

Land price prediction

July 15, 2024

```
[1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib
import warnings
warnings.filterwarnings('ignore')
matplotlib.rcParams["figure.figsize"]=(20,10)
```

```
[2]: df1=pd.read_csv("D:\\docs\\Bengaluru_House_Data.csv")
df1.head()
```

```
[2]:
```

		area_type	availability	location	size \
0	Super built-up	Area	19-Dec	Electronic City Phase II	2 BHK
1	Plot	Area	Ready To Move	Chikka Tirupathi	4 Bedroom
2	Built-up	Area	Ready To Move	Uttarahalli	3 BHK
3	Super built-up	Area	Ready To Move	Lingadheeranahalli	3 BHK
4	Super built-up	Area	Ready To Move	Kothanur	2 BHK

	society	total_sqft	bath	balcony	price
0	Coomee	1056	2.0	1.0	39.07
1	Theanmp	2600	5.0	3.0	120.00
2	NaN	1440	2.0	3.0	62.00
3	Soiewre	1521	3.0	1.0	95.00
4	NaN	1200	2.0	1.0	51.00

```
[3]: df1.shape
```

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

```
[4]: df1.groupby('area_type')['area_type'].agg('count')
```

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

```
[5]: df2=df1.drop(["area_type","society","balcony","availability"],axis="columns")
df2.head()
```

```
[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

```
[6]: df2.isnull().sum()
```

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

```
[7]: df3=df2.dropna()
df3.isnull().sum()
```

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

```
[8]: df3.shape
```

```
[8]: (13246, 5)
```

```
[9]: df3['size'].unique()
```

```
[9]: 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)
```

```
[10]: df3['bhk']=df3['size'].apply(lambda x: int(x.split(' ')[0]))
```

```
[11]: df3.head()
```

```
[11]:
```

	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

```
[12]: df3['bhk'].unique()
```

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

```
[13]: df3[df3.bhk>20]
```

	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

```
[14]: df3.total_sqft.unique()
```

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

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

```
[16]: df3[~df3["total_sqft"].apply(is_float)].head(10) # ~is not key that is it gives
        ↳all the toatl_sqft that is not in float
```

	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

```
[17]: def range_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

```

```
[18]: df4=df3.copy()
df4
```

```
[18]:
```

	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
...
13315	Whitefield	5 Bedroom	3453	4.0	231.00	5
13316	Richards Town	4 BHK	3600	5.0	400.00	4
13317	Raja Rajeshwari Nagar	2 BHK	1141	2.0	60.00	2
13318	Padmanabhanagar	4 BHK	4689	4.0	488.00	4
13319	Doddathoguru	1 BHK	550	1.0	17.00	1

[13246 rows x 6 columns]

```
[19]: df4["total_sqft"]=df4["total_sqft"].apply(range_to_num)
```

```
[20]: df4
```

```
[20]:
```

	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
...
13315	Whitefield	5 Bedroom	3453.0	4.0	231.00	5
13316	Richards Town	4 BHK	3600.0	5.0	400.00	4
13317	Raja Rajeshwari Nagar	2 BHK	1141.0	2.0	60.00	2
13318	Padmanabhanagar	4 BHK	4689.0	4.0	488.00	4
13319	Doddathoguru	1 BHK	550.0	1.0	17.00	1

[13246 rows x 6 columns]

```
[21]: df4.loc[30]
```

```
[21]: location      Yelahanka
size              4 BHK
total_sqft        2475.0
bath              4.0
```

```
price          186.0
bhk            4
Name: 30, dtype: object
```

```
[22]: df4.head()
```

```
[22]:
```

	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

```
[23]: df5=df4.copy()
```

```
[24]: df5["price_per_sqft"]=df5["price"]*100000/df5["total_sqft"]
df5.head()
```

```
[24]:
```

	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	

	price_per_sqft
0	3699.810606
1	4615.384615
2	4305.555556
3	6245.890861
4	4250.000000

```
[25]: len(df5["location"].unique())
```

```
[25]: 1304
```

```
[26]: df5.location= df5.location.apply(lambda x: x.strip())
location_stats=df5.groupby('location')['location'].agg('count').
    ↪sort_values(ascending=False)
location_stats
```

```
[26]: location
Whitefield          535
Sarjapur Road       392
Electronic City     304
Kanakpura Road      266
Thanisandra         236
...
```

```

adigondanhalli          1
akshaya nagar t c palya  1
anjananager magdi road  1
arudi                   1
2Electronic City Phase II 1
Name: location, Length: 1293, dtype: int64

```

```
[27]: len(location_stats[location_stats<=10])
```

```
[27]: 1052
```

```
[28]: location_stats_less_than_10=location_stats[location_stats<=10]
location_stats_less_than_10
```

```
[28]: location
Ganga Nagar          10
Gunjur Palya         10
BTM 1st Stage        10
Sadashiva Nagar      10
Kalkere              10
..
adigondanhalli       1
akshaya nagar t c palya 1
anjananager magdi road 1
arudi                1
2Electronic City Phase II 1
Name: location, Length: 1052, dtype: int64

```

```
[29]: df5.location=df5.location.apply(lambda x: 'other' if x in_
↳location_stats_less_than_10 else x)
len(df5.location.unique())
```

```
[29]: 242
```

```
[30]: df5.head(30)
```

```
[30]:
```

	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	
5	Whitefield	2 BHK	1170.0	2.0	38.00	2	
6	Old Airport Road	4 BHK	2732.0	4.0	204.00	4	
7	Rajaji Nagar	4 BHK	3300.0	4.0	600.00	4	
8	Marathahalli	3 BHK	1310.0	3.0	63.25	3	
9	other	6 Bedroom	1020.0	6.0	370.00	6	
10	Whitefield	3 BHK	1800.0	2.0	70.00	3	

11	Whitefield	4 Bedroom	2785.0	5.0	295.00	4
12	7th Phase JP Nagar	2 BHK	1000.0	2.0	38.00	2
13	Gottigere	2 BHK	1100.0	2.0	40.00	2
14	Sarjapur	3 Bedroom	2250.0	3.0	148.00	3
15	Mysore Road	2 BHK	1175.0	2.0	73.50	2
16	Bisuvanahalli	3 BHK	1180.0	3.0	48.00	3
17	Raja Rajeshwari Nagar	3 BHK	1540.0	3.0	60.00	3
18	other	3 BHK	2770.0	4.0	290.00	3
19	other	2 BHK	1100.0	2.0	48.00	2
20	Kengeri	1 BHK	600.0	1.0	15.00	1
21	Binny Pete	3 BHK	1755.0	3.0	122.00	3
22	Thanisandra	4 Bedroom	2800.0	5.0	380.00	4
23	Bellandur	3 BHK	1767.0	3.0	103.00	3
24	Thanisandra	1 RK	510.0	1.0	25.25	1
25	other	3 BHK	1250.0	3.0	56.00	3
26	Electronic City	2 BHK	660.0	1.0	23.10	2
27	Whitefield	3 BHK	1610.0	3.0	81.00	3
28	Ramagondanahalli	2 BHK	1151.0	2.0	48.77	2
29	Electronic City	3 BHK	1025.0	2.0	47.00	3

price_per_sqft

0	3699.810606
1	4615.384615
2	4305.555556
3	6245.890861
4	4250.000000
5	3247.863248
6	7467.057101
7	18181.818182
8	4828.244275
9	36274.509804
10	3888.888889
11	10592.459605
12	3800.000000
13	3636.363636
14	6577.777778
15	6255.319149
16	4067.796610
17	3896.103896
18	10469.314079
19	4363.636364
20	2500.000000
21	6951.566952
22	13571.428571
23	5829.088851
24	4950.980392
25	4480.000000

```

26      3500.000000
27      5031.055901
28      4237.185056
29      4585.365854

```

```
[31]: df5.shape
```

```
[31]: (13246, 7)
```

```
[32]: df6=df5[~(df5.total_sqft/df5.bhk<300)]
df6.shape
```

```
[32]: (12502, 7)
```

```
[33]: df6.price_per_sqft.describe()
```

```

[33]: count      12456.000000
mean         6308.502826
std          4168.127339
min           267.829813
25%          4210.526316
50%          5294.117647
75%          6916.666667
max         176470.588235
Name: price_per_sqft, dtype: float64

```

```

[34]: def remove_outlier(df):
    df_out=pd.DataFrame()
    for key,subdf in df.groupby('location'):
        m=np.mean(subdf.price_per_sqft)
        st=np.std(subdf.price_per_sqft)
        reduced_df=subdf[(subdf.price_per_sqft>(m-st))&(subdf.
        ↪price_per_sqft<=(m+st))]
        df_out=pd.concat([df_out,reduced_df],ignore_index=True)
    return df_out

df7=remove_outlier(df6)
df7.shape

```

```
[34]: (10241, 7)
```

```

[35]: def scatter_chart(df,location):
    bhk2=df[(df.location==location)&(df.bhk==2)]
    bhk3=df[(df.location==location)&(df.bhk==3)]
    matplotlib.rcParams['figure.figsize']=(15,10)
    plt.scatter(bhk2.total_sqft,bhk2.price,color='blue',label='2 BHK',s=50)
    plt.scatter(bhk3.total_sqft,bhk3.price,marker='+',color='green',label='3_
    ↪BHK',s=50)

```

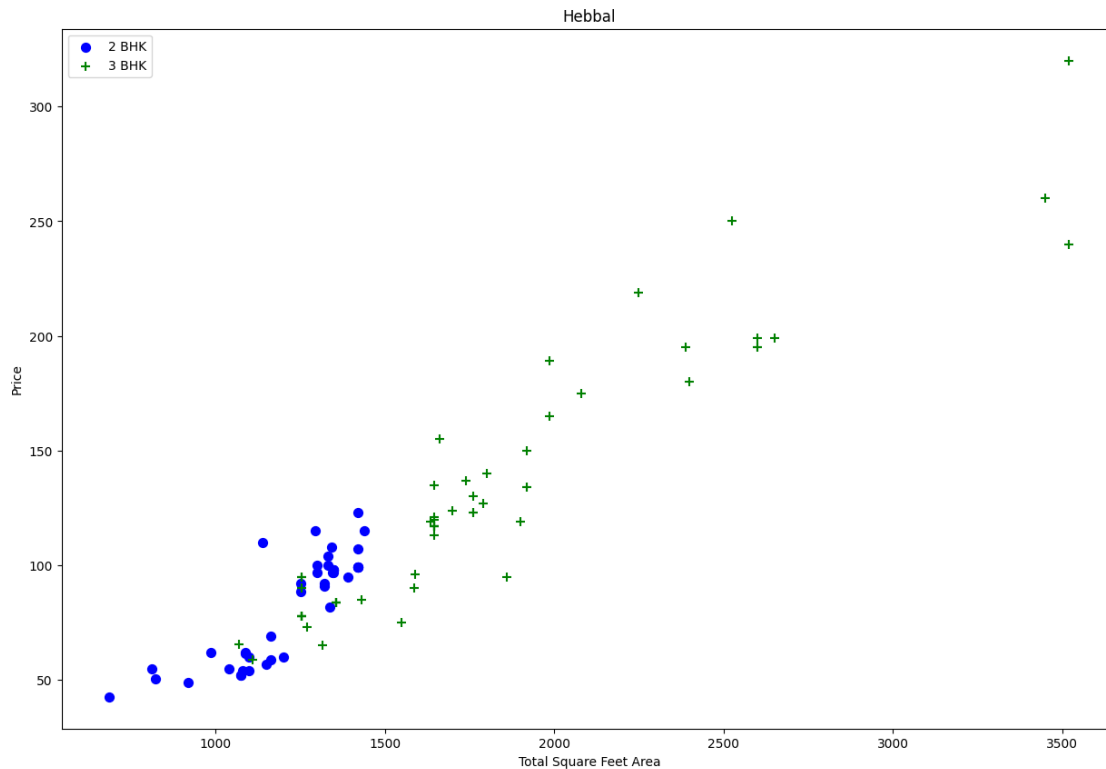


```

plt.xlabel("Total Square Feet Area")
plt.ylabel("Price")
plt.title(location)
plt.legend()

scatter_chart(df7,"Hebbal")

```



```

[36]: def remove_bhk_outliers(df):
    exclude_indices=np.array([])
    for location,location_df in df.groupby('location'):
        bhk_stats={}
        for bhk,bhk_df in location_df.groupby('bhk'):
            bhk_stats[bhk]={
                'mean':np.mean(bhk_df.price_per_sqft),
                'std':np.std(bhk_df.price_per_sqft),
                'count':bhk_df.shape[0]
            }
        for bhk,bhk_df in location_df.groupby('bhk'):
            stats=bhk_stats.get(bhk-1)
            if stats and stats['count']>5:
                exclude_indices=np.append(exclude_indices,bhk_df[bhk_df.
↪price_per_sqft<(stats['mean'])).index.values)

```

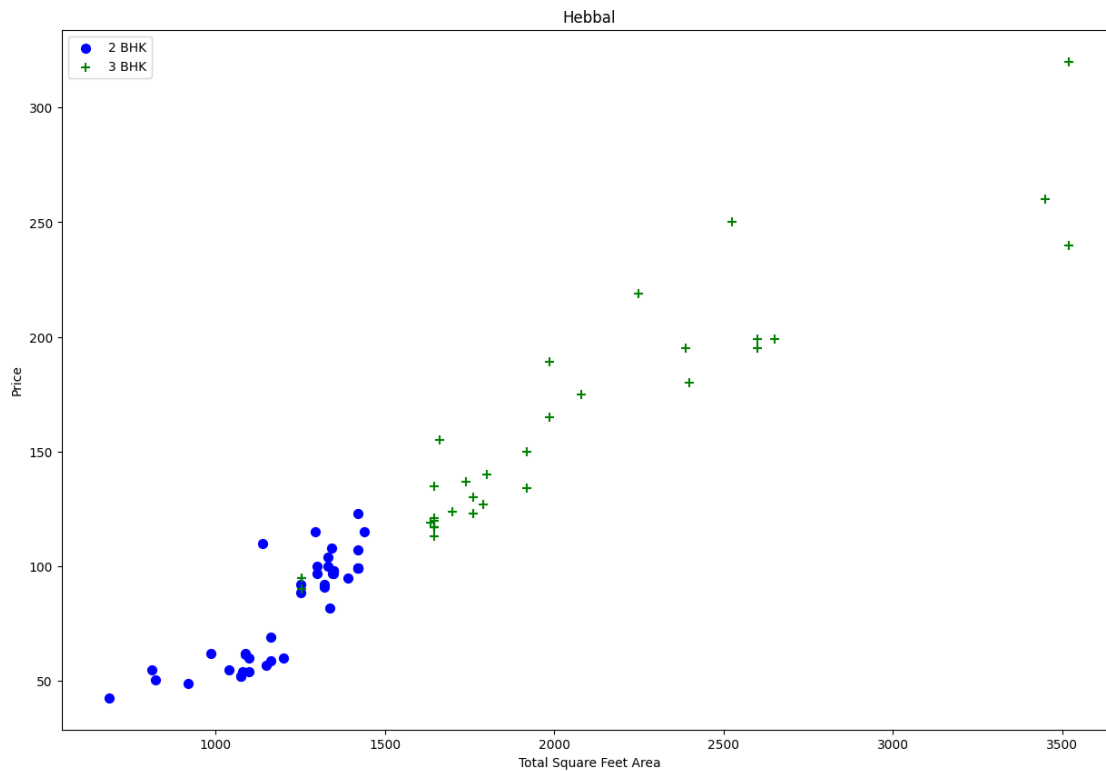
```

    return df.drop(exclude_indices,axis='index')
df8=remove_bhk_outliers(df7)
df8.shape

```

[36]: (7329, 7)

[37]: scatter_chart(df8,"Hebbal")

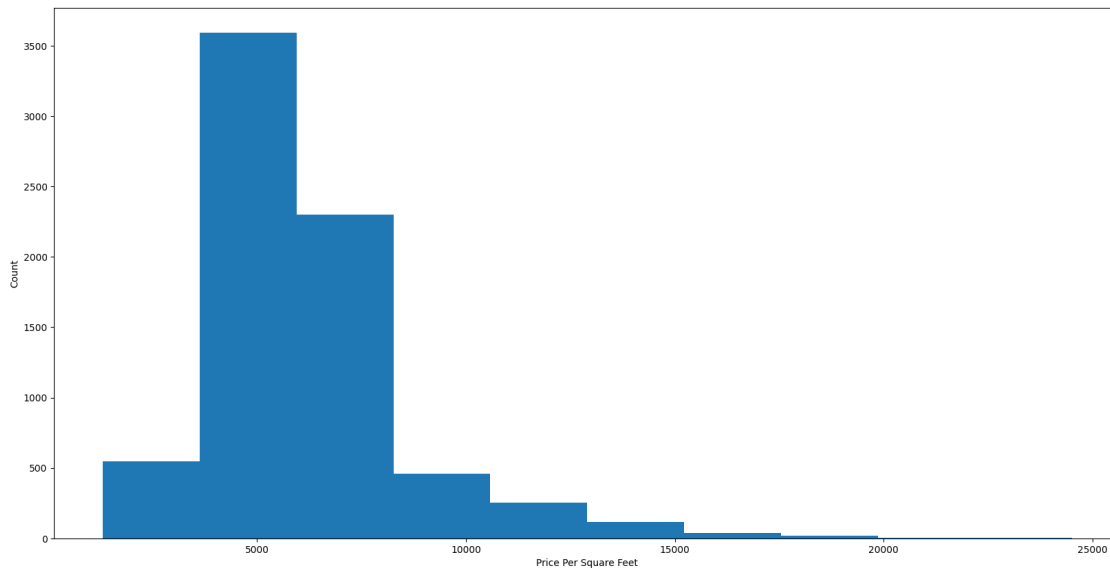


```

[38]: matplotlib.rcParams['figure.figsize']=(20,10)
plt.hist(df8.price_per_sqft)
plt.xlabel("Price Per Square Feet")
plt.ylabel("Count")

```

[38]: Text(0, 0.5, 'Count')



```
[39]: df8.bath.unique()
```

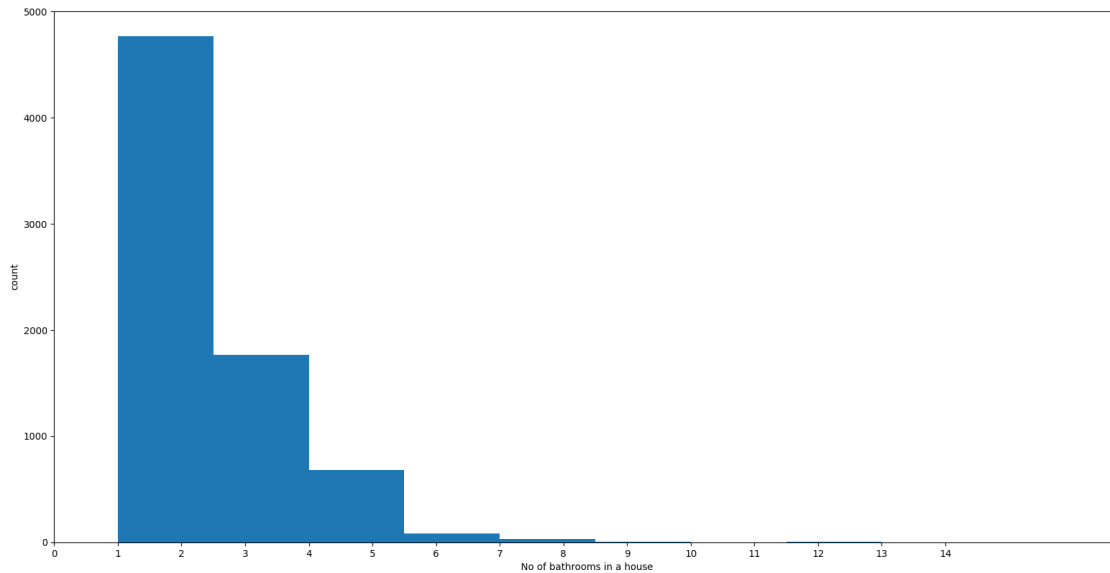
```
[39]: array([ 4.,  3.,  2.,  5.,  8.,  1.,  6.,  7.,  9., 12., 16., 13.])
```

```
[40]: df8[df8.bath>10]
```

```
[40]:
```

	location	size	total_sqft	bath	price	bhk	price_per_sqft
5277	Neeladri Nagar	10 BHK	4000.0	12.0	160.0	10	4000.000000
8486	other	10 BHK	12000.0	12.0	525.0	10	4375.000000
8575	other	16 BHK	10000.0	16.0	550.0	16	5500.000000
9308	other	11 BHK	6000.0	12.0	150.0	11	2500.000000
9639	other	13 BHK	5425.0	13.0	275.0	13	5069.124424

```
[41]: plt.hist(df8.bath)
plt.xlabel("No of bathrooms in a house")
plt.ylabel("count")
base=np.arange(0,15)
plt.xticks(base)
plt.show()
```



```
[42]: df9=df8[df8.bath<df8.bhk+2]
df9
```

```
[42]:
```

	location	size	total_sqft	bath	price	bhk	\
0	1st Block Jayanagar	4 BHK	2850.0	4.0	428.0	4	
1	1st Block Jayanagar	3 BHK	1630.0	3.0	194.0	3	
2	1st Block Jayanagar	3 BHK	1875.0	2.0	235.0	3	
3	1st Block Jayanagar	3 BHK	1200.0	2.0	130.0	3	
4	1st Block Jayanagar	2 BHK	1235.0	2.0	148.0	2	
...	
10232	other	2 BHK	1200.0	2.0	70.0	2	
10233	other	1 BHK	1800.0	1.0	200.0	1	
10236	other	2 BHK	1353.0	2.0	110.0	2	
10237	other	1 Bedroom	812.0	1.0	26.0	1	
10240	other	4 BHK	3600.0	5.0	400.0	4	

	price_per_sqft
0	15017.543860
1	11901.840491
2	12533.333333
3	10833.333333
4	11983.805668
...	...
10232	5833.333333
10233	11111.111111
10236	8130.081301
10237	3201.970443
10240	11111.111111

[7251 rows x 7 columns]

```
[43]: df10=df9.drop(["size","price_per_sqft"],axis="columns")
df10
```

```
[43]:
```

	location	total_sqft	bath	price	bhk
0	1st Block Jayanagar	2850.0	4.0	428.0	4
1	1st Block Jayanagar	1630.0	3.0	194.0	3
2	1st Block Jayanagar	1875.0	2.0	235.0	3
3	1st Block Jayanagar	1200.0	2.0	130.0	3
4	1st Block Jayanagar	1235.0	2.0	148.0	2
...
10232	other	1200.0	2.0	70.0	2
10233	other	1800.0	1.0	200.0	1
10236	other	1353.0	2.0	110.0	2
10237	other	812.0	1.0	26.0	1
10240	other	3600.0	5.0	400.0	4

[7251 rows x 5 columns]

```
[44]: df10[df10["location"]=="other"]
```

```
[44]:
```

	location	total_sqft	bath	price	bhk
7940	other	2770.0	4.0	290.0	3
7943	other	600.0	1.0	38.0	1
7946	other	1500.0	2.0	185.0	2
7947	other	840.0	2.0	45.0	2
7948	other	4395.0	3.0	240.0	3
...
10232	other	1200.0	2.0	70.0	2
10233	other	1800.0	1.0	200.0	1
10236	other	1353.0	2.0	110.0	2
10237	other	812.0	1.0	26.0	1
10240	other	3600.0	5.0	400.0	4

[1134 rows x 5 columns]

```
[45]: dummies=pd.get_dummies(df10.location,dtype=int)
```

```
[46]: df11=pd.concat([df10,dummies.drop('other',axis='columns')],axis="columns")
```

```
[47]: df11
```

```
[47]:
```

	location	total_sqft	bath	price	bhk	1st Block Jayanagar	\
0	1st Block Jayanagar	2850.0	4.0	428.0	4		1
1	1st Block Jayanagar	1630.0	3.0	194.0	3		1
2	1st Block Jayanagar	1875.0	2.0	235.0	3		1

3	1st Block Jayanagar	1200.0	2.0	130.0	3	1
4	1st Block Jayanagar	1235.0	2.0	148.0	2	1
...
10232	other	1200.0	2.0	70.0	2	0
10233	other	1800.0	1.0	200.0	1	0
10236	other	1353.0	2.0	110.0	2	0
10237	other	812.0	1.0	26.0	1	0
10240	other	3600.0	5.0	400.0	4	0

	1st Phase JP Nagar	2nd Phase Judicial Layout	2nd Stage Nagarbhavi	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	
...	
10232	0	0	0	
10233	0	0	0	
10236	0	0	0	
10237	0	0	0	
10240	0	0	0	

	5th Block Hbr Layout	...	Vijayanagar	Vishveshwarya Layout	\
0	0	...	0	0	
1	0	...	0	0	
2	0	...	0	0	
3	0	...	0	0	
4	0	...	0	0	
...	
10232	0	...	0	0	
10233	0	...	0	0	
10236	0	...	0	0	
10237	0	...	0	0	
10240	0	...	0	0	

	Vishwapriya Layout	Vittasandra	Whitefield	Yelachenahalli	Yelahanka	\
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	
...	
10232	0	0	0	0	0	
10233	0	0	0	0	0	
10236	0	0	0	0	0	
10237	0	0	0	0	0	
10240	0	0	0	0	0	

	Yelahanka New Town	Yelenahalli	Yeshwanthpur
0	0	0	0
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
...
10232	0	0	0
10233	0	0	0
10236	0	0	0
10237	0	0	0
10240	0	0	0

[7251 rows x 246 columns]

```
[48]: df12=df11.drop('location',axis="columns")
df12.head()
```

```
[48]: total_sqft  bath  price  bhk  1st Block Jayanagar  1st Phase JP Nagar \
0      2850.0    4.0  428.0    4                1                0
1      1630.0    3.0  194.0    3                1                0
2      1875.0    2.0  235.0    3                1                0
3      1200.0    2.0  130.0    3                1                0
4      1235.0    2.0  148.0    2                1                0
```

	2nd Phase Judicial Layout	2nd Stage Nagarbhavi	5th Block Hbr Layout	\
0		0	0	0
1		0	0	0
2		0	0	0
3		0	0	0
4		0	0	0

	5th Phase JP Nagar	...	Vijayanagar	Vishveshwarya Layout	\
0	0	...	0	0	
1	0	...	0	0	
2	0	...	0	0	
3	0	...	0	0	
4	0	...	0	0	

	Vishwapriya Layout	Vittasandra	Whitefield	Yelachenahalli	Yelahanka	\
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	

	Yelahanka New Town	Yelenahalli	Yeshwanthpur
0	0	0	0
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0

[5 rows x 245 columns]

```
[49]: X=df12.drop('price',axis="columns")
      y=df12["price"]
```

```
[50]: X.head()
```

```
[50]:
```

	total_sqft	bath	bhk	1st Block Jayanagar	1st Phase JP Nagar	\
0	2850.0	4.0	4	1	0	
1	1630.0	3.0	3	1	0	
2	1875.0	2.0	3	1	0	
3	1200.0	2.0	3	1	0	
4	1235.0	2.0	2	1	0	

	2nd Phase Judicial Layout	2nd Stage Nagarbhavi	5th Block Hbr Layout	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	

	5th Phase JP Nagar	6th Phase JP Nagar	... Vijayanagar	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	

	Vishveshwarya Layout	Vishwapriya Layout	Vittasandra	Whitefield	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	Yelachenahalli	Yelahanka	Yelahanka New Town	Yelenahalli	Yeshwanthpur
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0

4 0 0 0 0 0

[5 rows x 244 columns]

```
[51]: y.head()
```

```
[51]: 0    428.0
      1    194.0
      2    235.0
      3    130.0
      4    148.0
      Name: price, dtype: float64
```

```
[52]: 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)
```

```
[53]: from sklearn.linear_model import LinearRegression
      lr_clf=LinearRegression()
      lr_clf.fit(x_train,y_train)
      lr_clf.score(x_test,y_test)
```

```
[53]: 0.8452277697874374
```

```
[54]: from sklearn.model_selection import ShuffleSplit#it will randomize the samples
      from sklearn.model_selection import cross_val_score
      cv=ShuffleSplit(n_splits=5,test_size=0.2,random_state=0)
      cross_val_score(LinearRegression(),X,y,cv=cv)
```

```
[54]: array([0.82430186, 0.77166234, 0.85089567, 0.80837764, 0.83653286])
```

```
[55]: from sklearn.model_selection import GridSearchCV
      from sklearn.linear_model import Lasso
      from sklearn.tree import DecisionTreeRegressor
      def best_model(X,y):
          algos={
              'linear_regression':{
                  'model':LinearRegression(),
                  'params':{
                      'fit_intercept':[True,False]
                  }
              },
              'lasso':{
                  'model':Lasso(),
                  'params':{
                      'alpha':[1,2],
                      'selection':['random','cyclic']
                  }
              }
          }
```

```

    },
    'decision_tree':{
        'model':DecisionTreeRegressor(),
        'params':{
            'criterion':['mse','friedman_mse'],
            'splitter':['best','random']
        }
    }
}
scores=[]
cv=ShuffleSplit(n_splits=5,test_size=0.2,random_state=0)
for algo_name,config in algos.items():
    ↪gs=GridSearchCV(config['model'],config['params'],cv=cv,return_train_score=False)
    gs.fit(X,y)
    scores.append({
        'model':algo_name,
        'best_score':gs.best_score_,
        'best_params':gs.best_params_
    })
return pd.DataFrame(scores,columns=['model','best_score','best_params'])

```

```
[56]: best_model(X,y)
```

```

[56]:
      model  best_score  \
0  linear_regression    0.819001
1           lasso      0.687559
2  decision_tree      0.718272

      best_params
0  {'fit_intercept': False}
1  {'alpha': 1, 'selection': 'random'}
2  {'criterion': 'friedman_mse', 'splitter': 'best'}

```

```
[57]: X.columns
```

```

[57]: 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=244)

```

```
[58]: 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 lr_clf.predict([x])[0]
```

```
[59]: predict_price('1st Phase JP Nagar',1000,2,2)
```

```
[59]: np.float64(83.49904677204199)
```

```
[60]: import pickle  
with open("D:\\docs\\bengaluru_home_prices_model.pickle",'wb') as f:  
    pickle.dump(lr_clf,f)
```

```
[61]: import json  
columns={  
    'data_columns':[col.lower() for col in X.columns]  
}  
with open("D:\\docs\\columns.json","w") as f:  
    f.write(json.dumps(columns))
```