Project Development Phase
Model Performance Test

Date	18 October 2023
Team ID	NM2023TMID05638
Project Name	A Comprehensive Analysis of Social Media
Maximum Marks	10 Marks

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Screenshot / Values
1.	Dashboard design	No of Visualizations / Graphs -
2.	Data Responsiveness	
3.	Amount Data to Rendered (DB2 Metrics)	
4.	Utilization of Data Filters	
5.	Effective User Story	No of Scene Added -
6.	Descriptive Reports	No of Visulizations / Graphs -

October 18, 2023

[ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.animation as animation
import seaborn as sns
```

[ ]:
```
df = pd.read_csv('/content/House Price India.csv')
```

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

[ ]:
```
           id   Date  number of bedrooms  number of bathrooms  living area  \
0  6762810145  42491                   5                 2.50         3650
1  6762810635  42491                   4                 2.50         2920
2  6762810998  42491                   5                 2.75         2910
3  6762812605  42491                   4                 2.50         3310
4  6762812919  42491                   3                 2.00         2710
   lot area number of floors waterfront present number of views \
0      9050  2.0   0     4
1      4000  1.5   0     0
2      9480  1.5   0     0
3      42998 2.0   0     0 4   4500  1.5   0     0

   condition of the house … Built Year Renovation Year Postal Code \
0                    5 …    1921  0    122003
1                    5 …    1909  0    122004
2                    3 …    1939  0    122004
3                    3 …    2001  0    122005
4                    4 …    1929  0    122006

       Lattitude Longitude living_area_renov lot_area_renov \
0    52.8645     -114.557   2880  5400
1    52.8878     -114.470   2470  4000
2    52.8852     -114.468   2940  6600
3    52.9532     -114.321   3350  42847
4    52.9047     -114.485   2060  4500

    Number of schools nearby Distance from the airport Price
    0                     2                       58
                                                  2380000
```

```
1                      2                      51
                                              1400000
2                      1                      53
                                              1200000
3                      3                      76  838000
4                      1                      51  805000
[5 rows x 23 columns]
```

[ ]:
```
df.describe()
```

[ ]:
```
             id number of bedrooms number of bathrooms living area \
count 1.462000e+04     14620.000000      14620.000000
                                         14620.000000
mean   6.762821e+09         3.379343          2.129583 2098.262996
std    6.237575e+03         0.938719          0.769934  928.275721
min    6.762810e+09         1.000000          0.500000  370.000000
25%    6.762815e+09         3.000000          1.750000 1440.000000
50%    6.762821e+09         3.000000          2.250000 1930.000000
75%    6.762826e+09         4.000000          2.500000 2570.000000
max    6.762832e+09        33.000000          8.000000 13540.000000
     lot area number of floors waterfront present number of views \
count  1.462000e+04  14620.000000  14620.000000  14620.000000  mean
1.509328e+04 1.502360 0.007661 0.233105
std    3.791962e+04       0.540239          0.087193       0.766259
min    5.200000e+02       1.000000          0.000000       0.000000
25%    5.010750e+03       1.000000          0.000000       0.000000
50%    7.620000e+03       1.500000          0.000000       0.000000
75%    1.080000e+04       2.000000          0.000000       0.000000
max    1.074218e+06       3.500000          1.000000       4.000000

      condition of the house grade of the house …Built Year \
count         14620.000000     14620.000000 … 14620.000000
mean              3.430506         7.682421 …  1970.926402
std               0.664151         1.175033 …    29.493625
min               1.000000         4.000000 …  1900.000000
25%               3.000000         7.000000 …  1951.000000
50%               3.000000         7.000000 …  1975.000000
75%               4.000000         8.000000 …  1997.000000
max               5.000000        13.000000 …  2015.000000

      Renovation Year Postal Code   Lattitude    Longitude \
count   14620.00000014620.000000 14620.000000 14620.000000
mean       90.924008 122033.062244   52.792848 -114.404007
std       416.216661    19.082418    0.137522    0.141326
min         0.000000 122003.000000   52.385900 -114.709000
25%         0.000000 122017.000000   52.707600 -114.519000
50%         0.000000 122032.000000   52.806400 -114.421000
75%         0.000000 122048.000000   52.908900 -114.315000
```

```
max       2015.000000  122072.000000  53.007600  -113.505000
       living_area_renov  lot_area_renov  Number  of  schools
       nearby \
count     14620.000000  14620.000000           14620.000000
mean       1996.702257  12753.500068              2.012244
std         691.093366  26058.414467              0.817284
min         460.000000    651.000000              1.000000
25%        1490.000000   5097.750000              1.000000
50%        1850.000000   7620.000000              2.000000
75%        2380.000000  10125.000000              3.000000
max        6110.000000 560617.000000              3.000000
       Distance from the airport   Price

count             14620.000000 1.462000e+04

mean                 64.950958 5.389322e+05

std                   8.936008 3.675324e+05

min                  50.000000 7.800000e+04

25%                  57.000000 3.200000e+05

50%                  65.000000 4.500000e+05

75%                  73.000000 6.450000e+05

max                  80.000000 7.700000e+06

[8 rows x 22 columns]
```

[ ]:
```
df.shape
```

[ ]: (14620, 23)

DATA CLEANING

[ ]:
```

```

[ ]:
```
print(df.duplicated())
df.dropna(inplace = True)
```

```
0       False
1       False
2       False
3       False
4       False
        …
14615   False
14616   False
```

```
14617   False
14618   False
14619   False
Length: 14620, dtype: bool
```

[ ]:
```
df.drop_duplicates(inplace = True)
```

[ ]:
```
df['Date'] = pd.to_datetime(df['Date'])
```

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

[ ]:

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	condition of the house	…	Built Year	Renovation Year	Postal Code	Lattitude	Longitude	living_area_renov	lot_area_renov	Number of schools nearby
0	6762810145	1970-01-01 00:00:00.000042491	5	2.50	3650	9050	2.0	0	4	5	…	19210	122003	52.8645	-114.557	2880	5400	2	
1	6762810635	1970-01-01 00:00:00.000042491	4	2.50	2920	4000	1.5	0	0	5	…	19090	122004	52.8878	-114.470	2470	4000	2	
2	6762810998	1970-01-01 00:00:00.000042491	5	2.75	2910	9480	1.5	0	0	3	…	19390	122004	52.8852	-114.468				
3	6762812605	1970-01-01 00:00:00.000042491	4	2.50	3310	42998	2.0	0	0	3	…	20010	122005	52.9532	-114.321				
4	6762812919	1970-01-01 00:00:00.000042491	3	2.00	2710	4500	1.5	0	0	4	…	19290	122006	52.9047	-114.485				

```
2              2940    6600  1

3              3350    42847 3

4              2060    4500  1
```

```
      Distance from the airport  Price
    0                        58 2380000
    1                        51 1400000
    2                        53 1200000
    3                        76    838000
    4                        51    805000

    [5 rows x 23 columns]
```

[ ]:
```
df['Date'] = df['Date'].dt.date
```

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

[ ]:
```
           id      Date number of bedrooms number of bathrooms \
0  6762810145 1970-01-01    5     2.50

1  6762810635 1970-01-01    4     2.50

2  6762810998 1970-01-01    5     2.75

3  6762812605 1970-01-01    4     2.50

4  6762812919 1970-01-01    3     2.00

       living area lot area number of floors waterfront present \
0        3650    9050  2.0   0

1        2920    4000  1.5   0

2        2910    9480  1.5   0

3        3310    42998 2.0   0

4        2710    4500  1.5   0

  number of views condition of the house … Built Year Renovation Year \
0              4 5 …   1921  0

1              0 5 …   1909  0

2              0 3 …   1939  0

3              0 3 …   2001  0

4              0 4 …   1929  0

       Postal Code Lattitude Longitude living_area_renov lot_area_renov \
    0       122003     52.8645    -114.557   2880  5400
    1       122004     52.8878    -114.470   2470  4000
```

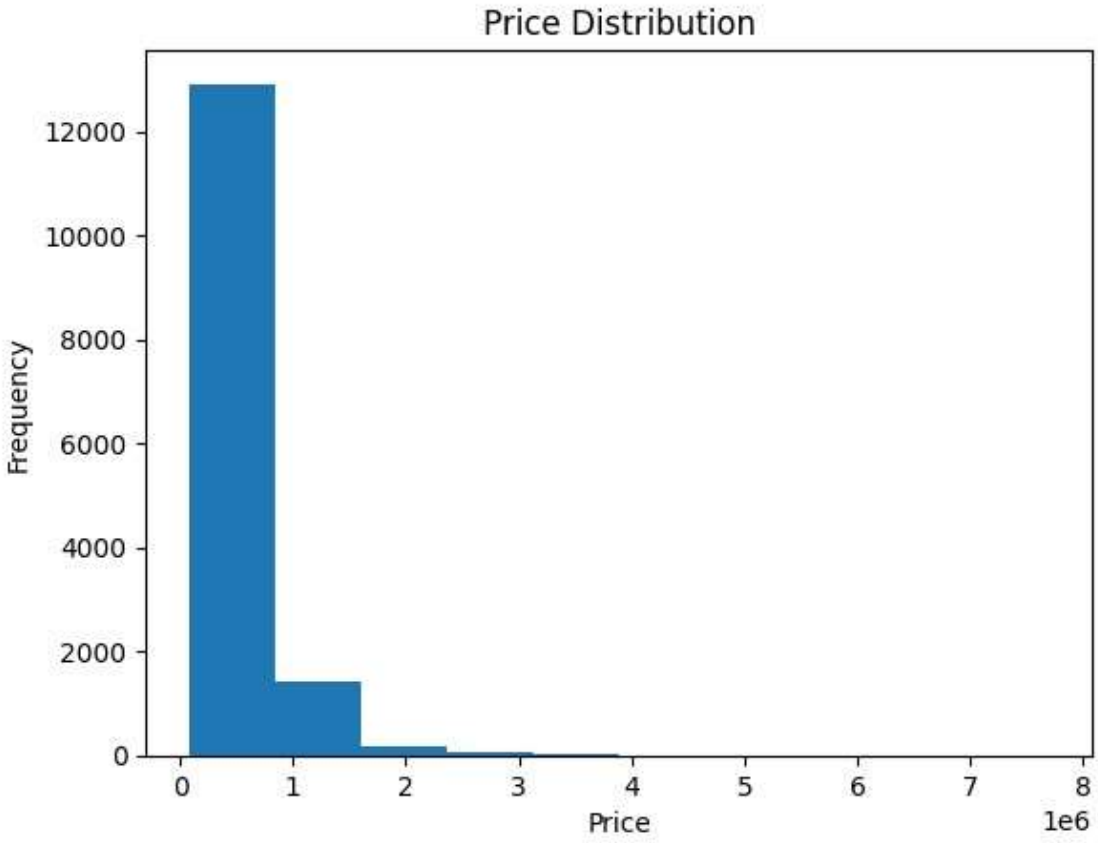
```
2      122004     52.8852    -114.468   2940  6600 3     122005
            52.9532    -114.321   3350  42847
4      122006   52.9047  -114.485             2060          4500
```

	Number of schools nearby	Distance from the airport	Price
0	2	58	2380000
1	2	51	1400000
2	1	53	1200000
3	3	76	838000
4	1	51	805000

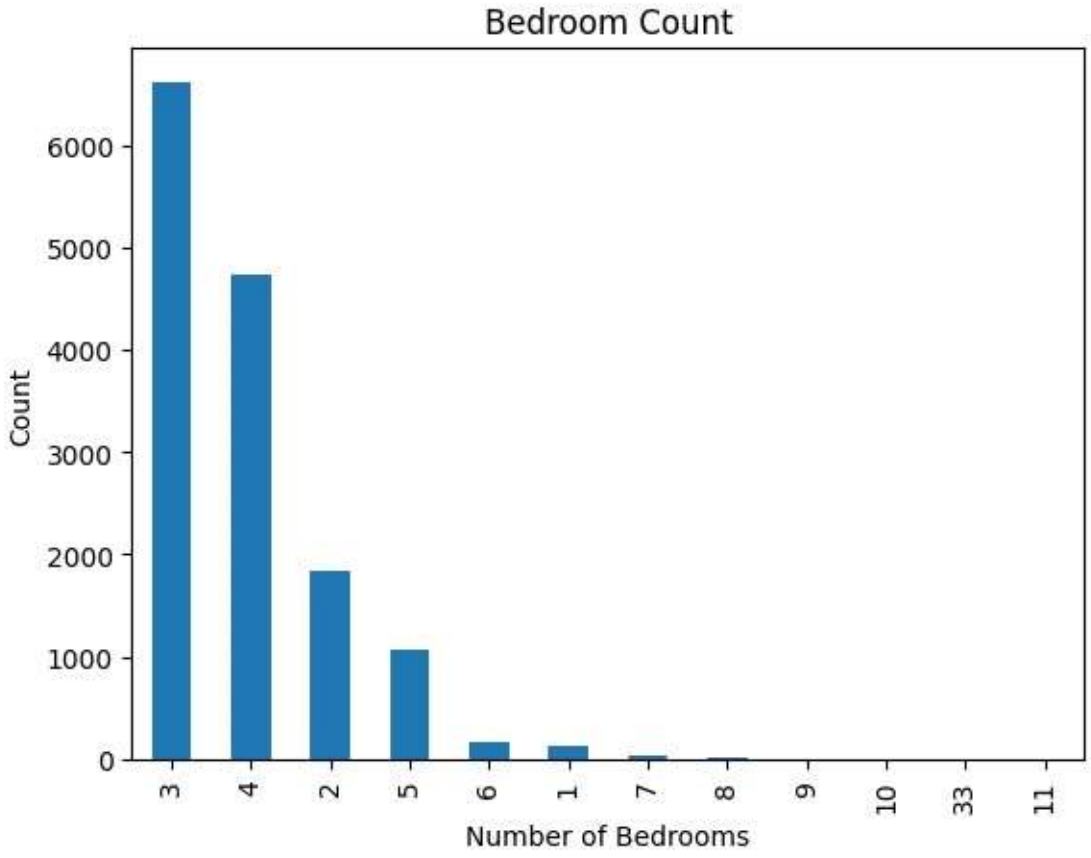
[5 rows x 23 columns]

UNIVARIATE

[ ]:
```
df['Price'].plot.hist()
plt.xlabel('Price')
plt.title('Price Distribution')
plt.show()
```



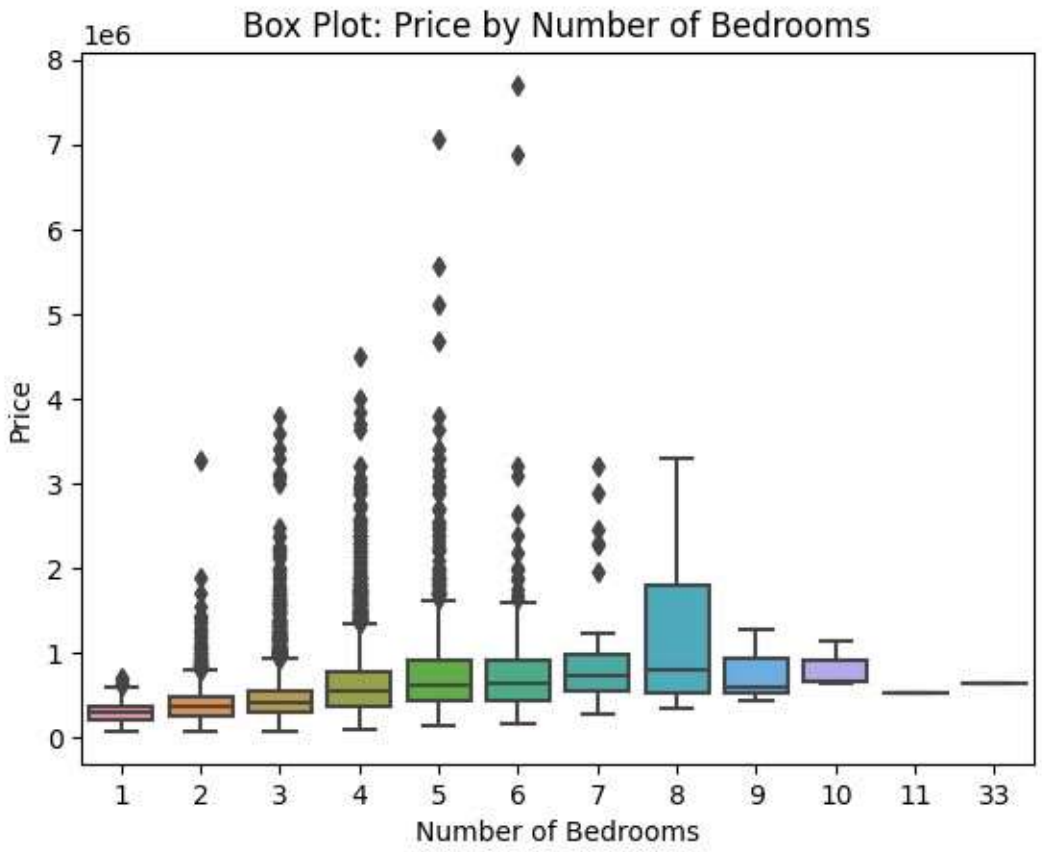
```
[ ]: df['number of bedrooms '].value_counts().plot(kind='bar')
     plt.xlabel('Number of Bedrooms ')
     plt.ylabel('Count')
     plt.title('Bedroom Count ')
     plt.show()
```



BI-VARIATE

```
[ ]: plt.scatter(df['living area'], df['Price'])
     plt.xlabel('Living Area')
     plt.ylabel('Price')
     plt.title('Scatter Plot: Living Area vs. Price ')
     plt.show()
```

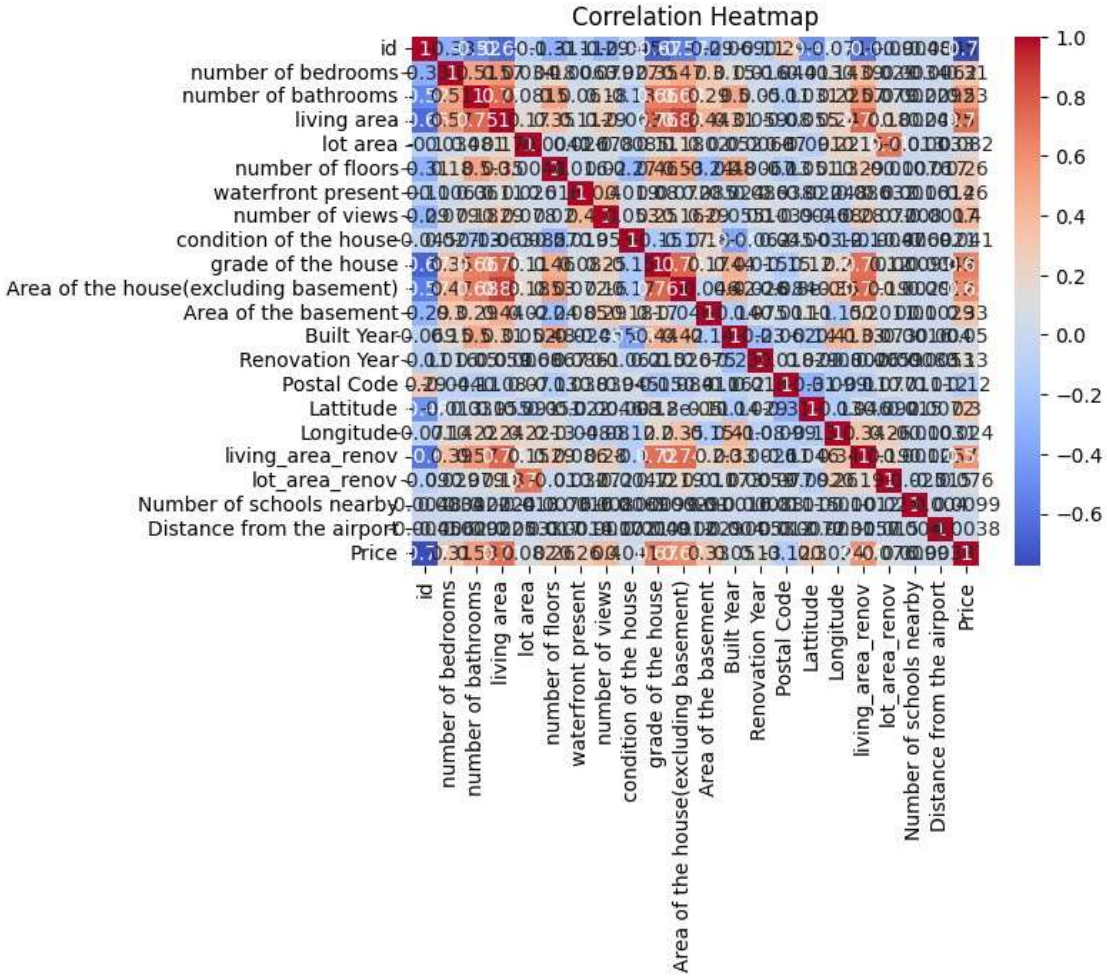

```
[ ]: sns.boxplot(x='number of bedrooms', y='Price',
data=df) plt.xlabel('Number of Bedrooms')
plt.ylabel('Price') plt.title('Box Plot: Price
by Number of Bedrooms') plt.show()
```



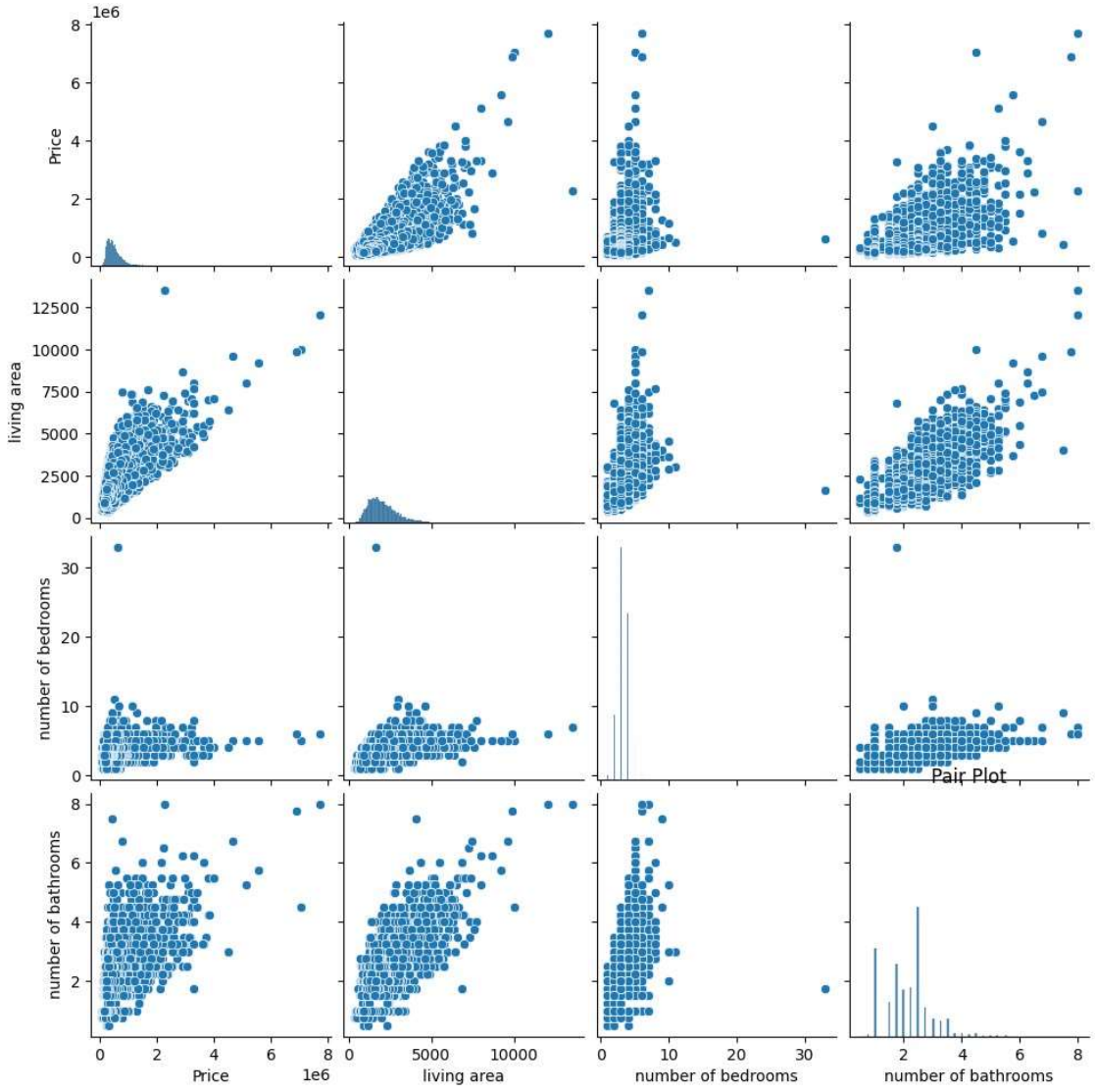
Multivariate

[ ]:
```
correlation_matrix = df.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

```
<ipython-input-40-182fd031f822>:1: FutureWarning: The default value
of numeric_only in DataFrame.corr is deprecated. In a future version,
it will default to False. Select only valid columns or specify the
value of numeric_only to silence this warning.
  correlation_matrix = df.corr()
```



[ ]:
```
sns.pairplot(df[['Price', 'living area', 'number of bedrooms', 'number of bathrooms']])
plt.title('Pair Plot')
plt.show()
```



DESCRIPTIVE STATISTICS

```
descriptive_stats = df.describe()
```

[ ]:
```
#Count of Non-null Values:
non_null_counts = df.count()
non_null_counts
```

[ ]: id 14620

[ ]: *#Basic Summary Statistics for Numerical Columns:*

Date 14620

number of bedrooms 14620

```
number of bathrooms                14620
living area 14620 lot area   14620 number
of floors   14620 waterfront present
 14620 number of views 14620 condition of
the house   14620 grade of the house
 14620
Area of the house(excluding basement) 14620
 Area of the basement                14620
 Built Year                          14620
 Renovation Year                     14620
 Postal Code                         14620
 Lattitude                           14620
 Longitude   14620 living_area_renov    14620
 lot_area_renov   14620
Number of schools nearby             14620
Distance from the airport            14620
 Price                               14620
 dtype: int64
```

[ ]:
```
#Frequency Count for Categorical Columns:
bedroom_counts = df['number of bedrooms'].value_counts()
bedroom_counts
```

[ ]:
```
3    6612
4    4724
2    1844
5    1079
6     176
1     136
7      30
8      11
9       3
10      3
33      1
11      1
Name: number of bedrooms, dtype: int64
```

[ ]:
```
#Grouping and Aggregating: avg_price_by_bedrooms =
df.groupby('number of bedrooms')['Price'].mean()
avg_price_by_bedrooms
```

[ ]: number of bedrooms
13.089638e+05 2
3.985476e+05
34.632776e+05
46.361988e+05 5 7.752550e+05 6
8.375815e+05 7 1.016544e+06 8 1.208455e+06
9 7.766663e+05
10 8.200000e+05
11 5.200000e+05
33 6.400000e+05
Name: Price, dtype: float64

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: