In [213…
```
import warnings
warnings.filterwarnings('ignore')
from operator import itemgetter
import pandas as pd #dataframe
import numpy as np #mathematical computations
import matplotlib.pyplot as plt #visualization
import matplotlib
import seaborn as sns #visualization
import json #exporting columns
import pickle #saving the model
from sklearn.linear_model import LinearRegression #Linear Regression
from sklearn.linear_model import Lasso #Lasso Regression
from sklearn.tree import DecisionTreeRegressor #Decision Tree Regression
from sklearn.ensemble import RandomForestRegressor #Random Forest Regression
from sklearn.model_selection import train_test_split #Splitting the dataset
from sklearn.model_selection import ShuffleSplit #Random shuffling
from sklearn.model_selection import cross_val_score #Score cross validation
from sklearn.model_selection import GridSearchCV #Hyper parameter tuning
from warnings import simplefilter #Filtering warnings
import seaborn as sns
import missingno as msno
import statsmodels.api as sm
from datetime import datetime
from scipy import stats
```

Observe the data

Import the data set and show the title

In [214…
```
Orginal_data = pd.read_csv('./Combined.csv',encoding = "ISO-8859-1")
Causes_data = pd.read_csv('./Causes.csv',encoding = "ISO-8859-1")
```

In [215…
```
Orginal_data.columns
```

Out[215]:
```
Index(['Campaign_ID', 'Campagin_Title ', 'Receiving_NPO_name ',
       'Receiving_NPO_Id', 'NPO_Status_orignal', 'NPO_Status',
       'Number_campaigns_NPO', 'Public_Campaign_Access', 'Creator_Type',
       'Creator_Id', 'Campaign_Status', 'Actual_Donation_Amount',
       'Distinct_Donors', 'Campaign_Goal', 'Campaign_Completion_Rate',
       'Days_Left_for_Campaign', 'Campaign_Start_Date', 'Campaign_End_Dat
e',
       'NPO_Tax_Deductibility', 'Campaign_Image1', 'Campaign_Image2',
       'Campaign_Image3', 'Campaign_Image4', 'Campaign_Image5',
       'Campaign_Video', 'Impact_Message1', 'Impact_Message2',
       'Impact_Message3', 'Impact_Message4', 'Impact_Message5',
       'Custom_Amount1', 'Custom_Amount2', 'Custom_Amount3', 'Custom_Amoun
t4',
       'Description_Campaign', 'Description_NPO'],
      dtype='object')
```

In [216…
```
Causes_data= Causes_data.fillna(0)
```

In [217…
```
combined_data = pd.merge(Orginal_data, Causes_data, how='left', on=['Campaig
```

In [218…
```
Total_Rows = combined_data.shape[0]
```

```
print(Total_Rows)
```

```
15979
```

In [219…

```
print(combined_data.columns)
```

```
Index(['Campaign_ID', 'Campagin_Title ', 'Receiving_NPO_name ',
       'Receiving_NPO_Id', 'NPO_Status_orignal', 'NPO_Status',
       'Number_campaigns_NPO', 'Public_Campaign_Access', 'Creator_Type',
       'Creator_Id', 'Campaign_Status', 'Actual_Donation_Amount',
       'Distinct_Donors', 'Campaign_Goal', 'Campaign_Completion_Rate',
       'Days_Left_for_Campaign', 'Campaign_Start_Date', 'Campaign_End_Dat
e',
       'NPO_Tax_Deductibility', 'Campaign_Image1', 'Campaign_Image2',
       'Campaign_Image3', 'Campaign_Image4', 'Campaign_Image5',
       'Campaign_Video', 'Impact_Message1', 'Impact_Message2',
       'Impact_Message3', 'Impact_Message4', 'Impact_Message5',
       'Custom_Amount1', 'Custom_Amount2', 'Custom_Amount3', 'Custom_Amount
4',
       'Description_Campaign', 'Description_NPO', 'Campaign_Title',
       'Org_Cause_Animal_Welfare', 'Org_Cause_Arts_Heritage',
       'Org_Cause_Children_Youth', 'Org_Cause_Community',
       'Org_Cause_Disability', 'Org_Cause_Education', 'Org_Cause_Elderly',
       'Org_Cause_Environment', 'Org_Cause_Families', 'Org_Cause_Health',
       'Org_Cause_Humanitarian', 'Org_Cause_Social Service',
       'Org_Cause_Sports', 'Org_Cause_Women_Girls', 'Cam_Cause_Animal_Welfa
re',
       'Cam_Cause_Arts_Heritage', 'Cam_Cause_Children_Youth',
       'Cam_Cause_Community', 'Cam_Cause_Disability', 'Cam_Cause_Educatio
n',
       'Cam_Cause_Elderly', 'Cam_Cause_Environment', 'Cam_Cause_Families',
       'Cam_Cause_Health', 'Cam_Cause_Humanitarian',
       'Cam_Cause_Social_Service', 'Cam_Cause_Sports', 'Cam_Cause_Women_Gir
ls',
       'Pub_Enquiry_Person', 'Pub_Enquiry_Contact', 'Pub_Enquiry_Email',
       'Web_URL', 'Facebook_Link'],
      dtype='object')
```

I found there is no "Organizational Causes" and "Campaign Causes" in this data set.

Here are all variables I plan to operate, ignnore other columns temporarily

In [220…

```
Need_variable = ["Actual_Donation_Amount","NPO_Tax_Deductibility", "Distinct
"Campaign_Goal","Campaign_Start_Date", "Campaign_End_Date",
"Campaign_Image1", "Campaign_Image2","Campaign_Image3",
"Campaign_Image4","Campaign_Image5","Campaign_Video",
"Impact_Message1","Impact_Message2","Impact_Message3","Impact_Message4",
"Impact_Message5","Custom_Amount1","Custom_Amount2","Custom_Amount3",
"Custom_Amount4","Description_Campaign","Description_NPO",
'Org_Cause_Animal_Welfare', 'Org_Cause_Arts_Heritage',
'Org_Cause_Children_Youth', 'Org_Cause_Community',
'Org_Cause_Disability', 'Org_Cause_Education', 'Org_Cause_Elderly',
'Org_Cause_Environment', 'Org_Cause_Families', 'Org_Cause_Health',
'Org_Cause_Humanitarian', 'Org_Cause_Social Service',
'Org_Cause_Sports', 'Org_Cause_Women_Girls', 'Cam_Cause_Animal_Welfare',
'Cam_Cause_Arts_Heritage', 'Cam_Cause_Children_Youth',
```

```
'Cam_Cause_Community', 'Cam_Cause_Disability', 'Cam_Cause_Education',
'Cam_Cause_Elderly', 'Cam_Cause_Environment', 'Cam_Cause_Families',
'Cam_Cause_Health', 'Cam_Cause_Humanitarian',
'Cam_Cause_Social_Service', 'Cam_Cause_Sports', 'Cam_Cause_Women_Girls'
]
extract_data = combined_data[Need_variable]
extract_data
```

Out[220]:

	Actual_Donation_Amount	NPO_Tax_Deductibility	Distinct_Donors	Campaign_Goal
0	5561.0	True	66	50000
1	2810.0	True	32	20000
2	1118.0	True	22	30000
3	2800.0	True	7	2000
4	2030.0	True	27	5000
...	...	...	...	...
15974	10.0	True	1	5000
15975	150.0	True	4	10000
15976	1000.0	True	10	1000
15977	120.0	True	2	3000
15978	120.0	True	2	40000

15979 rows × 51 columns

"Actual_Donation_Amount" "Campaign_Video" "Impact_Message1" "Impact_Message2" "Impact_Message3" "Impact_Message4" and "Impact_Message5" are many missing data, fill them first so that it's more convenient to operate.

In [221…

```
extract_data['NPO_Tax_Deductibility'] = extract_data['NPO_Tax_Deductibility'
extract_data['Campaign_Start_Date'] = extract_data['Campaign_Start_Date'].fi
extract_data['Campaign_End_Date'] = extract_data['Campaign_End_Date'].fillna
extract_data['Actual_Donation_Amount'] = extract_data['Actual_Donation_Amoun
extract_data['Actual_Donation_Amount'] = pd.to_numeric( extract_data['Actual
extract_data['Distinct_Donors'] = extract_data['Distinct_Donors'].fillna('0'
extract_data['Distinct_Donors'] = pd.to_numeric( extract_data['Distinct_Donc
extract_data['Campaign_Video'] = extract_data['Campaign_Video'].fillna('0')
extract_data['Impact_Message1'] = extract_data['Impact_Message1'].fillna('0'
extract_data['Impact_Message2'] = extract_data['Impact_Message2'].fillna('0'
extract_data['Impact_Message3'] = extract_data['Impact_Message3'].fillna('0'
extract_data['Impact_Message4'] = extract_data['Impact_Message4'].fillna('0'
extract_data['Impact_Message5'] = extract_data['Impact_Message5'].fillna('0'
```

In [222…

```
extract_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 15979 entries, 0 to 15978
Data columns (total 51 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Actual_Donation_Amount    15979 non-null  float64
 1   NPO_Tax_Deductibility     15979 non-null  object 
 2   Distinct_Donors           15979 non-null  int64  
 3   Campaign_Goal             15979 non-null  int64  
 4   Campaign_Start_Date       15979 non-null  object 
 5   Campaign_End_Date         15979 non-null  object 
 6   Campaign_Image1           15979 non-null  int64  
 7   Campaign_Image2           15979 non-null  int64  
 8   Campaign_Image3           15979 non-null  int64  
 9   Campaign_Image4           15979 non-null  int64  
 10  Campaign_Image5           15979 non-null  int64  
 11  Campaign_Video            15979 non-null  object 
 12  Impact_Message1           15979 non-null  object 
 13  Impact_Message2           15979 non-null  object 
 14  Impact_Message3           15979 non-null  object 
 15  Impact_Message4           15979 non-null  object 
 16  Impact_Message5           15979 non-null  object 
 17  Custom_Amount1            15979 non-null  int64  
 18  Custom_Amount2            15979 non-null  int64  
 19  Custom_Amount3            15979 non-null  int64  
 20  Custom_Amount4            15979 non-null  int64  
 21  Description_Campaign      15971 non-null  object 
 22  Description_NPO           13270 non-null  object 
 23  Org_Cause_Animal_Welfare  15979 non-null  object 
 24  Org_Cause_Arts_Heritage   15979 non-null  object 
 25  Org_Cause_Children_Youth  15979 non-null  object 
 26  Org_Cause_Community       15979 non-null  object 
 27  Org_Cause_Disability      15979 non-null  object 
 28  Org_Cause_Education       15979 non-null  object 
 29  Org_Cause_Elderly         15979 non-null  object 
 30  Org_Cause_Environment     15979 non-null  object 
 31  Org_Cause_Families        15979 non-null  object 
 32  Org_Cause_Health          15979 non-null  object 
 33  Org_Cause_Humanitarian    15979 non-null  object 
 34  Org_Cause_Social Service  15979 non-null  object 
 35  Org_Cause_Sports          15979 non-null  object 
 36  Org_Cause_Women_Girls     15979 non-null  object 
 37  Cam_Cause_Animal_Welfare  15979 non-null  object 
 38  Cam_Cause_Arts_Heritage   15979 non-null  object 
 39  Cam_Cause_Children_Youth  15979 non-null  object 
 40  Cam_Cause_Community       15979 non-null  object 
 41  Cam_Cause_Disability      15979 non-null  object 
 42  Cam_Cause_Education       15979 non-null  object 
 43  Cam_Cause_Elderly         15979 non-null  object 
 44  Cam_Cause_Environment     15979 non-null  object 
 45  Cam_Cause_Families        15979 non-null  object 
 46  Cam_Cause_Health          15979 non-null  object 
 47  Cam_Cause_Humanitarian    15979 non-null  object 
 48  Cam_Cause_Social_Service  15979 non-null  object 
 49  Cam_Cause_Sports          15979 non-null  object 
 50  Cam_Cause_Women_Girls     15979 non-null  object 
dtypes: float64(1), int64(11), object(39)
memory usage: 6.3+ MB
```

There is no donations per donor, So add a columns of

donations per donor

In [223…
```
extract_data.columns
```

Out[223]:
```
Index(['Actual_Donation_Amount', 'NPO_Tax_Deductibility', 'Distinct_Donor
s',
       'Campaign_Goal', 'Campaign_Start_Date', 'Campaign_End_Date',
       'Campaign_Image1', 'Campaign_Image2', 'Campaign_Image3',
       'Campaign_Image4', 'Campaign_Image5', 'Campaign_Video',
       'Impact_Message1', 'Impact_Message2', 'Impact_Message3',
       'Impact_Message4', 'Impact_Message5', 'Custom_Amount1',
       'Custom_Amount2', 'Custom_Amount3', 'Custom_Amount4',
       'Description_Campaign', 'Description_NPO', 'Org_Cause_Animal_Welfar
e',
       'Org_Cause_Arts_Heritage', 'Org_Cause_Children_Youth',
       'Org_Cause_Community', 'Org_Cause_Disability', 'Org_Cause_Educatio
n',
       'Org_Cause_Elderly', 'Org_Cause_Environment', 'Org_Cause_Families',
       'Org_Cause_Health', 'Org_Cause_Humanitarian',
       'Org_Cause_Social Service', 'Org_Cause_Sports', 'Org_Cause_Women_Gi
rls',
       'Cam_Cause_Animal_Welfare', 'Cam_Cause_Arts_Heritage',
       'Cam_Cause_Children_Youth', 'Cam_Cause_Community',
       'Cam_Cause_Disability', 'Cam_Cause_Education', 'Cam_Cause_Elderly',
       'Cam_Cause_Environment', 'Cam_Cause_Families', 'Cam_Cause_Health',
       'Cam_Cause_Humanitarian', 'Cam_Cause_Social_Service',
       'Cam_Cause_Sports', 'Cam_Cause_Women_Girls'],
      dtype='object')
```

In [225…
```
# I am not sure Distinct_Donors is the total donors or not ?
num_deductibility = 0
extract_data['Donation_per_donor'] = 0

for j in range(len(extract_data["Actual_Donation_Amount"])):
    if extract_data["Distinct_Donors"].iloc[j] != 0:
        extract_data['Donation_per_donor'].iloc[j] = extract_data['Actual_Do
    else:
        extract_data['Donation_per_donor'].iloc[j] = 0

    if extract_data['NPO_Tax_Deductibility'].iloc[j] == True:
        extract_data.loc[j, 'NPO_Tax_Deductibility'] = 1
        num_deductibility += 1
    else:
        extract_data.loc[j, 'NPO_Tax_Deductibility'] = 0
print("Number of deductibility:", num_deductibility)
```

```
Number of deductibility: 14998
```

Here is the Number of deducbility:
14998/15979 = 93.86%

Sum the numbers of org_causes and camp_causes

In [226…
```
Org_causes = ['Org_Cause_Animal_Welfare', 'Org_Cause_Arts_Heritage',
       'Org_Cause_Children_Youth', 'Org_Cause_Community',
       'Org_Cause_Disability', 'Org_Cause_Education', 'Org_Cause_Elderly',
```

```
       'Org_Cause_Environment', 'Org_Cause_Families', 'Org_Cause_Health',
       'Org_Cause_Humanitarian', 'Org_Cause_Social Service',
       'Org_Cause_Sports', 'Org_Cause_Women_Girls', 'Cam_Cause_Animal_Welfar
       ]
Cam_causes = ['Cam_Cause_Arts_Heritage', 'Cam_Cause_Children_Youth',
       'Cam_Cause_Community', 'Cam_Cause_Disability', 'Cam_Cause_Education',
       'Cam_Cause_Elderly', 'Cam_Cause_Environment', 'Cam_Cause_Families',
       'Cam_Cause_Health', 'Cam_Cause_Humanitarian',
       'Cam_Cause_Social_Service', 'Cam_Cause_Sports',
       'Cam_Cause_Women_Girls']
Length_Org_causes = len(Org_causes)
Length_Cam_causes = len(Cam_causes)
extract_data['Org_causes'] = 0
extract_data['Cam_causes'] = 0

for j in range(Total_Rows):
    num_Org_causes = 0
    num_Cam_causes = 0
    for position1 in range(Length_Org_causes):
        num_Org_causes += 1 if extract_data[Org_causes[position1]].iloc[j] !
    extract_data['Org_causes'].iloc[j] = num_Org_causes
    for position2 in range(Length_Cam_causes):
        num_Cam_causes += 1 if extract_data[Cam_causes[position2]].iloc[j] !
    extract_data['Org_causes'].iloc[j] = num_Org_causes
    extract_data['Cam_causes'].iloc[j] = num_Cam_causes
```

Add a columns of numbers of images

In [227…
```
Add_Campaign_Image_num = lambda x0,x1,x2,x3,x4: (x0 != 0).astype(np.int) +(x
extract_data["Campaign_Image_num"] = Add_Campaign_Image_num(extract_data["Ca
```

Classfy video into “0” and ”1“ two categories

In [228…
```
Video_or_not = lambda x0: (x0 != '0').astype(np.int)
extract_data["Campaign_Video"] = Video_or_not(extract_data["Campaign_Video"]
```

The format of the date needs to be modified and the duration will be calculated below

In [229…
```
month_dictionary = {'Jan':'1',
 'Feb':'2',
 'Mar':'3',
 'Apr':'4',
 'May':'5',
 'Jun':'6',
 'Jul':'7',
 'Aug':'8',
 'Sep':'9',
 'Oct':'10',
 'Nov':'11',
 'Dec':'12'}
extract_data['Campaign_Start_Day'] = '0'
extract_data['Campaign_Start_Month'] = '0'
extract_data['Campaign_Start_Year'] = '0'
extract_data['Campaign_End_Day'] = '0'
```

```
extract_data['Campaign_End_Month'] = '0'
extract_data['Campaign_End_Year'] = '0'
extract_data['Campaign_Start'] = '0'
extract_data['Campaign_End'] = '0'
extract_data['duration_day'] = '0'
i = 0

for row in extract_data['Campaign_Start_Date']:
    extract_data.loc[i, 'Campaign_Start_Day'] = extract_data.loc[i, 'Campaig
    extract_data.loc[i, 'Campaign_Start_Month'] = month_dictionary[ extract_
    extract_data.loc[i, 'Campaign_Start_Year'] = '20'+ extract_data.loc[i, '
    extract_data.loc[i, 'Campaign_End_Day'] = extract_data.loc[i, 'Campaign_
    extract_data.loc[i, 'Campaign_End_Month'] = month_dictionary[extract_dat
    extract_data.loc[i, 'Campaign_End_Year'] = '20' + extract_data.loc[i, 'C
    extract_data.loc[i, 'Campaign_Start'] = extract_data.loc[i, 'Campaign_St
    extract_data.loc[i, 'Campaign_End'] = extract_data['Campaign_End_Year'].
    extract_data.loc[i, 'duration_day'] = (datetime.strptime(extract_data.lo
    #if extract_data.loc[i, 'duration_day'] < 0:
    #    extract_data.loc[i, 'duration_day'] = 0
    i += 1
```

Here delete the rows where duration_day less than 0

In [230…

```
num = 0
for index, row in extract_data.iterrows():
    if extract_data.loc[index,'duration_day'] <= 0:
        extract_data.drop(index, inplace=True)
        num += 1
print("delete numbers", num)
Total_Rows = extract_data.shape[0]
```

```
delete numbers 106
```

See more infomation about every columns

Check whether there are missing data

Divide 5 messages into one category

In [231…

```
#extract_data['Msg1_category'] = 0
#extract_data['Msg2_category'] = 0
#extract_data['Msg3_category'] = 0
#extract_data['Msg4_category'] = 0
#extract_data['Msg5_category'] = 0
#
extract_data['Msg_category'] = 0

Impact_msg_list = ['Impact_Message1','Impact_Message2','Impact_Message3','Im
#Msg_category_list = ['Msg1_category','Msg2_category','Msg3_category','Msg4_
def sentence_length(s):
    return len([i for i in s.split(' ') if i])

#for j in range(len(Impact_msg_list)):
```

```
#    cnt=0
#    for s in extract_data[Impact_msg_list[j]]:
#        extract_data[Msg_category_list[j]].iloc[cnt] = 0 if sentence_length
#        cnt += 1
cnt=0

for index,row in extract_data.iterrows():
    last_category = 0
    current_category = 0
    final_category = 0
    for index_col in Impact_msg_list:
        s = extract_data.loc[index, index_col]
        if(sentence_length(s)<=2): # Write nothing
            current_category = 0
        else:
            if(sentence_length(s)<=10): # Write very little
                current_category = 1
            else:
                if(sentence_length(s)<20): # Write very little
                    current_category = 2
                else:
                    current_category = 3
        if(current_category == last_category):
            last_category = current_category
        else:
            last_category = current_category if current_category>last_catego
    #Msg_length = sentence_length(s)
    extract_data.loc[index,'Msg_category'] =  last_category#0 if sentence_le
    cnt += 1
```

In [232…

```
# Nunber of description words
extract_data['Num_desc_cam'] = 0
extract_data['Num_desc_NPO'] = 0
for index, row in extract_data.iterrows():
    extract_data.loc[index, 'Num_desc_cam'] = sentence_length( str(extract_d
    extract_data.loc[index, 'Num_desc_NPO'] = sentence_length( str(extract_d
extract_data
```

Out[232]:

	Actual_Donation_Amount	NPO_Tax_Deductibility	Distinct_Donors	Campaign_Goal
0	5561.0	1	66	50000
1	2810.0	1	32	20000
2	1118.0	1	22	30000
3	2800.0	1	7	2000
4	2030.0	1	27	5000
...	...	...	...	...
15974	10.0	1	1	5000
15975	150.0	1	4	10000
15976	1000.0	1	10	1000
15977	120.0	1	2	3000
15978	120.0	1	2	40000

15873 rows × 67 columns

Convert to numeric type

In [234…
```
extract_data['duration_day'] = pd.to_numeric( extract_data['duration_day'])
#extract_data['Total_Msg_polarity'] = pd.to_numeric( extract_data['Total_Msg
extract_data['NPO_Tax_Deductibility'] = pd.to_numeric(extract_data['NPO_Tax_
```

In [235…
```
#numeric_features Store the following variables that need to draw correlatic
numeric_feature = ['Actual_Donation_Amount',  'Campaign_Goal', 'duration_day
                   'Campaign_Image_num',  'Campaign_Video',
                   'Num_desc_cam', 'Num_desc_NPO' ]

numeric_features1 = ['Actual_Donation_Amount', 'Campaign_Goal', 'NPO_Tax_Ded
                   'duration_day', 'Campaign_Video', 'Msg_category',
                    'Campaign_Image_num','Num_desc_cam', 'Num_desc_NPO', 'C

numeric_features2 = ['Actual_Donation_Amount', 'Campaign_Goal', 'NPO_Tax_Ded
                     'duration_day','Campaign_Image_num', 'Campaign_Video',
                     'Num_desc_NPO', 'Org_causes', 'Cam_causes','Custom_Amo
                     'Custom_Amount3', 'Custom_Amount4']

#Correlation analysis
price_numeric = extract_data[numeric_feature]
correlation = price_numeric.corr()
y_train = Orginal_data['Actual_Donation_Amount']
corr = plt.subplots(figsize = (18,16), dpi=128)
corr= sns.heatmap(price_numeric.assign(Y=y_train).corr(method='spearman'), a
```

Variance inflation factor (Two methods to test make sure they are right)

In [236…
```
def vif(df, col_i):
        from statsmodels.formula.api import ols
        cols = list(df.columns)
        cols.remove(col_i)
        cols_noti = cols
        formula = col_i + '~' + '+'.join(cols_noti)
        r2 = ols(formula, df).fit().rsquared
        return 1.0 / (1.0 - r2)

test_data = extract_data[numeric_features2]
for i in numeric_features2:
        print(i, "\t", vif(df=test_data, col_i=i))
```

```
Actual_Donation_Amount   1.4607270504837995
Campaign_Goal    1.5606029814038935
NPO_Tax_Deductibility    1.0230017008899082
duration_day     1.1031988461197293
Campaign_Image_num       1.0806321625039117
Campaign_Video   1.0554341535826517
Msg_category     1.1483663145873253
Num_desc_cam     1.0639399773065488
Num_desc_NPO     2.3239655321198116
Org_causes       2.424761098456672
Cam_causes       1.155119043819482
Custom_Amount1   1.638546087942614
Custom_Amount2   14.204521143982552
Custom_Amount3   40.63111497814649
Custom_Amount4   34.43555194014789
```

Based on the result only the "Custom_Amount1-4" 's multi collinearity is high. Other variables seem reasonable.

In [237…
```
from statsmodels.stats.outliers_influence import variance_inflation_factor
from statsmodels.tools.tools import add_constant
X = add_constant(test_data)
ds=pd.Series([variance_inflation_factor(X.values, i)
               for i in range(X.shape[1])],
              index=X.columns)
print(ds)
```

```
const                      38.706812
Actual_Donation_Amount      1.460727
Campaign_Goal               1.560603
NPO_Tax_Deductibility       1.023002
duration_day                1.103199
Campaign_Image_num          1.080632
Campaign_Video              1.055434
Msg_category                1.148366
Num_desc_cam                1.063940
Num_desc_NPO                2.323966
Org_causes                  2.424761
Cam_causes                  1.155119
Custom_Amount1              1.638546
Custom_Amount2             14.204521
Custom_Amount3             40.631115
Custom_Amount4             34.435552
dtype: float64
```

Modeling verfication

Model1 'Org_causes' 'Cam_causes'

In [238…
```
variable_list1 = numeric_feature #['Actual_Donation_Amount', 'Campaign_Goal'
                 #'Campaign_Image_num', 'Num_desc_cam', 'Num_desc_NPO' ]
variables_data1 = extract_data[variable_list1]
```

Model2

In [239…
```
variable_list2 = numeric_features1 #['Actual_Donation_Amount', 'NPO_Tax_Dedu
                  #   'Msg_category', 'Num_desc_cam', 'Num_desc_NPO' ]
variables_data2 = extract_data[variable_list2]
```

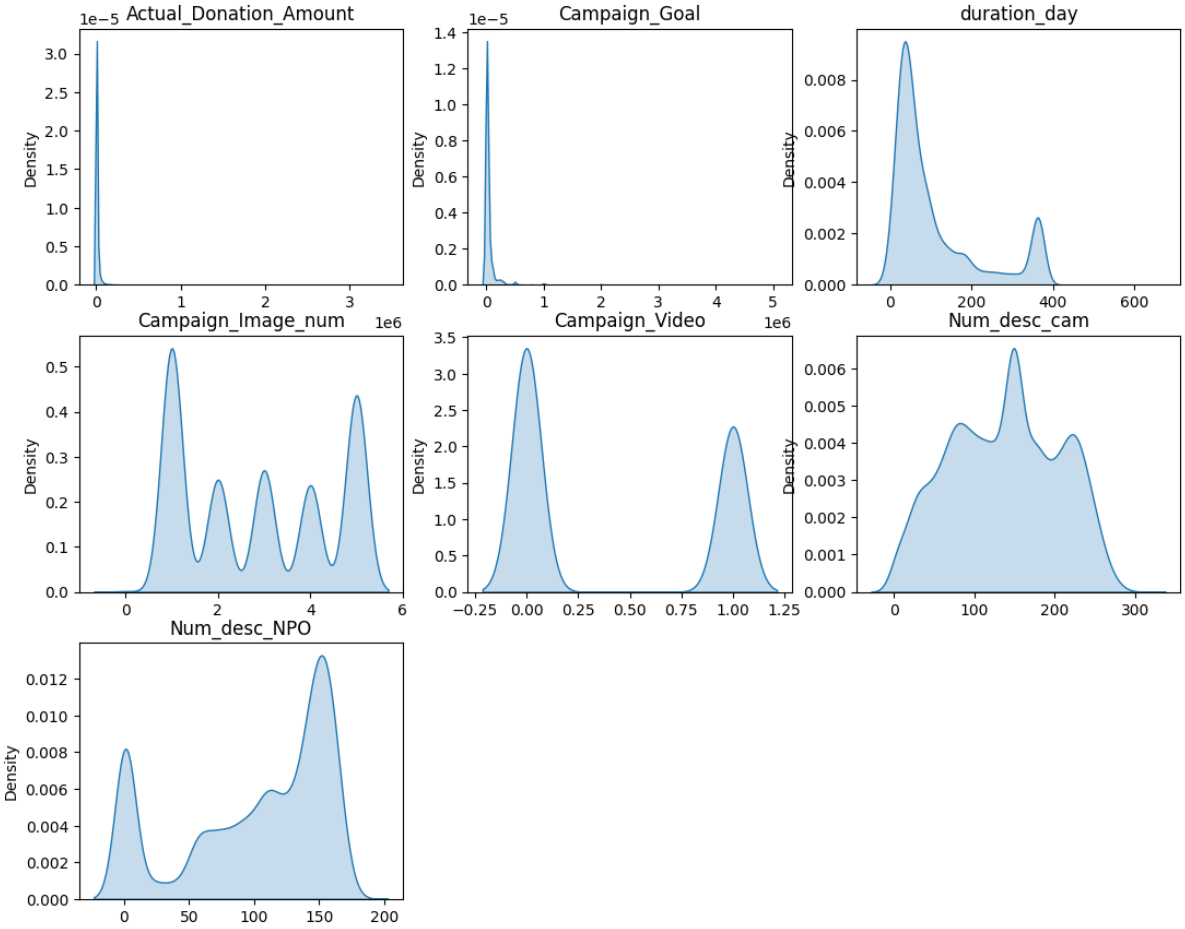
Model3

In [240…
```
variable_list3 = numeric_features2 #['Actual_Donation_Amount',  'Campaign_Go
                                   #   'Campaign_Video', 'Msg_category', 'N
variables_data3 = extract_data[variable_list3]
```

Variance, Average, Max, Min, Median calculation

In [241…
```
i = 0
plt.figure(figsize=(13, 14))
plt.xticks([])
for title in variable_list1:
    plt.subplot(4,3,i+1)
    plt.title(title)
    sns.kdeplot(extract_data[title], shade=True)
    plt.xlabel(" ")
    i += 1


#plt.hist(extract_data['Campaign_Goal'], bins=80, histtype="stepfilled", alp
```



In [242…

```
for title in variable_list2:
    extract_data[title] = pd.to_numeric( extract_data[title])
    print( title, "Average:",np.average(extract_data[title]))
    print( title, "Variance:" ,np.var(extract_data[title]))
    print( title, "Min:" ,np.min(extract_data[title]))
    print( title, "Max:" ,np.max(extract_data[title]))
    print( title, "Median:",np.median(extract_data[title]))
```

```
Actual_Donation_Amount Average: 9877.115731115731
Actual_Donation_Amount Variance: 3992094033.9647455
Actual_Donation_Amount Min: 0.0
Actual_Donation_Amount Max: 3431670.0
Actual_Donation_Amount Median: 1310.0
Campaign_Goal Average: 44845.93914193914
Campaign_Goal Variance: 23695726676.82714
Campaign_Goal Min: 100
Campaign_Goal Max: 5000000
Campaign_Goal Median: 5000.0
NPO_Tax_Deductibility Average: 0.9388899388899389
NPO_Tax_Deductibility Variance: 0.05737562154118571
NPO_Tax_Deductibility Min: 0
NPO_Tax_Deductibility Max: 1
NPO_Tax_Deductibility Median: 1.0
duration_day Average: 108.46355446355446
duration_day Variance: 12086.688412162612
duration_day Min: 1
duration_day Max: 630
duration_day Median: 60.0
Campaign_Video Average: 0.4042084042084042
Campaign_Video Variance: 0.24082397017569954
Campaign_Video Min: 0
Campaign_Video Max: 1
Campaign_Video Median: 0.0
Msg_category Average: 1.1348201348201348
Msg_category Variance: 1.5551871046106542
Msg_category Min: 0
Msg_category Max: 3
Msg_category Median: 1.0
Campaign_Image_num Average: 2.8696528696528696
Campaign_Image_num Variance: 2.5253771679778105
Campaign_Image_num Min: 0
Campaign_Image_num Max: 5
Campaign_Image_num Median: 3.0
Num_desc_cam Average: 137.56561456561457
Num_desc_cam Variance: 4554.391980875065
Num_desc_cam Min: 1
Num_desc_cam Max: 309
Num_desc_cam Median: 144.0
Num_desc_NPO Average: 101.07163107163107
Num_desc_NPO Variance: 3146.5967085914167
Num_desc_NPO Min: 1
Num_desc_NPO Max: 179
Num_desc_NPO Median: 115.0
Org_causes Average: 3.0334530334530334
Org_causes Variance: 2.3598082554801514
Org_causes Min: 0
Org_causes Max: 5
Org_causes Median: 4.0
Cam_causes Average: 3.455994455994456
Cam_causes Variance: 0.963870227903492
Cam_causes Min: 0
Cam_causes Max: 4
Cam_causes Median: 4.0
```

The Linear regression of selected variables Model 1

In [243…

```
import statsmodels.formula.api as smf
```

```
model = smf.ols(formula = 'Actual_Donation_Amount ~ Campaign_Goal  + duratic
    Campaign_Image_num  + Campaign_Video + Num_desc_cam + Num_desc_NPO', dat

results1 = model.summary()
predicts = model._results
print(results1)
```

```
                            OLS Regression Results
==============================================================================
=======
Dep. Variable:     Actual_Donation_Amount   R-squared:
0.308
Model:                                OLS   Adj. R-squared:
0.308
Method:                     Least Squares   F-statistic:
1180.
Date:                    Wed, 16 Nov 2022   Prob (F-statistic):
0.00
Time:                            22:11:14   Log-Likelihood:            -1.9
505e+05
No. Observations:                   15873   AIC:                         3.
901e+05
Df Residuals:                       15866   BIC:                         3.
902e+05
Df Model:                               6
Covariance Type:                nonrobust
==============================================================================
===========
                         coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-----------
Intercept          -4149.3718   1400.602     -2.963      0.003   -6894.711
-1404.033
Campaign_Goal          0.2290      0.003     82.825      0.000       0.224
0.234
duration_day         -17.1478      3.920     -4.374      0.000     -24.832
-9.463
Campaign_Image_num  -203.1075    270.361     -0.751      0.453    -733.045
326.830
Campaign_Video      2361.5457    860.060      2.746      0.006     675.730
4047.362
Num_desc_cam          13.3409      6.224      2.144      0.032       1.142
25.540
Num_desc_NPO          33.7381      7.748      4.355      0.000      18.551
48.925
==============================================================================
===
Omnibus:                    35397.741   Durbin-Watson:                   1.
942
Prob(Omnibus):                  0.000   Jarque-Bera (JB):        493131896.
805
Skew:                          20.455   Prob(JB):
0.00
Kurtosis:                     865.521   Cond. No.                     5.56e
+05
==============================================================================
===

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is corr
ectly specified.
[2] The condition number is large, 5.56e+05. This might indicate that there
are
strong multicollinearity or other numerical problems.
```

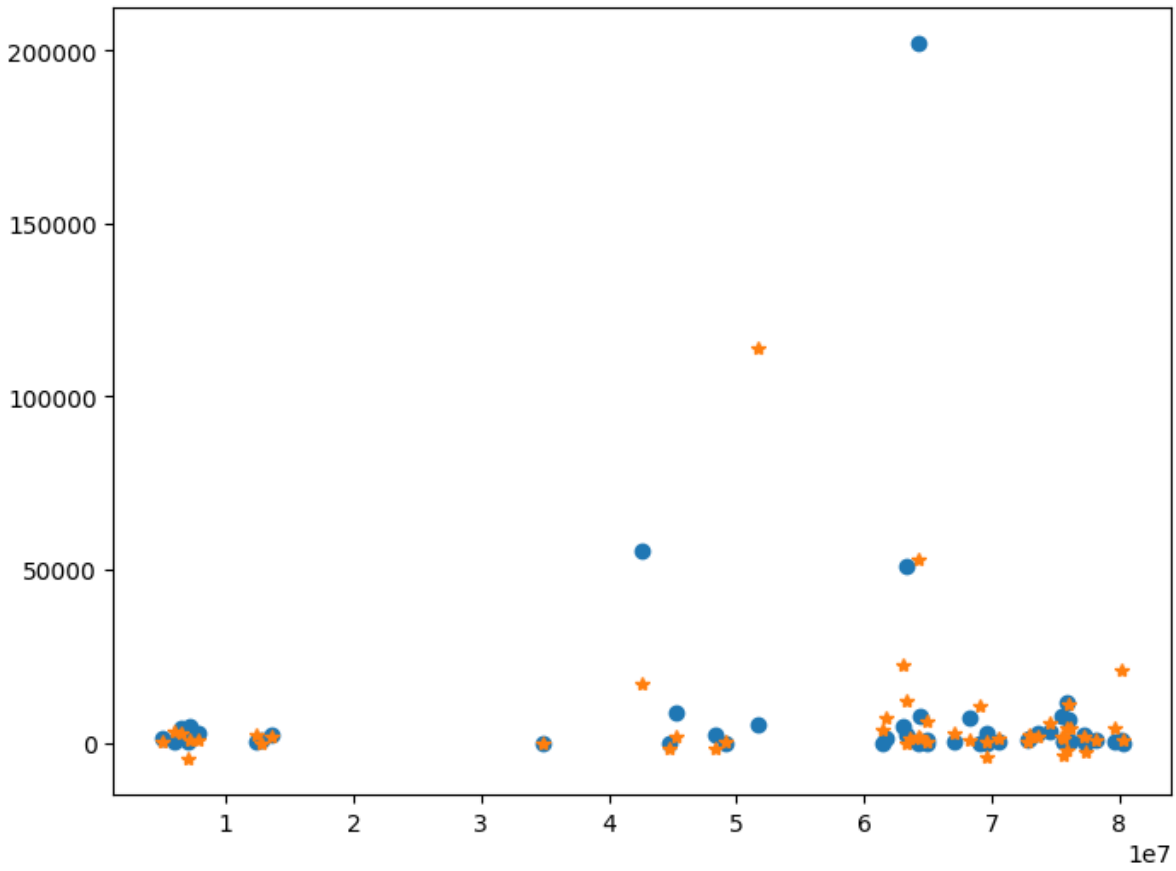
Randomly choose 50 points of prediction and actual data to

compare

Circle is actual donation star is regression result

In [244…
```
from random import sample
mysample = sample(range(0,Total_Rows), 50)
x = combined_data['Campaign_ID'][mysample]
y = extract_data['Actual_Donation_Amount'][mysample]
y_fitted = model.fittedvalues
fig, ax = plt.subplots(figsize=(8,6))
ax.plot(x, y, 'o', label='data')
ax.plot(x, y_fitted[mysample],'*',label='OLS')
```

Out[244]: [<matplotlib.lines.Line2D at 0x7fb742b23d68>]



Test normality.

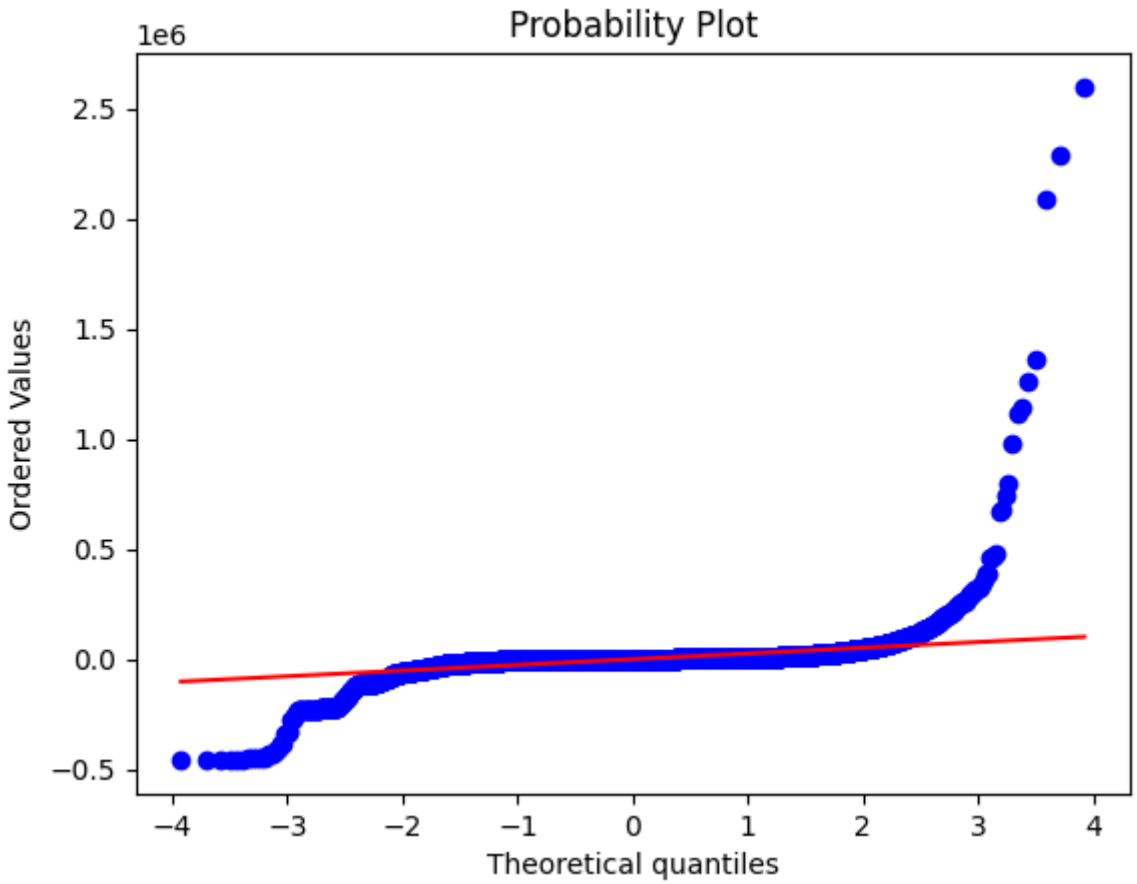
In [245…
```
import openturns as ot
from statsmodels.stats.diagnostic import lilliefors
model_resid = model.resid
result = lilliefors(list(model_resid))
print(result)
```

```
(0.3234836697207625, 0.0009999999999998899)
```

In [246…
```
# Example of the Anderson-Darling Normality Test
from scipy.stats import anderson
result = anderson(list(model_resid), dist='norm')
print('stat=%.3f' % (result.statistic))
print('significance_level:', (result.significance_level))
```

```
stat=3446.816
significance_level: [15.  10.   5.   2.5  1. ]
```

In [247…
```
#stats.probplot(sample, dist=stats.norm, plot=plt)
res = stats.probplot(list(model_resid),  dist=stats.norm, plot=plt)
```



Model 2

In [248…
```
import statsmodels.formula.api as smf

model2 = smf.ols(formula = 'Actual_Donation_Amount ~ Campaign_Goal  + NPO_Ta
    Campaign_Image_num  + Campaign_Video + Msg_category+\
     Num_desc_cam + Num_desc_NPO', data = variables_data2).fit()

results2 = model2.summary()
print(results2)
```

OLS Regression Results

Dep. Variable:	Actual_Donation_Amount	R-squared:	0.309
Model:	OLS	Adj. R-squared:	0.309
Method:	Least Squares	F-statistic:	888.2
Date:	Wed, 16 Nov 2022	Prob (F-statistic):	0.00
Time:	22:11:16	Log-Likelihood:	-1.9504e+05
No. Observations:	15873	AIC:	3.901e+05
Df Residuals:	15864	BIC:	3.902e+05
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-5476.9040	2174.328	-2.519	0.012	-9738.833	-1214.975
Campaign_Goal	0.2308	0.003	82.524	0.000	0.225	0.236
NPO_Tax_Deductibility	1899.5230	1751.120	1.085	0.278	-1532.871	5331.917
duration_day	-14.2829	3.971	-3.597	0.000	-22.067	-6.499
Campaign_Image_num	-134.2941	270.980	-0.496	0.620	-665.446	396.858
Campaign_Video	1970.1831	864.123	2.280	0.023	276.404	3663.963
Msg_category	-1526.7151	357.467	-4.271	0.000	-2227.391	-826.039
Num_desc_cam	19.2753	6.368	3.027	0.002	6.794	31.756
Num_desc_NPO	34.0062	7.773	4.375	0.000	18.769	49.243

Omnibus:	35337.317	Durbin-Watson:	1.943
Prob(Omnibus):	0.000	Jarque-Bera (JB):	489754007.723
Skew:	20.369	Prob(JB):	0.00
Kurtosis:	862.563	Cond. No.	1.01e+06

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.01e+06. This might indicate that there

```
are
strong multicollinearity or other numerical problems.
```

In [249…
```
model_resid2 = model2.resid
result = lilliefors(list(model_resid2))
print(result)
```

```
(0.32829910268106566, 0.0009999999999998899)
```

In [250…
```
variables_data3
```

Out[250]:

	Actual_Donation_Amount	Campaign_Goal	NPO_Tax_Deductibility	duration_day	Ca
0	5561.0	50000	1	252	
1	2810.0	20000	1	89	
2	1118.0	30000	1	58	
3	2800.0	2000	1	88	
4	2030.0	5000	1	50	
...	...	...	...	...	
15974	10.0	5000	1	62	
15975	150.0	10000	1	30	
15976	1000.0	1000	1	30	
15977	120.0	3000	1	61	
15978	120.0	40000	1	117	

15873 rows × 15 columns

Model 3

In [251…
```
variables_data3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 15873 entries, 0 to 15978
Data columns (total 15 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Actual_Donation_Amount  15873 non-null  float64
 1   Campaign_Goal           15873 non-null  int64
 2   NPO_Tax_Deductibility   15873 non-null  int64
 3   duration_day            15873 non-null  int64
 4   Campaign_Image_num      15873 non-null  int64
 5   Campaign_Video          15873 non-null  int64
 6   Msg_category            15873 non-null  int64
 7   Num_desc_cam            15873 non-null  int64
 8   Num_desc_NPO            15873 non-null  int64
 9   Org_causes              15873 non-null  int64
 10  Cam_causes              15873 non-null  int64
 11  Custom_Amount1          15873 non-null  int64
 12  Custom_Amount2          15873 non-null  int64
 13  Custom_Amount3          15873 non-null  int64
 14  Custom_Amount4          15873 non-null  int64
dtypes: float64(1), int64(14)
memory usage: 2.6 MB
```

In [252…

```
model3 = smf.ols(formula = 'Actual_Donation_Amount ~ Campaign_Goal  + NPO_Ta
    Campaign_Image_num  + Campaign_Video + Msg_category+\
    Num_desc_cam + Num_desc_NPO + Org_causes + Cam_causes', data = variables

results3 = model3.summary()
print(results3)
```

OLS Regression Results

Dep. Variable:	Actual_Donation_Amount	R-squared:	0.311
Model:	OLS	Adj. R-squared:	0.310
Method:	Least Squares	F-statistic:	715.4
Date:	Wed, 16 Nov 2022	Prob (F-statistic):	0.00
Time:	22:11:16	Log-Likelihood:	-1.9503e+05
No. Observations:	15873	AIC:	3.901e+05
Df Residuals:	15862	BIC:	3.902e+05
Df Model:	10		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-6913.8169	2588.905	-2.671	0.008	-1.2e+04	-1839.270
Campaign_Goal	0.2305	0.003	82.452	0.000	0.225	0.236
NPO_Tax_Deductibility	2167.1466	1756.816	1.234	0.217	-1276.412	5610.705
duration_day	-14.8688	3.974	-3.741	0.000	-22.659	-7.079
Campaign_Image_num	-244.7545	272.368	-0.899	0.369	-778.627	289.118
Campaign_Video	2494.5029	871.401	2.863	0.004	786.457	4202.549
Msg_category	-1431.5808	357.540	-4.004	0.000	-2132.399	-730.762
Num_desc_cam	18.2633	6.363	2.870	0.004	5.790	30.736
Num_desc_NPO	-12.9861	11.316	-1.148	0.251	-35.167	9.195
Org_causes	2418.5795	421.660	5.736	0.000	1592.077	3245.082
Cam_causes	-342.7775	455.861	-0.752	0.452	-1236.317	550.762

Omnibus:	35349.036	Durbin-Watson:	1.945
Prob(Omnibus):	0.000	Jarque-Bera (JB):	490668088.832
Skew:	20.385	Prob(JB):	0.00
Kurtosis:	863.365	Cond. No.	1.10e+06

Notes:

```
[1] Standard Errors assume that the covariance matrix of the errors is corr
ectly specified.
[2] The condition number is large, 1.1e+06. This might indicate that there
are
strong multicollinearity or other numerical problems.
```

Residuals

In [253…
```
model.resid
```

Out[253]:
```
0        -6411.307333
1        -6370.989712
2       -10164.110968
3         3777.481846
4          100.908581
              ...
15974    -1669.528960
15975     -346.144204
15976     2564.808072
15977      923.325187
15978    -4963.810834
Length: 15873, dtype: float64
```