In [1]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline
import matplotlib
matplotlib.rcParams["figure.figsize"] = (20,10)
```

In [2]:
```
df = pd.read_csv('C:\\Users\\Atif\\Desktop\\python learning\\bengaluru_house_prices.csv')
df.head(7)
```

Out[2]:

	area_type	availability	location	size	society	total_sqft	bath	balcony	price
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056	2.0	1.0	39.07
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600	5.0	3.0	120.00
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	NaN	1440	2.0	3.0	62.00
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	Soiewre	1521	3.0	1.0	95.00
4	Super built-up Area	Ready To Move	Kothanur	2 BHK	NaN	1200	2.0	1.0	51.00
5	Super built-up Area	Ready To Move	Whitefield	2 BHK	DuenaTa	1170	2.0	1.0	38.00
6	Super built-up Area	18-May	Old Airport Road	4 BHK	Jaades	2732	4.0	NaN	204.00

In [3]:
```
df.shape
```

Out[3]:
```
(13320, 9)
```

In [4]:
```
df.groupby('area_type')['area_type'].agg('count')
#This group the the area type and aggregate the count of area type
```

Out[4]:
```
area_type
Built-up  Area          2418
Carpet  Area              87
Plot  Area              2025
Super built-up  Area    8790
Name: area_type, dtype: int64
```

In [5]:
```
df2 = df.drop(['area_type','availability','society','balcony'],axis='columns')
df2.head()
```

Out[5]:

	location	size	total_sqft	bath	price
0	Electronic City Phase II	2 BHK	1056	2.0	39.07
1	Chikka Tirupathi	4 Bedroom	2600	5.0	120.00
2	Uttarahalli	3 BHK	1440	2.0	62.00
3	Lingadheeranahalli	3 BHK	1521	3.0	95.00
4	Kothanur	2 BHK	1200	2.0	51.00

In [6]:
```
df2.isnull().sum()
#this show where every features have how many null values
```

Out[6]:
```
location       1
size          16
total_sqft     0
bath          73
price          0
dtype: int64
```

In [7]:
```
df3 = df2.dropna()
df3.isnull().sum()
#this will drop the all values who have null values
```

Out[7]:
```
location      0
size          0
total_sqft    0
bath          0
price         0
dtype: int64
```

In [8]:
```
df3['size'].unique()
```

Out[8]:
```
array(['2 BHK', '4 Bedroom', '3 BHK', '4 BHK', '6 Bedroom', '3 Bedroom',
       '1 BHK', '1 RK', '1 Bedroom', '8 Bedroom', '2 Bedroom',
       '7 Bedroom', '5 BHK', '7 BHK', '6 BHK', '5 Bedroom', '11 BHK',
       '9 BHK', '9 Bedroom', '27 BHK', '10 Bedroom', '11 Bedroom',
       '10 BHK', '19 BHK', '16 BHK', '43 Bedroom', '14 BHK', '8 BHK',
       '12 Bedroom', '13 BHK', '18 Bedroom'], dtype=object)
```

In [9]:
```
df3['bhk'] = df3['size'].apply(lambda x: int(x.split(' ')[0]))
#create new column and apply lambda and split the string value that is in columns size a
```

```
C:\Users\Atif\AppData\Local\Temp\ipykernel_7488\2729291521.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df3['bhk'] = df3['size'].apply(lambda x: int(x.split(' ')[0]))
```

In [10]:
```
df3.head()
```

Out[10]:

	location	size	total_sqft	bath	price	bhk
0	Electronic City Phase II	2 BHK	1056	2.0	39.07	2
1	Chikka Tirupathi	4 Bedroom	2600	5.0	120.00	4
2	Uttarahalli	3 BHK	1440	2.0	62.00	3
3	Lingadheeranahalli	3 BHK	1521	3.0	95.00	3
4	Kothanur	2 BHK	1200	2.0	51.00	2

In [11]:
```
df3['bhk'].unique()
```

Out[11]:
```
array([ 2,  4,  3,  6,  1,  8,  7,  5, 11,  9, 27, 10, 19, 16, 43, 14, 12,
       13, 18], dtype=int64)
```

In [12]:
```
df3[df3.bhk>20]
```

Out[12]:

	location	size	total_sqft	bath	price	bhk
1718	2Electronic City Phase II	27 BHK	8000	27.0	230.0	27
4684	Munnekollal	43 Bedroom	2400	40.0	660.0	43

In [ ]:
```
ft.unique()
```

```
# total_sqft have some range value now convert it into simple number
```

Out[13]:
```
array(['1056', '2600', '1440', ..., '1133 - 1384', '774', '4689'],
      dtype=object)
```

In [14]:
```
def is_float(x):
    try:
        float(x)
    except:
        return False
    return True
```

In [15]:
```
df3[~df3['total_sqft'].apply(is_float)].head(10)
```

Out[15]:

	location	size	total_sqft	bath	price	bhk
30	Yelahanka	4 BHK	2100 - 2850	4.0	186.000	4
122	Hebbal	4 BHK	3067 - 8156	4.0	477.000	4
137	8th Phase JP Nagar	2 BHK	1042 - 1105	2.0	54.005	2
165	Sarjapur	2 BHK	1145 - 1340	2.0	43.490	2
188	KR Puram	2 BHK	1015 - 1540	2.0	56.800	2
410	Kengeri	1 BHK	34.46Sq. Meter	1.0	18.500	1
549	Hennur Road	2 BHK	1195 - 1440	2.0	63.770	2
648	Arekere	9 Bedroom	4125Perch	9.0	265.000	9
661	Yelahanka	2 BHK	1120 - 1145	2.0	48.130	2
672	Bettahalsoor	4 Bedroom	3090 - 5002	4.0	445.000	4

In [16]:
```
def covert_sqft_to_num(x):
    tokens=x.split("-")
    if len(tokens)==2:
        return (float(tokens[0])+float(tokens[1]))/2
    try:
        return float(x)
    except:
        return None
#this is use to convert range into simple float number after taking average of range
```

In [17]:
```
covert_sqft_to_num('1015 - 1540')
```

Out[17]:
```
1277.5
```

In [18]:
```
df4 = df3.copy()
df4['total_sqft'] = df4['total_sqft'].apply(covert_sqft_to_num)
df4.head()
```

Out[18]:

	location	size	total_sqft	bath	price	bhk
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.07	2
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.00	3
4	Kothanur	2 BHK	1200.0	2.0	51.00	2

In [19]:
```
df4.loc[30]
```

Out[19]:
```
location      Yelahanka
size              4 BHK
total_sqft       2475.0
bath                4.0
price             186.0
bhk                   4
Name: 30, dtype: object
```

In [20]:
```
df4['location'].unique()
```

Out[20]:
```
array(['Electronic City Phase II', 'Chikka Tirupathi', 'Uttarahalli', ...,
       '12th cross srinivas nagar banshankari 3rd stage',
       'Havanur extension', 'Abshot Layout'], dtype=object)
```

In [21]:
```
len(df4.location.unique())
```

Out[21]:
```
1304
```

In [22]:
```
df5 = df4.copy()
df5['price_per_sqft'] = df5['price']*1000000/df5['total_sqft']
df5.head()
#make a new column beacuse in real eastate we need price per square ft
```

Out[22]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.07	2	36998.106061
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4	46153.846154
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3	43055.555556
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.00	3	62458.908613
4	Kothanur	2 BHK	1200.0	2.0	51.00	2	42500.000000

In [23]:
```
df5.location = df5.location.apply(lambda x: x.strip())
#this is use to remove space from location columns
location_stat = df5.groupby('location')['location'].agg('count').sort_values(ascending=F
location_stat
#this will show the values of loction with number of rows.
```

Out[23]:
```
location
Whitefield               535
Sarjapur  Road           392
Electronic City          304
Kanakpura Road           266
Thanisandra              236
                        ...
1 Giri Nagar               1
Kanakapura Road,           1
Kanakapura main  Road      1
Karnataka Shabarimala      1
whitefiled                 1
Name: location, Length: 1293, dtype: int64
```

In [24]:
```
len(location_stat[location_stat<=10])
#total length is 1293 and 1052 is less then 10 values
```

Out[24]:
```
1052
```

In [25]:
```
locaton_less_10 = (location_stat[location_stat<=10])
locaton_less_10
```

Out[25]:
```
location
Basapura                 10
1st Block Koramangala    10
Gunjur Palya             10
Kalkere                  10
Sector 1 HSR Layout      10
                         ..
1 Giri Nagar              1
Kanakapura Road,          1
Kanakapura main  Road     1
Karnataka Shabarimala     1
whitefiled                1
Name: location, Length: 1052, dtype: int64
```

In [26]:
```
df5.location = df5.location.apply(lambda x:'other' if x in locaton_less_10 else x)
len(df5.location.unique())
```

Out[26]:
```
242
```

In [27]:
```
df5.head(10)
```

Out[27]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.07	2	36998.106061
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4	46153.846154
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3	43055.555556
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.00	3	62458.908613
4	Kothanur	2 BHK	1200.0	2.0	51.00	2	42500.000000
5	Whitefield	2 BHK	1170.0	2.0	38.00	2	32478.632479
6	Old Airport Road	4 BHK	2732.0	4.0	204.00	4	74670.571010
7	Rajaji Nagar	4 BHK	3300.0	4.0	600.00	4	181818.181818
8	Marathahalli	3 BHK	1310.0	3.0	63.25	3	48282.442748
9	other	6 Bedroom	1020.0	6.0	370.00	6	362745.098039

In [28]:
```
min_thresold , max_thresold = df5.total_sqft.quantile([0.01,0.99])
min_thresold , max_thresold
```

Out[28]:
```
(500.0, 5000.0)
```

In [29]:
```
df5[df5.total_sqft < min_thresold]
```

Out[29]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
78	Kaval Byrasandra	2 BHK	460.0	1.0	22.0	2	4.782609e+04
119	Hennur Road	2 Bedroom	276.0	3.0	23.0	2	8.333333e+04
171	Attibele	1 BHK	450.0	1.0	11.0	1	2.444444e+04
177	Nagavara	1 Bedroom	400.0	1.0	14.0	1	3.500000e+04
349	other	3 Bedroom	11.0	3.0	74.0	3	6.727273e+06
...	...	...	...	...	...	...	...
12579	Chandapura	1 BHK	410.0	1.0	10.0	1	2.439024e+04
12666	Marsur	2 BHK	497.0	1.0	20.0	2	4.024145e+04
12895	other	1 BHK	450.0	1.0	20.0	1	4.444444e+04
13112	Nagavara	3 Bedroom	440.0	3.0	35.0	3	7.954545e+04
13216	other	1 BHK	250.0	2.0	40.0	1	1.600000e+05

120 rows × 7 columns

In [30]:
```
df5[df5.total_sqft > max_thresold]
```

Out[30]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
62	Whitefield	4 Bedroom	5700.0	5.0	650.0	4	114035.087719
122	Hebbal	4 BHK	5611.5	4.0	477.0	4	85004.009623
408	Rajaji Nagar	7 BHK	12000.0	6.0	2200.0	7	183333.333333
440	Whitefield	4 Bedroom	11890.0	4.0	700.0	4	58873.002523
514	Banashankari Stage III	4 Bedroom	8500.0	4.0	145.0	4	17058.823529
...	...	...	...	...	...	...	...
13095	other	4 BHK	6652.0	6.0	660.0	4	99218.280216
13119	other	4 Bedroom	6688.0	6.0	700.0	4	104665.071770
13197	other	4 Bedroom	9200.0	4.0	2600.0	4	282608.695652
13200	other	6 Bedroom	8000.0	6.0	2800.0	6	350000.000000
13226	Raja Rajeshwari Nagar	8 Bedroom	6000.0	8.0	215.0	8	35833.333333

126 rows × 7 columns

In [31]:
```
df6 = df5[(df5.total_sqft < max_thresold) & (df5.total_sqft > min_thresold)]
df6.shape
```

Out[31]:
```
(12900, 7)
```

In [32]:
```
min_thresold , max_thresold = df6.price_per_sqft.quantile([0.1,0.9])
min_thresold , max_thresold
```

Out[32]:
```
(35004.77047264908, 113644.56842796196)
```

In [33]:
```
df6[df6.price_per_sqft < min_thresold]
```

Out[33]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
5	Whitefield	2 BHK	1170.0	2.0	38.00	2	32478.632479
20	Kengeri	1 BHK	600.0	1.0	15.00	1	25000.000000
26	Electronic City	2 BHK	660.0	1.0	23.10	2	35000.000000
31	Bisuvanahalli	3 BHK	1075.0	2.0	35.00	3	32558.139535
33	Raja Rajeshwari Nagar	3 BHK	1693.0	3.0	57.39	3	33898.405198
...	...	...	...	...	...	...	...
13244	Kereguddadahalli	2 BHK	1015.0	2.0	35.00	2	34482.758621
13275	Kothannur	4 Bedroom	1600.0	4.0	45.00	4	28125.000000
13291	other	1 Bedroom	812.0	1.0	26.00	1	32019.704433
13304	Raja Rajeshwari Nagar	2 BHK	1187.0	2.0	40.14	2	33816.343724
13319	Doddathoguru	1 BHK	550.0	1.0	17.00	1	30909.090909

1290 rows × 7 columns

In [34]:
```
df6[df6.price_per_sqft > max_thresold]
```

Out[34]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
7	Rajaji Nagar	4 BHK	3300.0	4.0	600.0	4	181818.181818
9	other	6 Bedroom	1020.0	6.0	370.0	6	362745.098039
22	Thanisandra	4 Bedroom	2800.0	5.0	380.0	4	135714.285714
45	HSR Layout	8 Bedroom	600.0	9.0	200.0	8	333333.333333
48	KR Puram	2 Bedroom	800.0	1.0	130.0	2	162500.000000
...	...	...	...	...	...	...	...
13262	other	2 BHK	1140.0	1.0	185.0	2	162280.701754
13277	other	7 Bedroom	1400.0	7.0	218.0	7	155714.285714
13296	Cox Town	2 BHK	1200.0	2.0	140.0	2	116666.666667
13306	other	4 Bedroom	1200.0	5.0	325.0	4	270833.333333
13311	Ramamurthy Nagar	7 Bedroom	1500.0	9.0	250.0	7	166666.666667

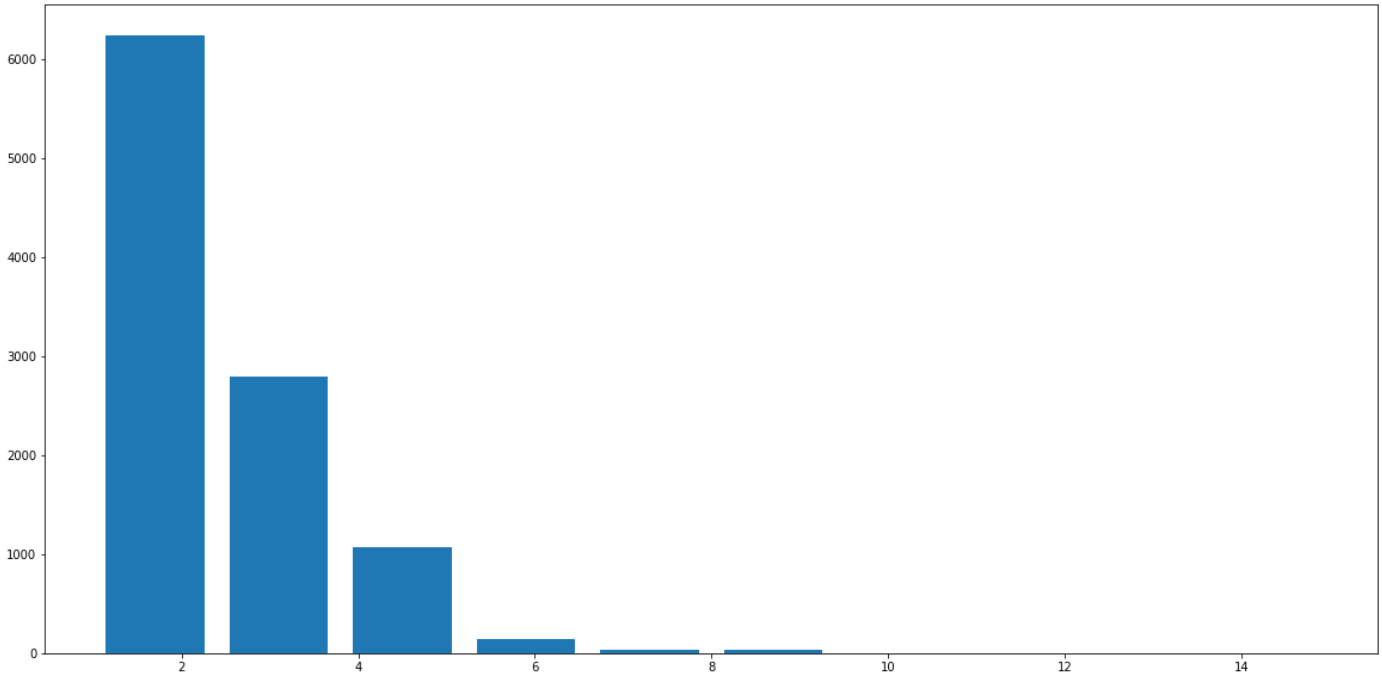
1290 rows × 7 columns

In [35]:
```
df7 = df6[(df6.price_per_sqft < max_thresold) & (df6.price_per_sqft > min_thresold)]
df7.shape
```

Out[35]:
```
(10320, 7)
```

In [36]:
```
plt.hist(df7.bath,rwidth=0.8)
```

Out[36]:
```
(array([6.242e+03, 2.796e+03, 1.065e+03, 1.380e+02, 3.600e+01, 3.600e+01,
        3.000e+00, 3.000e+00, 0.000e+00, 1.000e+00]),
 array([ 1. ,  2.4,  3.8,  5.2,  6.6,  8. ,  9.4, 10.8, 12.2, 13.6, 15. ]),
 <BarContainer object of 10 artists>)
```



In [37]:
```
min_thresold , max_thresold = df6.bath.quantile([0.01,0.93])
min_thresold , max_thresold
```

Out[37]:
```
(1.0, 5.0)
```

In [38]:
```
df7[df7.bath < min_thresold]
```

Out[38]:

location	size	total_sqft	bath	price	bhk	price_per_sqft

In [39]:
```
df7[df7.bath > max_thresold]
```

Out[39]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
64	Bommanahalli	8 Bedroom	3000.0	8.0	140.0	8	46666.666667
68	Devarachikkanahalli	8 Bedroom	1350.0	7.0	85.0	8	62962.962963
79	ISRO Layout	6 Bedroom	4400.0	6.0	250.0	6	56818.181818
85	Hegde Nagar	6 Bedroom	3000.0	7.0	210.0	6	70000.000000
145	Vishveshwarya Layout	7 BHK	4000.0	7.0	225.0	7	56250.000000
...	...	...	...	...	...	...	...
13180	Sarakki Nagar	4 BHK	3124.0	6.0	349.0	4	111715.749040
13208	Hebbal	4 BHK	4000.0	6.0	370.0	4	92500.000000
13219	Laggere	7 Bedroom	1590.0	9.0	132.0	7	83018.867925
13221	other	9 Bedroom	1178.0	9.0	75.0	9	63667.232598
13300	Hosakerehalli	5 Bedroom	1500.0	6.0	145.0	5	96666.666667

217 rows × 7 columns

In [40]:
```
df8 = df7[(df7.bath < max_thresold) & (df7.bath > min_thresold)]
df8.shape
```

Out[40]:
```
(9337, 7)
```

In [41]:
```
df8.head(10)
```

Out[41]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.07	2	36998.106061
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3	43055.555556
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.00	3	62458.908613
4	Kothanur	2 BHK	1200.0	2.0	51.00	2	42500.000000
6	Old Airport Road	4 BHK	2732.0	4.0	204.00	4	74670.571010
8	Marathahalli	3 BHK	1310.0	3.0	63.25	3	48282.442748
10	Whitefield	3 BHK	1800.0	2.0	70.00	3	38888.888889
12	7th Phase JP Nagar	2 BHK	1000.0	2.0	38.00	2	38000.000000
13	Gottigere	2 BHK	1100.0	2.0	40.00	2	36363.636364
14	Sarjapur	3 Bedroom	2250.0	3.0	148.00	3	65777.777778

In [42]:
```
df9 = df8.drop(['size','price_per_sqft'],axis='columns')
df9.head()
```

Out[42]:

	location	total_sqft	bath	price	bhk
0	Electronic City Phase II	1056.0	2.0	39.07	2
2	Uttarahalli	1440.0	2.0	62.00	3
3	Lingadheeranahalli	1521.0	3.0	95.00	3
4	Kothanur	1200.0	2.0	51.00	2
6	Old Airport Road	2732.0	4.0	204.00	4

In [43]:
```
dummies = pd.get_dummies(df9.location)
dummies.head()
```

Out[43]:

	1st Block Jayanagar	1st Phase JP Nagar	2nd Phase Judicial Layout	2nd Stage Nagarbhavi	5th Block Hbr Layout	5th Phase JP Nagar	6th Phase JP Nagar	7th Phase JP Nagar	8th Phase JP Nagar	9th Phase JP Nagar	...	Vishveshwarya Layout
0	0	0	0	0	0	0	0	0	0	0	...	0
2	0	0	0	0	0	0	0	0	0	0	...	0
3	0	0	0	0	0	0	0	0	0	0	...	0
4	0	0	0	0	0	0	0	0	0	0	...	0
6	0	0	0	0	0	0	0	0	0	0	...	0

5 rows × 241 columns

In [44]:
```
df10 = pd.concat([df9,dummies.drop('other',axis='columns')],axis='columns')
df10.head(10)
```

Out[44]:

	location	total_sqft	bath	price	bhk	1st Block Jayanagar	1st Phase JP Nagar	2nd Phase Judicial Layout	2nd Stage Nagarbhavi	5th Block Hbr Layout	...	Vijaya
0	Electronic City Phase II	1056.0	2.0	39.07	2	0	0	0	0	0	...	
2	Uttarahalli	1440.0	2.0	62.00	3	0	0	0	0	0	...	
3	Lingadheeranahalli	1521.0	3.0	95.00	3	0	0	0	0	0	...	
4	Kothanur	1200.0	2.0	51.00	2	0	0	0	0	0	...	
6	Old Airport Road	2732.0	4.0	204.00	4	0	0	0	0	0	...	
8	Marathahalli	1310.0	3.0	63.25	3	0	0	0	0	0	...	
10	Whitefield	1800.0	2.0	70.00	3	0	0	0	0	0	...	
12	7th Phase JP Nagar	1000.0	2.0	38.00	2	0	0	0	0	0	...	
13	Gottigere	1100.0	2.0	40.00	2	0	0	0	0	0	...	
14	Sarjapur	2250.0	3.0	148.00	3	0	0	0	0	0	...	

10 rows × 245 columns

In [45]:
```
df11 = df10.drop('location',axis='columns')
df11.head(20)
```

Out[45]:

	total_sqft	bath	price	bhk	1st Block Jayanagar	1st Phase JP Nagar	2nd Phase Judicial Layout	2nd Stage Nagarbhavi	5th Block Hbr Layout	5th Phase JP Nagar	...	Vijayanagar	Vis
0	1056.0	2.0	39.07	2	0	0	0	0	0	0	...	0	
2	1440.0	2.0	62.00	3	0	0	0	0	0	0	...	0	
3	1521.0	3.0	95.00	3	0	0	0	0	0	0	...	0	
4	1200.0	2.0	51.00	2	0	0	0	0	0	0	...	0	
6	2732.0	4.0	204.00	4	0	0	0	0	0	0	...	0	
8	1310.0	3.0	63.25	3	0	0	0	0	0	0	...	0	
10	1800.0	2.0	70.00	3	0	0	0	0	0	0	...	0	
12	1000.0	2.0	38.00	2	0	0	0	0	0	0	...	0	
13	1100.0	2.0	40.00	2	0	0	0	0	0	0	...	0	
14	2250.0	3.0	148.00	3	0	0	0	0	0	0	...	0	
15	1175.0	2.0	73.50	2	0	0	0	0	0	0	...	0	
16	1180.0	3.0	48.00	3	0	0	0	0	0	0	...	0	
17	1540.0	3.0	60.00	3	0	0	0	0	0	0	...	0	
18	2770.0	4.0	290.00	3	0	0	0	0	0	0	...	0	
19	1100.0	2.0	48.00	2	0	0	0	0	0	0	...	0	
21	1755.0	3.0	122.00	3	0	0	0	0	0	0	...	0	
23	1767.0	3.0	103.00	3	0	0	0	0	0	0	...	0	
25	1250.0	3.0	56.00	3	0	0	0	0	0	0	...	0	
27	1610.0	3.0	81.00	3	0	0	0	0	0	0	...	0	
28	1151.0	2.0	48.77	2	0	0	0	0	0	0	...	0	

20 rows × 244 columns

In [46]:
```
df11.shape
```

Out[46]:
```
(9337, 244)
```

In [47]:
```
X = df11.drop('price',axis='columns')
X
```

Out[47]:

	total_sqft	bath	bhk	1st Block Jayanagar	1st Phase JP Nagar	2nd Phase Judicial Layout	2nd Stage Nagarbhavi	5th Block Hbr Layout	5th Phase JP Nagar	6th Phase JP Nagar	...	Vijayanagar
0	1056.0	2.0	2	0	0	0	0	0	0	0	...	0
2	1440.0	2.0	3	0	0	0	0	0	0	0	...	0
3	1521.0	3.0	3	0	0	0	0	0	0	0	...	0
4	1200.0	2.0	2	0	0	0	0	0	0	0	...	0
6	2732.0	4.0	4	0	0	0	0	0	0	0	...	0
...	...	...	...	...	...	...	...	...	...	...	...	...
13313	1345.0	2.0	3	0	0	0	0	0	0	0	...	0
13314	1715.0	3.0	3	0	0	0	0	0	0	0	...	0
13315	3453.0	4.0	5	0	0	0	0	0	0	0	...	0
13317	1141.0	2.0	2	0	0	0	0	0	0	0	...	0
13318	4689.0	4.0	4	0	0	0	0	0	0	0	...	0

9337 rows × 243 columns

In [48]:
```
Y = df11.price
```

In [49]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2,random_state=10)
```

In [50]:
```
from sklearn import linear_model
reg = linear_model.LinearRegression()
reg.fit(X_train,Y_train)
reg.score(X_test,Y_test)
```

Out[50]:
```
0.7702466549754435
```

In [51]:
```
def predict_price(location,sqft,bath,bhk):
    loc_index = np.where(X.columns==location)[0][0]

    x = np.zeros(len(X.columns))
    x[0] = sqft
    x[1] = bath
    x[2] = bhk
    if loc_index >= 0:
        x[loc_index] = 1

    return reg.predict([x])[0]
```

In [52]:
```
X.columns
```

Out[52]:
```
Index(['total_sqft', 'bath', 'bhk', '1st Block Jayanagar',
       '1st Phase JP Nagar', '2nd Phase Judicial Layout',
       '2nd Stage Nagarbhavi', '5th Block Hbr Layout', '5th Phase JP Nagar',
       '6th Phase JP Nagar',
       ...
       'Vijayanagar', 'Vishveshwarya Layout', 'Vishwapriya Layout',
       'Vittasandra', 'Whitefield', 'Yelachenahalli', 'Yelahanka',
       'Yelahanka New Town', 'Yelenahalli', 'Yeshwanthpur'],
      dtype='object', length=243)
```

In [53]:
```
np.where(X.columns=='2nd Phase Judicial Layout')[0][0]
```

```
#this will show the index number of  2nd Phase Judicial Layout
```

Out[53]:
```
5
```

In [61]:
```
predict_price('Electronic City Phase II',1056,2,2)
```

```
C:\Users\Atif\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:
450: UserWarning: X does not have valid feature names, but LinearRegression was fitted w
ith feature names
  warnings.warn(
```

Out[61]:
```
44.56377937806113
```

In [59]:
```
predict_price('Indira Nagar',1500,4,2)
```

```
C:\Users\Atif\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:
450: UserWarning: X does not have valid feature names, but LinearRegression was fitted w
ith feature names
  warnings.warn(
```

Out[59]:
```
146.4725951608555
```

In [58]:
```
predict_price('1st Phase JP Nagar',1000,4,3)
```

```
C:\Users\Atif\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:
450: UserWarning: X does not have valid feature names, but LinearRegression was fitted w
ith feature names
  warnings.warn(
```

Out[58]:
```
88.8864408761639
```

In [ ]:
```

```