LOAN ANALYSIS EDA EXERCISE

VARSHINI M S

USHA RANI T

THE PROBLEM

What is the problem?

This case study aims to identify patterns which indicate if a client has difficulty paying their instalments which may be used for taking actions such as denying the loan, reducing the amount of loan, lending (to risky applicants) at a higher interest rate, etc. This will ensure that the consumers capable of repaying the loan are not rejected. Identification of such applicants using EDA is the aim of this case study.

Dataset Used

1. 'application_data.csv' contains all the information of the client at the time of application.
2. 'previous_application.csv' contains information about the client's previous loan data..
3. 'columns_description.csv' is data dictionary which describes the meaning of the variables.

BACKGROUND INFORMATION

RESULT EXPECTED:

1. IDENTIFY THE MISSING DATA AND USE APPROPRIATE METHOD TO DEAL WITH IT. (REMOVE COLUMNS/OR REPLACE IT WITH AN APPROPRIATE VALUE)
2. IDENTIFY IF THERE ARE OUTLIERS IN THE DATASET. ALSO, MENTION WHY DO YOU THINK IT IS AN OUTLIER. AGAIN, REMEMBER THAT FOR THIS EXERCISE, IT IS NOT NECESSARY TO REMOVE ANY DATA POINTS.
3. IDENTIFY IF THERE IS DATA IMBALANCE IN THE DATA. FIND THE RATIO OF DATA IMBALANCE.
4. EXPLAIN THE RESULTS OF UNIVARIATE, SEGMENTED UNIVARIATE, BIVARIATE ANALYSIS, ETC. IN BUSINESS TERMS.
5. FIND THE TOP 10 CORRELATION FOR THE CLIENT WITH PAYMENT DIFFICULTIES AND ALL OTHER CASES (TARGET VARIABLE). NOTE THAT YOU HAVE TO FIND THE TOP CORRELATION BY SEGMENTING THE DATA FRAME W.R.T TO THE TARGET VARIABLE AND THEN FIND THE TOP CORRELATION FOR EACH OF THE SEGMENTED DATA AND FIND IF ANY INSIGHT IS THERE. SAY, THERE ARE 5+1(TARGET) VARIABLES IN A DATASET: VAR1, VAR2, VAR3, VAR4, VAR5, TARGET. AND IF YOU HAVE TO FIND TOP 3 CORRELATION, IT CAN BE: VAR1 & VAR2, VAR2 & VAR3, VAR1 & VAR3. TARGET VARIABLE WILL NOT FEATURE IN THIS CORRELATION AS IT IS A CATEGORICAL VARIABLE AND NOT A CONTINUOUS VARIABLE WHICH IS INCREASING OR DECREASING.

DATA SET 1: APPLICATION_DATA.CSV

1.MISSING/NULL VALUES

```
In [166]: df.isnull().sum().sort_values(ascending=False)
```

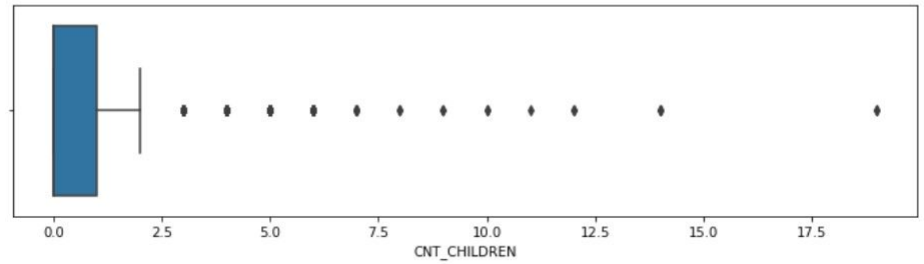
```
Out[166]: COMMONAREA_MEDI                214865
          COMMONAREA_AVG                 214865
          COMMONAREA_MODE                214865
          NONLIVINGAPARTMENTS_MODE       213514
          NONLIVINGAPARTMENTS_MEDI       213514
          NONLIVINGAPARTMENTS_AVG        213514
          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
                                           ...
```

```
In [14]: df.dropna(axis=1,inplace=True)

In [15]: df.shape
Out[15]: (307511, 55)
```

2.ANALYSIS OF OUTLIERS

only single extreme high value data point is present as outlier in CNT_CHILDREN

```
In [177]: plt.figure(figsize = (12,3))
          sns.boxplot(df['CNT_CHILDREN'])
          plt.show()
```

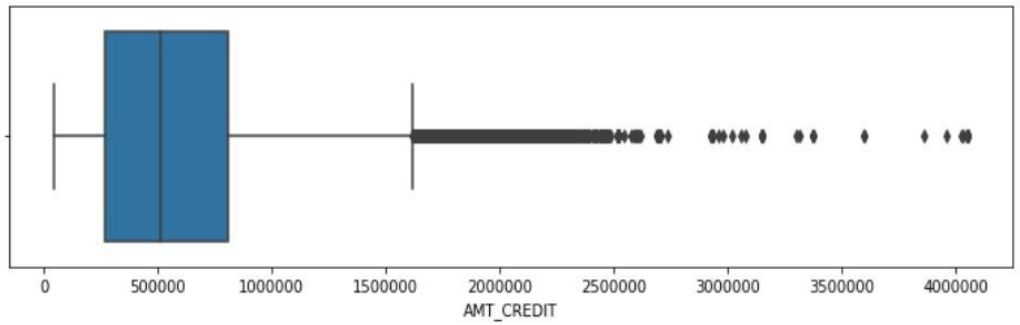



```
plt.figure(figsize = (12,3))
sns.boxplot(appda['AMT_INCOME_TOTAL'])
plt.show()
```

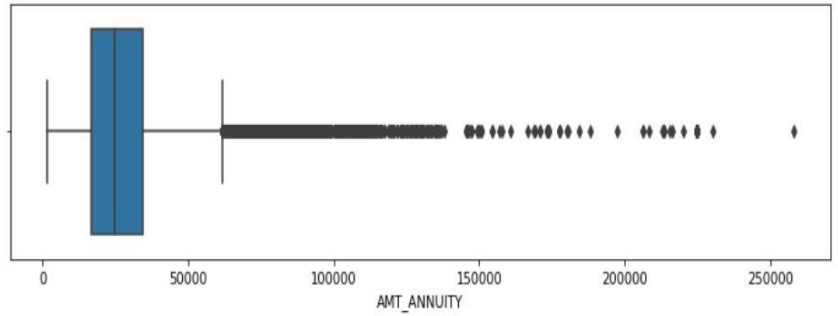
In AMT_INCOME_TOTAL only single high value data point is present as outlier

AMT_CREDIT HAS LITTLE BIT MORE OUTLIERS

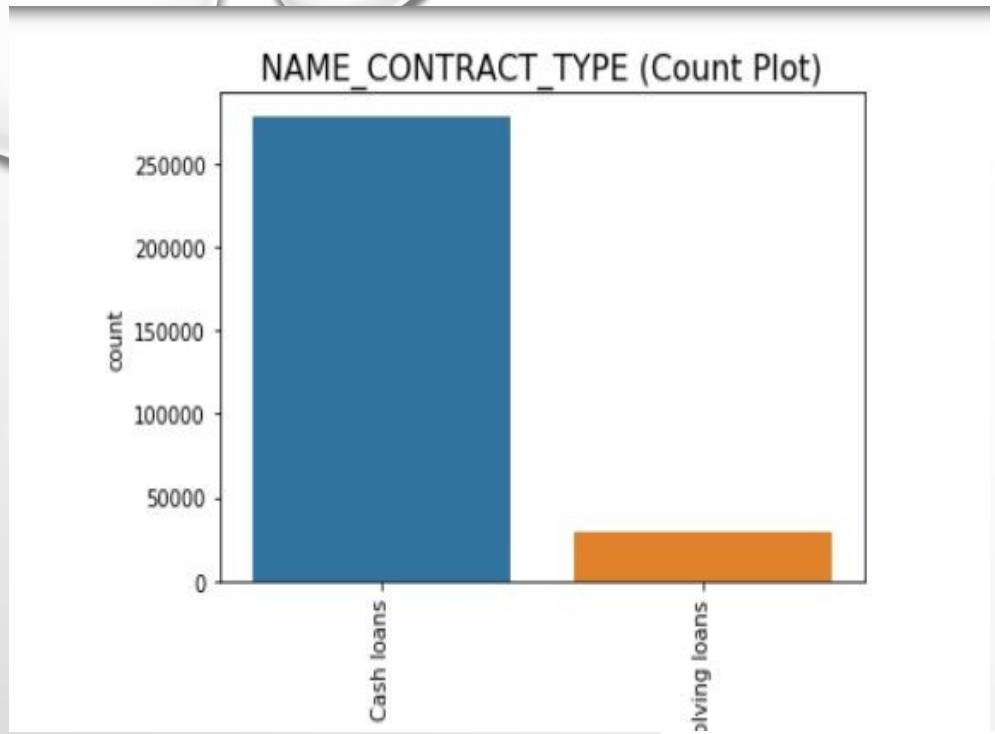
```
In [179]: plt.figure(figsize = (12,3))
          sns.boxplot(df['AMT_CREDIT'])
          plt.show()
```



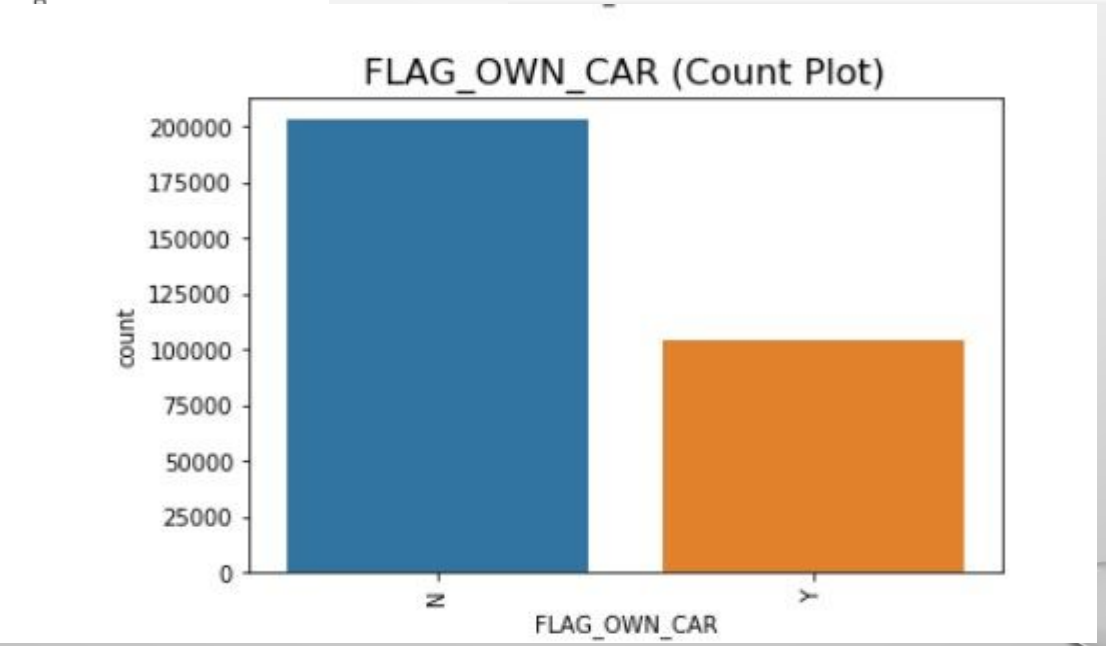

```
In [180]: plt.figure(figsize = (12,3))
          sns.boxplot(df['AMT_ANNUITY'])
          plt.show()
```

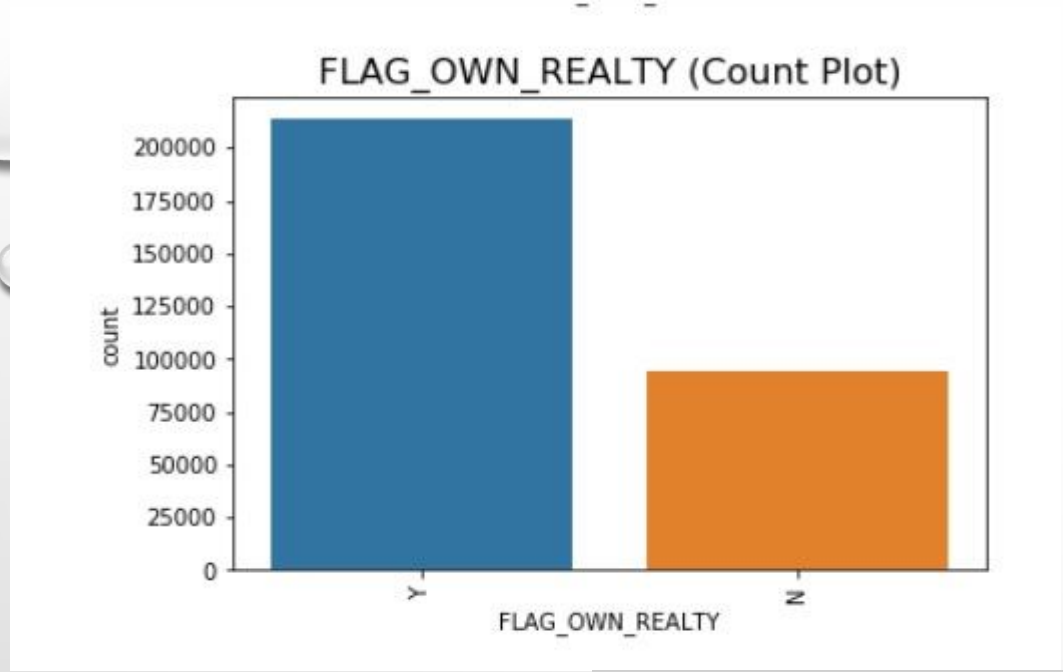


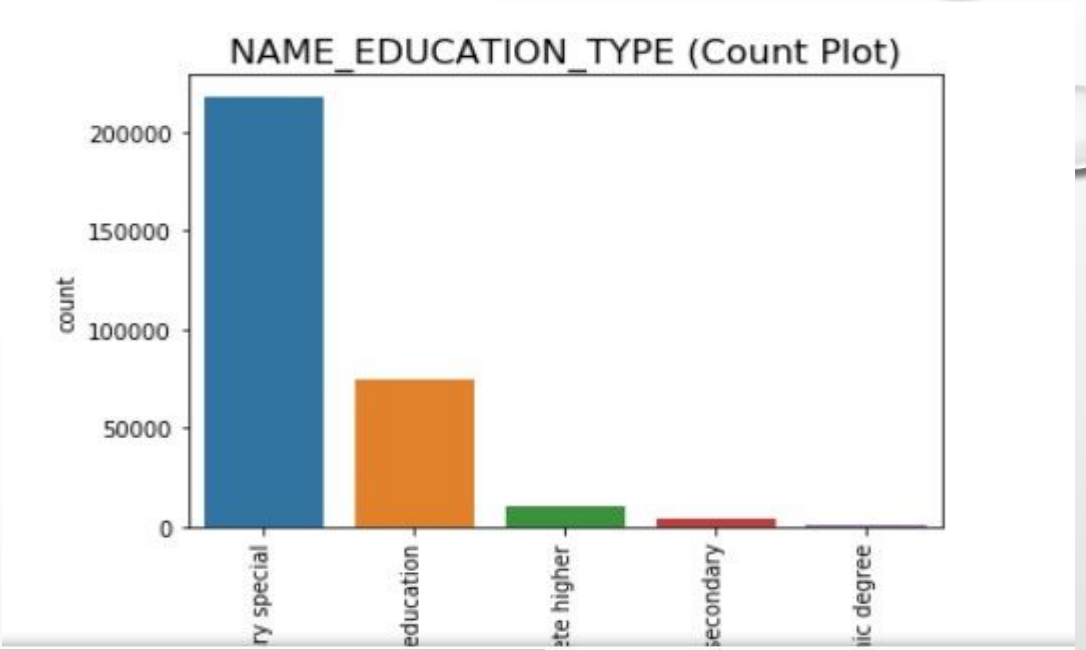
1st quartiles and 3rd quartile for AMT_ANNUITY is moved towards first quartile.

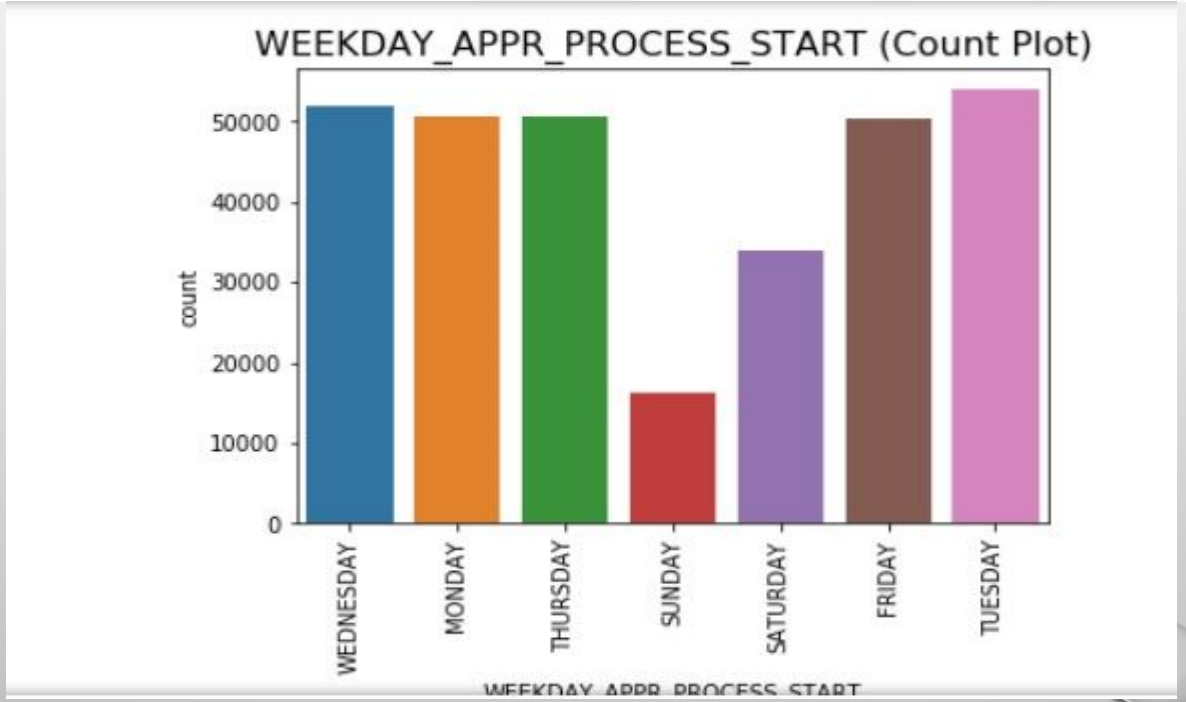

NAME_CONTRACT_TYPE (Count Plot)
count
250000
200000
150000
100000
50000
0
Cash loans


CODE_GENDER (Count Plot)
count
200000
175000
150000
125000
100000
75000
50000
25000
0
M
F
XNA
CODE_GENDER


FLAG_OWN_CAR (Count Plot)
count
200000
175000
150000
125000
100000
75000
50000
25000
0
N
Y
FLAG_OWN_CAR


FLAG_OWN_REALTY (Count Plot)
count
200000
175000
150000
125000
100000
75000
50000
25000
0
Y
N
FLAG_OWN_REALTY

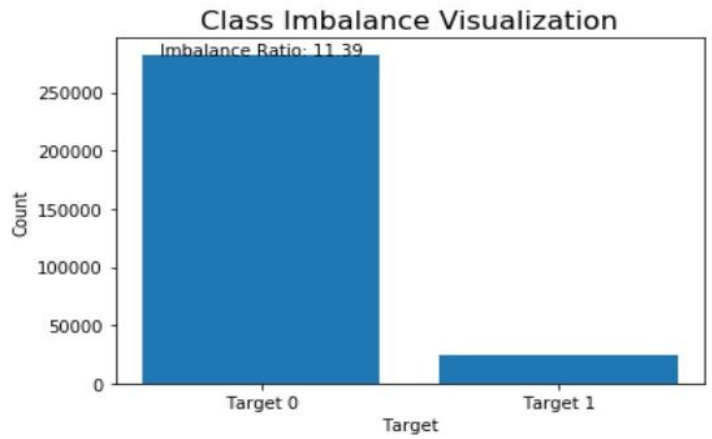

NAME_EDUCATION_TYPE (Count Plot)
count
200000
150000
100000
50000
0


WEEKDAY_APPR_PROCESS_START (Count Plot)
count
50000
40000
30000
20000
10000
0
WEDNESDAY
MONDAY
THURSDAY
SUNDAY
SATURDAY
FRIDAY
TUESDAY

3. IMBALANCE PERCENTAGE

Imbalance percenta

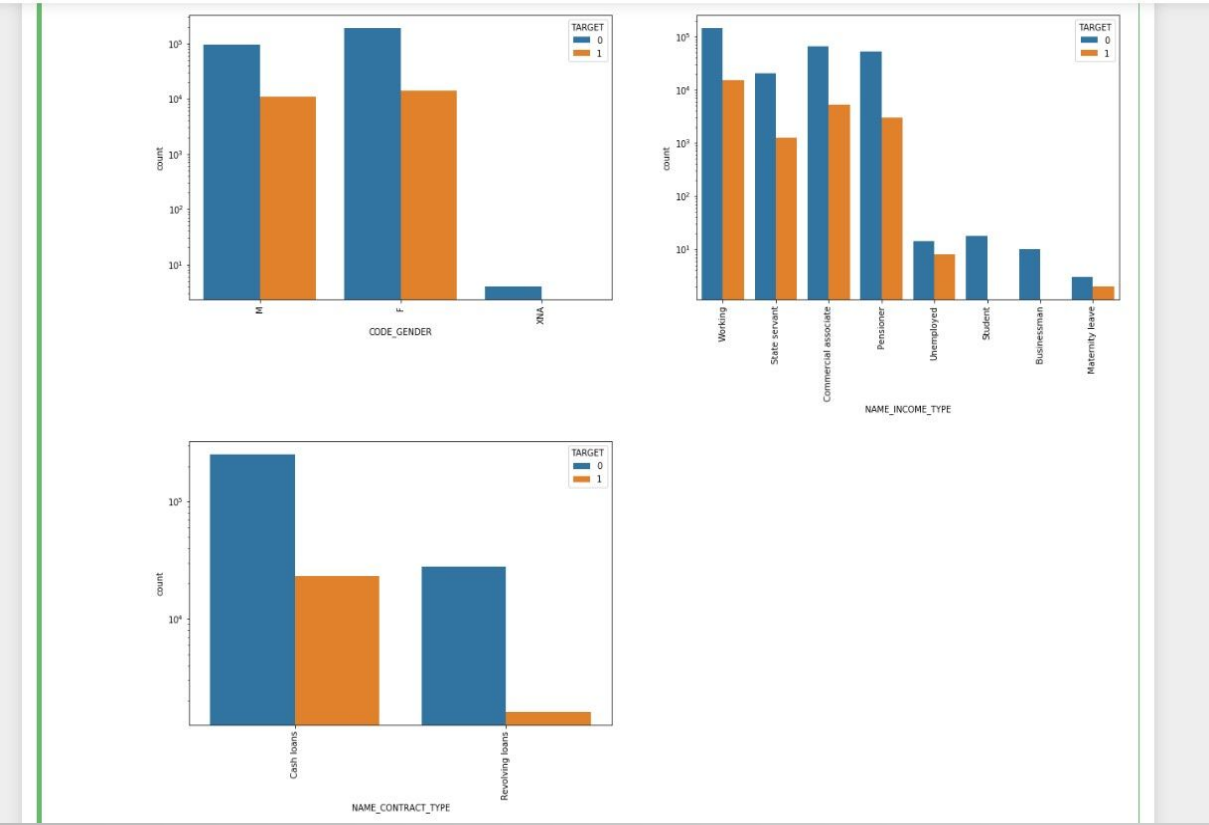
```
In [182]: plt.figure(figsize=(6, 4))
          plt.bar(['Target 0', 'Target 1'], [len(Target0), len(Target1)])
          plt.xlabel('Target')
          plt.ylabel('Count')
          plt.title('Class Imbalance Visualization')
          plt.text(0, len(Target0), f'Imbalance Ratio: {Imbalance}', ha='center')
          plt.show()
```



4.Univariate And Bivariate Analysis

Univariate

```
In [183]: flow = ['CODE_GENDER', 'NAME_INCOME_TYPE','NAME_CONTRACT_TYPE']
          plt.figure(figsize = (20, 15))

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

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

              plt.xticks(rotation = 90)
              plt.yscale('log')
```



CODE_GENDER:

The % of defaulters are more in Male than Female

NAME_INCOME_TYPE:

Student and business are higher in percentage of loan repayment. - Working, State servent and Commercial associates are higher in default percentage.

NAME_CONTRACT_TYPE

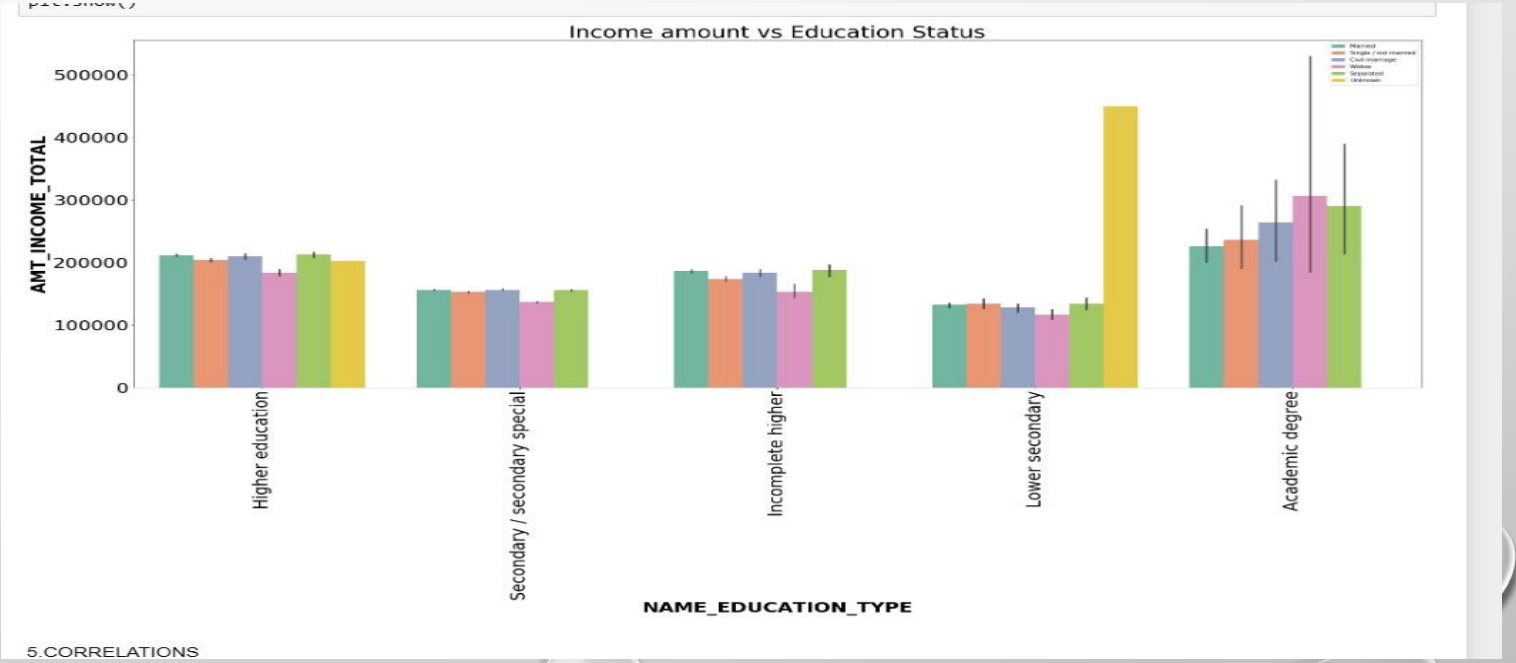
For contract type ‘Cash loans’ are high in number of credits than ‘Revolving loans’ contract type.

Bivariate analysis

```
In [184]:
plt.figure(figsize=(35,14))


sns.barplot(data =Target0, x='NAME_EDUCATION_TYPE',y='AMT_INCOME_TOTAL',
            hue ='NAME_FAMILY_STATUS',orient='v',palette='Set2')


plt.legend( loc = 'upper right')
plt.title('Income amount vs Education Status',fontsize=35 )
plt.xlabel("NAME_EDUCATION_TYPE",fontsize= 30, fontweight="bold")
plt.ylabel("AMT_INCOME_TOTAL",fontsize= 30, fontweight="bold")
plt.xticks(rotation=90, fontsize=30)
plt.yticks(rotation=360, fontsize=30)

plt.show()
```

Bivariate

- Clients having Higher Education, Incomplete Higher Education, Lower Secondary Education and Academic degree have a higher number of outliers.



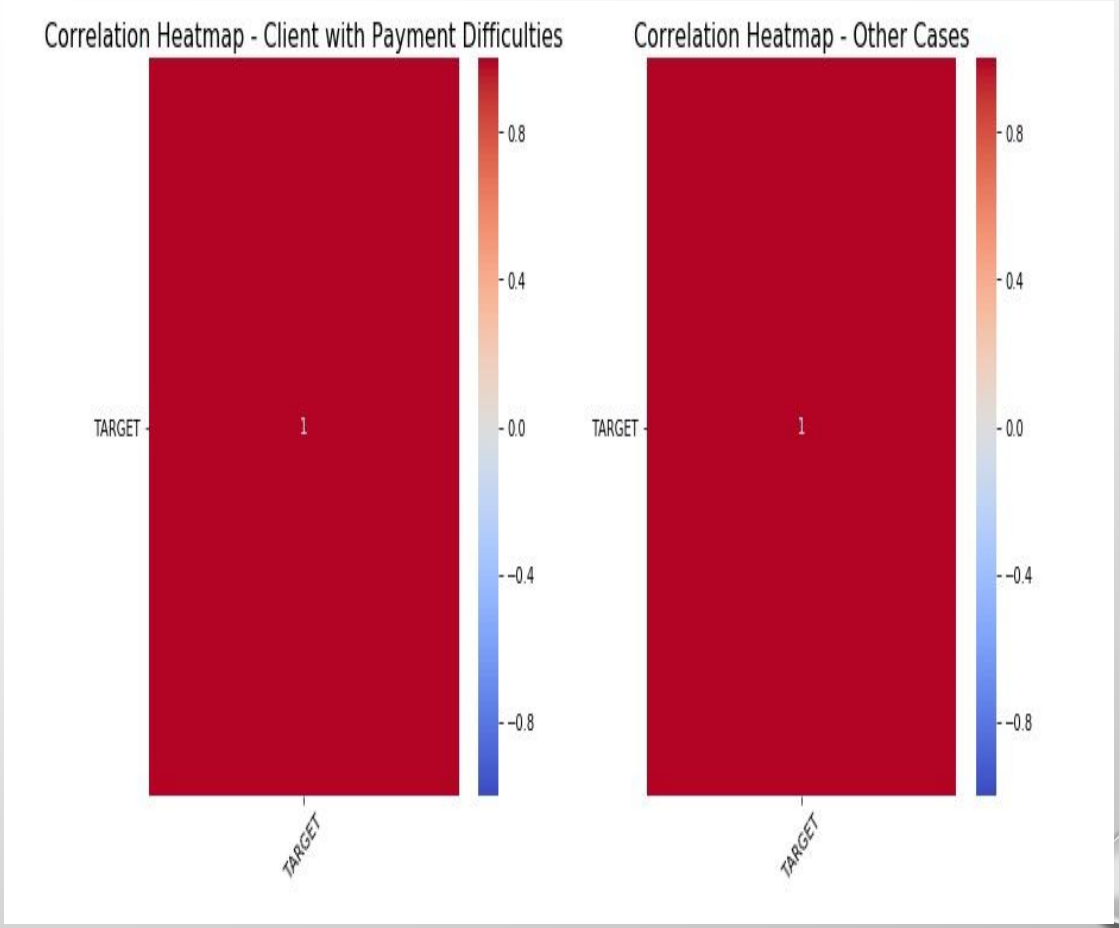
5.CORRELATIONS

5.Correlation

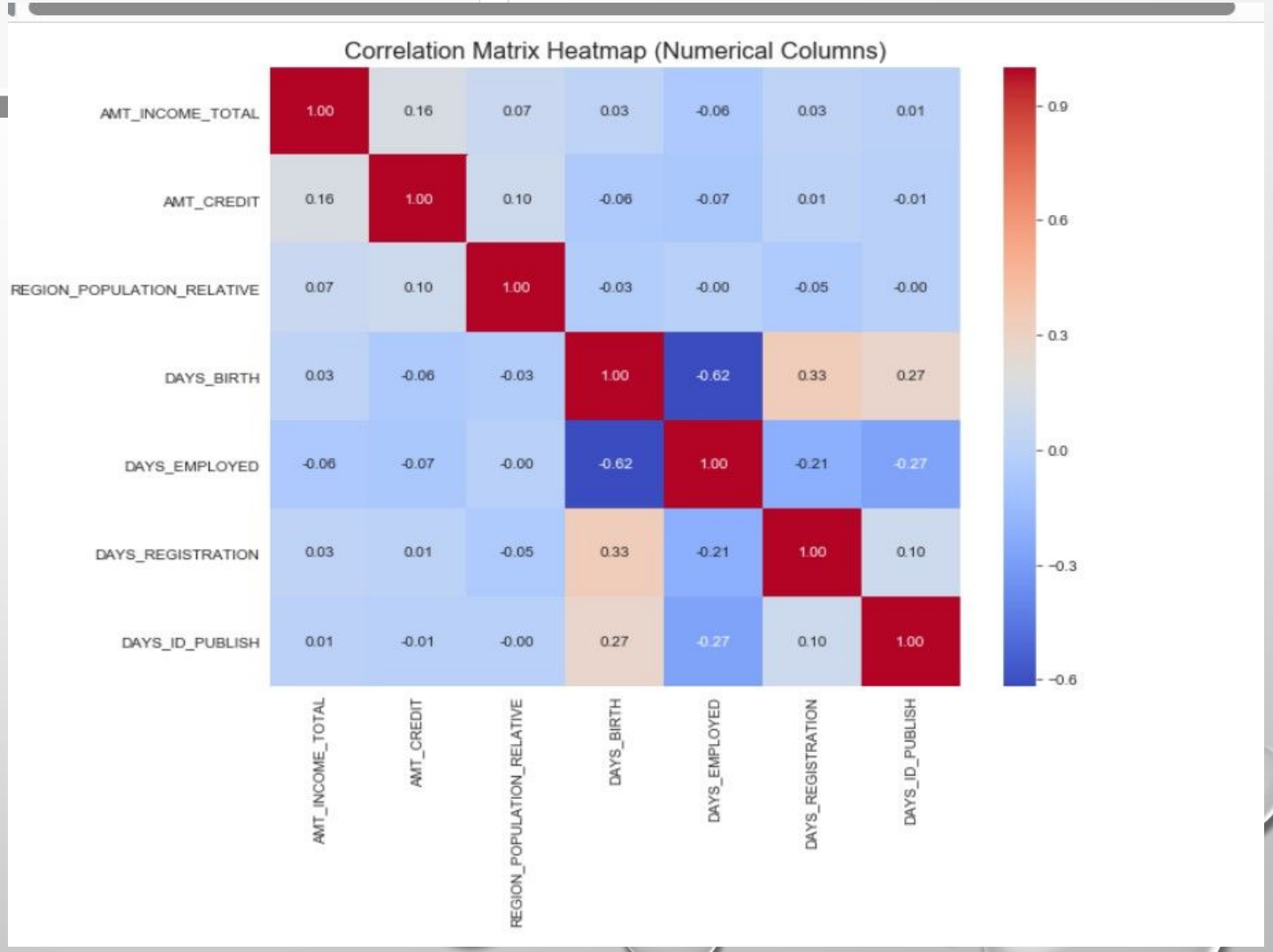
In [186]:

```
# Create a DataFrame for the top correlations in each subset
difficulties_top_corr_df = df[list(difficulties_correlations.index) + ['TARGET']]
other_cases_top_corr_df = df[list(other_cases_correlations.index) + ['TARGET']]

# Calculate the correlation matrices for the top correlations in each subset
difficulties_corr_matrix = difficulties_top_corr_df.corr()
other_cases_corr_matrix = other_cases_top_corr_df.corr()

# Plot the correlation heatmaps
plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
sns.heatmap(difficulties_corr_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Correlation Heatmap - Client with Payment Difficulties')
plt.xticks(rotation=45)
plt.yticks(rotation=0)

plt.subplot(1, 2, 2)
sns.heatmap(other_cases_corr_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Correlation Heatmap - Other Cases')
plt.xticks(rotation=45)
plt.yticks(rotation=0)

plt.tight_layout()
plt.show()
```

Target is highly correlated



```
In [64]: # Select numerical columns
numerical_columns = ['AMT_INCOME_TOTAL', 'AMT_CREDIT', 'REGION_POPULATION_RELATIVE', 'DAYS_BIRTH', 'DAYS_EMPLOYED', 'DAYS_REGISTI

# Create correlation matrix
correlation_matrix = df[numerical_columns].corr()

# Generate heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix Heatmap (Numerical Columns)')
plt.show()
```



DATASET2: Previous application.csv

1.MISSING VALUE

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

```
Out[8]: RATE_INTEREST_PRIVILEGED      1664263
        RATE_INTEREST_PRIMARY         1664263
        RATE_DOWN_PAYMENT              895844
        AMT_DOWN_PAYMENT               895844
        NAME_TYPE_SUITE                820405
        DAYS_TERMINATION               673065
        NFLAG_INSURED_ON_APPROVAL      673065
        DAYS_FIRST_DRAWING             673065
        DAYS_FIRST_DUE                 673065
        DAYS_LAST_DUE_1ST_VERSION      673065
        DAYS_LAST_DUE                  673065
        AMT_GOODS_PRICE                385515
        AMT_ANNUITY                    372235
        CNT_PAYMENT                    372230
        PRODUCT_COMBINATION               346
```

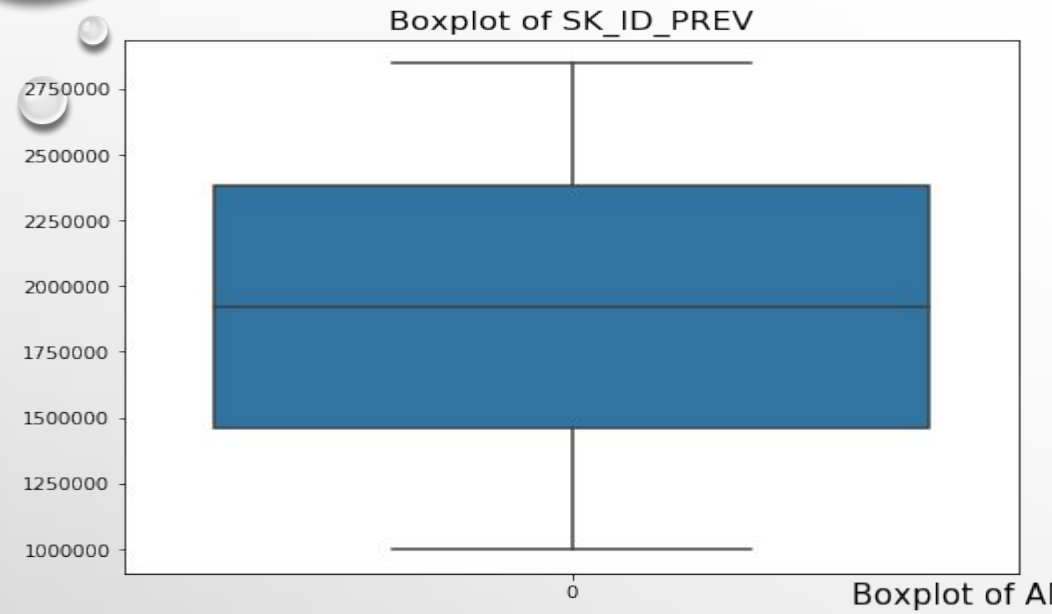
```
In [37]: data.drop(['RATE_INTEREST_PRIVILEGED','RATE_INTEREST_PRIMARY','RATE_DOWN_PAYMENT','AMT_DOWN_PAYMENT','NAME_TYPE_SUITE','DAYS_TERI
```

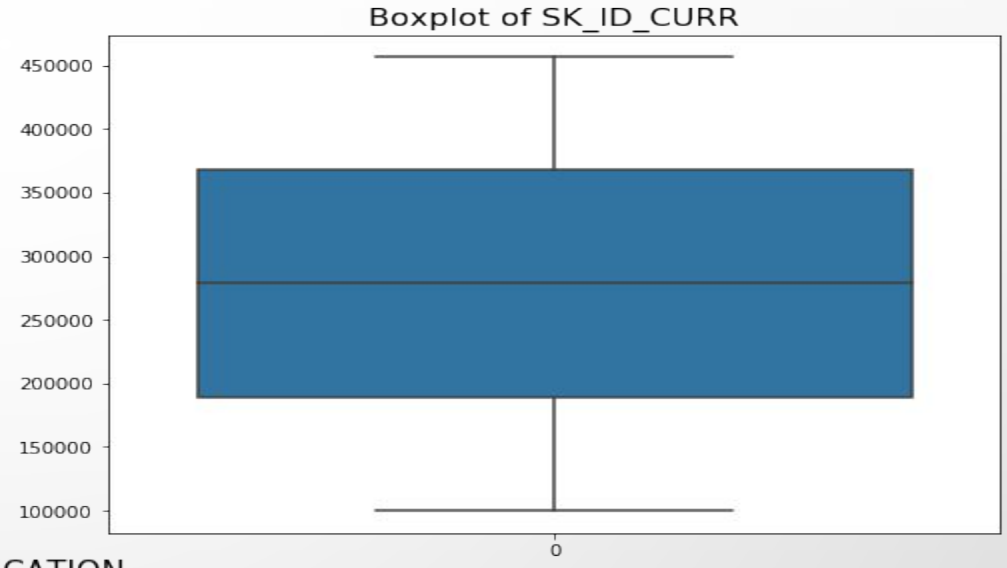
```
In [38]: data.shape

Out[38]: (1670214, 23)
```

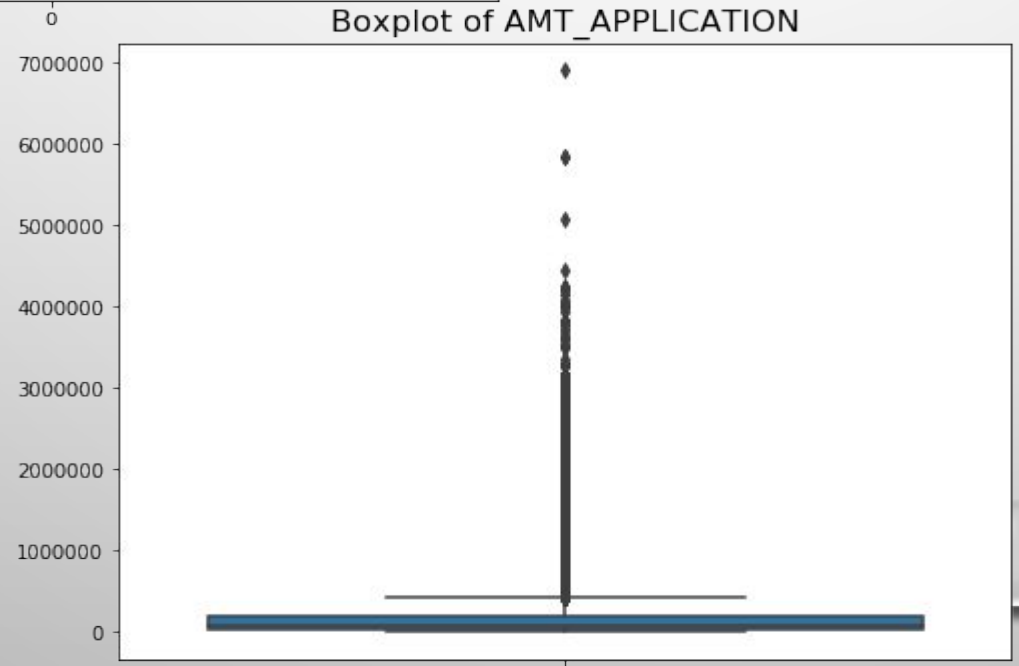
```
In [39]: mean_value = data['AMT_CREDIT'].mean()
         data['AMT_CREDIT'].fillna(mean_value, inplace=True)
```

```
In [40]: mode_value = data['PRODUCT_COMBINATION'].mode()[0]
         data['PRODUCT_COMBINATION'].fillna(mode_value, inplace=True)
```

2.OUTLIERS


Boxplot of SK_ID_PREV
2750000
2500000
2250000
2000000
1750000
1500000
1250000
1000000
0

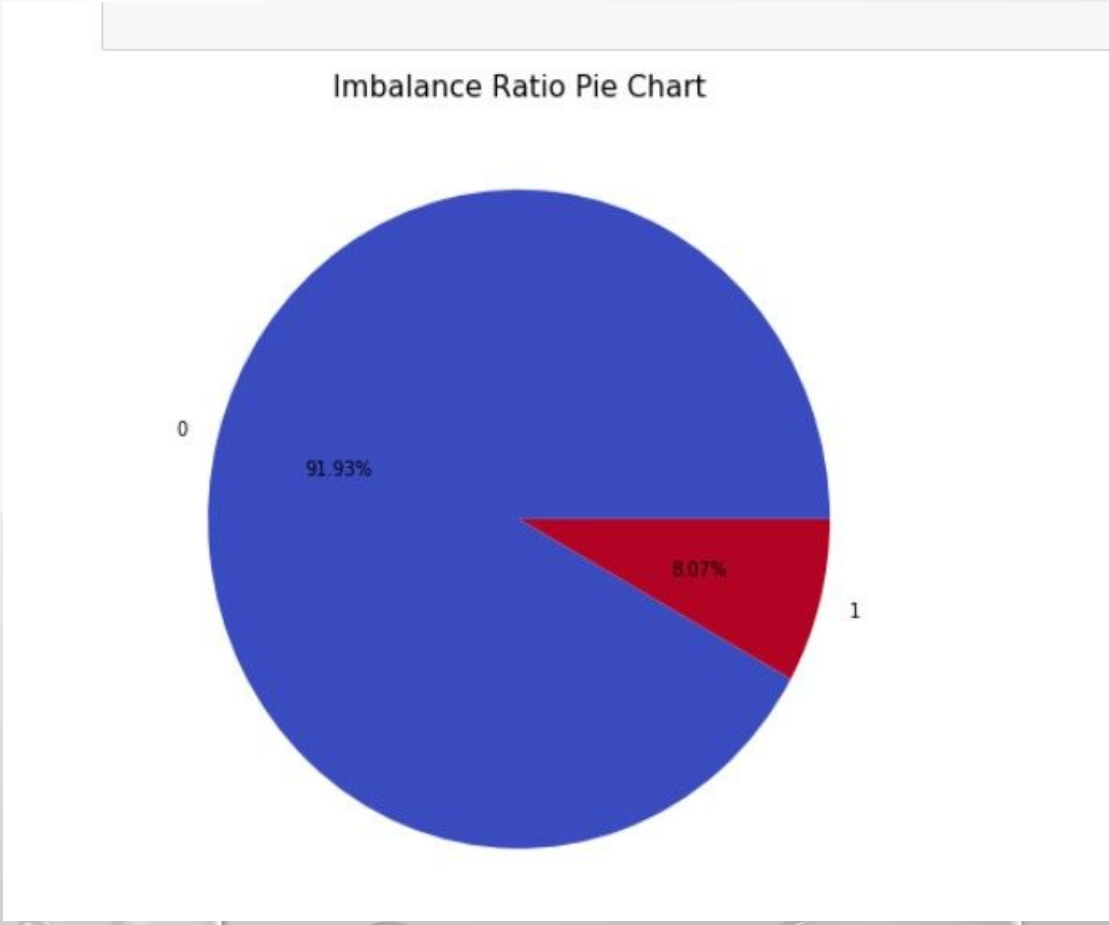

Boxplot of SK_ID_CURR
450000
400000
350000
300000
250000
200000
150000
100000
0


Boxplot of AMT_APPLICATION
7000000
6000000
5000000
4000000
3000000
2000000
1000000
0

3.IMBALANCE PERCENTAGE

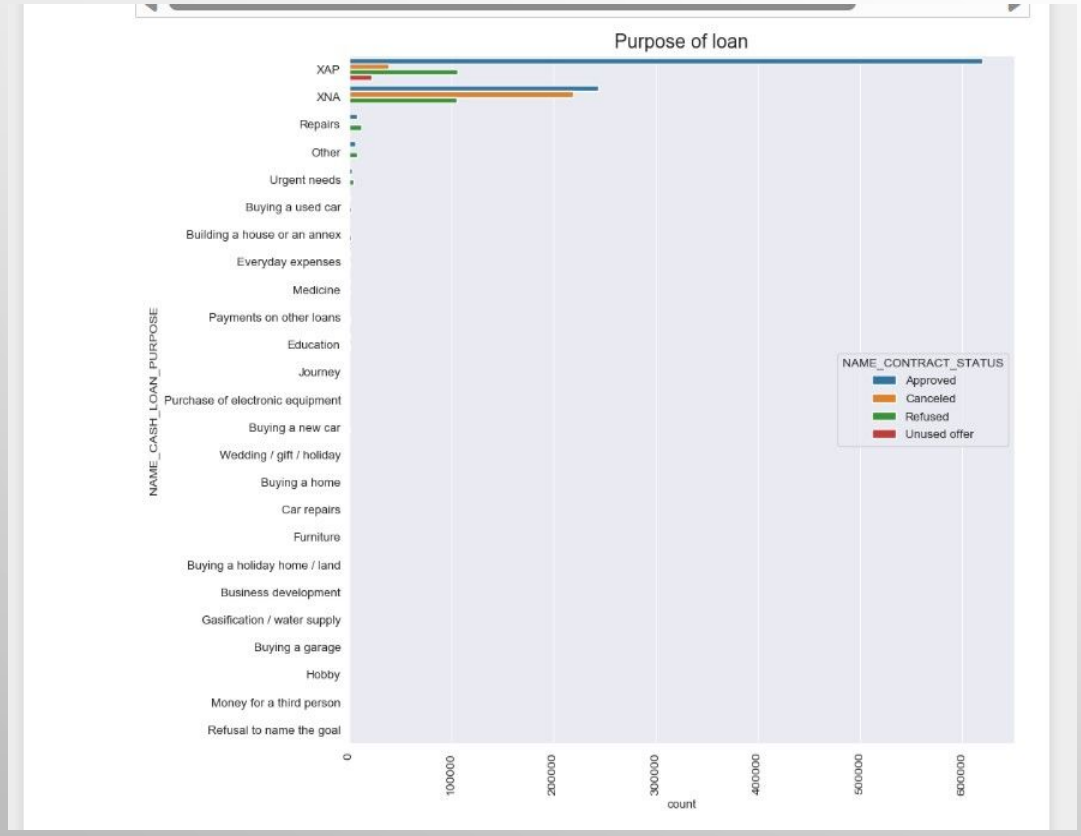
[51]:

```
# Create a DataFrame from the imbalance_ratio Series
imbalance_df = pd.DataFrame({'Imbalance Ratio': imbalance_ratio})

# Plot the pie chart
plt.figure(figsize=(8, 8))
imbalance_df['Imbalance Ratio'].plot.pie(autopct='%.2f%%', cmap='coolwarm')

# Set the title and labels
plt.title('Imbalance Ratio Pie Chart')
plt.ylabel('')

# Show the plot
plt.show()
```

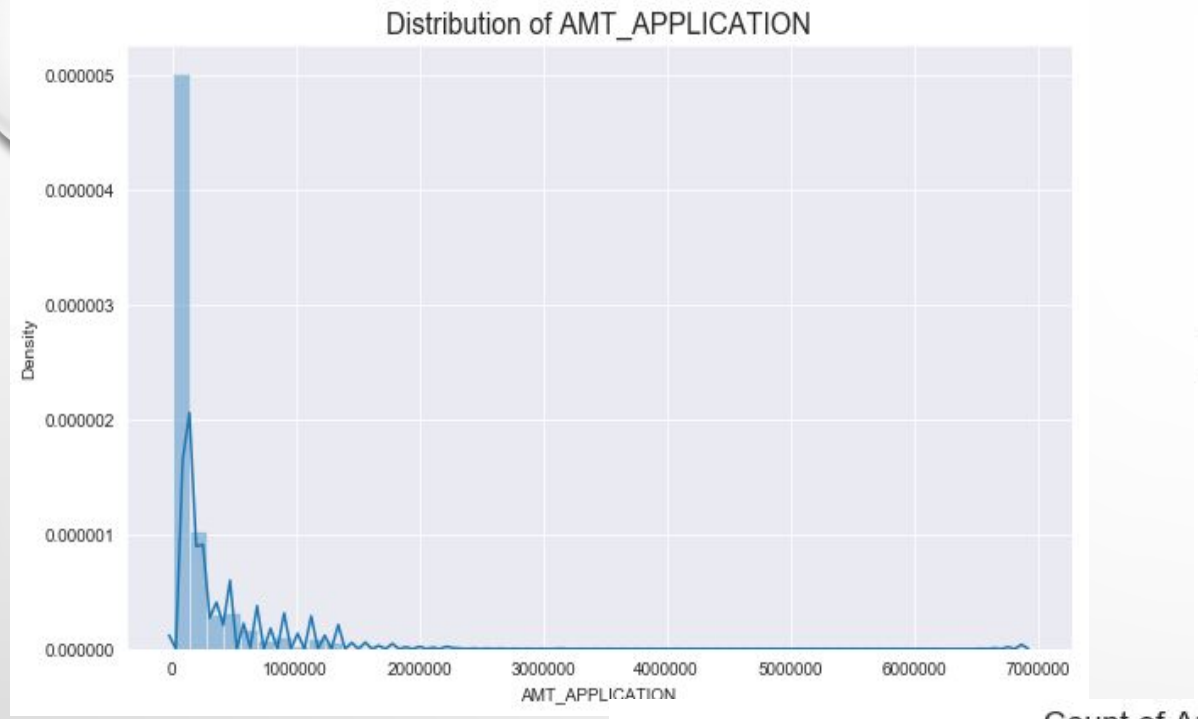


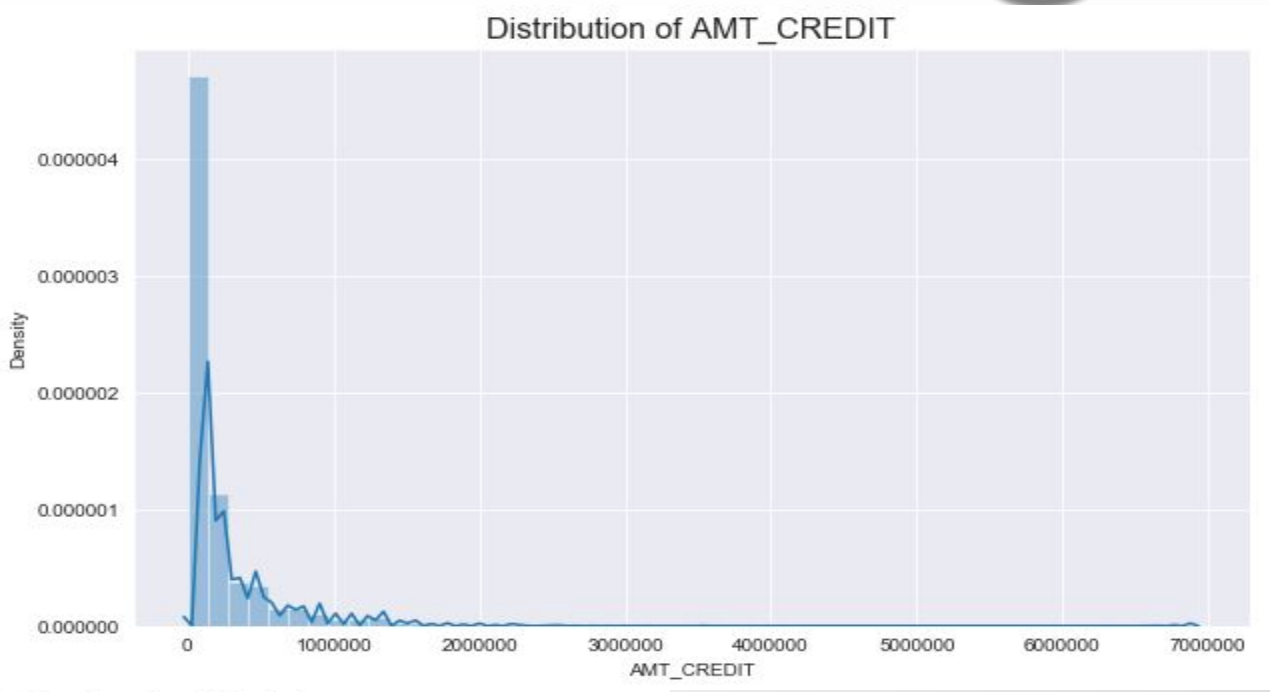
3.Univariate analysis

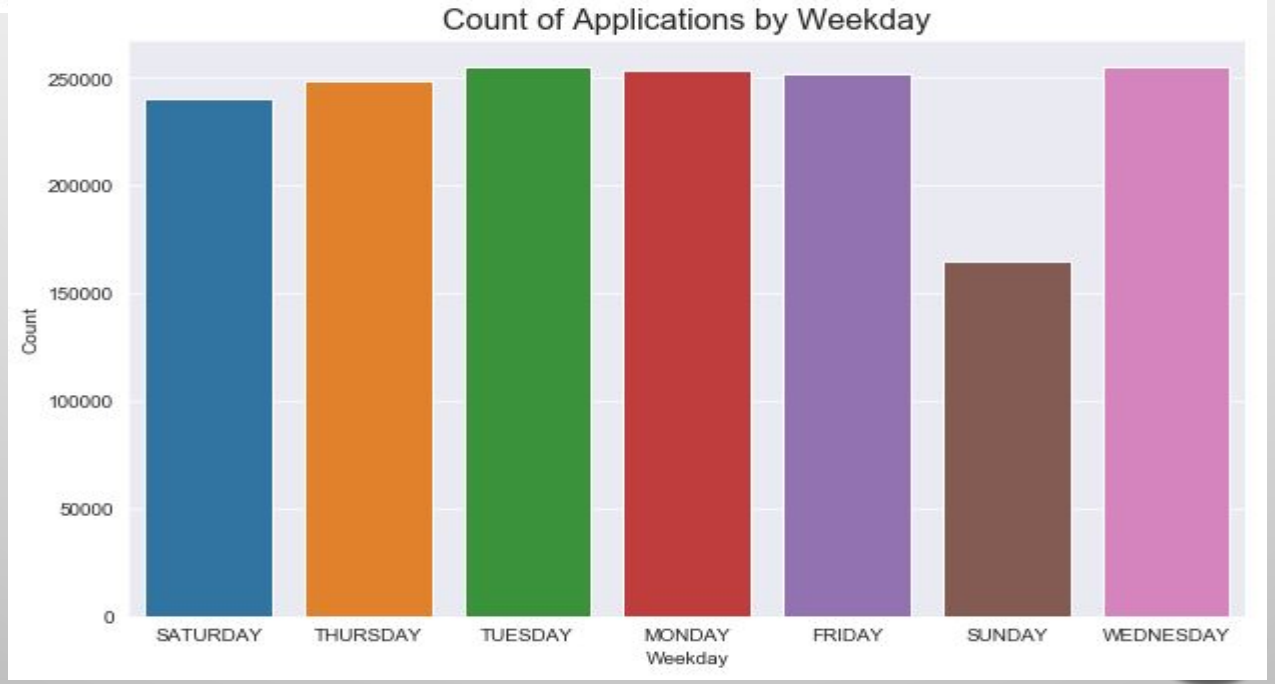
```
In [228]: plt.figure(figsize=(10,10),dpi =100)
          plt.xticks(rotation=90)
          plt.title('Purpose of loan')
          sns.set_style('darkgrid')
          ax = sns.countplot(data=dfdata,y= 'NAME_CASH_LOAN_PURPOSE', order=dfdata['NAME_CASH_LOAN_PURPOSE'].value_counts().index,hue = 'NA
          plt.show()
```

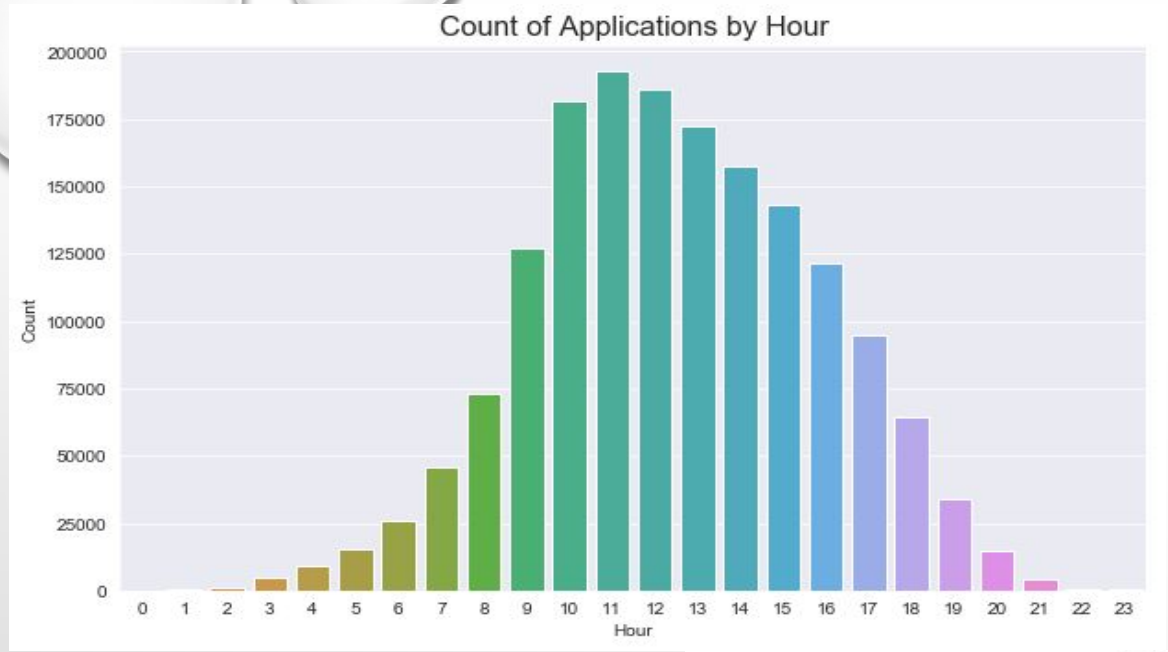


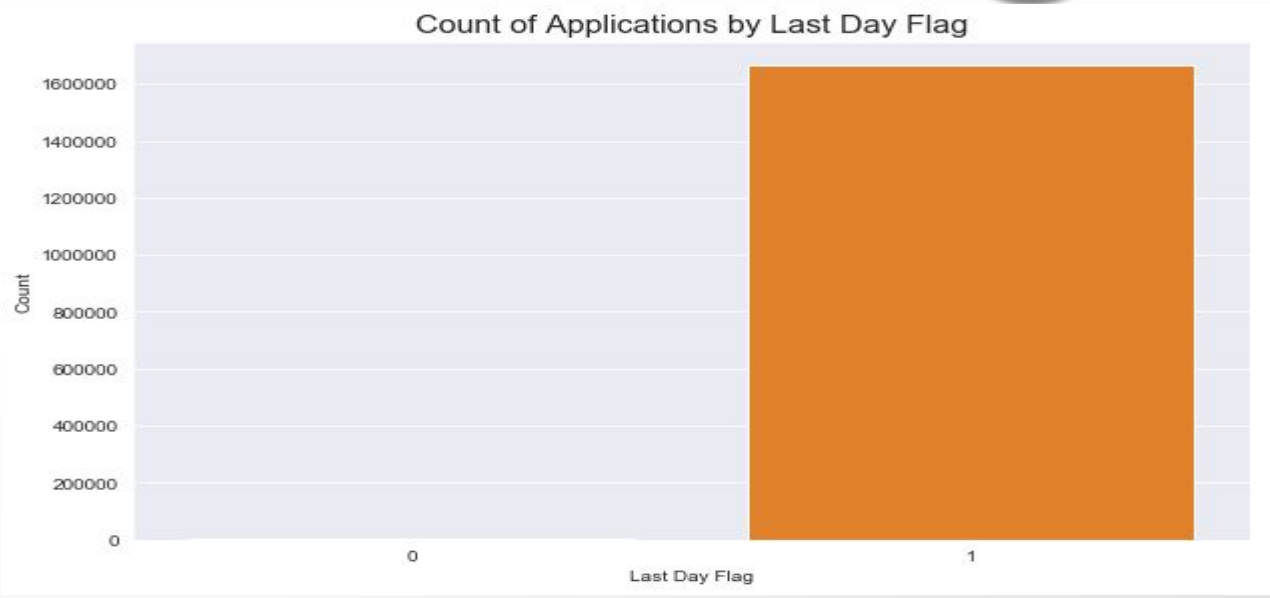
Most of loan rejection was from 'repairs'

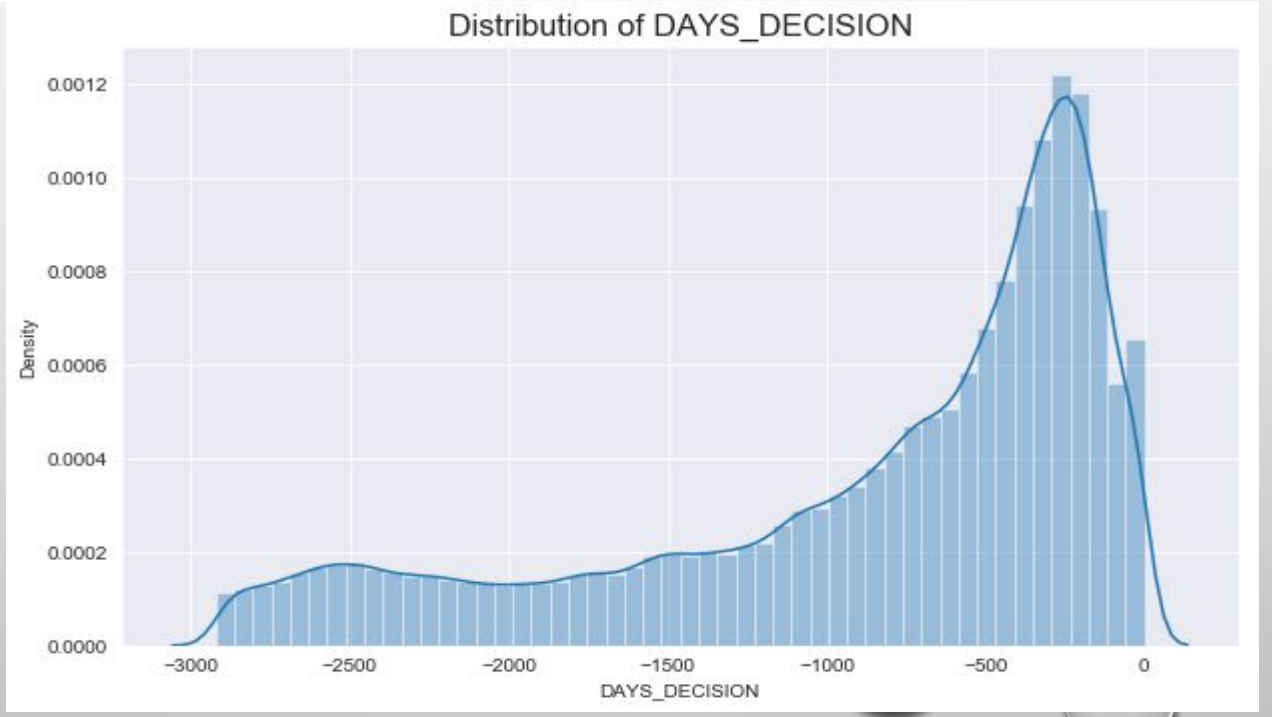

Distribution of AMT_APPLICATION
Density
0.000005
0.000004
0.000003
0.000002
0.000001
0.000000
0
1000000
2000000
3000000
4000000
5000000
6000000
7000000
AMT_APPLICATION


Distribution of AMT_CREDIT
Density
0.000004
0.000003
0.000002
0.000001
0.000000
0
1000000
2000000
3000000
4000000
5000000
6000000
7000000
AMT_CREDIT

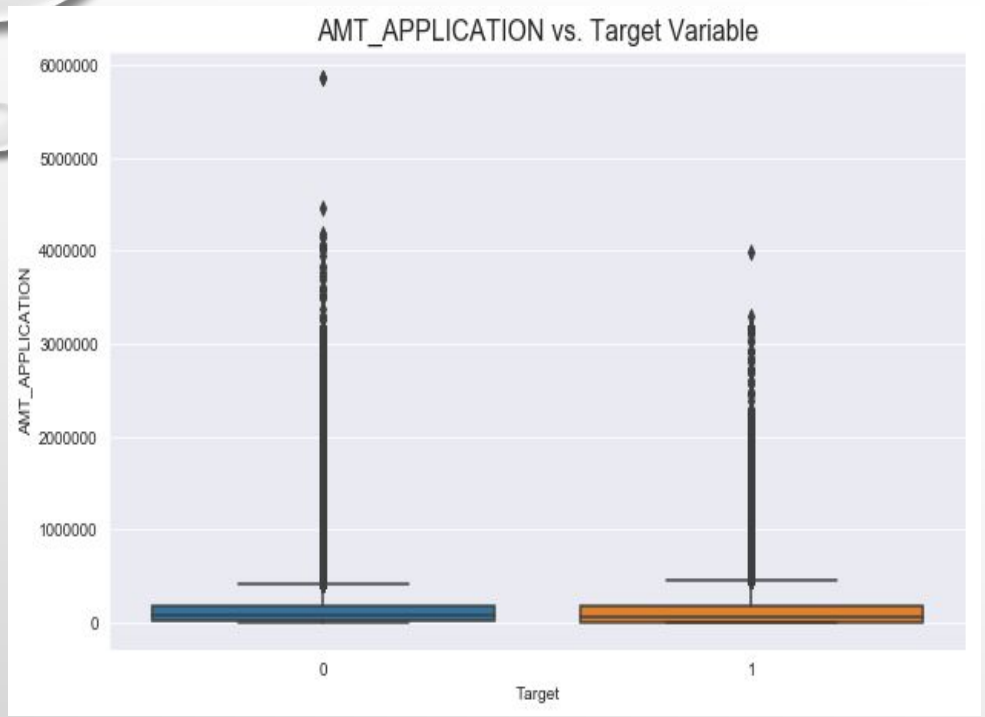

Count of Applications by Weekday
Count
250000
200000
150000
100000
50000
0
SATURDAY
THURSDAY
TUESDAY
MONDAY
FRIDAY
SUNDAY
WEDNESDAY
Weekday

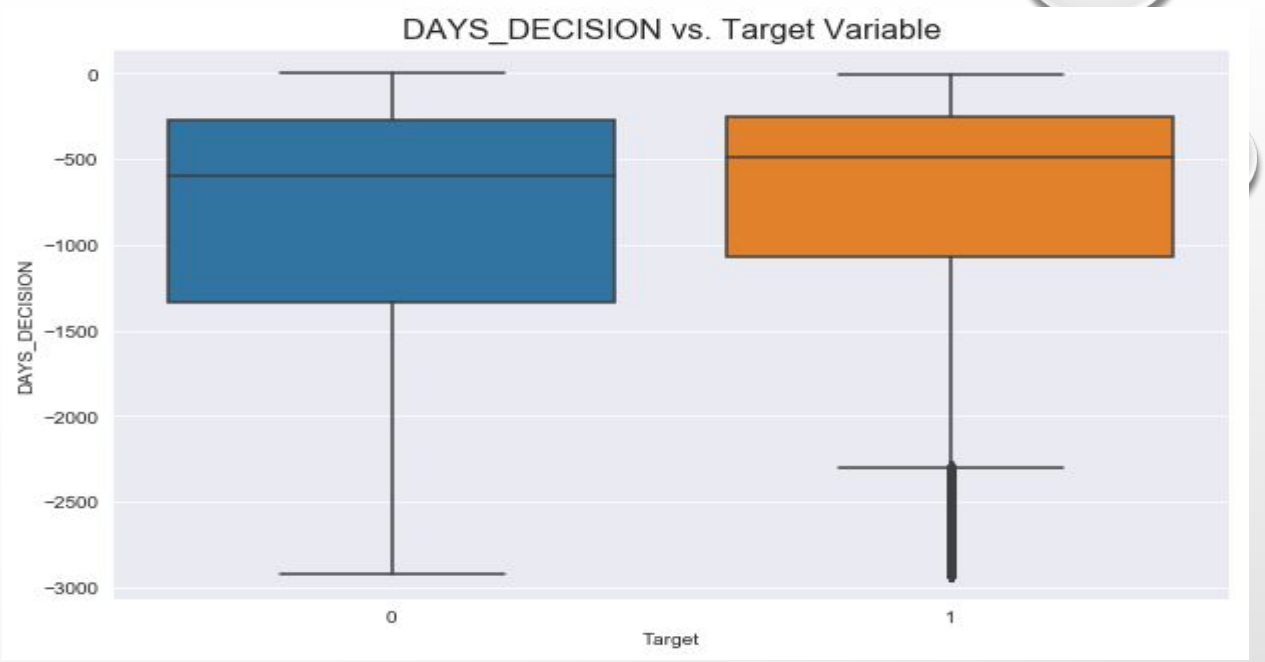

Count of Applications by Hour
Count
Hour
0
25000
50000
75000
100000
125000
150000
175000
200000
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23

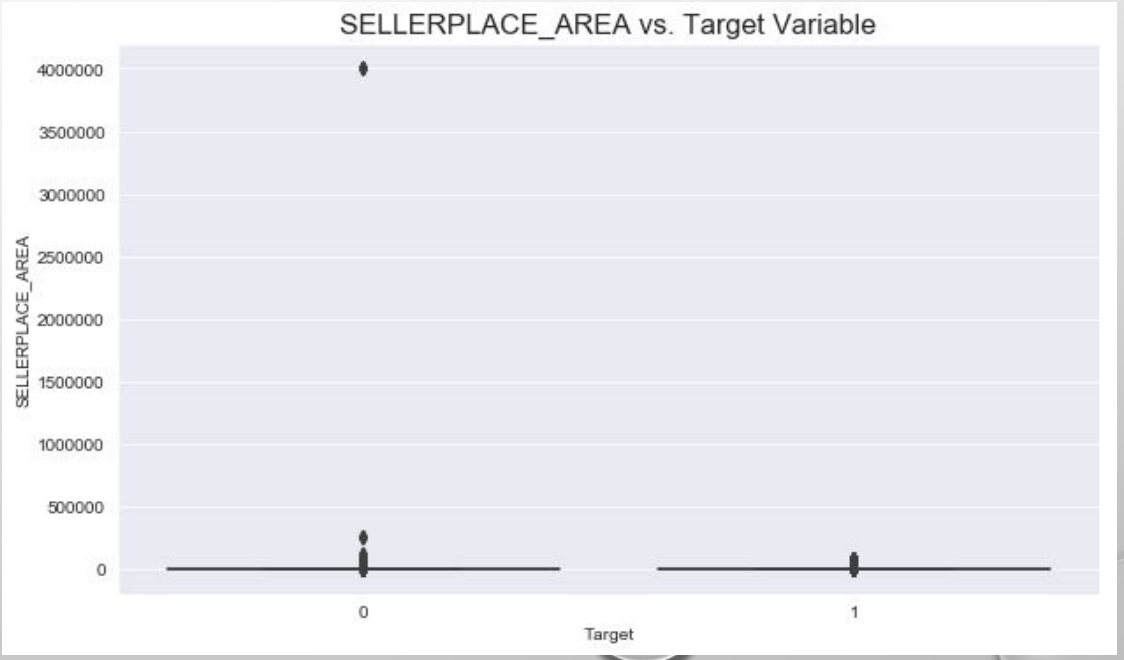

Count of Applications by Last Day Flag
Count
Last Day Flag
0
200000
400000
600000
800000
1000000
1200000
1400000
1600000
0
1


Distribution of DAYS_DECISION
Density
DAYS_DECISION
0.0000
0.0002
0.0004
0.0006
0.0008
0.0010
0.0012
−3000
−2500
−2000
−1500
−1000
−500
0

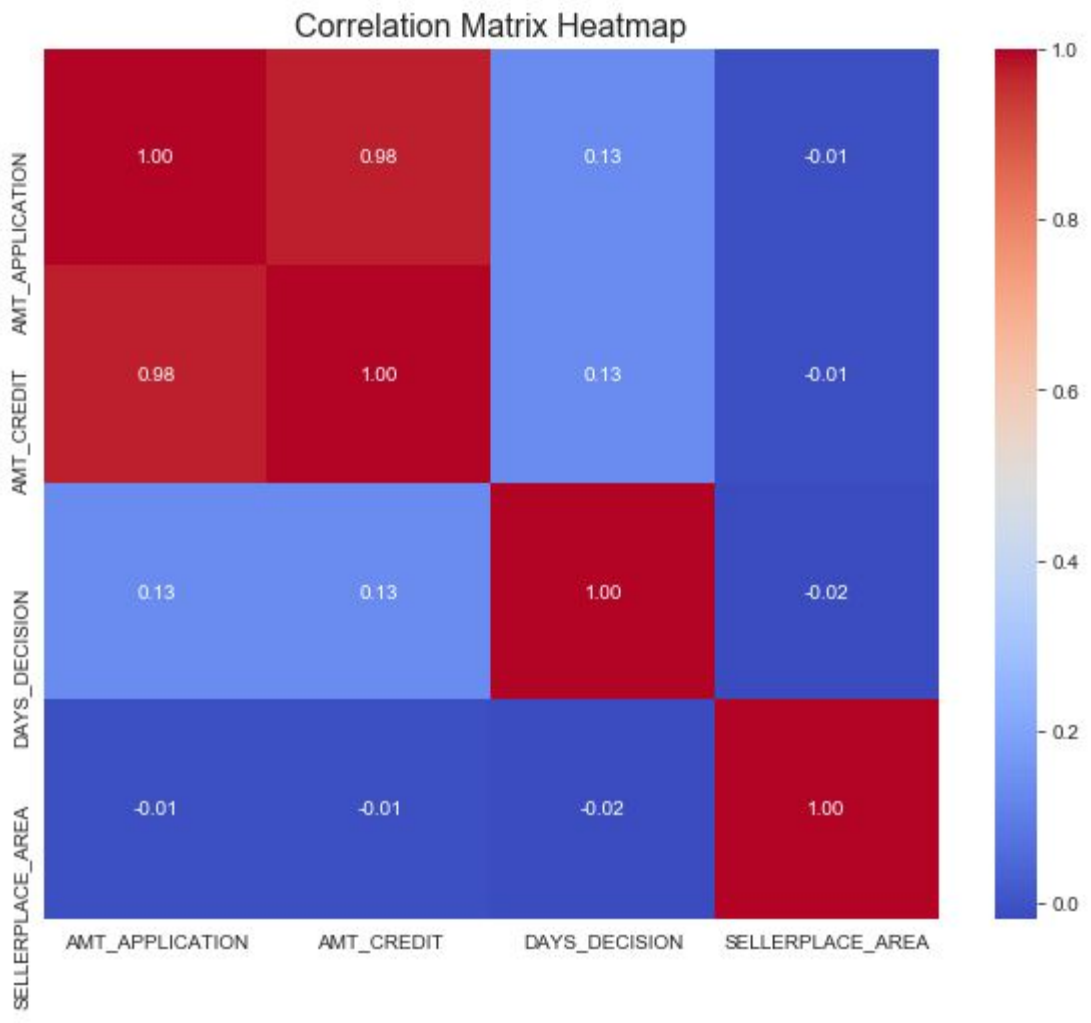
BIVARIATE


AMT_APPLICATION vs. Target Variable
AMT_APPLICATION
6000000
5000000
4000000
3000000
2000000
1000000
0
0
1
Target


DAYS_DECISION vs. Target Variable
DAYS_DECISION
0
−500
−1000
−1500
−2000
−2500
−3000
0
1
Target


SELLERPLACE_AREA vs. Target Variable
SELLERPLACE_AREA
4000000
3500000
3000000
2500000
2000000
1500000
1000000
500000
0
0
1
Target

5.CORRELATION



CORRELATION

In [62]:
```
numerical_columns = ['AMT_APPLICATION', 'AMT_CREDIT', 'DAYS_DECISION', 'SELLERPLACE_AREA']

# Create correlation matrix
correlation_matrix = data[numerical_columns].corr()

# Generate heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix Heatmap')
plt.show()
```