Vikash Kumar Maheshwari

Project- Loan Case Study

Description :

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

Approach :

Firstly we will go through the data and check the missing data or null one. After that we will remove the outliers and using charts we will show our results.

Tech-Stack Used :

Python

With the help of Google colab we have analyze and EDA on our dataset

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

```
df = pd.read_csv('appdata.csv', on_bad_lines='skip')
```

Insights

1. Cleaning the data

```
df.shape
```

```
(307511, 122)
```

```
app=df.drop(['DAYS_REGISTRATION','FLAG_MOBIL','FLAG_EMP_PHONE','FLAG_WORK_PHONE',
'FLAG_CONT_MOBILE','FLAG_PHONE','FLAG_EMAIL','WEEKDAY_APPR_PROCESS_START','HOUR_APPR_PROCESS_START','LIVE_REGION_NOT_WORK_REGION',
'REG_CITY_NOT_LIVE_CITY','REG_CITY_NOT_WORK_CITY','LIVE_CITY_NOT_WORK_CITY','DAYS_LAST_PHONE_CHANGE',
'OBS_30_CNT_SOCIAL_CIRCLE','DEF_30_CNT_SOCIAL_CIRCLE','OBS_60_CNT_SOCIAL_CIRCLE','DEF_60_CNT_SOCIAL_CIRCLE', 'NAME_TYPE_SUITE',
'OWN_CAR_AGE', 'OCCUPATION_TYPE', 'EXT_SOURCE_1',
'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'], axis=1)
```

```
app.isnull().sum()
```

```
SK_ID_CURR                      0
TARGET                          0
NAME_CONTRACT_TYPE              0
CODE_GENDER                     0
FLAG_OWN_CAR                    0
FLAG_OWN_REALTY                 0
CNT_CHILDREN                    0
AMT_INCOME_TOTAL                0
AMT_CREDIT                      0
AMT_ANNUITY                    12
AMT_GOODS_PRICE               278
NAME_INCOME_TYPE                0
NAME_EDUCATION_TYPE             0
NAME_FAMILY_STATUS              0
NAME_HOUSING_TYPE               0
REGION_POPULATION_RELATIVE      0
DAYS_BIRTH                      0
DAYS_EMPLOYED                   0
DAYS_ID_PUBLISH                 0
CNT_FAM_MEMBERS                 2
REGION_RATING_CLIENT            0
REGION_RATING_CLIENT_W_CITY     0
REG_REGION_NOT_LIVE_REGION      0
REG_REGION_NOT_WORK_REGION      0
ORGANIZATION_TYPE               0
EXT_SOURCE_2                  660
EXT_SOURCE_3                60965
FLAG_DOCUMENT_2                 0
FLAG_DOCUMENT_3                 0
FLAG_DOCUMENT_4                 0
FLAG_DOCUMENT_5                 0
FLAG_DOCUMENT_6                 0
FLAG_DOCUMENT_7                 0
```

```
FLAG_DOCUMENT_8                 0
FLAG_DOCUMENT_9                 0
FLAG_DOCUMENT_10                0
FLAG_DOCUMENT_11                0
FLAG_DOCUMENT_12                0
FLAG_DOCUMENT_13                0
FLAG_DOCUMENT_14                0
FLAG_DOCUMENT_15                0
FLAG_DOCUMENT_16                0
FLAG_DOCUMENT_17                0
FLAG_DOCUMENT_18                0
FLAG_DOCUMENT_19                0
FLAG_DOCUMENT_20                0
FLAG_DOCUMENT_21                0
AMT_REQ_CREDIT_BUREAU_HOUR  41519
AMT_REQ_CREDIT_BUREAU_DAY   41519
AMT_REQ_CREDIT_BUREAU_WEEK  41519
AMT_REQ_CREDIT_BUREAU_MON   41519
AMT_REQ_CREDIT_BUREAU_QRT   41519
AMT_REQ_CREDIT_BUREAU_YEAR  41519
dtype: int64
```

```
missing_rows = app[app.isnull().any(axis=1)]
```

```
appl = app.dropna(axis=0)
appl.head()
```

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_C
0	100002	1	Cash loans	M	N	Y	
2	100004	0	Revolving loans	M	Y	Y	
5	100008	0	Cash loans	M	N	Y	
6	100009	0	Cash loans	F	Y	Y	
7	100010	0	Cash loans	M	Y	Y	

5 rows × 53 columns

```
appl.shape
```

```
(245895, 53)
```

```
appl.isnull().sum()
```

```
SK_ID_CURR                  0
TARGET                      0
NAME_CONTRACT_TYPE          0
CODE_GENDER                 0
FLAG_OWN_CAR                0
FLAG_OWN_REALTY             0
CNT_CHILDREN                0
AMT_INCOME_TOTAL            0
AMT_CREDIT                  0
AMT_ANNUITY                 0
AMT_GOODS_PRICE             0
NAME_INCOME_TYPE            0
NAME_EDUCATION_TYPE         0
NAME_FAMILY_STATUS          0
NAME_HOUSING_TYPE           0
REGION_POPULATION_RELATIVE  0
DAYS_BIRTH                  0
DAYS_EMPLOYED               0
DAYS_ID_PUBLISH             0
CNT_FAM_MEMBERS             0
REGION_RATING_CLIENT        0
REGION_RATING_CLIENT_W_CITY 0
REG_REGION_NOT_LIVE_REGION  0
REG_REGION_NOT_WORK_REGION  0
ORGANIZATION_TYPE           0
EXT_SOURCE_2                0
EXT_SOURCE_3                0
FLAG_DOCUMENT_2             0
FLAG_DOCUMENT_3             0
FLAG_DOCUMENT_4             0
FLAG_DOCUMENT_5             0
FLAG_DOCUMENT_6             0
FLAG_DOCUMENT_7             0
FLAG_DOCUMENT_8             0
FLAG_DOCUMENT_9             0
FLAG_DOCUMENT_10            0
FLAG_DOCUMENT_11            0
FLAG_DOCUMENT_12            0
FLAG_DOCUMENT_13            0
FLAG_DOCUMENT_14            0
FLAG_DOCUMENT_15            0
FLAG_DOCUMENT_16            0
FLAG_DOCUMENT_17            0
FLAG_DOCUMENT_18            0
FLAG_DOCUMENT_19            0
FLAG_DOCUMENT_20            0
FLAG_DOCUMENT_21            0
AMT_REQ_CREDIT_BUREAU_HOUR  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
dtype: int64
```

```
appl.info()
```

```
Data columns (total 53 columns):
 #   Column                       Non-Null Count   Dtype
---  ------                       --------------   -----
 0   SK_ID_CURR                   245895 non-null  int64
 1   TARGET                       245895 non-null  int64
 2   NAME_CONTRACT_TYPE           245895 non-null  object
 3   CODE_GENDER                  245895 non-null  object
 4   FLAG_OWN_CAR                 245895 non-null  object
 5   FLAG_OWN_REALTY              245895 non-null  object
 6   CNT_CHILDREN                 245895 non-null  int64
 7   AMT_INCOME_TOTAL             245895 non-null  float64
 8   AMT_CREDIT                   245895 non-null  float64
 9   AMT_ANNUITY                  245895 non-null  float64
```

```
 12  NAME_EDUCATION_TYPE           245895 non-null  object
 13  NAME_FAMILY_STATUS            245895 non-null  object
 14  NAME_HOUSING_TYPE             245895 non-null  object
 15  REGION_POPULATION_RELATIVE    245895 non-null  float64
 16  DAYS_BIRTH                    245895 non-null  int64
 17  DAYS_EMPLOYED                 245895 non-null  int64
 18  DAYS_ID_PUBLISH               245895 non-null  int64
 19  CNT_FAM_MEMBERS               245895 non-null  float64
 20  REGION_RATING_CLIENT          245895 non-null  int64
 21  REGION_RATING_CLIENT_W_CITY   245895 non-null  int64
 22  REG_REGION_NOT_LIVE_REGION    245895 non-null  int64
 23  REG_REGION_NOT_WORK_REGION    245895 non-null  int64
 24  ORGANIZATION_TYPE             245895 non-null  object
 25  EXT_SOURCE_2                  245895 non-null  float64
 26  EXT_SOURCE_3                  245895 non-null  float64
 27  FLAG_DOCUMENT_2               245895 non-null  int64
 28  FLAG_DOCUMENT_3               245895 non-null  int64
 29  FLAG_DOCUMENT_4               245895 non-null  int64
 30  FLAG_DOCUMENT_5               245895 non-null  int64
 31  FLAG_DOCUMENT_6               245895 non-null  int64
 32  FLAG_DOCUMENT_7               245895 non-null  int64
 33  FLAG_DOCUMENT_8               245895 non-null  int64
 34  FLAG_DOCUMENT_9               245895 non-null  int64
 35  FLAG_DOCUMENT_10              245895 non-null  int64
 36  FLAG_DOCUMENT_11              245895 non-null  int64
 37  FLAG_DOCUMENT_12              245895 non-null  int64
 38  FLAG_DOCUMENT_13              245895 non-null  int64
 39  FLAG_DOCUMENT_14              245895 non-null  int64
 40  FLAG_DOCUMENT_15              245895 non-null  int64
 41  FLAG_DOCUMENT_16              245895 non-null  int64
 42  FLAG_DOCUMENT_17              245895 non-null  int64
 43  FLAG_DOCUMENT_18              245895 non-null  int64
 44  FLAG_DOCUMENT_19              245895 non-null  int64
 45  FLAG_DOCUMENT_20              245895 non-null  int64
 46  FLAG_DOCUMENT_21              245895 non-null  int64
 47  AMT_REQ_CREDIT_BUREAU_HOUR    245895 non-null  float64
 48  AMT_REQ_CREDIT_BUREAU_DAY     245895 non-null  float64
 49  AMT_REQ_CREDIT_BUREAU_WEEK    245895 non-null  float64
 50  AMT_REQ_CREDIT_BUREAU_MON     245895 non-null  float64
 51  AMT_REQ_CREDIT_BUREAU_QRT     245895 non-null  float64
 52  AMT_REQ_CREDIT_BUREAU_YEAR    245895 non-null  float64
dtypes: float64(14), int64(30), object(9)
memory usage: 101.3+ MB
```

```
appl.describe()
```

	SK_ID_CURR	TARGET	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNU
count	245895.000000	245895.000000	245895.000000	2.458950e+05	2.458950e+05	245895.000
mean	278186.442091	0.077667	0.424262	1.720402e+05	6.084651e+05	27186.262
std	102823.258275	0.267648	0.726667	2.568701e+05	4.048787e+05	14333.483
min	100002.000000	0.000000	0.000000	2.610000e+04	4.500000e+04	1615.500
25%	189031.500000	0.000000	0.000000	1.125000e+05	2.745000e+05	16677.000
50%	278111.000000	0.000000	0.000000	1.575000e+05	5.212800e+05	25015.500
75%	367241.000000	0.000000	1.000000	2.025000e+05	8.140410e+05	34767.000
max	456255.000000	1.000000	19.000000	1.170000e+08	4.050000e+06	258025.500

8 rows × 44 columns

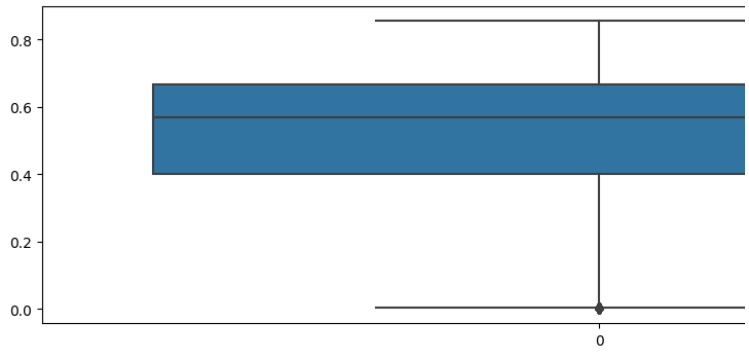
Insights

2. Identify if there are outliers in the dataset

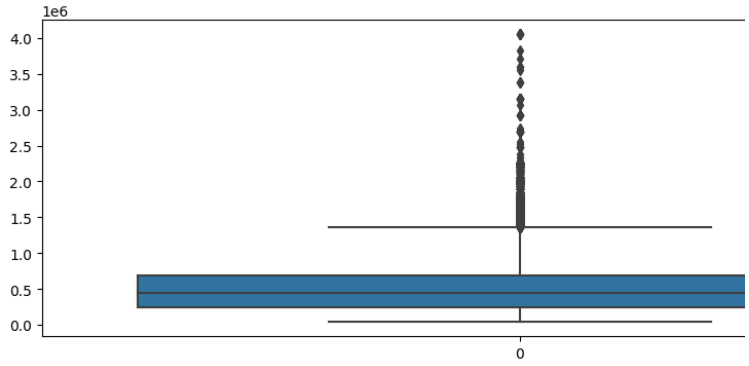
```
Q1 = appl.quantile(0.25)
Q3 = appl.quantile(0.75)
IQR = Q3 - Q1
print(IQR)
```

```
SK_ID_CURR                    178209.500000
TARGET                             0.000000
CNT_CHILDREN                       1.000000
AMT_INCOME_TOTAL               90000.000000
AMT_CREDIT                    539541.000000
AMT_ANNUITY                    18090.000000
AMT_GOODS_PRICE               450000.000000
REGION_POPULATION_RELATIVE         0.018657
DAYS_BIRTH                      7125.000000
DAYS_EMPLOYED                   2577.000000
DAYS_ID_PUBLISH                 2503.000000
CNT_FAM_MEMBERS                    1.000000
REGION_RATING_CLIENT               0.000000
REGION_RATING_CLIENT_W_CITY        0.000000
REG_REGION_NOT_LIVE_REGION         0.000000
REG_REGION_NOT_WORK_REGION         0.000000
EXT_SOURCE_2                       0.264126
EXT_SOURCE_3                       0.298407
FLAG_DOCUMENT_2                    0.000000
FLAG_DOCUMENT_3                    1.000000
FLAG_DOCUMENT_4                    0.000000
FLAG_DOCUMENT_5                    0.000000
FLAG_DOCUMENT_6                    0.000000
FLAG_DOCUMENT_7                    0.000000
FLAG_DOCUMENT_8                    0.000000
FLAG_DOCUMENT_9                    0.000000
FLAG_DOCUMENT_10                   0.000000
FLAG_DOCUMENT_11                   0.000000
FLAG_DOCUMENT_12                   0.000000
FLAG_DOCUMENT_13                   0.000000
FLAG_DOCUMENT_14                   0.000000
FLAG_DOCUMENT_15                   0.000000
FLAG_DOCUMENT_16                   0.000000
FLAG_DOCUMENT_17                   0.000000
FLAG_DOCUMENT_18                   0.000000
FLAG_DOCUMENT_19                   0.000000
FLAG_DOCUMENT_20                   0.000000
FLAG_DOCUMENT_21                   0.000000
AMT_REQ_CREDIT_BUREAU_HOUR         0.000000
AMT_REQ_CREDIT_BUREAU_DAY          0.000000
AMT_REQ_CREDIT_BUREAU_WEEK         0.000000
AMT_REQ_CREDIT_BUREAU_MON          0.000000
AMT_REQ_CREDIT_BUREAU_QRT          0.000000
AMT_REQ_CREDIT_BUREAU_YEAR         3.000000
dtype: float64
```

```
# Box plot for continuious variable
plt.figure(figsize=(14,4))
sns.boxplot(appl['EXT_SOURCE_2'])
plt.show()
```




```
plt.figure(figsize=(12,4))
sns.boxplot(appl['AMT_GOODS_PRICE'])
plt.show()
```




```
# For xna Code Gender column

print('CODE_GENDER: ',appl['CODE_GENDER'].unique())
print('No of values: ',appl[appl['CODE_GENDER']=='XNA'].shape[0])

XNA_count = appl[appl['CODE_GENDER']=='XNA'].shape[0]
per_XNA = round(XNA_count/len(appl.index)*100,3)

print('% of XNA Values:',  per_XNA)

print('maximum frequency data :', appl['CODE_GENDER'].describe().top)
```

```
CODE_GENDER:  ['M' 'F' 'XNA']
No of values:  4
% of XNA Values: 0.002
maximum frequency data : F
```

```
# Dropping the XNA value in column 'CODE_GENDER' with "F" for the dataset

appl = appl.drop(appl.loc[appl['CODE_GENDER']=='XNA'].index)
appl[appl['CODE_GENDER']=='XNA'].shape
```

```
(0, 53)
```

```
# For Organization column
print('No of XNA values: ', appl[appl['ORGANIZATION_TYPE']=='XNA'].shape[0])

XNA_count = appl[appl['ORGANIZATION_TYPE']=='XNA'].shape[0]
per_XNA = round(XNA_count/len(appl.index)*100,3)

print('% of XNA Values:',  per_XNA)

appl['ORGANIZATION_TYPE'].describe()
```

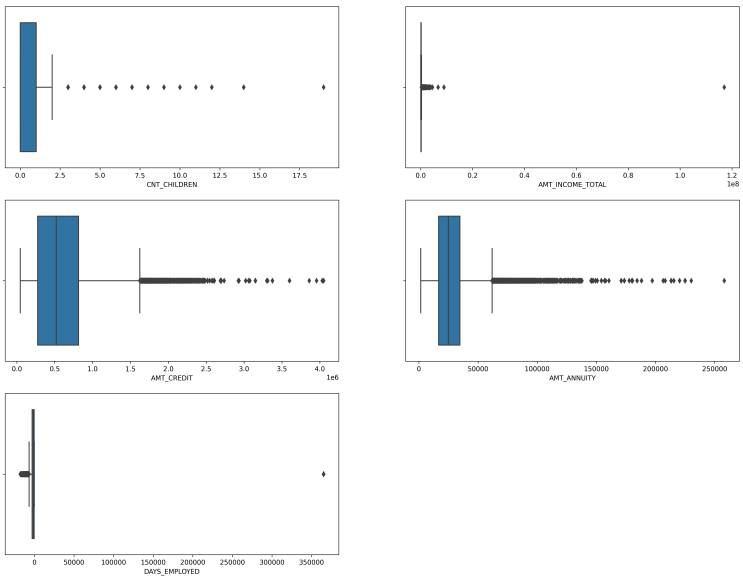
```
No of XNA values:  44176
% of XNA Values: 17.966
count                    245891
unique                       58
top       Business Entity Type 3
freq                      53625
Name: ORGANIZATION_TYPE, dtype: object
```

```
# Casting variable into numeric in the dataset

numeric_columns=['TARGET','CNT_CHILDREN','AMT_INCOME_TOTAL','AMT_CREDIT','AMT_ANNUITY','REGION_POPULATION_RELATIVE',
                 'DAYS_BIRTH','DAYS_EMPLOYED','DAYS_ID_PUBLISH']

appl[numeric_columns]=appl[numeric_columns].apply(pd.to_numeric)
appl.head(5)
```

```
appl[numeric_columns].describe()
```

	TARGET	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	REGION_POI
count	245891.000000	245891.000000	2.458910e+05	2.458910e+05	245891.000000	
mean	0.077669	0.424257	1.720400e+05	6.084685e+05	27186.379601	
std	0.267650	0.726665	2.568721e+05	4.048809e+05	14333.557448	
min	0.000000	0.000000	2.610000e+04	4.500000e+04	1615.500000	
25%	0.000000	0.000000	1.125000e+05	2.745000e+05	16677.000000	
50%	0.000000	0.000000	1.575000e+05	5.212800e+05	25015.500000	
75%	0.000000	1.000000	2.025000e+05	8.140410e+05	34767.000000	
max	1.000000	19.000000	1.170000e+08	4.050000e+06	258025.500000	

```
# Box plot for selected columns
features = ['CNT_CHILDREN', 'AMT_INCOME_TOTAL','AMT_CREDIT','AMT_ANNUITY','DAYS_EMPLOYED',]

plt.figure(figsize = (20, 15), dpi=300)
for i in enumerate(features):
    plt.subplot(3, 2, i[0]+1)
    sns.boxplot(x = i[1], data = appl)
plt.show()
```




```
# Dividing the dataset into two dataset of  target=1(client with payment difficulties) and target=0(all other)

target0_df=appl.loc[appl["TARGET"]==0]
target1_df=appl.loc[appl["TARGET"]==1]
```

```
# insights from number of target values

percentage_defaulters= round(100*len(target1_df)/(len(target0_df)+len(target1_df)),2)

percentage_nondefaulters=round(100*len(target0_df)/(len(target0_df)+len(target1_df)),2)

print('Count of target0_df:', len(target0_df))
print('Count of target1_df:', len(target1_df))


print('Percentage of people who paid their loan are: ', percentage_nondefaulters, '%' )
print('Percentage of people who did not paid their loan are: ', percentage_defaulters, '%' )
```

```
Count of target0_df: 226793
Count of target1_df: 19098
```

```
Percentage of people who paid their loan are:  92.23 %
Percentage of people who did not paid their loan are:  7.77 %
```

Insights

3. Find the ratio of data imbalance.

```
# Calculating Imbalance percentage

# Since the majority is target0 and minority is target1

imb_ratio = round(len(target0_df)/len(target1_df),2)

print('Imbalance Ratio:', imb_ratio)
```

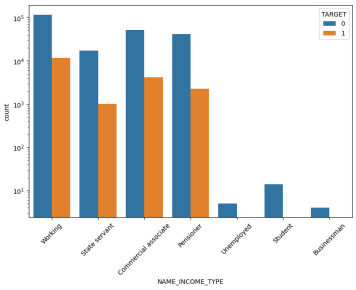
```
Imbalance Ratio: 11.88
```

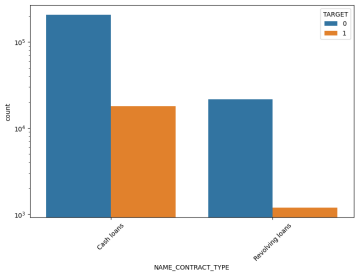
Insights

4. Results of univariate, bivariate analysis

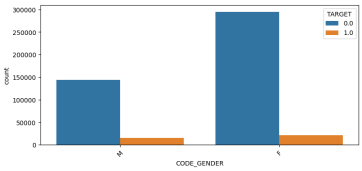
```
# Count plotting in logarithmic scale

def uniplot(df,col,title,hue =None):

    sns.set_style('whitegrid')
    sns.set_context('talk')
    plt.rcParams["axes.labelsize"] = 14
    plt.rcParams['axes.titlesize'] = 16
    plt.rcParams['axes.titlepad'] = 14


    temp = pd.Series(data = hue)
    fig, ax = plt.subplots()
    width = len(df[col].unique()) + 7 + 4*len(temp.unique())
    fig.set_size_inches(width , 8)
    plt.xticks(rotation=45)
    plt.yscale('log')
    plt.title(title)
    ax = sns.countplot(data = df, x= col, order=df[col].value_counts().index,hue = hue)

    plt.show()
```

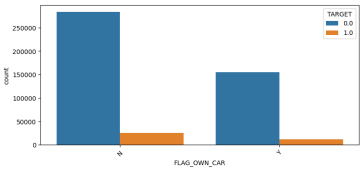
```
# Categoroical Univariate Analysis in logarithmic scale

features = ['NAME_INCOME_TYPE','NAME_CONTRACT_TYPE','NAME_FAMILY_STATUS']
plt.figure(figsize = (20, 15))

for i in enumerate(features):
    plt.subplot(2, 2, i[0]+1)
    plt.subplots_adjust(hspace=0.5)
    sns.countplot(x = i[1], hue = 'TARGET', data = appl)

    plt.rcParams['axes.titlesize'] = 16

    plt.xticks(rotation = 45)
    plt.yscale('log')
```



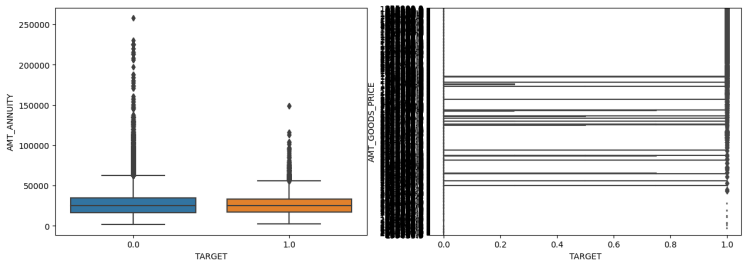


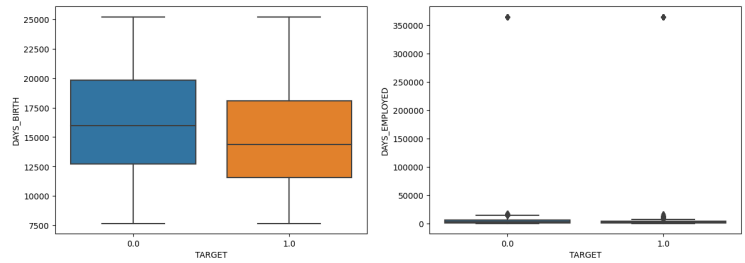

```
# Categoroical Univariate Analysis in Value scale

features·=·['CODE_GENDER','FLAG_OWN_CAR']
plt.figure(figsize·=·(20,·10))

for i in enumerate(features):
    plt.subplot(2, 2, i[0]+1)
    plt.subplots_adjust(hspace=0.5)
    sns.countplot(x = i[1], hue = 'TARGET', data = appl)

    plt.rcParams['axes.titlesize'] = 16
    plt.xticks(rotation = 45)
#     plt.yscale('log')
```





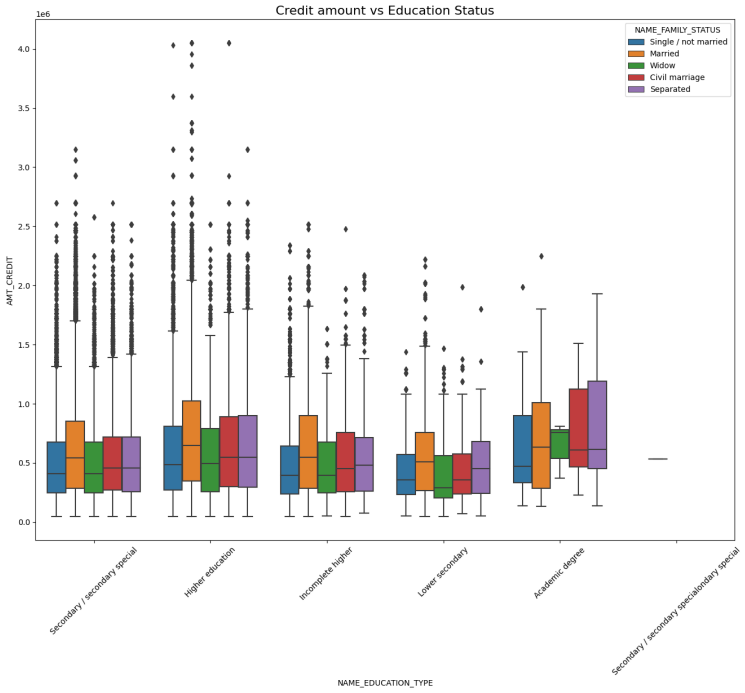

```
# Univariate Analysis for continous variable

features = ['AMT_ANNUITY','AMT_GOODS_PRICE','DAYS_BIRTH','DAYS_EMPLOYED','DAYS_ID_PUBLISH']
plt.figure(figsize = (15, 20))

for i in enumerate(features):
    plt.subplot(3, 2, i[0]+1)
    plt.subplots_adjust(hspace=0.5)
    sns.boxplot(x = 'TARGET', y = i[1], data = appl)
```





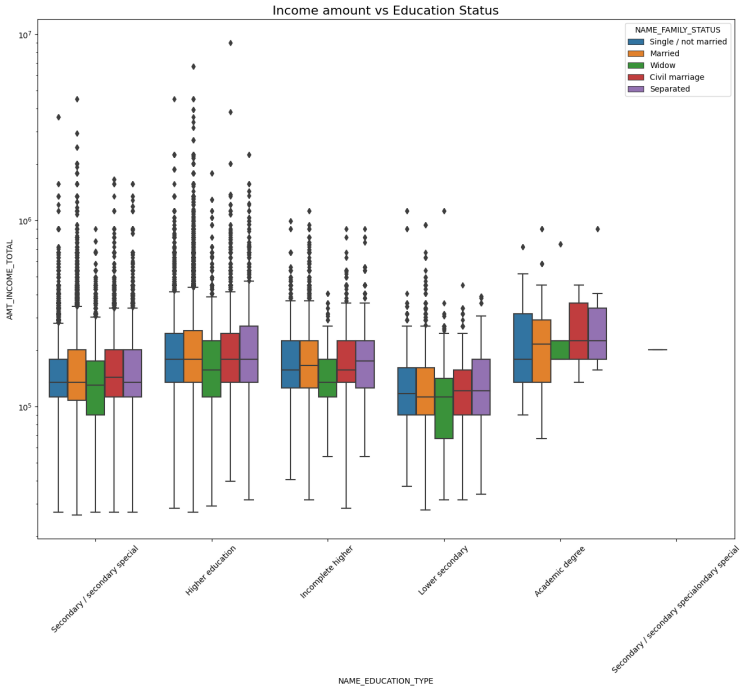
3.b. Bivariate analysis for numerical variables For Target 0

```
# Box plotting for Credit amount

plt.figure(figsize=(16,12))
plt.xticks(rotation=45)
sns.boxplot(data =target0_df, x='NAME_EDUCATION_TYPE',y='AMT_CREDIT', hue ='NAME_FAMILY_STATUS',orient='v')
plt.title('Credit amount vs Education Status')
plt.show()
```



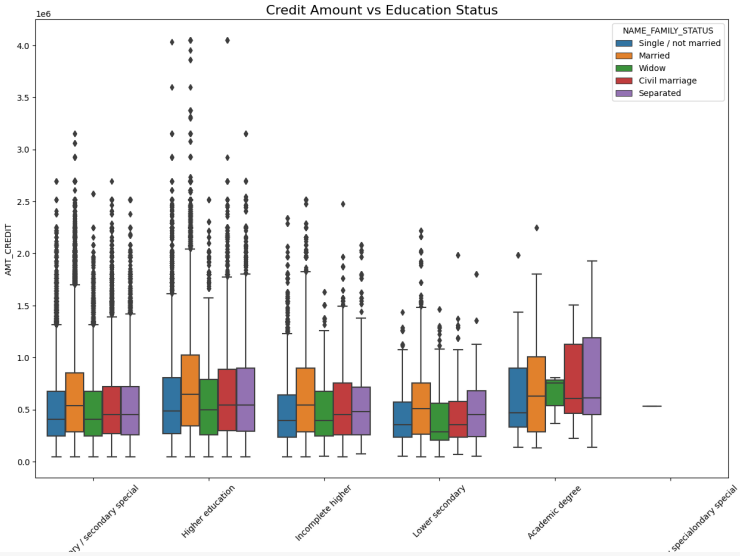

```
# Box plotting for Income amount in logarithmic scale
```

```
plt.figure(figsize=(16,12))
plt.xticks(rotation=45)
plt.yscale('log')
sns.boxplot(data =target0_df, x='NAME_EDUCATION_TYPE',y='AMT_INCOME_TOTAL', hue ='NAME_FAMILY_STATUS',orient='v')
plt.title('Income amount vs Education Status')
plt.show()
```

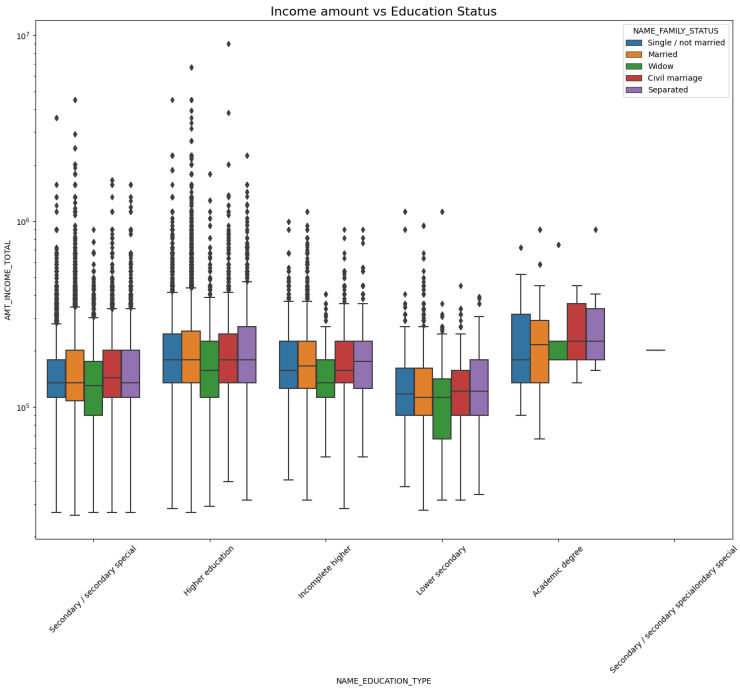


For Target 1

```
# Box plotting for credit amount

plt.figure(figsize=(15,10))
plt.xticks(rotation=45)
sns.boxplot(data =target0_df, x='NAME_EDUCATION_TYPE',y='AMT_CREDIT', hue ='NAME_FAMILY_STATUS',orient='v')
plt.title('Credit Amount vs Education Status')
plt.show()
```




```
# Box plotting for Income amount in logarithmic scale

plt.figure(figsize=(16,12))
plt.xticks(rotation=45)
plt.yscale('log')
sns.boxplot(data =target0_df, x='NAME_EDUCATION_TYPE',y='AMT_INCOME_TOTAL', hue ='NAME_FAMILY_STATUS',orient='v')
plt.title('Income amount vs Education Status')
plt.show()
```



Insights

5. Find the top 10 correlation

```
# Top 10 correlated variables: target 0 dataaframe
```

```
corr = target0_df.corr()
corrdf = corr.where(np.triu(np.ones(corr.shape), k=1).astype(np.bool))
corrdf = corrdf.unstack().reset_index()
corrdf.columns = ['Var1', 'Var2', 'Correlation']
corrdf.dropna(subset = ['Correlation'], inplace = True)
corrdf['Correlation'] = round(corrdf['Correlation'], 2)
corrdf['Correlation'] = abs(corrdf['Correlation'])
corrdf.sort_values(by = 'Correlation', ascending = False).head(10)
```

```
<ipython-input-133-d7b1279b2ebd>:4: DeprecationWarning: `np.bool` is a deprecated alias for
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/
  corrdf = corr.where(np.triu(np.ones(corr.shape), k=1).astype(np.bool))
```

	Var1	Var2	Correlation
184	AMT_ANNUITY	AMT_CREDIT	0.77
295	DAYS_EMPLOYED	DAYS_BIRTH	0.63
512	FLAG_DOCUMENT_6	DAYS_EMPLOYED	0.59
515	FLAG_DOCUMENT_6	FLAG_DOCUMENT_3	0.49
587	FLAG_DOCUMENT_8	FLAG_DOCUMENT_3	0.47
183	AMT_ANNUITY	AMT_INCOME_TOTAL	0.44
511	FLAG_DOCUMENT_6	DAYS_BIRTH	0.41
147	AMT_CREDIT	AMT_INCOME_TOTAL	0.36
254	DAYS_BIRTH	CNT_CHILDREN	0.35
332	DAYS_ID_PUBLISH	DAYS_EMPLOYED	0.27

```
# Top 10 correlated variables: target 1 dataaframe

corr = target1_df.corr()
corrdf = corr.where(np.triu(np.ones(corr.shape), k=1).astype(np.bool))
corrdf = corrdf.unstack().reset_index()
corrdf.columns = ['Var1', 'Var2', 'Correlation']
corrdf.dropna(subset = ['Correlation'], inplace = True)
corrdf['Correlation'] = round(corrdf['Correlation'], 2)
corrdf['Correlation'] = abs(corrdf['Correlation'])
corrdf.sort_values(by = 'Correlation', ascending = False).head(10)
```

```
<ipython-input-134-83c9c380cf9a>:4: DeprecationWarning: `np.bool` is a deprecated alias for
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/
  corrdf = corr.where(np.triu(np.ones(corr.shape), k=1).astype(np.bool))
```

	Var1	Var2	Correlation
184	AMT_ANNUITY	AMT_CREDIT	0.75
512	FLAG_DOCUMENT_6	DAYS_EMPLOYED	0.62
295	DAYS_EMPLOYED	DAYS_BIRTH	0.58
587	FLAG_DOCUMENT_8	FLAG_DOCUMENT_3	0.54
515	FLAG_DOCUMENT_6	FLAG_DOCUMENT_3	0.49
511	FLAG_DOCUMENT_6	DAYS_BIRTH	0.40
254	DAYS_BIRTH	CNT_CHILDREN	0.27
404	FLAG_DOCUMENT_3	DAYS_EMPLOYED	0.27
331	DAYS_ID_PUBLISH	DAYS_BIRTH	0.23
479	FLAG_DOCUMENT_5	FLAG_DOCUMENT_3	0.23

Thank You