Python Mini Project: Predicting Real Estate Prices in Bangalore

Dataset : Bengaluru_House_Data

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

Data Load: Load bangalore home prices into a dataframe

```
df1 = pd.read_csv("Bengaluru_House_Data.csv")
df1.head()
```

	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

```
df1.shape
```

```
(13320, 9)
```

```
df1.columns
```

```
Index(['area_type', 'availability', 'location', 'size', 'society',
       'total_sqft', 'bath', 'balcony', 'price'],
      dtype='object')
```

```
df1['area_type'].unique()
```

```
array(['Super built-up  Area', 'Plot  Area', 'Built-up  Area',
       'Carpet  Area'], dtype=object)
```

```
df1['area_type'].value_counts()
```

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

Drop the features which are not required to build our model.

```
df2 = df1.drop(['area_type','society','balcony','availability'],axis='columns')
df2.shape
```

```
(13320, 5)
```

Data Cleaning : Handling NA Values

```
df2.isnull().sum()
```

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

```
df2.shape
```

```
(13320, 5)
```

```
df3 = df2.dropna()
df3.isnull().sum()
```

```
location      0
size          0
total_sqft    0
bath          0
price         0
dtype: int64
```

```
df3.shape
```

```
(13246, 5)
```

Feature Engineering :

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

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

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

Explore total_sqft column

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

```
is_float(4.0)
```

```
True
```

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

	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

Above shows that total_sqft can be a range (e.g. 2100-2850). For such cases we can just take average of min and max value in the range. There are other cases such as 34.46Sq. Meter and 4125Perch which one can convert to square ft using unit conversion or just drop such cases and try to keep things simple.

```
def convert_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
```

```
convert_sqft_to_num('2560')
```

```
2560.0
```

```
convert_sqft_to_num('2100 - 2850')
```

```
2475.0
```

Start coding or generate with AI.

Start coding or generate with AI.

```
df4 = df3.copy()
df4.total_sqft = df4.total_sqft.apply(convert_sqft_to_num)
df4 = df4[df4.total_sqft.notnull()]
df4.head(10)
```

	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	Gandhi Bazar	6 Bedroom	1020.0	6.0	370.00	6

For the below row it shows 2475.0 as an average of the range 2100 - 2850

```
df4.loc[30]
```

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

Feature Engineering

Add new feature - price per square feet

```
df5 = df4.copy()
df5['price_per_sqft'] = df5['price']*100000/df5['total_sqft']
df5.head(10)
```

	location	size	total_sqft	bath	price	bhk	price_per_sqft
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
5	Whitefield	2 BHK	1170.0	2.0	38.00	2	3247.863248
6	Old Airport Road	4 BHK	2732.0	4.0	204.00	4	7467.057101
7	Rajaji Nagar	4 BHK	3300.0	4.0	600.00	4	18181.818182
8	Marathahalli	3 BHK	1310.0	3.0	63.25	3	4828.244275
9	Gandhi Bazar	6 Bedroom	1020.0	6.0	370.00	6	36274.509804

```
df5['price_per_sqft'].describe()
```

```
count    1.320000e+04
mean     7.920759e+03
std      1.067272e+05
min      2.678298e+02
25%      4.267701e+03
50%      5.438331e+03
75%      7.317073e+03
max      1.200000e+07
Name: price_per_sqft, dtype: float64
```

Examine locations which is a categorical variable.

We are applying dimensionality reduction technique here to reduce number of locations.

```
df5.location = df5.location.apply(lambda x: x.strip())
location_stats = df5['location'].value_counts(ascending=False)
location_stats
```

```
Whitefield                 533
Sarjapur  Road             392
Electronic City            304
Kanakpura Road             264
Thanisandra                235
                          ...
Ramanagara Channapatna       1
Badrappa Layout              1
Chikkaballapur               1
Nagarbhavi  BDA Complex      1
Pillanna Gardens             1
Name: location, Length: 1287, dtype: int64
```

```
len(location_stats)
```

```
1287
```

```
len(location_stats[location_stats>10])
```

```
240
```

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

```
1047
```

Dimensionality Reduction

Any location having less than 10 data points should be tagged as "other" location. In this way number of categories can be reduced by huge amount. Later on when we do one hot encoding, it will help us with having fewer dummy columns

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

```
Ganga Nagar                10
Kalkere                    10
Sector 1 HSR Layout        10
Thyagaraja Nagar           10
Basapura                   10
                           ..
```

```
Ramanagara Channapatna      1
Badrappa Layout             1
Chikkaballapur              1
Nagarbhavi  BDA Complex     1
Pillanna Gardens            1
Name: location, Length: 1047, dtype: int64
```

```
len(df5.location.unique())
```

```
1287
```

```
df5.location = df5.location.apply(lambda x: 'Other' if x in location_stats_less_than_10 else x)
len(df5.location.unique())
```

```
241
```

```
df5.head(15)
```

	location	size	total_sqft	bath	price	bhk	price_per_sqft
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
5	Whitefield	2 BHK	1170.0	2.0	38.00	2	3247.863248
6	Old Airport Road	4 BHK	2732.0	4.0	204.00	4	7467.057101
7	Rajaji Nagar	4 BHK	3300.0	4.0	600.00	4	18181.818182
8	Marathahalli	3 BHK	1310.0	3.0	63.25	3	4828.244275
9	Other	6 Bedroom	1020.0	6.0	370.00	6	36274.509804
10	Whitefield	3 BHK	1800.0	2.0	70.00	3	3888.888889
11	Whitefield	4 Bedroom	2785.0	5.0	295.00	4	10592.459605
12	7th Phase JP Nagar	2 BHK	1000.0	2.0	38.00	2	3800.000000
13	Gottigere	2 BHK	1100.0	2.0	40.00	2	3636.363636
14	Sarjapur	3 Bedroom	2250.0	3.0	148.00	3	6577.777778

Outlier Removal for Business

As a data scientist when you have a conversation with your business manager (who has expertise in real estate), he will tell you that normally square ft per bedroom is 300 (i.e. 2 bhk apartment is minimum 600 sqft. If you have for example 400 sqft apartment with 2 bhk than that seems suspicious and can be removed as an outlier. We will remove such outliers by keeping our minimum thresold per bhk to be 300 sqft

```
df5[df5.total_sqft/df5.bhk<300].head(10)
```

	location	size	total_sqft	bath	price	bhk	price_per_sqft
9	Other	6 Bedroom	1020.0	6.0	370.0	6	36274.509804
45	HSR Layout	8 Bedroom	600.0	9.0	200.0	8	33333.333333
58	Murugeshpalya	6 Bedroom	1407.0	4.0	150.0	6	10660.980810
68	Devarachikkanahalli	8 Bedroom	1350.0	7.0	85.0	8	6296.296296
70	Other	3 Bedroom	500.0	3.0	100.0	3	20000.000000
78	Kaval Byrasandra	2 BHK	460.0	1.0	22.0	2	4782.608696
89	Rajaji Nagar	6 Bedroom	710.0	6.0	160.0	6	22535.211268
119	Hennur Road	2 Bedroom	276.0	3.0	23.0	2	8333.333333
129	Vishwapriya Layout	7 Bedroom	950.0	7.0	115.0	7	12105.263158
149	Other	6 Bedroom	1034.0	5.0	185.0	6	17891.682785

```
df5.shape
```

```
(13200, 7)
```

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

```
(12456, 7)
```

Outlier Removal Using Standard Deviation and Mean

```
df6.price_per_sqft.describe()
```

```
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
```

Here we find that min price per sqft is 267 whereas max is 12000000, this is a wide variation in property prices. We should remove outliers per location using mean and one standard deviation

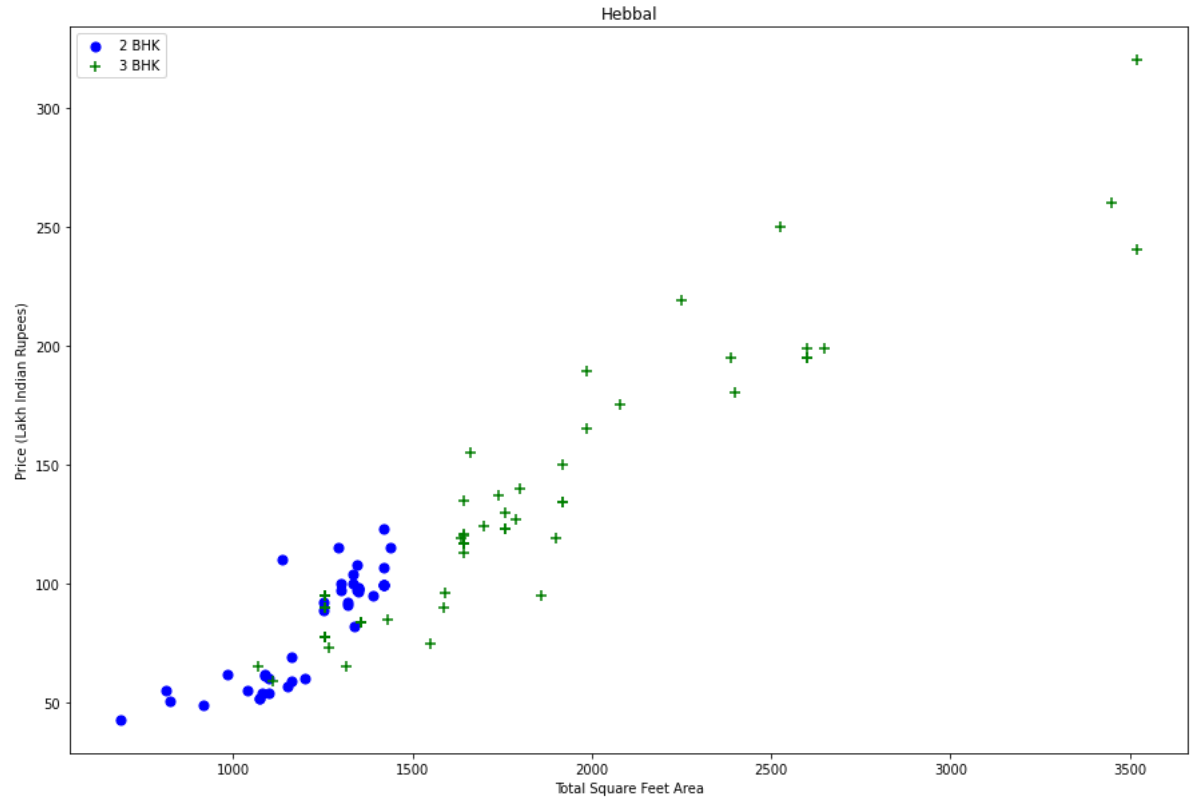
```
def remove_pps_outliers(df):
    df_out = pd.DataFrame()
    for key, subdf in df.groupby('location'):
        m = np.mean(subdf.price_per_sqft)
        st = np.std(subdf.price_per_sqft)
        reduced_df = subdf[(subdf.price_per_sqft>(m-st)) & (subdf.price_per_sqft<=(m+st))]
        df_out = pd.concat([df_out,reduced_df],ignore_index=True)
    return df_out
df7 = remove_pps_outliers(df6)
df7.shape
```

```
(10242, 7)
```

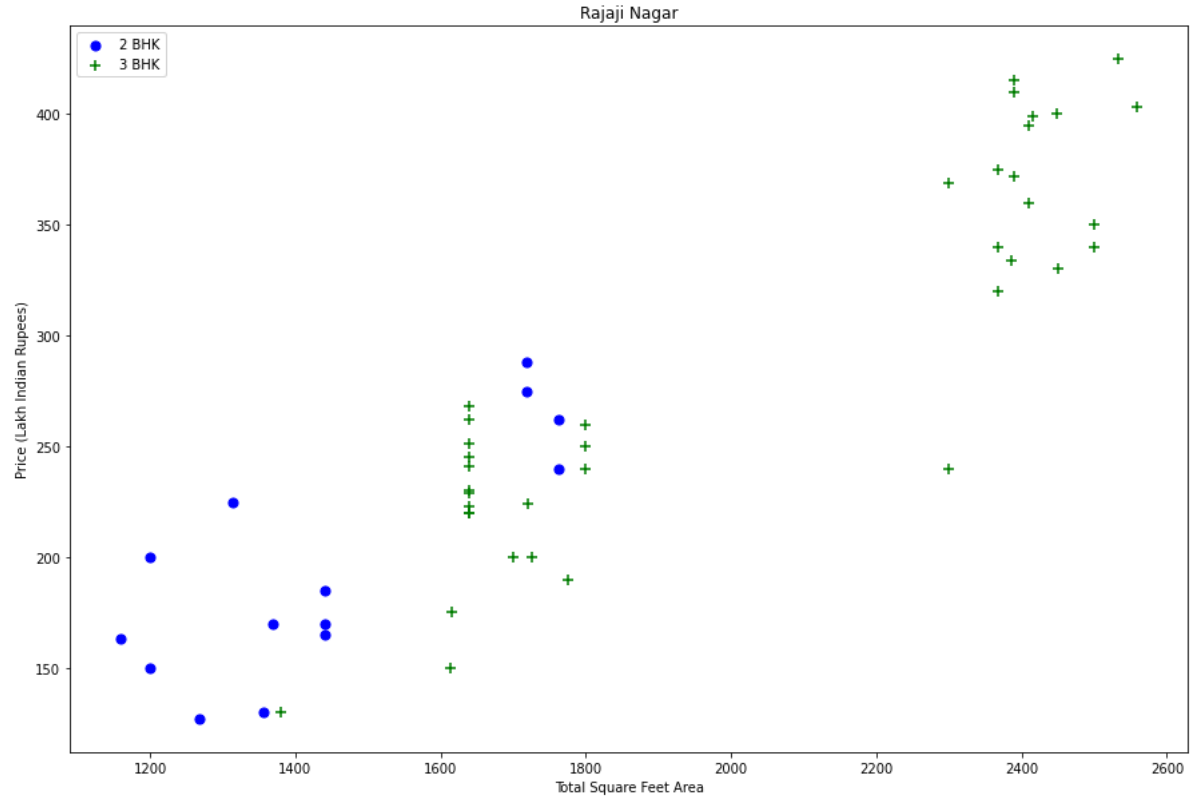
Now Let's check if for a given location how does the 2 BHK and 3 BHK property prices look like

```
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'] = (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 (Lakh Indian Rupees)")
    plt.title(location)
    plt.legend()
```

```
plot_scatter_chart(df7,"Hebbal")
```




```
plot_scatter_chart(df7,"Rajaji Nagar")
```



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).

So for a given location, we will try to build a dictionary of stats per bhk, i.e.

{ '1' : { 'mean': 4000, 'std: 2000, 'count': 34 }, '2' : { 'mean': 4300, 'std: 2300, 'count': 22 }, }

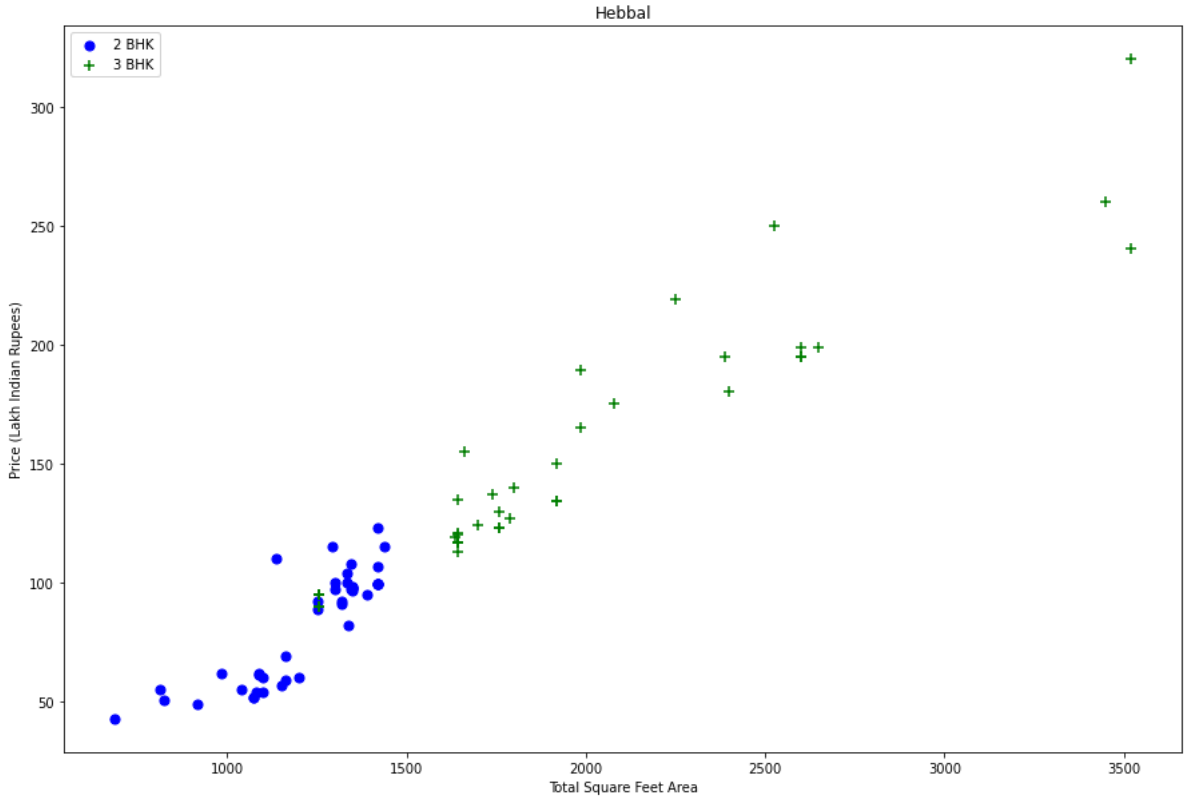
Now we should remove those 2 BHK apartments whose price_per_sqft is less than mean price_per_sqft of 1 BHK apartment

```
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')
```

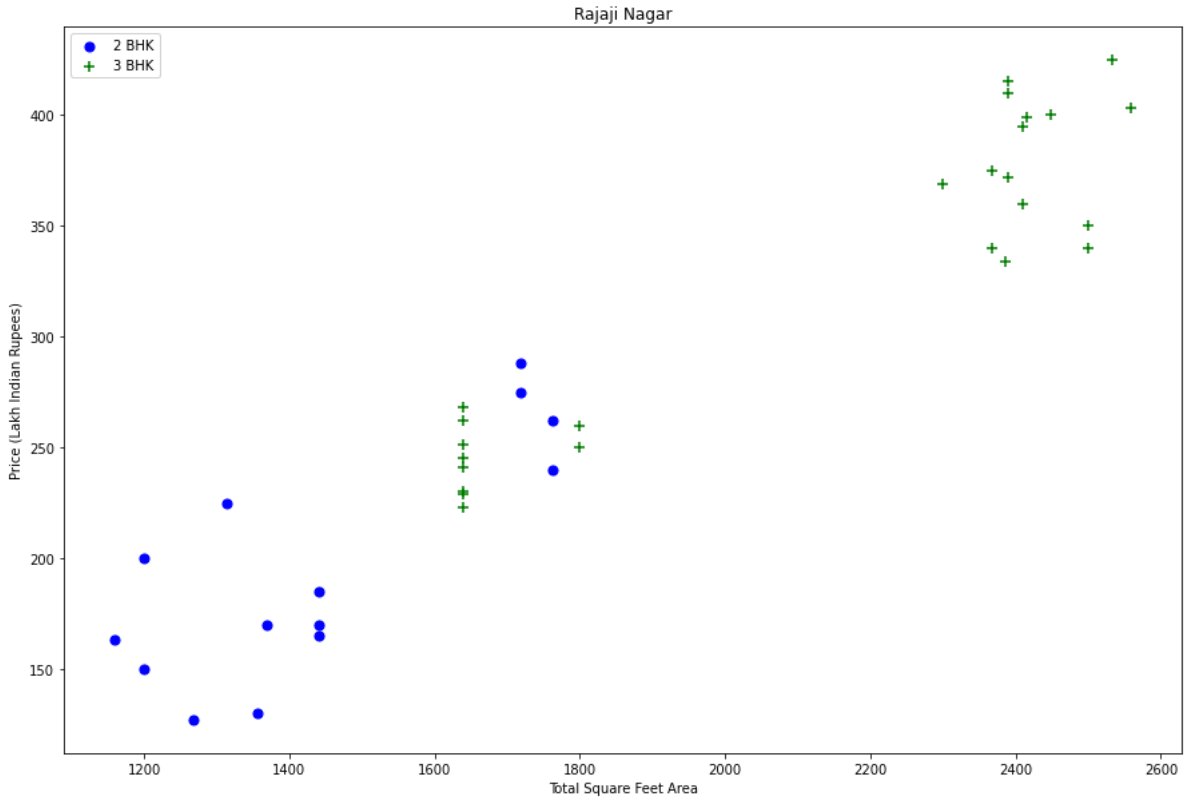
```
df8 = remove_bhk_outliers(df7)
df8.shape
```

```
(7317, 7)
```

```
plot_scatter_chart(df8,"Hebbal")
```

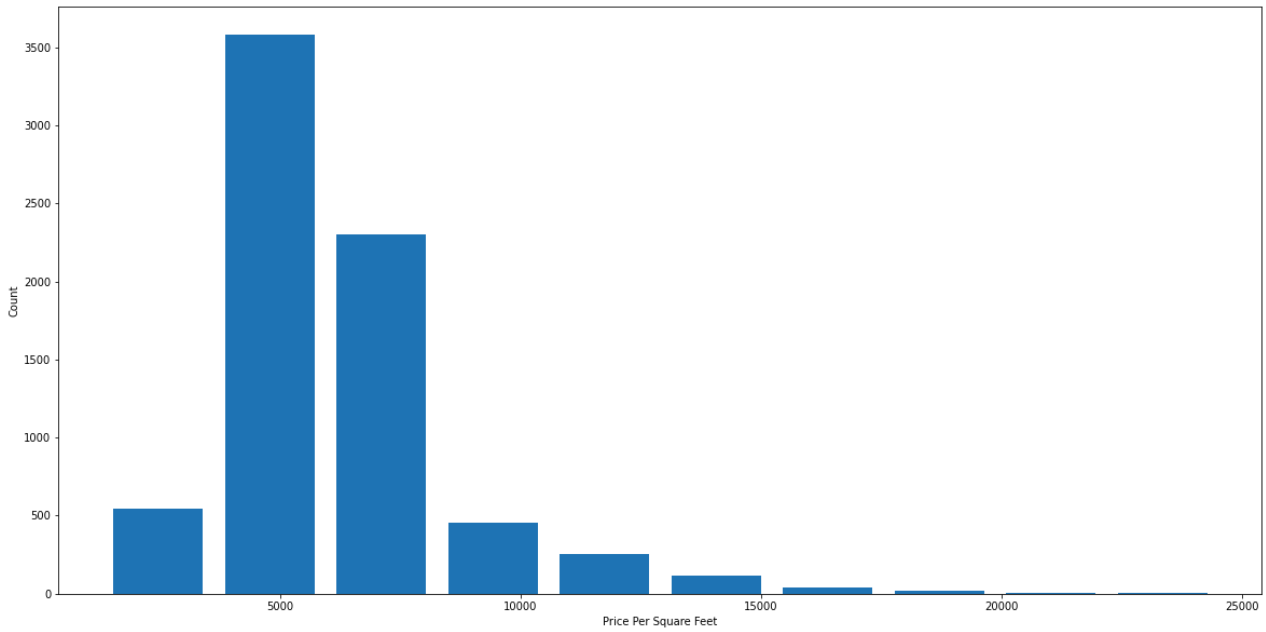



```
plot_scatter_chart(df8,"Rajaji Nagar")
```




```
import matplotlib
matplotlib.rcParams["figure.figsize"] = (20,10)
plt.hist(df8.price_per_sqft,rwidth=0.8)
plt.xlabel("Price Per Square Feet")
plt.ylabel("Count")
```

```
Text(0, 0.5, 'Count')
```



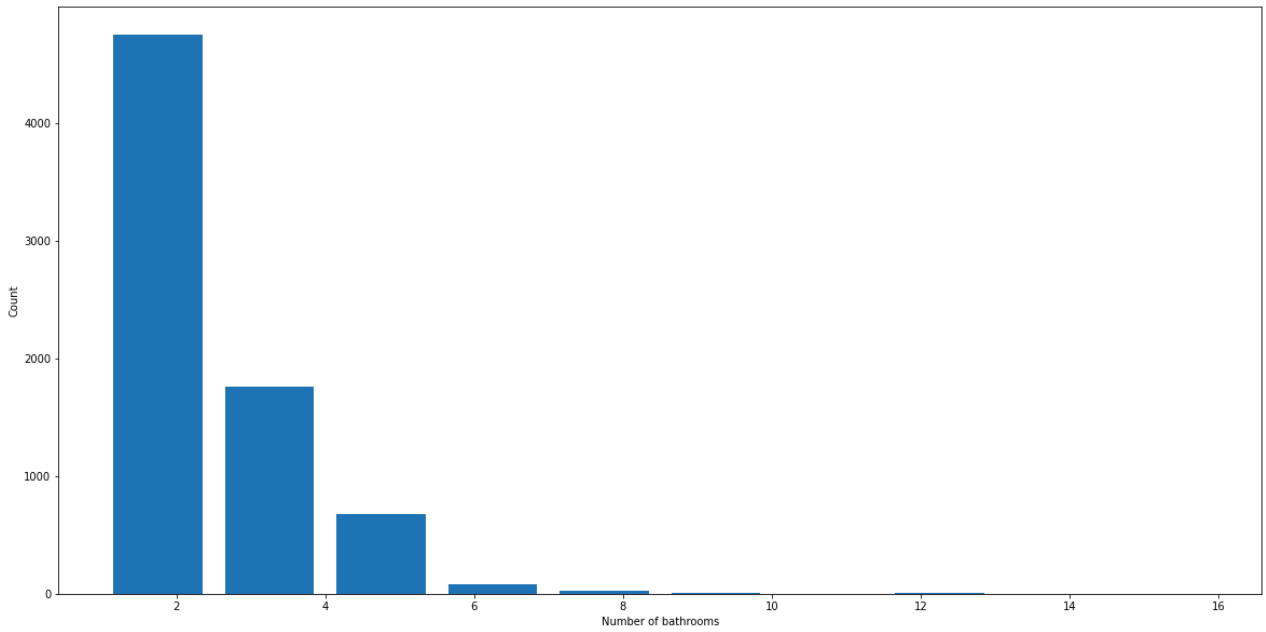
Outlier Removal Using Bathrooms Feature

```
df8.bath.unique()
```

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

```
plt.hist(df8.bath,rwidth=0.8)
plt.xlabel("Number of bathrooms")
plt.ylabel("Count")
```

```
Text(0, 0.5, 'Count')
```




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

	location	size	total_sqft	bath	price	bhk	price_per_sqft
5277	Neeladri Nagar	10 BHK	4000.0	12.0	160.0	10	4000.000000
5926	Other	10 BHK	12000.0	12.0	525.0	10	4375.000000
6015	Other	16 BHK	10000.0	16.0	550.0	16	5500.000000
6749	Other	11 BHK	6000.0	12.0	150.0	11	2500.000000
7080	Other	13 BHK	5425.0	13.0	275.0	13	5069.124424

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

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

	location	size	total_sqft	bath	price	bhk	price_per_sqft
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
5851	Other	6 BHK	11338.0	9.0	1000.0	6	8819.897689
9017	Thanisandra	3 BHK	1806.0	6.0	116.0	3	6423.034330

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

```
(7239, 7)
```

```
df9.head(10)
```

	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	1st Block Jayanagar	4 BHK	2850.0	4.0	428.0	4	15017.543860
1	1st Block Jayanagar	3 BHK	1630.0	3.0	194.0	3	11901.840491
2	1st Block Jayanagar	3 BHK	1875.0	2.0	235.0	3	12533.333333
3	1st Block Jayanagar	3 BHK	1200.0	2.0	130.0	3	10833.333333
4	1st Block Jayanagar	2 BHK	1235.0	2.0	148.0	2	11983.805668
5	1st Block Jayanagar	4 BHK	2750.0	4.0	413.0	4	15018.181818
6	1st Block Jayanagar	4 BHK	2450.0	4.0	368.0	4	15020.408163
8	1st Phase JP Nagar	3 BHK	1875.0	3.0	167.0	3	8906.666667
9	1st Phase JP Nagar	5 Bedroom	1500.0	5.0	85.0	5	5666.666667
10	1st Phase JP Nagar	3 BHK	2065.0	4.0	210.0	3	10169.491525

Drop unnecessary columns

```
df10 = df9.drop(['size','price_per_sqft'],axis='columns')
df10.head(10)
```

	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
5	1st Block Jayanagar	2750.0	4.0	413.0	4
6	1st Block Jayanagar	2450.0	4.0	368.0	4
8	1st Phase JP Nagar	1875.0	3.0	167.0	3
9	1st Phase JP Nagar	1500.0	5.0	85.0	5
10	1st Phase JP Nagar	2065.0	4.0	210.0	3

One Hot Encoding for Location

```
dummies = pd.get_dummies(df10.location)
dummies.head(10)
```

	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	...	Vijayanagar	Vishveshwarya Layout	Vishw
0	1	0	0	0	0	0	0	0	0	0	...	0	0	
1	1	0	0	0	0	0	0	0	0	0	...	0	0	
2	1	0	0	0	0	0	0	0	0	0	...	0	0	
3	1	0	0	0	0	0	0	0	0	0	...	0	0	
4	1	0	0	0	0	0	0	0	0	0	...	0	0	
5	1	0	0	0	0	0	0	0	0	0	...	0	0	
6	1	0	0	0	0	0	0	0	0	0	...	0	0	
8	0	1	0	0	0	0	0	0	0	0	...	0	0	
9	0	1	0	0	0	0	0	0	0	0	...	0	0	
10	0	1	0	0	0	0	0	0	0	0	...	0	0	

10 rows × 241 columns

```
df11 = pd.concat([df10,dummies.drop('Other',axis='columns')],axis='columns')
df11.head(10)
```

	location	total_sqft	bath	price	bhk	1st Block Jayanagar	1st Phase JP Nagar	2nd Phase Judicial Layout	2nd Stage Nagarbhavi	5th Block Hbr Layout	...	Vijayanagar	Vishveshwarya Layout
0	1st Block Jayanagar	2850.0	4.0	428.0	4	1	0	0	0	0	...	0	0
1	1st Block Jayanagar	1630.0	3.0	194.0	3	1	0	0	0	0	...	0	0
2	1st Block Jayanagar	1875.0	2.0	235.0	3	1	0	0	0	0	...	0	0
3	1st Block Jayanagar	1200.0	2.0	130.0	3	1	0	0	0	0	...	0	0
4	1st Block Jayanagar	1235.0	2.0	148.0	2	1	0	0	0	0	...	0	0
5	1st Block Jayanagar	2750.0	4.0	413.0	4	1	0	0	0	0	...	0	0
6	1st Block Jayanagar	2450.0	4.0	368.0	4	1	0	0	0	0	...	0	0
8	1st Phase JP Nagar	1875.0	3.0	167.0	3	0	1	0	0	0	...	0	0
9	1st Phase JP Nagar	1500.0	5.0	85.0	5	0	1	0	0	0	...	0	0
10	1st Phase JP Nagar	2065.0	4.0	210.0	3	0	1	0	0	0	...	0	0

10 rows × 245 columns

```
df12 = df11.drop('location',axis='columns')
df12.head(10)
```

	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	Vishveshwarya Layout	Vis
0	2850.0	4.0	428.0	4	1	0	0	0	0	0	...	0	0	
1	1630.0	3.0	194.0	3	1	0	0	0	0	0	...	0	0	
2	1875.0	2.0	235.0	3	1	0	0	0	0	0	...	0	0	
3	1200.0	2.0	130.0	3	1	0	0	0	0	0	...	0	0	
4	1235.0	2.0	148.0	2	1	0	0	0	0	0	...	0	0	
5	2750.0	4.0	413.0	4	1	0	0	0	0	0	...	0	0	
6	2450.0	4.0	368.0	4	1	0	0	0	0	0	...	0	0	
8	1875.0	3.0	167.0	3	0	1	0	0	0	0	...	0	0	
9	1500.0	5.0	85.0	5	0	1	0	0	0	0	...	0	0	
10	2065.0	4.0	210.0	3	0	1	0	0	0	0	...	0	0	

10 rows × 244 columns

Model Building

```
df12.shape
```

```
(7239, 244)
```

```
X = df12.drop(['price'],axis='columns')
X.head(10)
```

	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	Vis
0	2850.0	4.0	4	1	0	0	0	0	0	0	...	0	0	
1	1630.0	3.0	3	1	0	0	0	0	0	0	...	0	0	
2	1875.0	2.0	3	1	0	0	0	0	0	0	...	0	0	
3	1200.0	2.0	3	1	0	0	0	0	0	0	...	0	0	
4	1235.0	2.0	2	1	0	0	0	0	0	0	...	0	0	
5	2750.0	4.0	4	1	0	0	0	0	0	0	...	0	0	
6	2450.0	4.0	4	1	0	0	0	0	0	0	...	0	0	
8	1875.0	3.0	3	0	1	0	0	0	0	0	...	0	0	
9	1500.0	5.0	5	0	1	0	0	0	0	0	...	0	0	
10	2065.0	4.0	3	0	1	0	0	0	0	0	...	0	0	

10 rows × 243 columns

```
X.shape
```

```
(7239, 243)
```

```
y = df12.price
y.head(10)
```

```
0     428.0
1     194.0
2     235.0
3     130.0
4     148.0
5     413.0
6     368.0
8     167.0
9      85.0
10    210.0
Name: price, dtype: float64
```

```
len(y)
```

```
7239
```

```
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)
```

```
from sklearn.linear_model import LinearRegression

lr = LinearRegression()
lr.fit(X_train, y_train)
lr_score = lr.score(X_test, y_test)
lr_score
```

```
0.8697077447864602
```

```
from sklearn.svm import SVR

svr = SVR()
svr.fit(X_train,y_train)
svr_score=svr.score(X_test,y_test)
svr_score
```

```
0.6807978422767961
```

```
from sklearn.ensemble import RandomForestRegressor

rfr = RandomForestRegressor()
rfr.fit(X_train,y_train)
rfr_score=rfr.score(X_test,y_test)
rfr_score
```

```
0.7862114205698892
```

```
from sklearn.linear_model import Lasso

lr_lasso = Lasso()
lr_lasso.fit(X_train, y_train)
lr_lasso_score=lr_lasso.score(X_test, y_test)
lr_lasso_score
```

```
0.7344307377283503
```

```
def predict_price_1(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]
```

```
def predict_price_2(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 svr.predict([x])[0]
```

```
def predict_price_3(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 rfr.predict([x])[0]
```

```
def predict_price_4(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_lasso.predict([x])[0]
```

```
predict_price_1('1st Block Jayanagar',1875, 2, 3)  # Linear Regression
```

```
271.14459907853814
```

```
predict_price_2('1st Block Jayanagar',1875, 2, 3)  # Support Vector
```

```
126.1083460634477
```

```
predict_price_3('1st Block Jayanagar',1875, 2, 3)  # Random Forest Regression
```