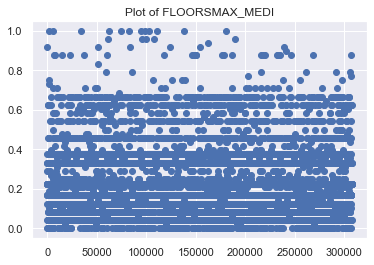
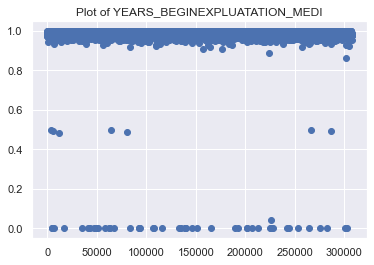
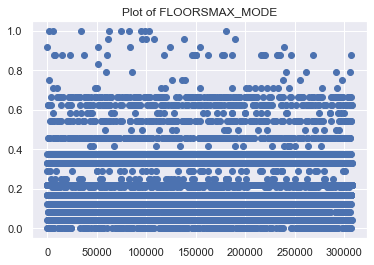
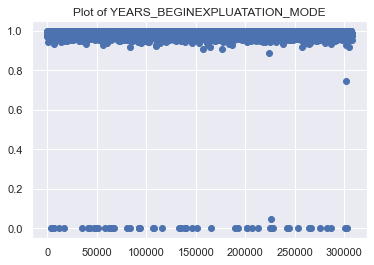
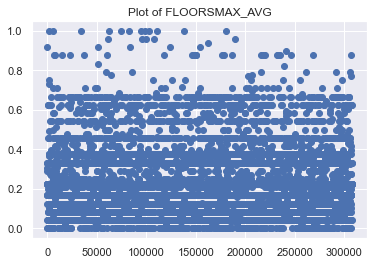
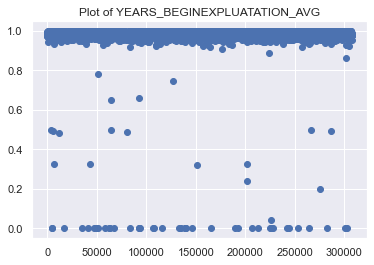
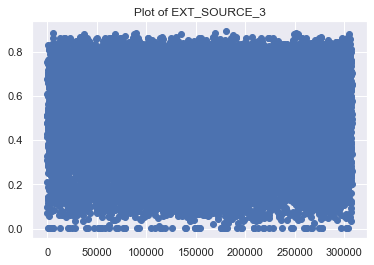
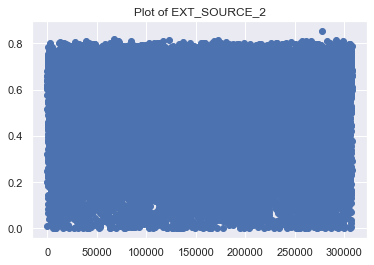
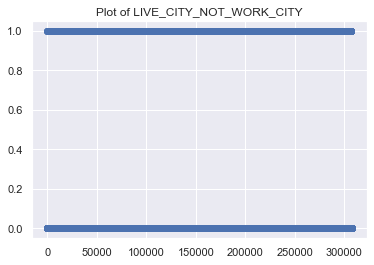
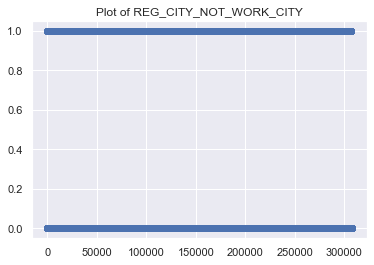
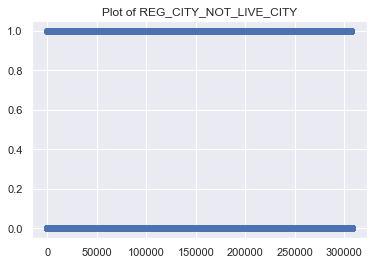
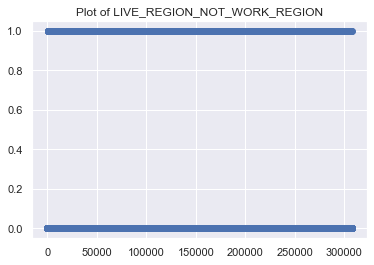
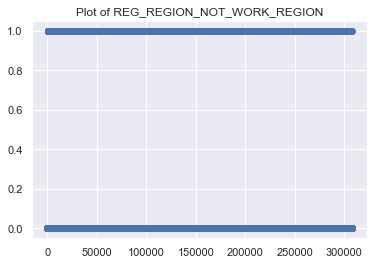
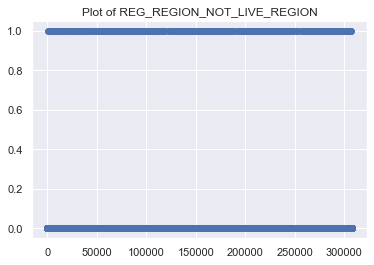
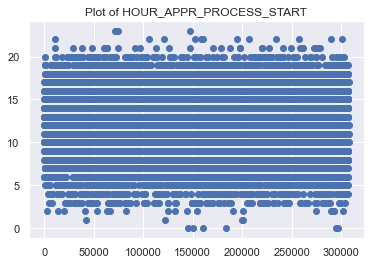
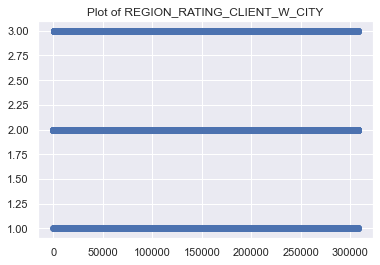
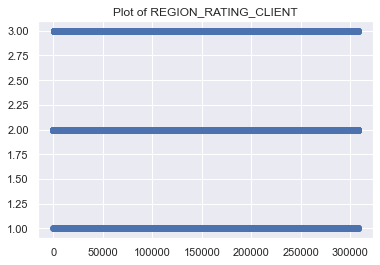
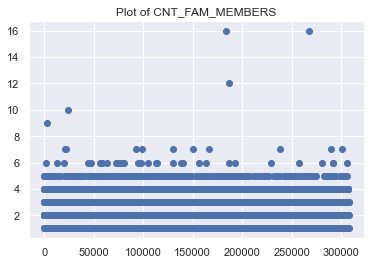
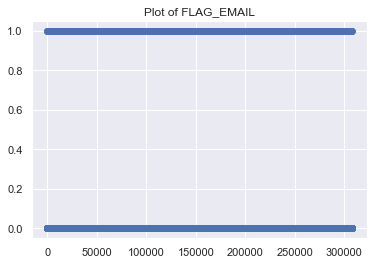
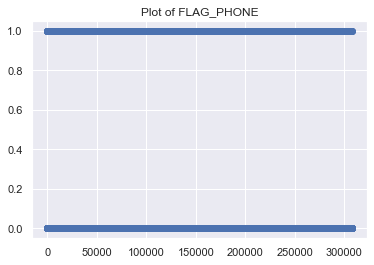
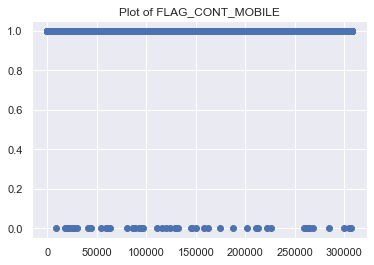
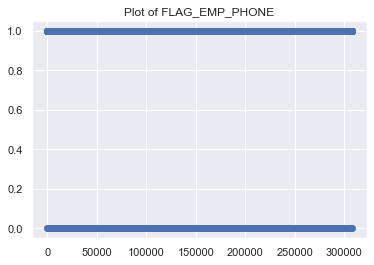
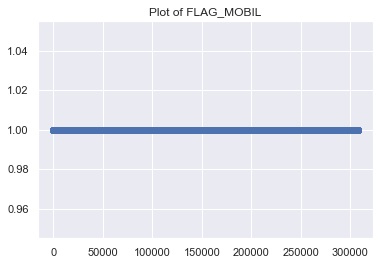
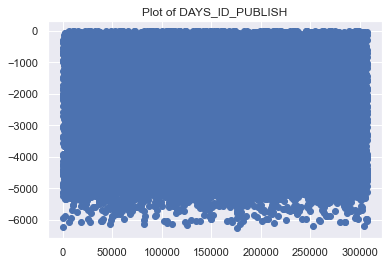
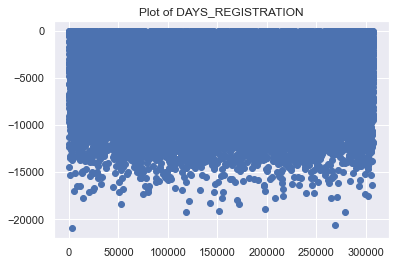
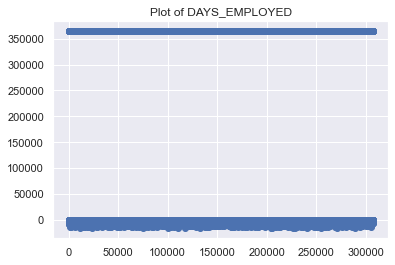
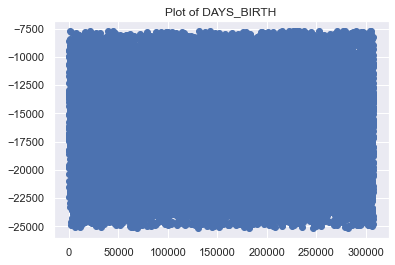
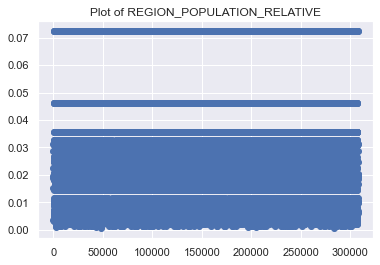
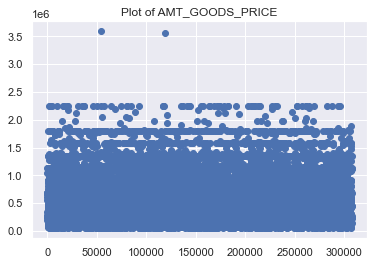
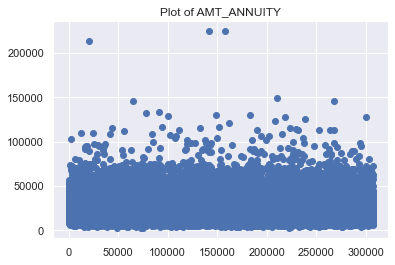
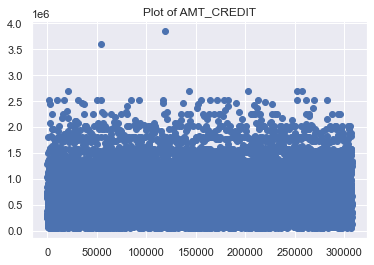
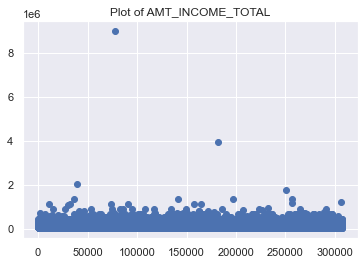
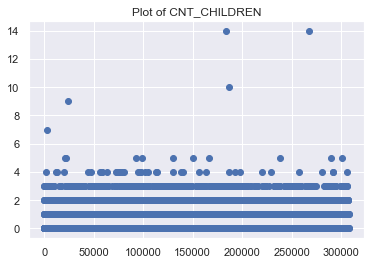
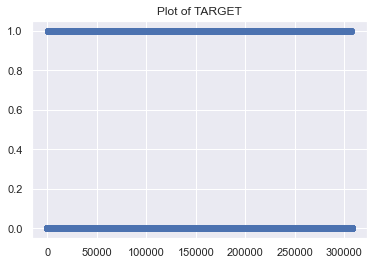
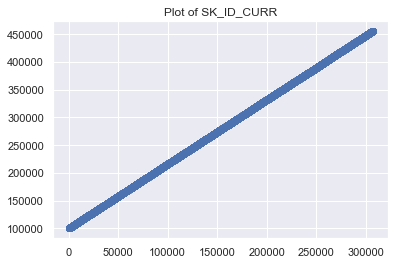
Index(['SK\_ID\_CURR', 'TARGET', 'CNT\_CHILDREN', 'AMT\_INCOME\_TOTAL',  
 'AMT\_CREDIT', 'AMT\_ANNUITY', 'AMT\_GOODS\_PRICE',  
 'REGION\_POPULATION\_RELATIVE', 'DAYS\_BIRTH', 'DAYS\_EMPLOYED',  
 'DAYS\_REGISTRATION', 'DAYS\_ID\_PUBLISH', 'FLAG\_MOBIL', 'FLAG\_EMP\_PHONE',  
 'FLAG\_WORK\_PHONE', 'FLAG\_CONT\_MOBILE', 'FLAG\_PHONE', 'FLAG\_EMAIL',  
 'CNT\_FAM\_MEMBERS', 'REGION\_RATING\_CLIENT',  
 'REGION\_RATING\_CLIENT\_W\_CITY', 'HOUR\_APPR\_PROCESS\_START',  
 'REG\_REGION\_NOT\_LIVE\_REGION', 'REG\_REGION\_NOT\_WORK\_REGION',  
 'LIVE\_REGION\_NOT\_WORK\_REGION', 'REG\_CITY\_NOT\_LIVE\_CITY',  
 'REG\_CITY\_NOT\_WORK\_CITY', 'LIVE\_CITY\_NOT\_WORK\_CITY', 'EXT\_SOURCE\_2',  
 'EXT\_SOURCE\_3', 'YEARS\_BEGINEXPLUATATION\_AVG', 'FLOORSMAX\_AVG',  
 'YEARS\_BEGINEXPLUATATION\_MODE', 'FLOORSMAX\_MODE',  
 'YEARS\_BEGINEXPLUATATION\_MEDI', 'FLOORSMAX\_MEDI', 'TOTALAREA\_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', 'FLAG\_DOCUMENT\_2', 'FLAG\_DOCUMENT\_3',  
 'FLAG\_DOCUMENT\_4', 'FLAG\_DOCUMENT\_5', 'FLAG\_DOCUMENT\_6',  
 'FLAG\_DOCUMENT\_7', 'FLAG\_DOCUMENT\_8', 'FLAG\_DOCUMENT\_9',  
 'FLAG\_DOCUMENT\_10', 'FLAG\_DOCUMENT\_11', 'FLAG\_DOCUMENT\_12',  
 'FLAG\_DOCUMENT\_13', 'FLAG\_DOCUMENT\_14', 'FLAG\_DOCUMENT\_15',  
 'FLAG\_DOCUMENT\_16', 'FLAG\_DOCUMENT\_17', 'FLAG\_DOCUMENT\_18',  
 'FLAG\_DOCUMENT\_19', 'FLAG\_DOCUMENT\_20', 'FLAG\_DOCUMENT\_21',  
 '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')

**Analysis for the outliers**

Potting the numerial data based on the index **and** analysing **if** there are outliers **in** any of the column.

[104]

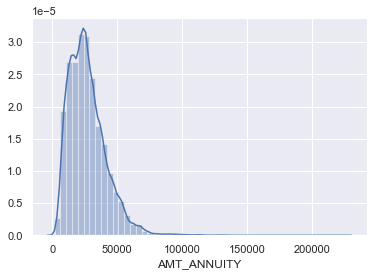
**for** column **in** train\_categorical:  
    title = "Plot of "+column  
    plt.scatter(application\_train.index, application\_train[column])  
    plt.title(title)  
    plt.show()



**Converting a numerical data to categorical for analysis**

[105]

<matplotlib.axes.\_subplots.AxesSubplot at 0x25d626f0a00>



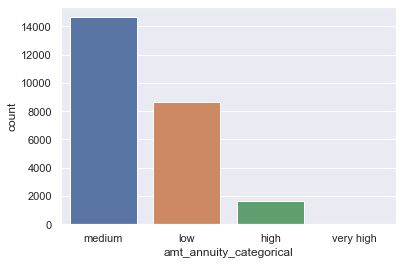
[106]

**def** amt\_annuity(x):  
    **if** x<20000:  
        **return** "low"  
    **elif** x>20k **and** <=50k  
    medium  
    **elif** x>50k **and** <=1lakh  
    high  
    **else**:  
        very high  
  
application\_train['amt\_annuity\_categorical'] = apply the above function

[107]

*# countplot*

<matplotlib.axes.\_subplots.AxesSubplot at 0x25d64295a90>



**Univariate Analysis for numerical data**

For univariate analysis of the numerical columns, we will plot the histogram and the distribution plot.

[108]

**for** column **in** train\_categorical:  
    title = "Plot of "+column  
    print(title)  
    plt.hist(train\_0[column], alpha=0.5, label='0')  
    plt.hist(train\_1[column], alpha=0.5, label='1')  
    plt.show()  
      
    sns.distplot(train\_0[column].dropna(), label='0')  
    sns.distplot(train\_1[column].dropna(),  label='1')  
      
    plt.show()  
    *#box\_plot(train\_0, train\_1, column)*  
    print("------------------------------------------------------------------------")

Plot of SK\_ID\_CURR

------------------------------------------------------------------------  
Plot of TARGET

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must have variance to compute a kernel density estimate.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must have variance to compute a kernel density estimate.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of CNT\_CHILDREN

------------------------------------------------------------------------  
Plot of AMT\_INCOME\_TOTAL

------------------------------------------------------------------------  
Plot of AMT\_CREDIT

------------------------------------------------------------------------  
Plot of AMT\_ANNUITY

------------------------------------------------------------------------  
Plot of AMT\_GOODS\_PRICE

------------------------------------------------------------------------  
Plot of REGION\_POPULATION\_RELATIVE

------------------------------------------------------------------------  
Plot of DAYS\_BIRTH

------------------------------------------------------------------------  
Plot of DAYS\_EMPLOYED

------------------------------------------------------------------------  
Plot of DAYS\_REGISTRATION

------------------------------------------------------------------------  
Plot of DAYS\_ID\_PUBLISH

------------------------------------------------------------------------  
Plot of FLAG\_MOBIL

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must have variance to compute a kernel density estimate.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must have variance to compute a kernel density estimate.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of FLAG\_EMP\_PHONE

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of FLAG\_WORK\_PHONE

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of FLAG\_CONT\_MOBILE

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of FLAG\_PHONE

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of FLAG\_EMAIL

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of CNT\_FAM\_MEMBERS

------------------------------------------------------------------------  
Plot of REGION\_RATING\_CLIENT

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of REGION\_RATING\_CLIENT\_W\_CITY

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of HOUR\_APPR\_PROCESS\_START

------------------------------------------------------------------------  
Plot of REG\_REGION\_NOT\_LIVE\_REGION

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of REG\_REGION\_NOT\_WORK\_REGION

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of LIVE\_REGION\_NOT\_WORK\_REGION

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of REG\_CITY\_NOT\_LIVE\_CITY

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of REG\_CITY\_NOT\_WORK\_CITY

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of LIVE\_CITY\_NOT\_WORK\_CITY

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of EXT\_SOURCE\_2

------------------------------------------------------------------------  
Plot of EXT\_SOURCE\_3

------------------------------------------------------------------------  
Plot of YEARS\_BEGINEXPLUATATION\_AVG

------------------------------------------------------------------------  
Plot of FLOORSMAX\_AVG

------------------------------------------------------------------------  
Plot of YEARS\_BEGINEXPLUATATION\_MODE

------------------------------------------------------------------------  
Plot of FLOORSMAX\_MODE

------------------------------------------------------------------------  
Plot of YEARS\_BEGINEXPLUATATION\_MEDI

------------------------------------------------------------------------  
Plot of FLOORSMAX\_MEDI

------------------------------------------------------------------------  
Plot of TOTALAREA\_MODE

------------------------------------------------------------------------  
Plot of OBS\_30\_CNT\_SOCIAL\_CIRCLE

------------------------------------------------------------------------  
Plot of DEF\_30\_CNT\_SOCIAL\_CIRCLE

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of OBS\_60\_CNT\_SOCIAL\_CIRCLE

------------------------------------------------------------------------  
Plot of DEF\_60\_CNT\_SOCIAL\_CIRCLE

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of DAYS\_LAST\_PHONE\_CHANGE

------------------------------------------------------------------------  
Plot of FLAG\_DOCUMENT\_2

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must have variance to compute a kernel density estimate.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of FLAG\_DOCUMENT\_3

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of FLAG\_DOCUMENT\_4

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must have variance to compute a kernel density estimate.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of FLAG\_DOCUMENT\_5

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of FLAG\_DOCUMENT\_6

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of FLAG\_DOCUMENT\_7

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must have variance to compute a kernel density estimate.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of FLAG\_DOCUMENT\_8

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of FLAG\_DOCUMENT\_9

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of FLAG\_DOCUMENT\_10

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must have variance to compute a kernel density estimate.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must have variance to compute a kernel density estimate.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of FLAG\_DOCUMENT\_11

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of FLAG\_DOCUMENT\_12

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must have variance to compute a kernel density estimate.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of FLAG\_DOCUMENT\_13

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of FLAG\_DOCUMENT\_14

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of FLAG\_DOCUMENT\_15

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must have variance to compute a kernel density estimate.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of FLAG\_DOCUMENT\_16

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of FLAG\_DOCUMENT\_17

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must have variance to compute a kernel density estimate.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of FLAG\_DOCUMENT\_18

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of FLAG\_DOCUMENT\_19

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of FLAG\_DOCUMENT\_20

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must have variance to compute a kernel density estimate.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of FLAG\_DOCUMENT\_21

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:283: UserWarning: Data must have variance to compute a kernel density estimate.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of AMT\_REQ\_CREDIT\_BUREAU\_HOUR

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of AMT\_REQ\_CREDIT\_BUREAU\_DAY

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of AMT\_REQ\_CREDIT\_BUREAU\_WEEK

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

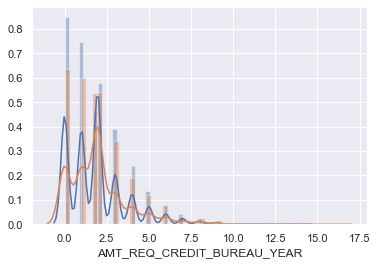
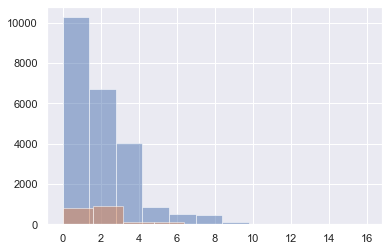
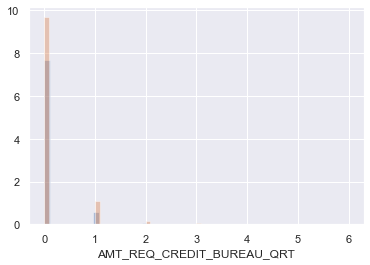
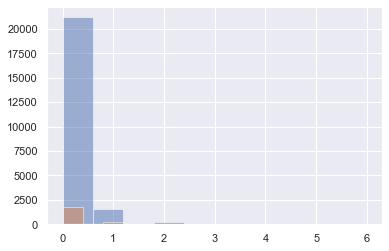
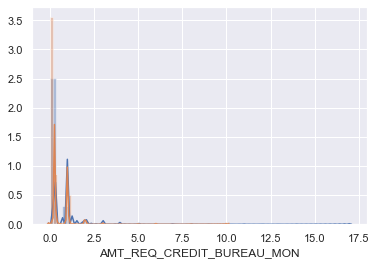
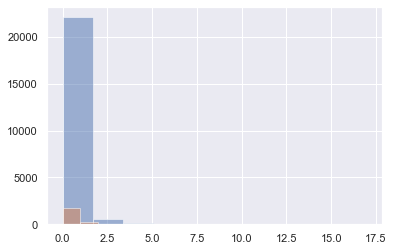
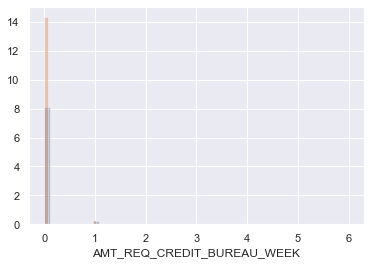
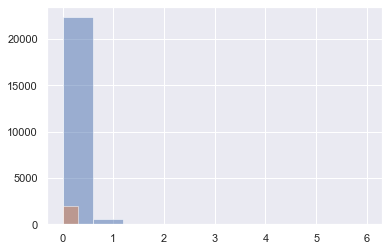
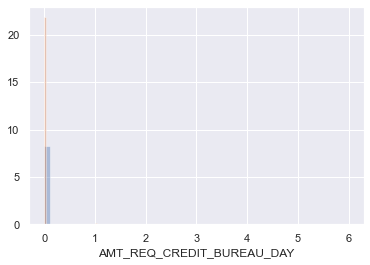
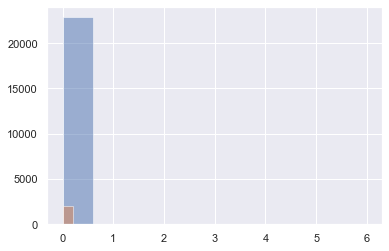
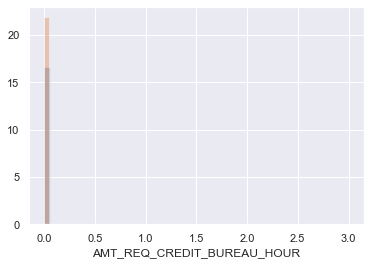
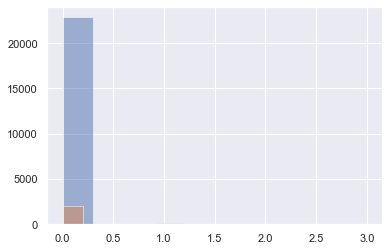
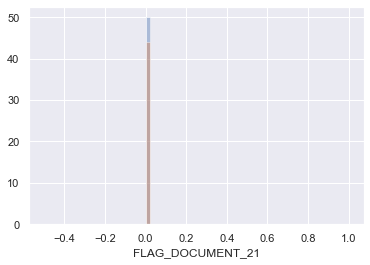
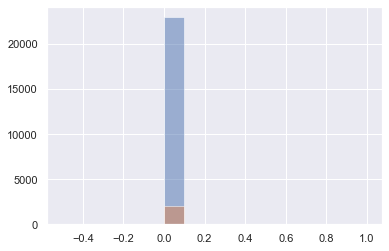
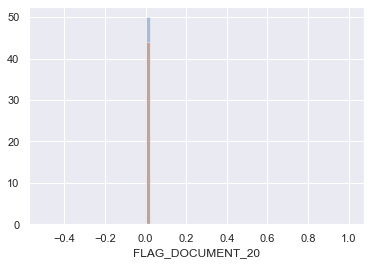
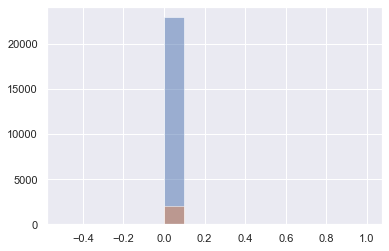
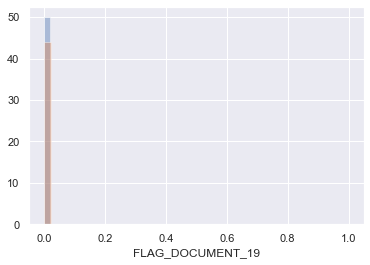
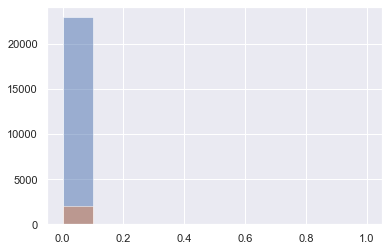
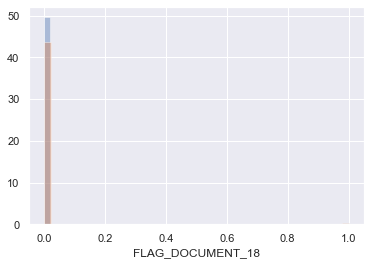
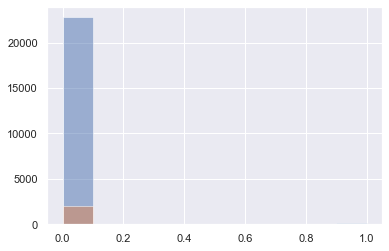
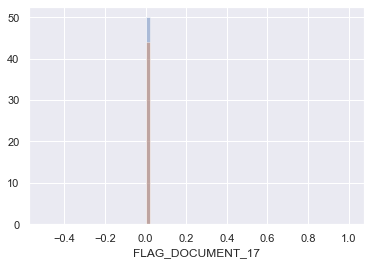
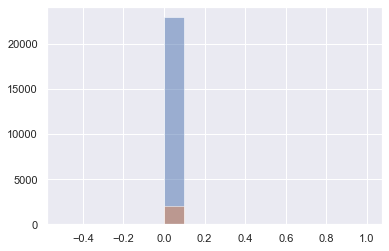
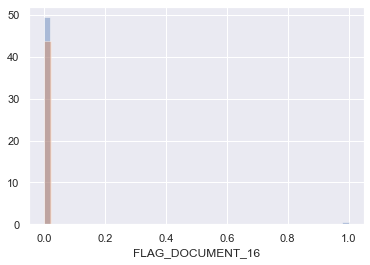
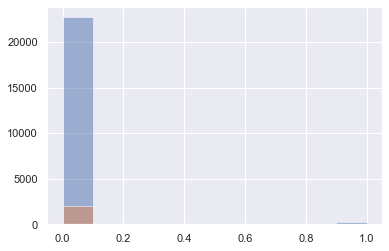
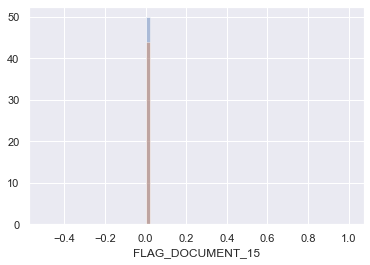
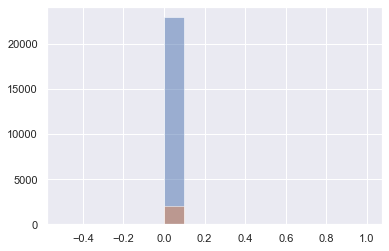
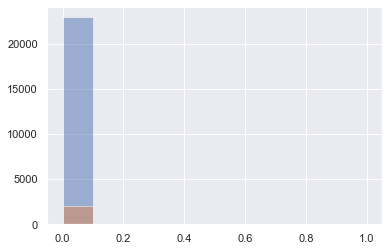
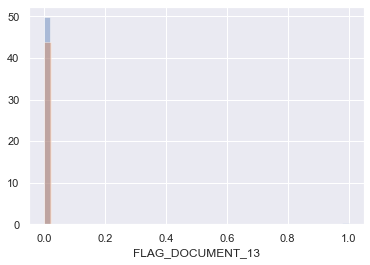
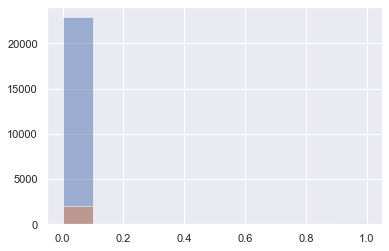
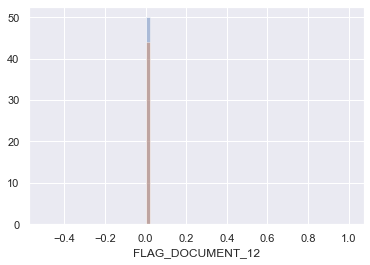
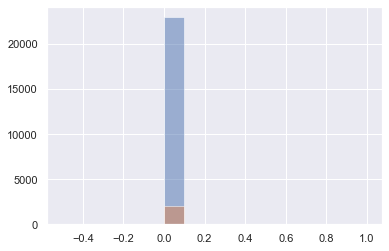
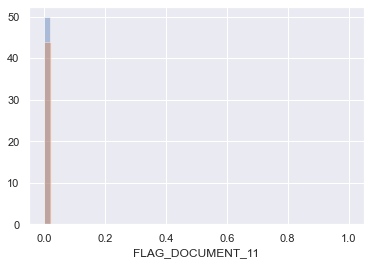
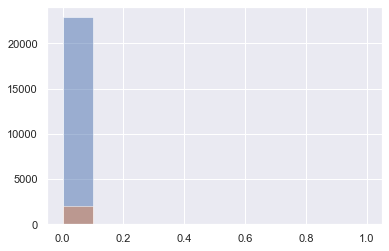
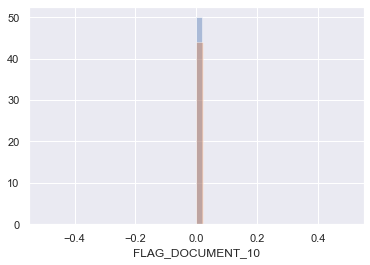
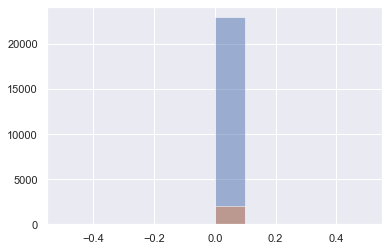
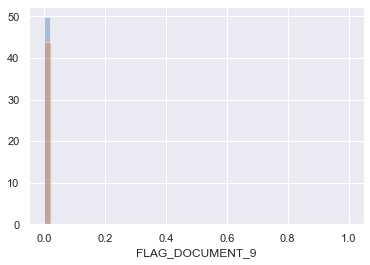
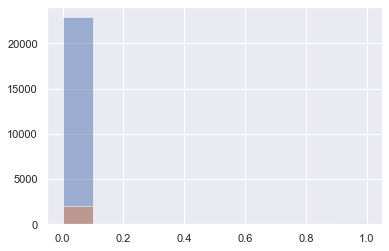
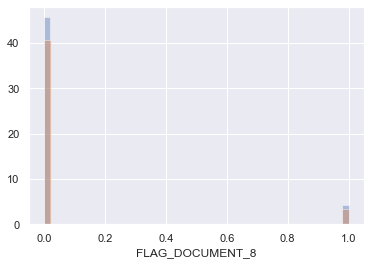
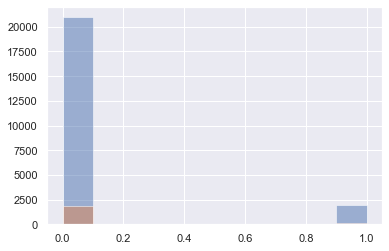
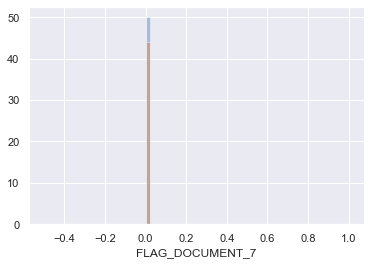
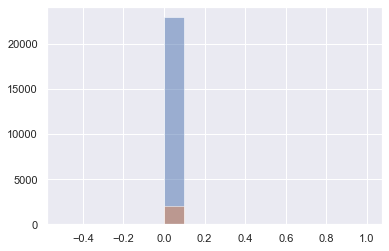
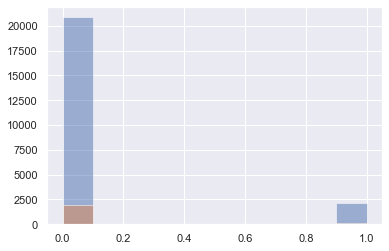
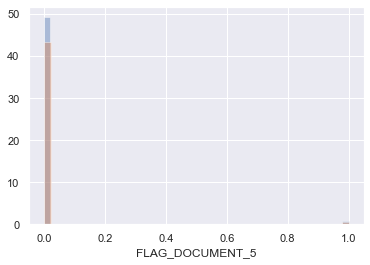
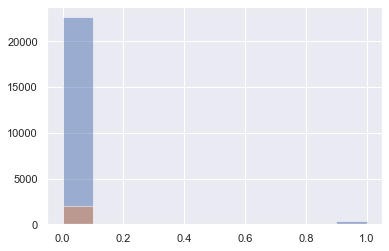
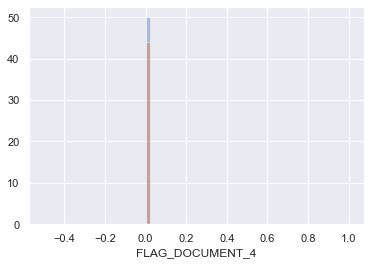
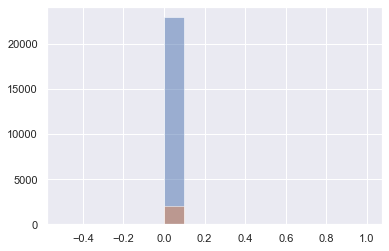
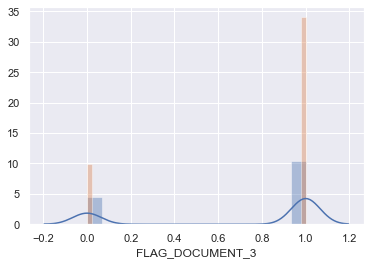
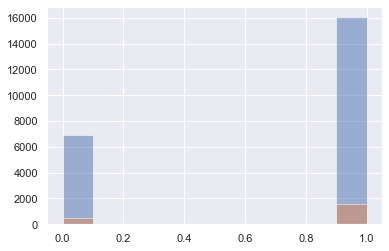
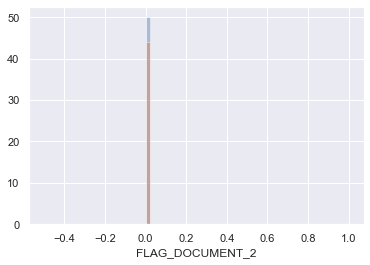
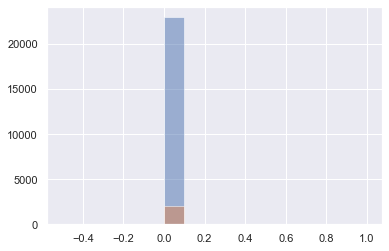
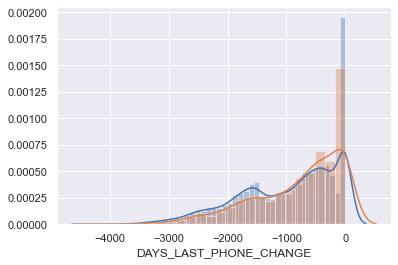
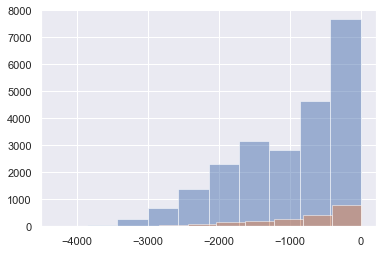
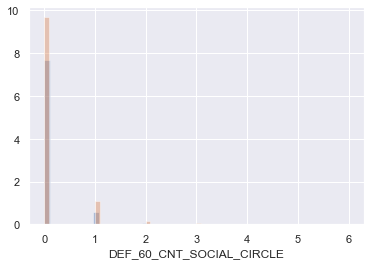
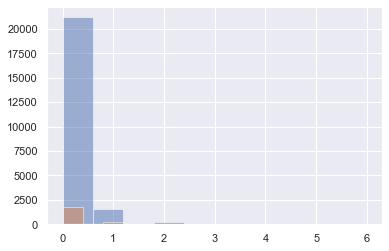
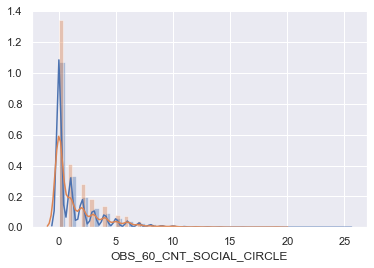
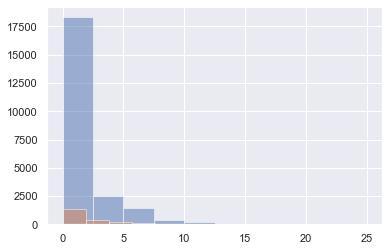
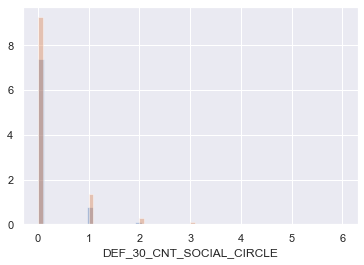
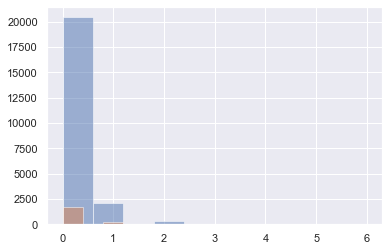
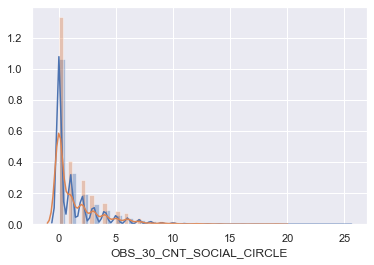
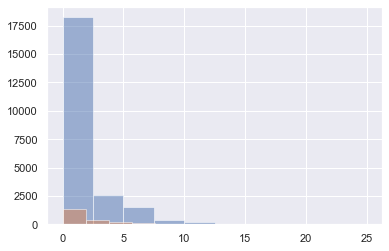
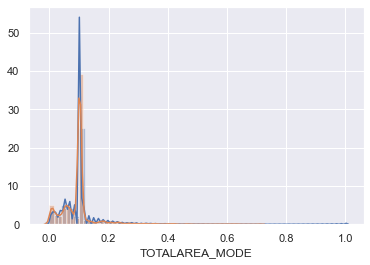
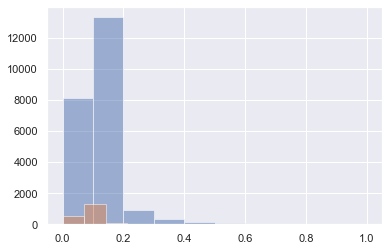
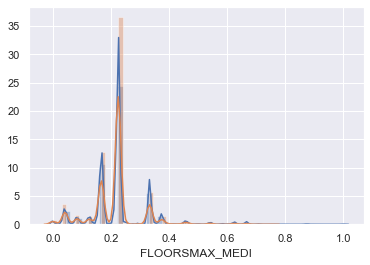
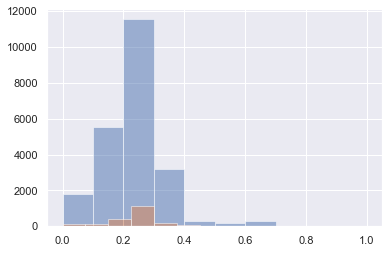
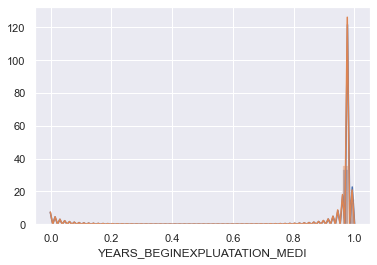
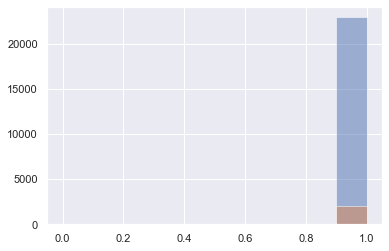
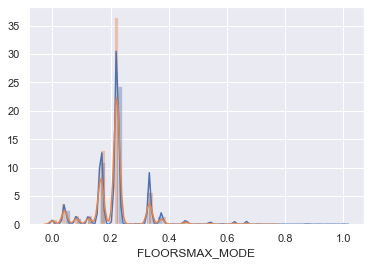
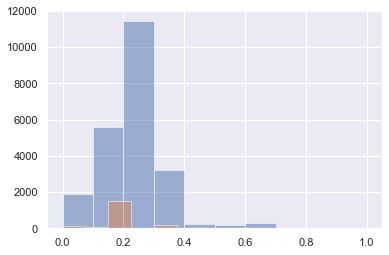
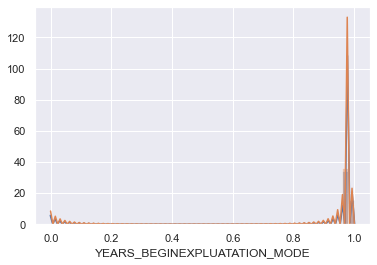
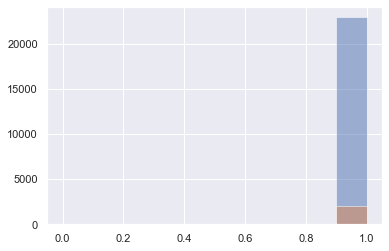
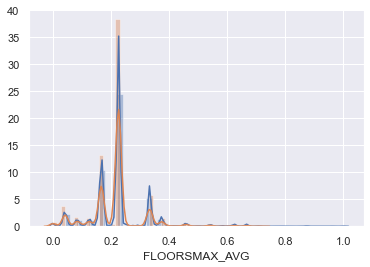
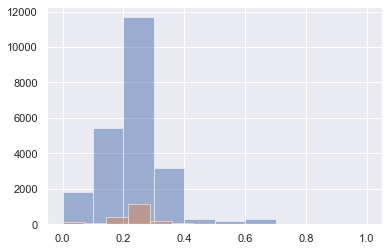
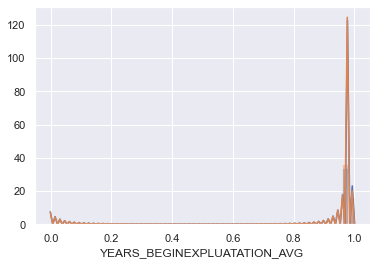
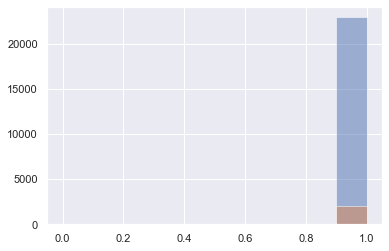
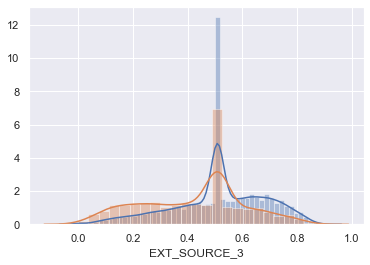
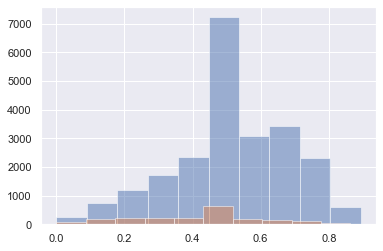
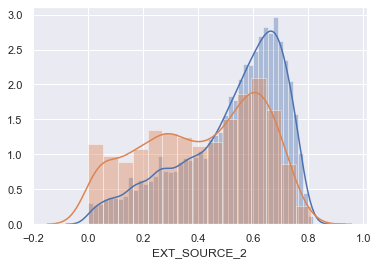
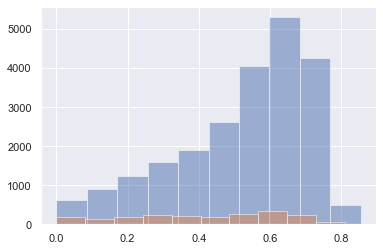
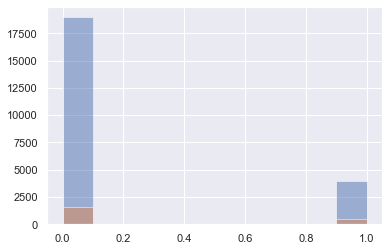
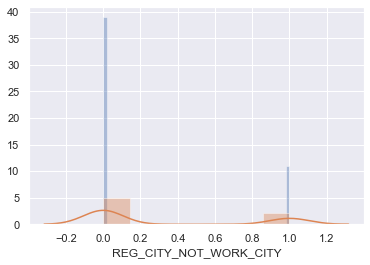
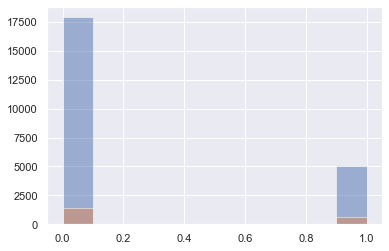
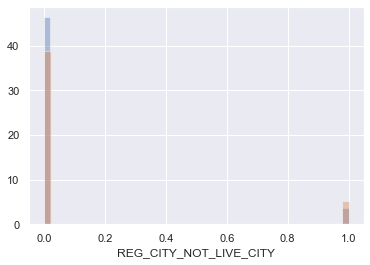
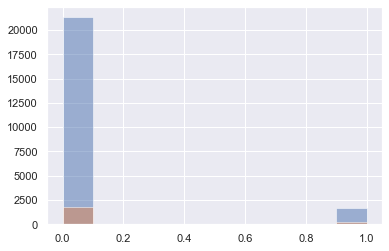
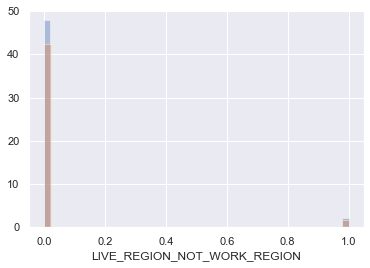
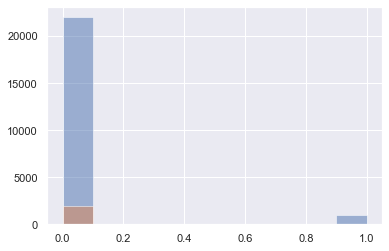
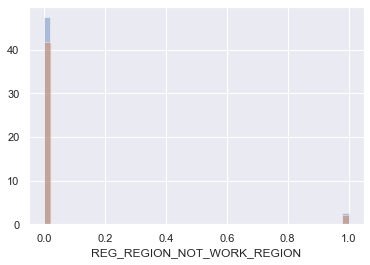
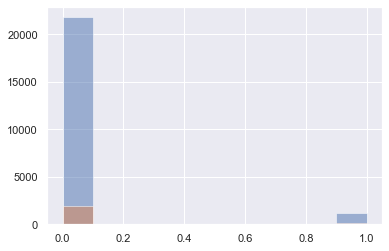
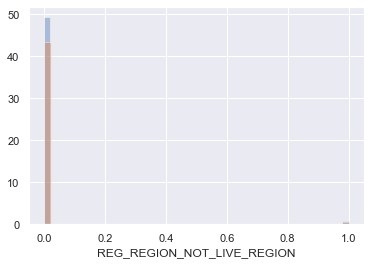
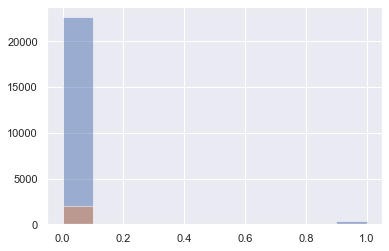
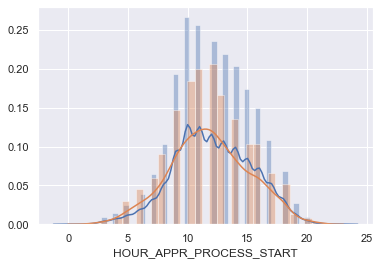
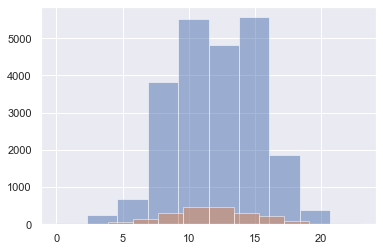
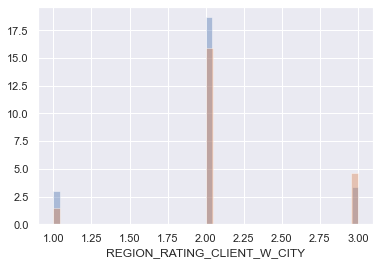
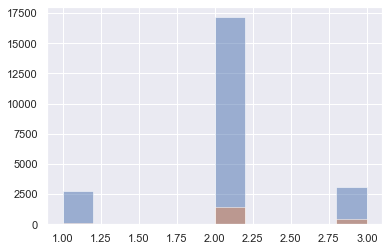
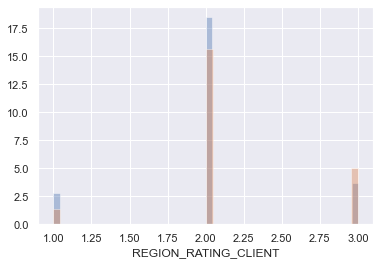
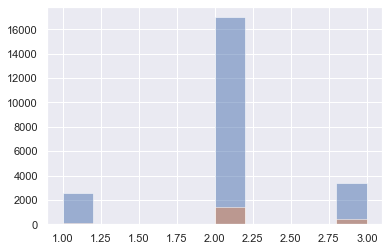
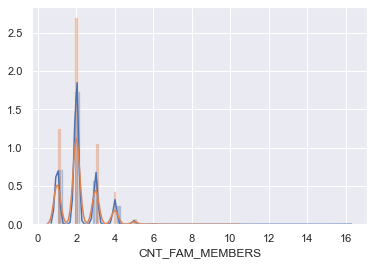
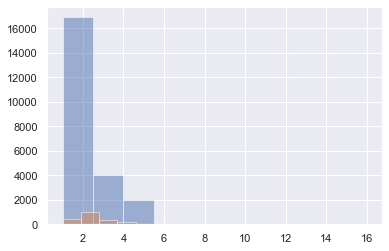
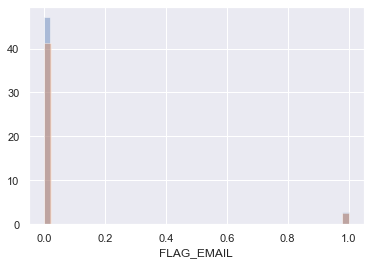
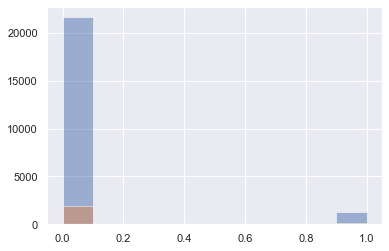
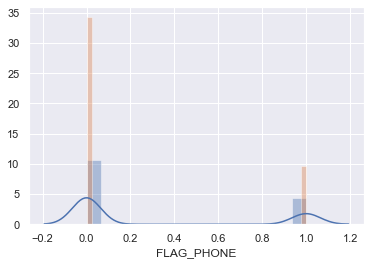
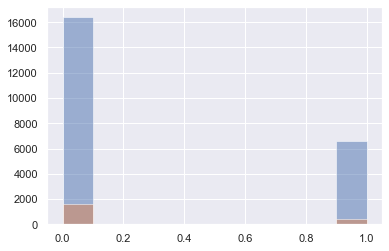
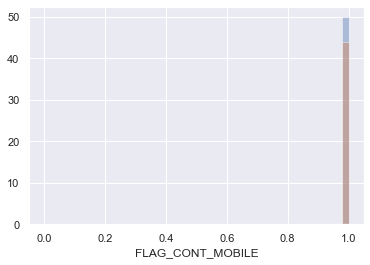
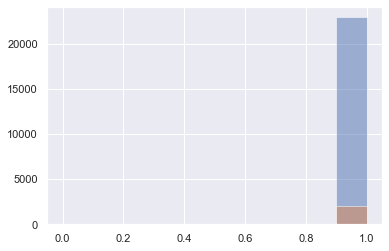
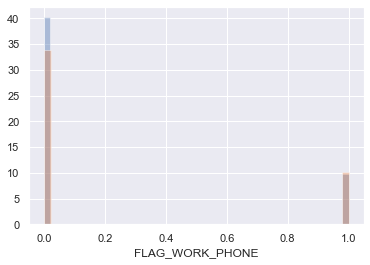
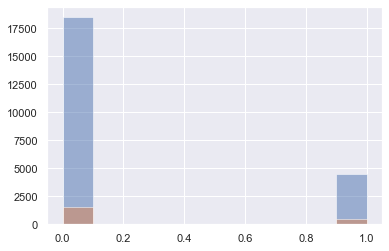
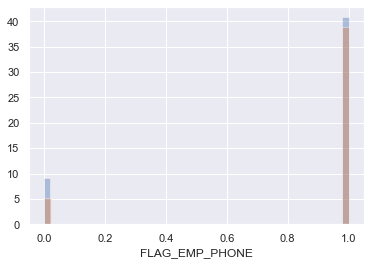
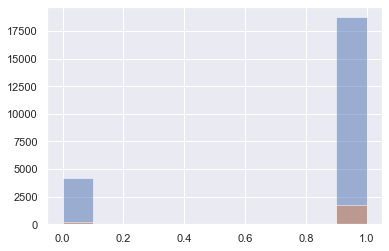
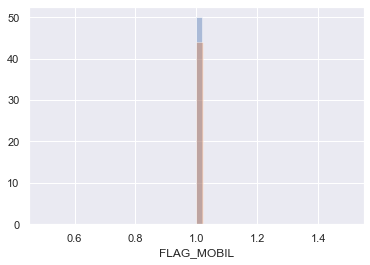
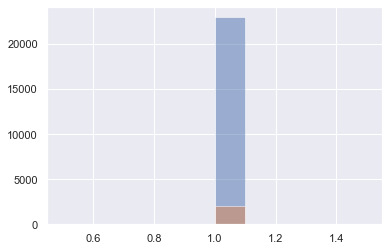
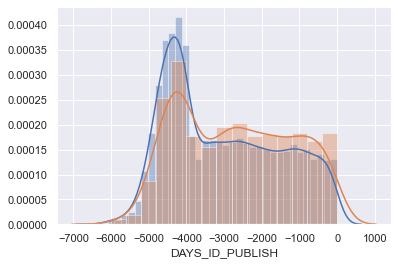
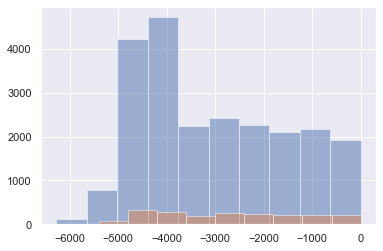
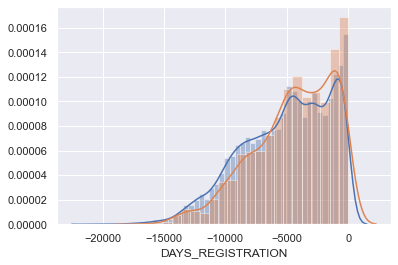
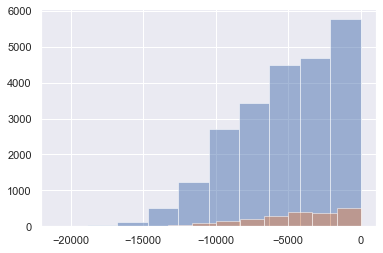
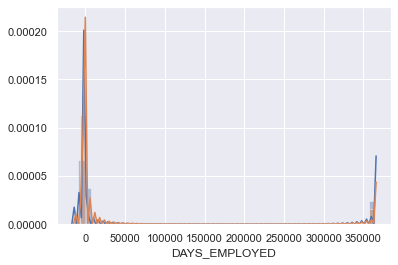
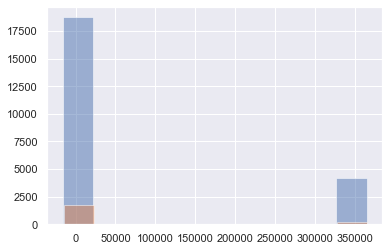
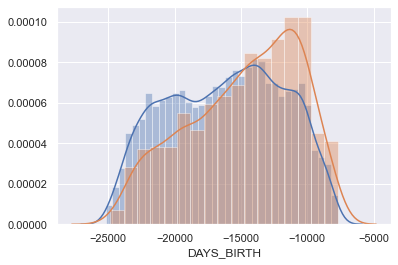
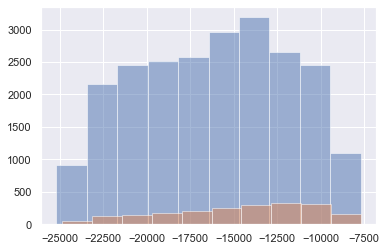
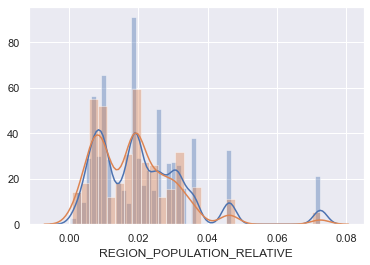
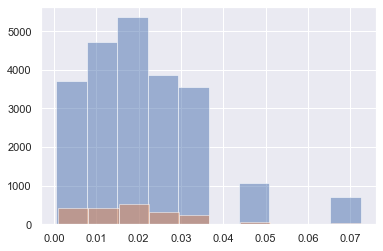
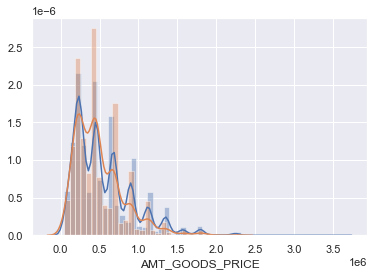
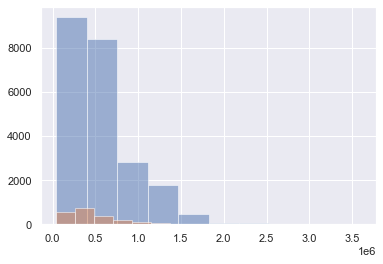
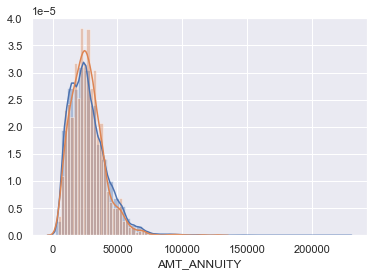
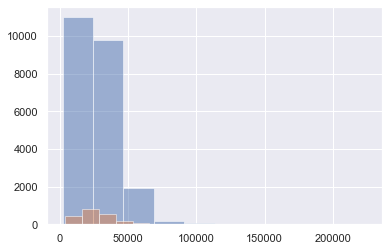
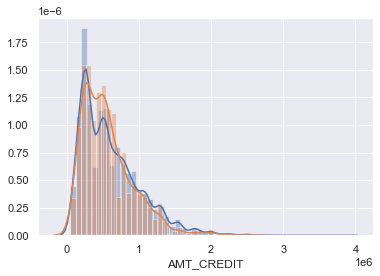
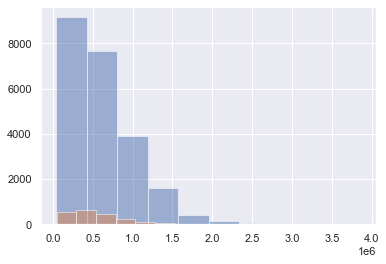
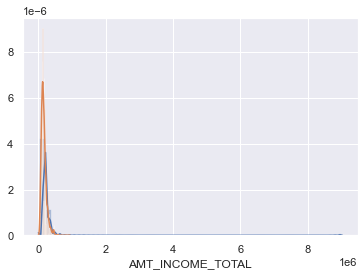
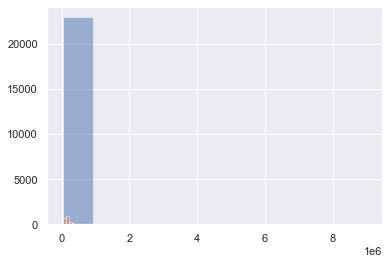
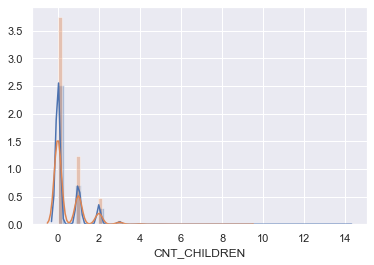
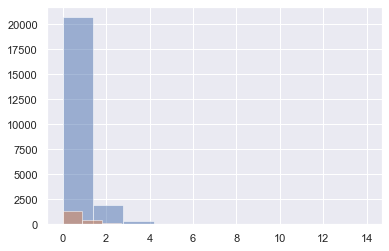
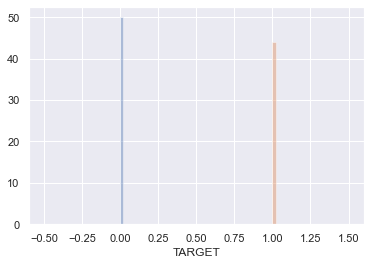
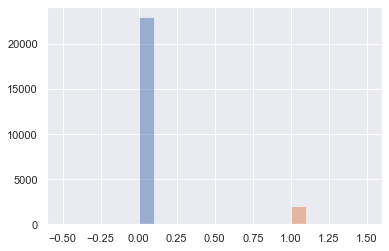
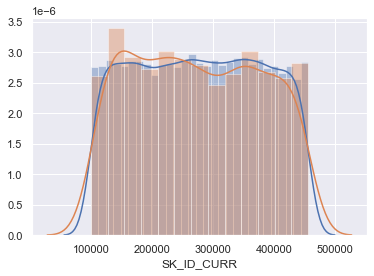
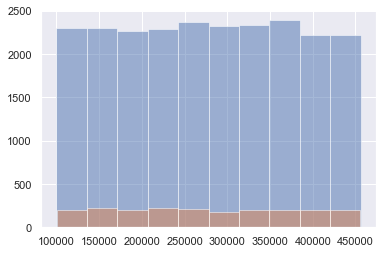
------------------------------------------------------------------------  
Plot of AMT\_REQ\_CREDIT\_BUREAU\_MON

------------------------------------------------------------------------  
Plot of AMT\_REQ\_CREDIT\_BUREAU\_QRT

C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)  
C:\Users\ingle\anaconda3\lib\site-packages\seaborn\distributions.py:369: UserWarning: Default bandwidth for data is 0; skipping density estimation.  
 warnings.warn(msg, UserWarning)

------------------------------------------------------------------------  
Plot of AMT\_REQ\_CREDIT\_BUREAU\_YEAR

------------------------------------------------------------------------



**Reading previous application**

[109]

previous\_application = pd.read\_csv('previous\_application.csv')  
previous\_application.head()

[110]

*# Shape of previous application*  
previous\_application.shape

(1670214, 37)

[111]

previous\_application = previous\_application.sample(25000)  
previous\_application.head()

[112]

previous\_application.shape

(25000, 37)

[113]

previous\_application1 = previous\_application  
previous\_application1.shape

(25000, 37)

**There are duplicate 'SK\_ID\_CURR' as a person could have taken loan multiple times**

[114]

*# Number of unique id in previous application*  
previous\_application.SK\_ID\_PREV.value\_counts()

1607679 1  
1029177 1  
1291346 1  
1516914 1  
2380399 1  
 ..  
1895751 1  
2683752 1  
2497699 1  
1258828 1  
1968129 1  
Name: SK\_ID\_PREV, Length: 25000, dtype: int64

[115]

*# Number of unique id in previous application*  
previous\_application.SK\_ID\_CURR.value\_counts()

152600 4  
327393 3  
244126 3  
387020 3  
211489 3  
 ..  
340086 1  
257079 1  
328762 1  
137390 1  
393216 1  
Name: SK\_ID\_CURR, Length: 23727, dtype: int64

**As you can see above, the shape of previous application is (1670214, 37) and length of SK\_ID\_PREV is also (1670214), but length of SK\_ID\_CURR is (338857), which is less than length of SK\_ID\_PREV, which tells us that there are duplicate number of SK\_ID\_PREV**

**Let's merge dataframe: train and previous application based on SK\_ID\_PREV**

After merging both the dataframes, the new dataframe will also have duplicate number of SK\_ID\_PREV. This should not be a problem, as we are trying to figure out if any pattern is present by including the cases if a lender has previously taken loan more than once.

[116]

previous\_train = application\_train.merge(previous\_application, left\_on='SK\_ID\_CURR',   
                                         right\_on='SK\_ID\_CURR', how='inner')

[117]

previous\_train.shape

(1748, 118)

[118]

previous\_train.head()

[119]

previous\_application.columns.values

array(['SK\_ID\_PREV', 'SK\_ID\_CURR', 'NAME\_CONTRACT\_TYPE', 'AMT\_ANNUITY',  
 'AMT\_APPLICATION', 'AMT\_CREDIT', 'AMT\_DOWN\_PAYMENT',  
 'AMT\_GOODS\_PRICE', 'WEEKDAY\_APPR\_PROCESS\_START',  
 'HOUR\_APPR\_PROCESS\_START', 'FLAG\_LAST\_APPL\_PER\_CONTRACT',  
 'NFLAG\_LAST\_APPL\_IN\_DAY', 'RATE\_DOWN\_PAYMENT',  
 'RATE\_INTEREST\_PRIMARY', 'RATE\_INTEREST\_PRIVILEGED',  
 'NAME\_CASH\_LOAN\_PURPOSE', 'NAME\_CONTRACT\_STATUS', 'DAYS\_DECISION',  
 'NAME\_PAYMENT\_TYPE', 'CODE\_REJECT\_REASON', 'NAME\_TYPE\_SUITE',  
 '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', 'DAYS\_FIRST\_DRAWING', 'DAYS\_FIRST\_DUE',  
 'DAYS\_LAST\_DUE\_1ST\_VERSION', 'DAYS\_LAST\_DUE', 'DAYS\_TERMINATION',  
 'NFLAG\_INSURED\_ON\_APPROVAL'], dtype=object)

**The merged dataframe also has multiple values for SK\_ID\_CURR**

[120]

previous\_application.SK\_ID\_CURR.value\_counts().head()

152600 4  
327393 3  
244126 3  
387020 3  
211489 3  
Name: SK\_ID\_CURR, dtype: int64

**Segregating the dataset on Target=0 and Target=1**

[121]

train\_0 = application\_train.loc[application\_train['TARGET'] == 0]  
train\_1 = application\_train.loc[application\_train['TARGET'] == 1]

[122]

ptrain\_0 = previous\_train.loc[previous\_train['TARGET'] == 0]  
ptrain\_1 = previous\_train.loc[previous\_train['TARGET'] == 1]

**Plotting data**

[123]

**def** plotting(column, hue):  
    col = column  
    hue = hue  
    fig = plt.figure(figsize=(13,10))  
  
    ax1 = plt.subplot(221)  
    application\_train[col].value\_counts().plot.pie(autopct = "%1.0f%%", ax=ax1)  
    plt.title('Plotting data for the column: '+ column)  
  
  
    ax2 = plt.subplot(222)  
    df = pd.DataFrame()  
    df['0']= ((train\_0[col].value\_counts())/len(train\_0))  
    df['1']= ((train\_1[col].value\_counts())/len(train\_1))  
    df.plot.bar(ax=ax2)  
    plt.title('Plotting data for target in terms of total count')  
  
  
    ax3 = plt.subplot(223)  
    sns.countplot(x=col, hue=hue, data=ptrain\_0, ax = ax3)  
    plt.xticks(rotation=90)  
    plt.title('Plotting data for Target=0 in terms of percentage')  
  
    ax4 = plt.subplot(224)  
    sns.countplot(x=col, hue=hue, data=ptrain\_1, ax = ax4)  
    plt.xticks(rotation=90)  
    plt.title('Plotting data for Target=1 in terms of percentage')  
  
  
  
    fig.tight\_layout() *# Or equivalently,  "plt.tight\_layout()"*  
  
    plt.show()

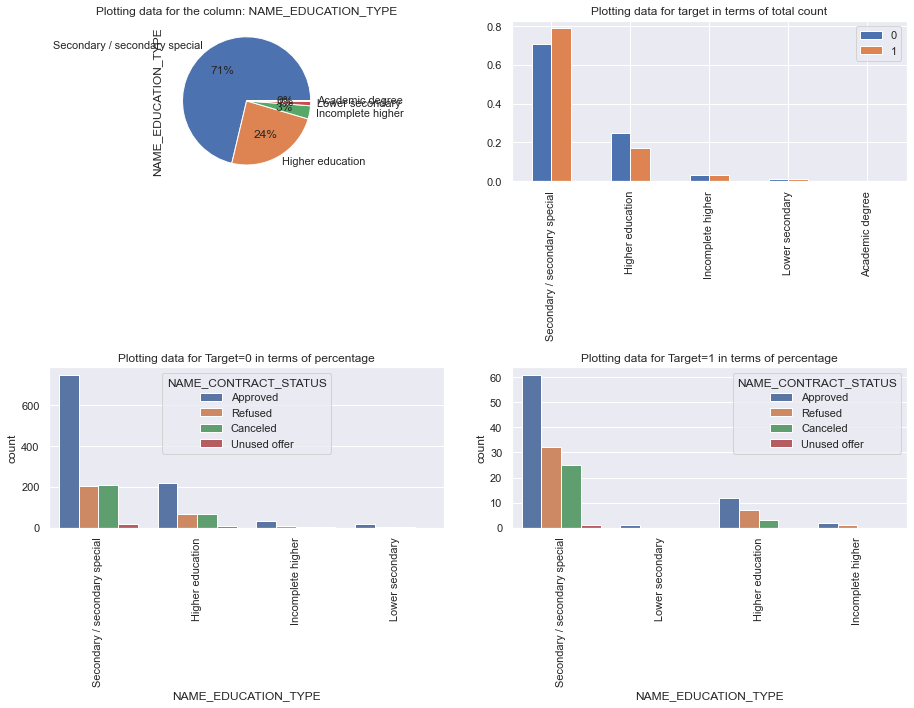
**Bivariate Analysis**

Here, plotting only **for** 3 columns, **as** plotting **in** loop **for** all columns was pretty heavy **for** this size of dataset.

**Plotting NAME\_EDUCATION\_TYPE**

[124]

plotting('NAME\_EDUCATION\_TYPE','NAME\_CONTRACT\_STATUS')



**Happy Learning**