Bengaluru_home_price_prediction

December 28, 2024

1 Data Science Project: Predicting Home Prices in Banglore

[340]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import matplotlib
%matplotlib inline
matplotlib.rcParams["figure.figsize"]=(20,10)
import warnings
warnings.filterwarnings("ignore")
import seaborn as sns
```

2 Loading the data

[202]:
```
df1=pd.read_csv("Bengaluru_House_Data.csv")
```

[204]:
```
df1.head()
```

[204]:

	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

[206]:
```
df1.shape
```

[206]:
```
(13320, 9)
```

[208]:
```
df1.groupby("area_type")["area_type"].agg("count")
```

[208]:
```
area_type
Built-up  Area          2418
Carpet  Area              87
Plot  Area              2025
Super built-up  Area    8790
Name: area_type, dtype: int64
```

2.0.1 Droping the features that are not required to build our model

[210]:
```
df2 = df1.drop(["area_type","society","balcony","availability"],axis="columns")
df2.head()
```

[210]:

	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

3 Data Cleaning

[212]:
```
df2.isnull().sum()
```

[212]:
```
location       1
size          16
total_sqft     0
bath          73
price          0
dtype: int64
```

[214]:
```
df3 = df2.dropna()
df3.isnull().sum()
```

[214]:
```
location      0
size          0
total_sqft    0
bath          0
price         0
dtype: int64
```

[216]:
```
df3.shape
```

[216]:
```
(13246, 5)
```

[218]:
```
df3["size"].unique()
```

[218]:
```
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)
```

4 Feature Engineering

4.0.1 Add new feature(integer) for bhk (Bedrooms Hall Kitchen)

[220]:
```
df3["bhk"]=df3["size"].apply(lambda x: int(x.split(" ")[0]))
#delilmeter(" ") ke through split kre or first index ki value liye[0] delimeter␣
 ↪se
```

[222]:
```
df3.head()
```

[222]:

	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

[224]:
```
df3["bhk"].unique()
```

[224]:
```
array([ 2,  4,  3,  6,  1,  8,  7,  5, 11,  9, 27, 10, 19, 16, 43, 14, 12,
       13, 18], dtype=int64)
```

[226]:
```
df3["total_sqft"].unique()
```

[226]:
```
array(['1056', '2600', '1440', …, '1133 - 1384', '774', '4689'],
      dtype=object)
```

[228]:
```
def is_float(x):
    try:
        float(x)
    except:
        return False
    return True
```

[230]:
```
df3[~df3["total_sqft"].apply(is_float)].head()
```

[230]:

	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
```

[232]:
```
def convert_sqrt_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

#jaha bhi value (123-1232) ye form me h we are taking the avg of it using the␣
 ↪above func
```

[234]:
```
convert_sqrt_to_num("2100")
```

[234]:
```
2100.0
```

[236]:
```
convert_sqrt_to_num("2100 - 2850")
```

[236]:
```
2475.0
```

[238]:
```
df4=df3.copy()
df4["total_sqft"]=df4["total_sqft"].apply(convert_sqrt_to_num)
df4.head()
```

[238]:

	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

[240]:
```
df4.loc[30]    # checking the value of total_sqrt for a particular index
```

[240]:
```
location      Yelahanka
size              4 BHK
total_sqft       2475.0
bath                4.0
price             186.0
bhk                   4
Name: 30, dtype: object
```

[242]:
```
(2100+2850)/2
```

[242]:
```
2475.0
```

5 Feature Engineering

[244]:
```
df5= df4.copy()
df5["price_per_sqrft"] = df5["price"]*100000 / df5["total_sqft"]
df5.head()
```

[244]:

	location	size	total_sqft	bath	price	bhk	price_per_sqrft
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.07	2	3699.810606
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4	4615.384615
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3	4305.555556
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.00	3	6245.890861
4	Kothanur	2 BHK	1200.0	2.0	51.00	2	4250.000000

[246]:
```
len(df5.location.unique())
```

[246]:
```
1304
```

5.1 Examine locations which is a categorical variable. We need to apply dimensionality reduction technique here to reduce number of locations

[248]:
```
df5.location = df5.location.apply(lambda x : x.strip())    #removing the extra␣
 ↪space from the loc col
location_stats  = df5.groupby("location")["location"].agg("count").
 ↪sort_values(ascending = False)
location_stats
```

[248]:
```
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
```

[250]:
```
len(location_stats[location_stats<=10])
```

[250]:
```
1052
```

6 Dimensionality Reduction

[357]:
```
location_stats_less_than_10 = location_stats[location_stats<=10]
location_stats_less_than_10
```

[357]:
```
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
```

[359]:
```
len(df5.location.unique())
```

[359]:
```
242
```

[256]:
```
df5.location = df5.location.apply(lambda x : "other" if x in␣
 ↪location_stats_less_than_10 else x)
len(df5.location.unique())
```

[256]:
```
242
```

[258]:
```
df5.head(10)
```

[258]:

	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

```
   price_per_sqrft
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
```

7 Outlier Removal

[260]:
```
df5[df5.total_sqft/df5.bhk<300].head()     #here we r removing the outliers ex
↪if toto_sqft/bhk is less than 300 we should remove it . this are all data
↪error
```

[260]:
```
               location       size  total_sqft  bath  price  bhk  \
9                 other  6 Bedroom      1020.0   6.0  370.0    6
45          HSR Layout  8 Bedroom       600.0   9.0  200.0    8
58        Murugeshpalya  6 Bedroom      1407.0   4.0  150.0    6
68  Devarachikkanahalli  8 Bedroom      1350.0   7.0   85.0    8
70                other  3 Bedroom       500.0   3.0  100.0    3

    price_per_sqrft
9      36274.509804
45     33333.333333
58     10660.980810
68      6296.296296
70     20000.000000
```

7.0.1 Check above data points. We have 6 bhk apartment with 1020 sqft. Another one is 8 bhk and total sqft is 600. These are clear data errors that can be removed safely

[262]:
```
df5.shape
```

[262]:
```
(13246, 7)
```

[264]:
```
df6 = df5[~(df5.total_sqft/df5.bhk<300)]
df6.shape
# removing the outliers
```

[264]:
```
(12502, 7)
```

8 Outlier Removal Using Standard Deviation and Mean

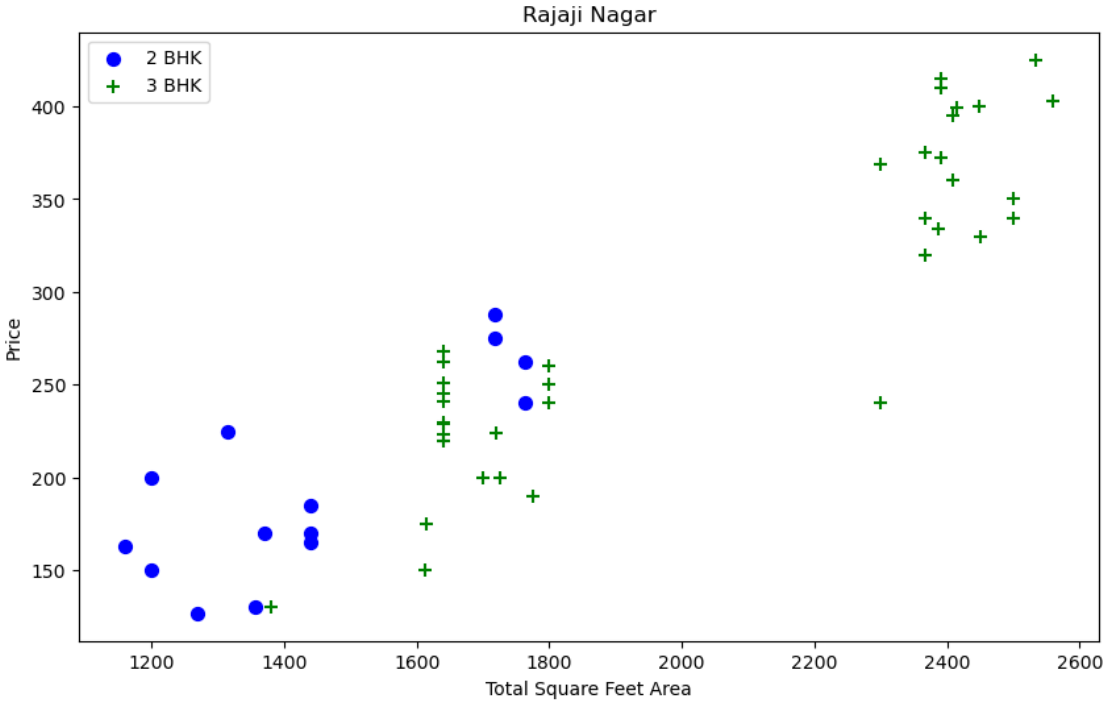
[364]:
```
df6.price_per_sqrft.describe()
```

[364]:
```
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_sqrft, dtype: float64
```

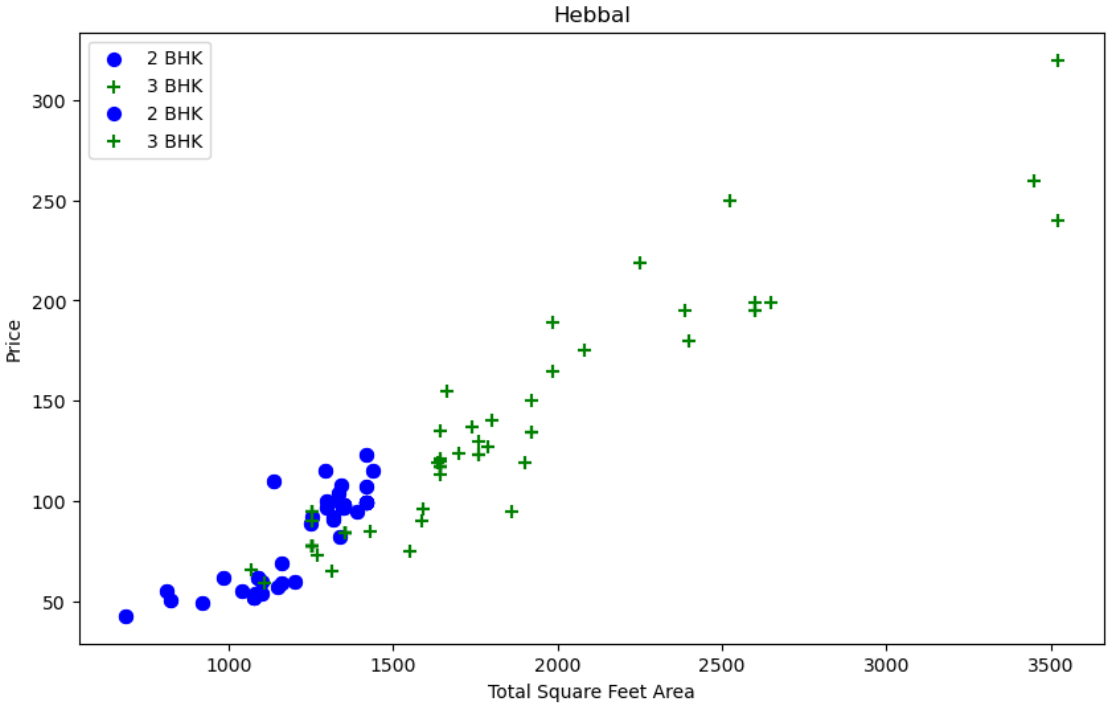
[366]:
```
def removing_pps_outliers(df):
    df_out = pd.DataFrame()
    for key,subdf in df.groupby("location"):
        m = np.mean(subdf.price_per_sqrft)
        st = np.std(subdf.price_per_sqrft)
        reduce_df = subdf[(subdf.price_per_sqrft>(m-st)) & (subdf.
 ↪price_per_sqrft<=(m+st))]
        df_out = pd.concat([df_out,reduce_df],ignore_index=True)
    return df_out

df7 = removing_pps_outliers(df6)
df7.shape
```

[366]:
```
(10241, 7)
```

[368]:
```
def plot_scatter_chart(df,location):
    bhk2 = df[(df.location==location)  & (df.bhk==2)]
    bhk3 = df[(df.location==location)  & (df.bhk==3)]
    matplotlib.rcParams["figure.figsize"] = (10,6)
    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()

plot_scatter_chart(df7 , "Rajaji Nagar")

plt.show()
```



[372]:
```
plot_scatter_chart(df7,"Hebbal")
plt.show()
```



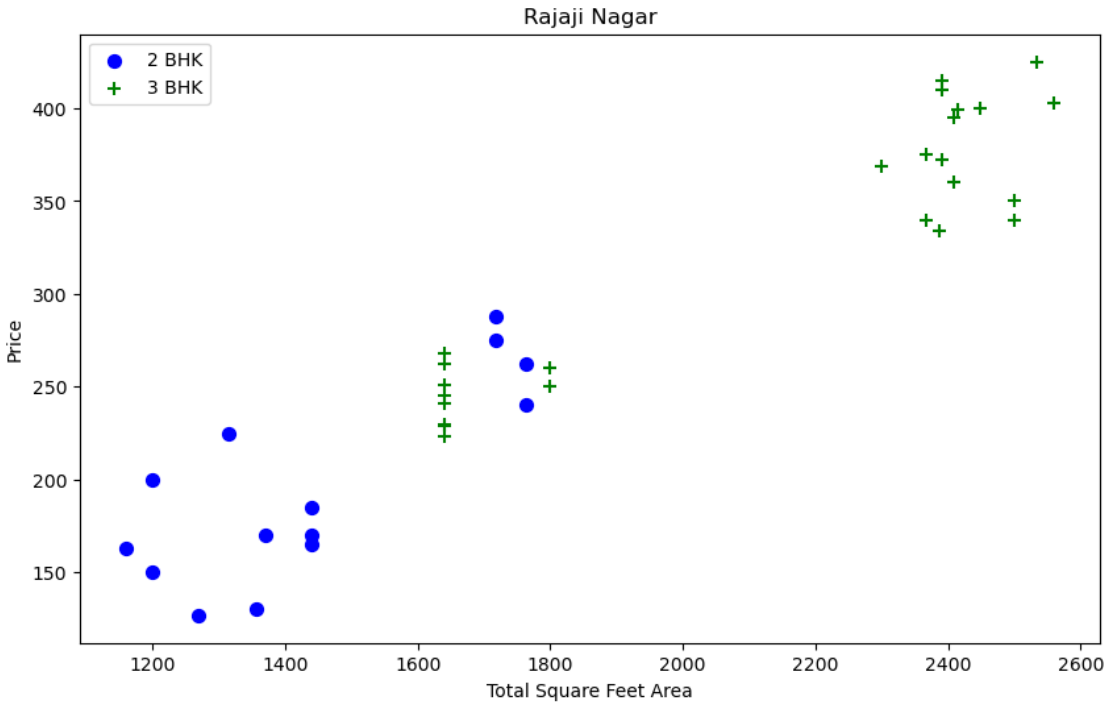
We should also remove properties where for same location, the price of (for example) 3 bedroom apartment is less than 2 bedroom apartment (with same square ft area). What we will do is for a given location, we will build a dictionary of stats per bhk, i.e.

{ '1' : { 'mean': 4000, 'std: 2000, 'count': 34 }, '2' : { 'mean': 4300, 'std: 2300, 'count': 22 }, } Now we can remove those 2 BHK apartments whose price_per_sqft is less than mean price_per_sqft of 1 BHK apartment

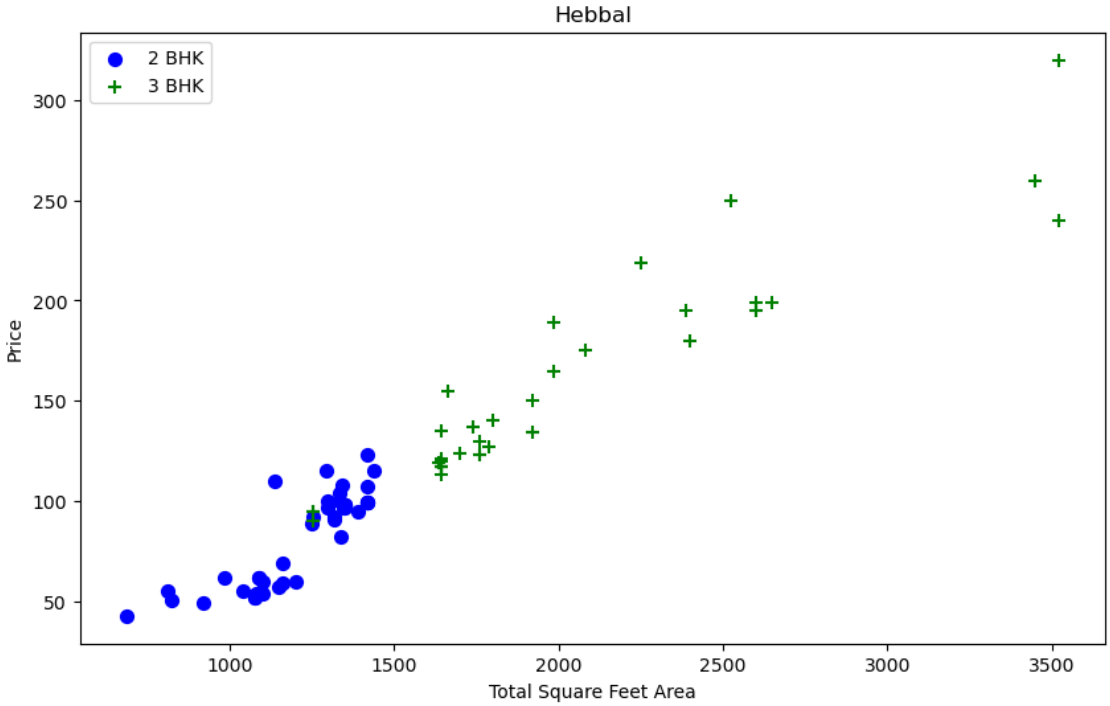
[384]:
```
# in this func we r trying to remove the outliers whose 2bhk price is more than␣
 ↪the 3 bhk price

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_sqrft),
                "std":np.std(bhk_df.price_per_sqrft),
                "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_sqrft<(stats["mean"])].index.values)
    return df.drop(exclude_indices,axis="index")

df8 = remove_bhk_outliers(df7)
df8.shape
```

[384]:
```
(7329, 7)
```

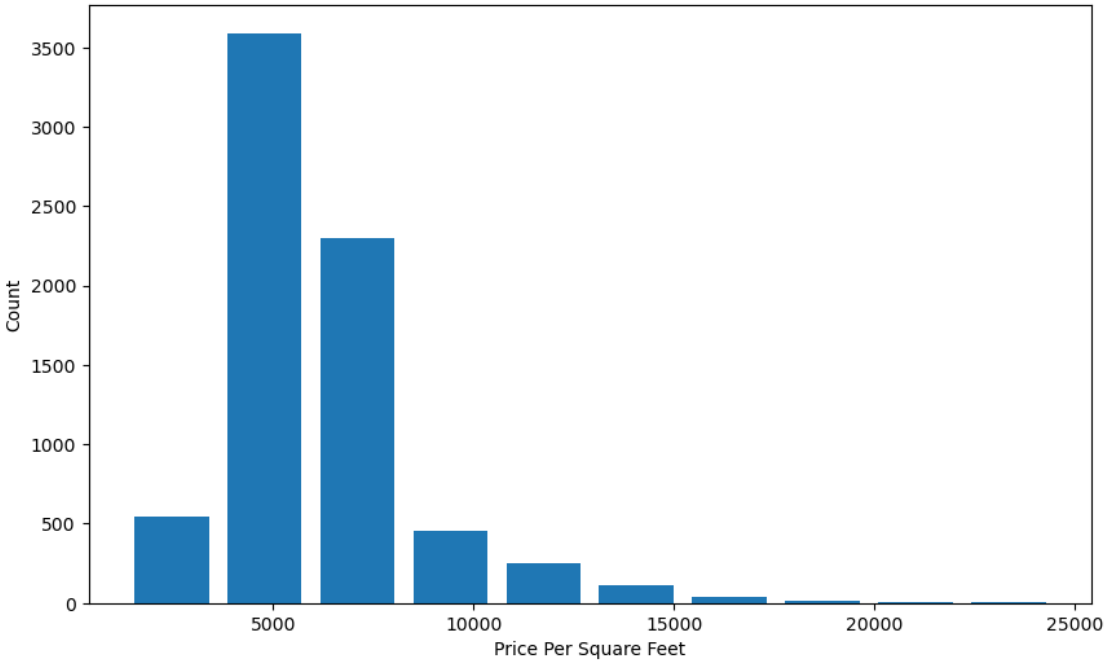
[386]:
```
plot_scatter_chart(df8 ,"Rajaji Nagar")
plt.show()
```



[388]:
```
plot_scatter_chart(df8 ,"Hebbal")
plt.show()
```



[330]:
```
plt.hist(df8.price_per_sqrft,rwidth=0.8)
plt.xlabel("Price Per Square Feet")
plt.ylabel("Count")

plt.show()
```



[282]:
```
df8.bath.unique()
```

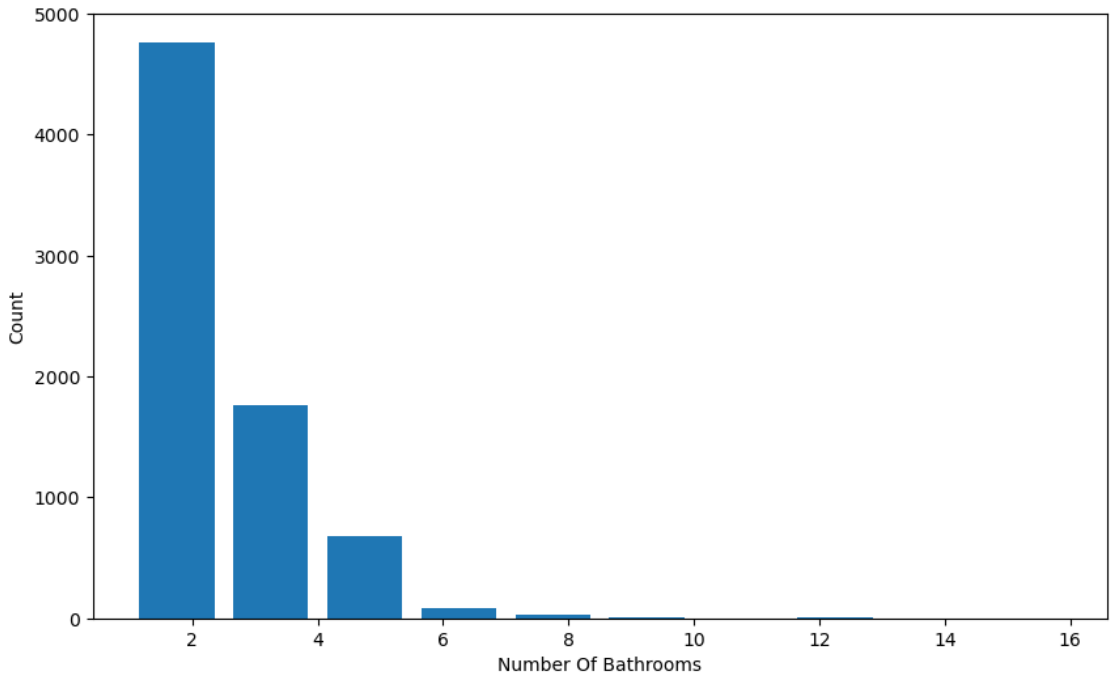
[282]:
```
array([ 4.,  3.,  2.,  5.,  8.,  1.,  6.,  7.,  9., 12., 16., 13.])
```

[284]:
```
df8[df8.bath>10]
```

[284]:

	location	size	total_sqft	bath	price	bhk	price_per_sqrft
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

[336]:
```
plt.hist(df8.bath,rwidth=0.8)
plt.xlabel("Number Of Bathrooms")
plt.ylabel("Count")
plt.show()
```



It is unusual to have 2 more bathrooms than number of bedrooms in a home

[288]:
```
df8[df8.bath>df8.bhk+2]
```

[288]:

	location	size	total_sqft	bath	price	bhk	price_per_sqrft
1626	Chikkabanavar	4 Bedroom	2460.0	7.0	80.0	4	3252.032520
5238	Nagasandra	4 Bedroom	7000.0	8.0	450.0	4	6428.571429
6711	Thanisandra	3 BHK	1806.0	6.0	116.0	3	6423.034330
8411	other	6 BHK	11338.0	9.0	1000.0	6	8819.897689

[290]:
```
df8.shape
```

[290]: (7329, 7)

[292]:
```
df9 = df8[df8.bath<df8.bhk+2]
df9.shape
```

[292]: (7251, 7)

[294]:
```
df10 = df9.drop(["size","price_per_sqrft"],axis="columns")
df10.head()
```

[294]:

	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
```

8.1 Use One Hot Encoding For Location

[296]:
```
dummies = pd.get_dummies(df10.location)
dummies
```

[296]:
```
       1st Block Jayanagar  1st Phase JP Nagar  2nd Phase Judicial Layout  \
0                     True               False                      False
1                     True               False                      False
2                     True               False                      False
3                     True               False                      False
4                     True               False                      False
…                      …                   …                          …
10232                False               False                      False
10233                False               False                      False
10236                False               False                      False
10237                False               False                      False
10240                False               False                      False

       2nd Stage Nagarbhavi  5th Block Hbr Layout  5th Phase JP Nagar  \
0                     False                 False               False
1                     False                 False               False
2                     False                 False               False
3                     False                 False               False
4                     False                 False               False
…                      …                     …                   …
10232                 False                 False               False
10233                 False                 False               False
10236                 False                 False               False
10237                 False                 False               False
10240                 False                 False               False

       6th Phase JP Nagar  7th Phase JP Nagar  8th Phase JP Nagar  \
0                   False               False               False
1                   False               False               False
2                   False               False               False
3                   False               False               False
4                   False               False               False
…                    …                   …                   …
10232               False               False               False
10233               False               False               False
10236               False               False               False
10237               False               False               False
10240               False               False               False
```

```
       9th Phase JP Nagar  …  Vishveshwarya Layout  Vishwapriya Layout  \
0                      False  …                 False               False
1                      False  …                 False               False
2                      False  …                 False               False
3                      False  …                 False               False
4                      False  …                 False               False
…                          …  …                   …                   …
10232                  False  …                 False               False
10233                  False  …                 False               False
10236                  False  …                 False               False
10237                  False  …                 False               False
10240                  False  …                 False               False

       Vittasandra  Whitefield  Yelachenahalli  Yelahanka  Yelahanka New Town  \
0            False       False           False      False               False
1            False       False           False      False               False
2            False       False           False      False               False
3            False       False           False      False               False
4            False       False           False      False               False
…                …           …               …          …                   …
10232        False       False           False      False               False
10233        False       False           False      False               False
10236        False       False           False      False               False
10237        False       False           False      False               False
10240        False       False           False      False               False

       Yelenahalli  Yeshwanthpur  other
0            False         False  False
1            False         False  False
2            False         False  False
3            False         False  False
4            False         False  False
…                …             …      …
10232        False         False   True
10233        False         False   True
10236        False         False   True
10237        False         False   True
10240        False         False   True

[7251 rows x 242 columns]
```

[298]:
```
df11 = pd.concat([df10,dummies.drop("other",axis="columns")],axis="columns")
df11.head()
```

[298]:
```
              location  total_sqft  bath  price  bhk  1st Block Jayanagar  \
0  1st Block Jayanagar      2850.0   4.0  428.0    4                 True
1  1st Block Jayanagar      1630.0   3.0  194.0    3                 True
```

```
2  1st Block Jayanagar      1875.0   2.0  235.0    3                 True
3  1st Block Jayanagar      1200.0   2.0  130.0    3                 True
4  1st Block Jayanagar      1235.0   2.0  148.0    2                 True

   1st Phase JP Nagar  2nd Phase Judicial Layout  2nd Stage Nagarbhavi  \
0               False                      False                 False
1               False                      False                 False
2               False                      False                 False
3               False                      False                 False
4               False                      False                 False

   5th Block Hbr Layout  …  Vijayanagar  Vishveshwarya Layout  \
0                 False  …        False                 False
1                 False  …        False                 False
2                 False  …        False                 False
3                 False  …        False                 False
4                 False  …        False                 False

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

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

[5 rows x 246 columns]
```

[300]:
```
df12 = df11.drop("location",axis="columns")
df12.head()
```

[300]:
```
   total_sqft  bath  price  bhk  1st Block Jayanagar  1st Phase JP Nagar  \
0      2850.0   4.0  428.0    4                 True               False
1      1630.0   3.0  194.0    3                 True               False
2      1875.0   2.0  235.0    3                 True               False
3      1200.0   2.0  130.0    3                 True               False
4      1235.0   2.0  148.0    2                 True               False

   2nd Phase Judicial Layout  2nd Stage Nagarbhavi  5th Block Hbr Layout  \
0                      False                 False                 False
1                      False                 False                 False
```

```
2                     False                False                 False
3                     False                False                 False
4                     False                False                 False

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

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

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

[5 rows x 245 columns]
```

8.2 Build a Model Now……..

[302]:
```
df12.shape
```

[302]:
```
(7251, 245)
```

[304]:
```
X=df12.drop("price",axis="columns")
X.head()

#separating the dependent and the independent variable
```

[304]:
```
   total_sqft  bath  bhk  1st Block Jayanagar  1st Phase JP Nagar  \
0      2850.0   4.0    4                 True               False
1      1630.0   3.0    3                 True               False
2      1875.0   2.0    3                 True               False
3      1200.0   2.0    3                 True               False
4      1235.0   2.0    2                 True               False

   2nd Phase Judicial Layout  2nd Stage Nagarbhavi  5th Block Hbr Layout  \
0                      False                 False                 False
```

```
1                     False                False                False
2                     False                False                False
3                     False                False                False
4                     False                False                False

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

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

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

[5 rows x 244 columns]
```

[306]:
```
y=df12.price
y.head()
```

[306]:
```
0    428.0
1    194.0
2    235.0
3    130.0
4    148.0
Name: price, dtype: float64
```

[308]:
```
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)
```

[310]:
```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(x_train,y_train)
lr.score(x_test,y_test)
```

[310]: 0.8452277697873772

8.3 Use K Fold cross validation to measure accuracy of our LinearRegression model

[312]:
```
from sklearn.model_selection import ShuffleSplit
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)
```

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

8.4 Find best model using GridSearchCV

[314]:
```
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Lasso
from sklearn.tree import DecisionTreeRegressor
import pandas as pd
from sklearn.model_selection import ShuffleSplit

def find_best_model_using_gridsearchcv(X, y):
    algos = {
        "linear_regression": {
        "model": LinearRegression(fit_intercept=True, copy_X=True,␣
 ↪positive=True),
        "params": {
            "n_jobs": [1, 2, 4]
            }
        },
        "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"])

find_best_model_using_gridsearchcv(X, y)
```

[314]:
```
                model  best_score  \
0  linear_regression    0.788689
1              lasso    0.687429
2      decision_tree    0.715706

                                        best_params
0                                    {'n_jobs': 1}
1                {'alpha': 1, 'selection': 'cyclic'}
2  {'criterion': 'friedman_mse', 'splitter': 'best'}
```

8.5 Test the model for few properties

[316]:
```
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.predict([x])[0]
```

[318]:
```
predict_price('1st Phase JP Nagar',1000, 2, 2)
```

[318]:
```
83.49904676965245
```

[320]:
```
predict_price('1st Phase JP Nagar',1000, 3, 3)
```

[320]:
```
86.80519394990591
```

[322]:
```
predict_price('Indira Nagar',1000, 2, 2)
```

[322]:
```
181.27815484010569
```

[324]:
```
predict_price('Indira Nagar',1000, 3, 3)
```

[324]:
```
184.5843020203592
```

8.6 Export the tested model to a pickle file

[267]:
```
import pickle
with open('Bengaluru_home_price_prediction.pickle','wb') as f:
    pickle.dump(lr,f)
```

8.7 Export location and column information to a file that will be useful later on in our prediction application

[269]:
```
import json
columns = {
    'data_columns' : [col.lower() for col in X.columns]
}
with open("columns.json","w") as f:
    f.write(json.dumps(columns))
```

[ ]:

[ ]:

[ ]: