

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Loading the raw Data

```
In [2]: data = pd.read_csv("C:/Users/Pratik/Desktop/Internship/task 4/house_prices.csv")
```

```
In [3]: data.head()
```

```
Out[3]:
```

| | area_type | availability | location | size | society | total_sqft | bath | balcony |
|---|---------------------|---------------|--------------------------|-----------|---------|------------|------|---------|
| 0 | Super built-up Area | 19-Dec | Electronic City Phase II | 2 BHK | Coomee | 1056 | 2.0 | 1.0 |
| 1 | Plot Area | Ready To Move | Chikka Tirupathi | 4 Bedroom | Theanmp | 2600 | 5.0 | 3.0 |
| 2 | Built-up Area | Ready To Move | Uttarahalli | 3 BHK | NaN | 1440 | 2.0 | 3.0 |
| 3 | Super built-up Area | Ready To Move | Lingadheeranahalli | 3 BHK | Soiewre | 1521 | 3.0 | 1.0 |
| 4 | Super built-up Area | Ready To Move | Kothanur | 2 BHK | NaN | 1200 | 2.0 | 1.0 |

```
In [4]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13320 entries, 0 to 13319
Data columns (total 9 columns):
#   Column          Non-Null Count  Dtype
---  -
0   area_type       13320 non-null  object
1   availability     13320 non-null  object
2   location        13319 non-null  object
3   size            13304 non-null  object
4   society         7818 non-null   object
5   total_sqft      13320 non-null  object
6   bath            13247 non-null  float64
7   balcony         12711 non-null  float64
8   price           13320 non-null  float64
dtypes: float64(3), object(6)
memory usage: 936.7+ KB
```

```
In [5]: data.columns
```

```
Out[5]: Index(['area_type', 'availability', 'location', 'size', 'society',
              'total_sqft', 'bath', 'balcony', 'price'],
              dtype='object')
```

```
In [6]: ## Data Cleaning
```

```
In [7]: data = data.drop(['area_type', 'availability', 'balcony', 'society'], axis=1)
data
```

```
Out[7]:
```

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

13320 rows × 5 columns

```
In [8]: data.isna().sum()
```

```
Out[8]: location      1
size      16
total_sqft    0
bath      73
price      0
dtype: int64
```

```
In [9]: data = data.dropna()
```

```
In [10]: data.isna().sum()
```

```
Out[10]: location      0
size      0
total_sqft    0
bath      0
price      0
dtype: int64
```

```
In [11]: data.shape
```

Out[11]: (13246, 5)

In [12]: `data['size'].unique()`

Out[12]: 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 [13]: `## feature Engineering`
`data['BHK']=data['size'].apply(lambda x: int(x.split(' ')[0]))`

C:\Users\Pratik\AppData\Local\Temp\ipykernel_15448\1969391074.py:2: 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

`data['BHK']=data['size'].apply(lambda x: int(x.split(' ')[0]))`

In [14]: `data.head()`

Out[14]:

| | 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 [15]: `data['BHK'].unique()`

Out[15]: array([2, 4, 3, 6, 1, 8, 7, 5, 11, 9, 27, 10, 19, 16, 43, 14, 12,
13, 18], dtype=int64)

In [16]: `data[data.BHK>20]`

Out[16]:

| | 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 [17]: `data.total_sqft.unique()`

Out[17]: array(['1056', '2600', '1440', ..., '1133 - 1384', '774', '4689'],
dtype=object)

```
In [18]: def isfloat(x):
          try:
              float(x)
          except:
              return False
          return True
```

```
In [19]: data[~data['total_sqft'].apply(isfloat)].head(10)
```

```
Out[19]:
```

| | 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 [20]: def convert_sqft_tonum(x):
          token=x.split('-')
          if len(token)==2:
              return (float(token[0])+float(token[1]))/2
          try:
              return float(x)
          except:
              return None
```

```
In [21]: data=data.copy()
          data['total_sqft']=data['total_sqft'].apply(convert_sqft_tonum)
```

```
In [22]: data.head(10)
```

Out[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 |
| 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 |

In [23]: `data.loc[40]`

Out[23]:

| | |
|------------|---------------|
| location | Murugeshpalya |
| size | 2 BHK |
| total_sqft | 1296.0 |
| bath | 2.0 |
| price | 81.0 |
| BHK | 2 |

Name: 40, dtype: object

In [24]: `## adding a feature called price per square feet`

In [25]:

```
data1=data.copy()
data1['price_per_sqft']=data1['price']*1000000/data1['total_sqft']
data1.head()
```

Out[25]:

| | 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 [26]: `len(data1.location.unique())`

Out[26]: 1304

In [27]:

```
data1.location=data1.location.apply(lambda x: x.strip())
location_stats=data1.groupby('location')['location'].agg('count').sort_values(ascen
```

```
location_stats
```

```
Out[27]: 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 [28]: len(location_stats[location_stats<=10])
```

```
Out[28]: 1052
```

```
In [29]: locationlessthan10=location_stats[location_stats<=10]
locationlessthan10
```

```
Out[29]: 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 [30]: len(data1.location.unique())
```

```
Out[30]: 1293
```

```
In [31]: data1.location=data1.location.apply(lambda x: 'other' if x in locationlessthan10 else
len(data1.location.unique())
```

```
Out[31]: 242
```

```
In [32]: data1.head(10)
```

Out[32]:

| | 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 [33]: data1[data1.total_sqft/data1.BHK<300].head()

Out[33]:

| | location | size | total_sqft | bath | price | BHK | price_per_sqft |
|----|---------------------|-----------|------------|------|-------|-----|----------------|
| 9 | other | 6 Bedroom | 1020.0 | 6.0 | 370.0 | 6 | 362745.098039 |
| 45 | HSR Layout | 8 Bedroom | 600.0 | 9.0 | 200.0 | 8 | 333333.333333 |
| 58 | Murugeshpalya | 6 Bedroom | 1407.0 | 4.0 | 150.0 | 6 | 106609.808102 |
| 68 | Devarachikkanahalli | 8 Bedroom | 1350.0 | 7.0 | 85.0 | 8 | 62962.962963 |
| 70 | other | 3 Bedroom | 500.0 | 3.0 | 100.0 | 3 | 200000.000000 |

In [34]: data2=data1[~(data1.total_sqft/data1.BHK<300)]
data2.head(10)

Out[34]:

| | 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 |
| 10 | Whitefield | 3 BHK | 1800.0 | 2.0 | 70.00 | 3 | 38888.888889 |

In [35]: data2.shape

Out[35]: (12502, 7)

In [37]: *## outlier removal using standard deviation and mean*

data2['price_per_sqft'].describe().apply(lambda x:format(x,'f'))

```
Out[37]: count      12456.000000
mean       63085.028260
std        41681.273385
min         2678.298133
25%        42105.263158
50%        52941.176471
75%        69166.666667
max       1764705.882353
Name: price_per_sqft, dtype: object
```

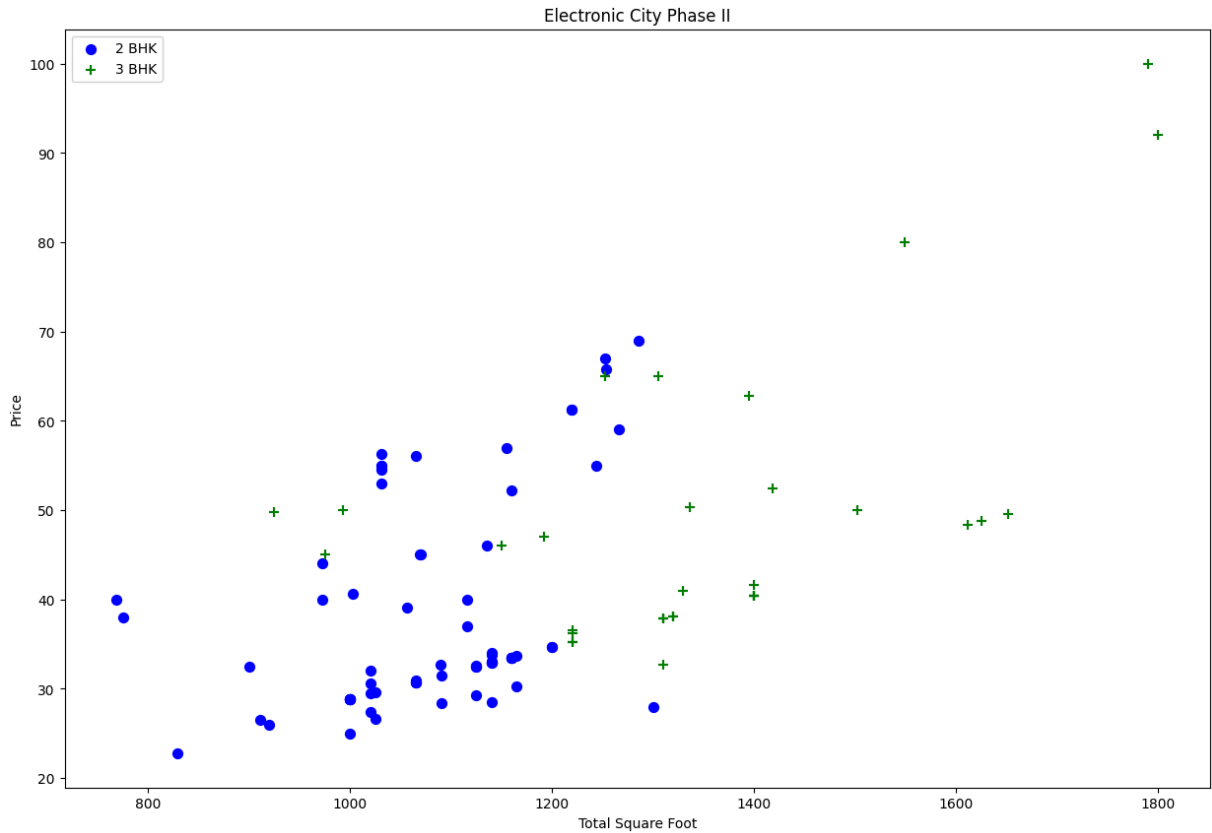
```
In [38]: def remove_pps_outliers(df):
df_out=pd.DataFrame()
for key,subdf in df.groupby('location'):
    m=np.mean(subdf.price_per_sqft)
    st=np.std(subdf.price_per_sqft)
    reduced_df=subdf[(subdf.price_per_sqft>(m-st))& (subdf.price_per_sqft<(m+st))]
    df_out=pd.concat([df_out,reduced_df],ignore_index=True)
return df_out
data3=remove_pps_outliers(data2)
data3.shape
```

Out[38]: (10241, 7)

```
In [42]: def plot_scatter_chart(df,location):
bhk2=df[(df.location==location)&(df.BHK==2)]
bhk3=df[(df.location==location)&(df.BHK==3)]
```



```
plt.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,color='green',marker='+',label='3 BHK',s=50)
plt.xlabel('Total Square Foot')
plt.ylabel('Price')
plt.title(location)
plt.legend()
plot_scatter_chart(data3,"Electronic City Phase II")
```

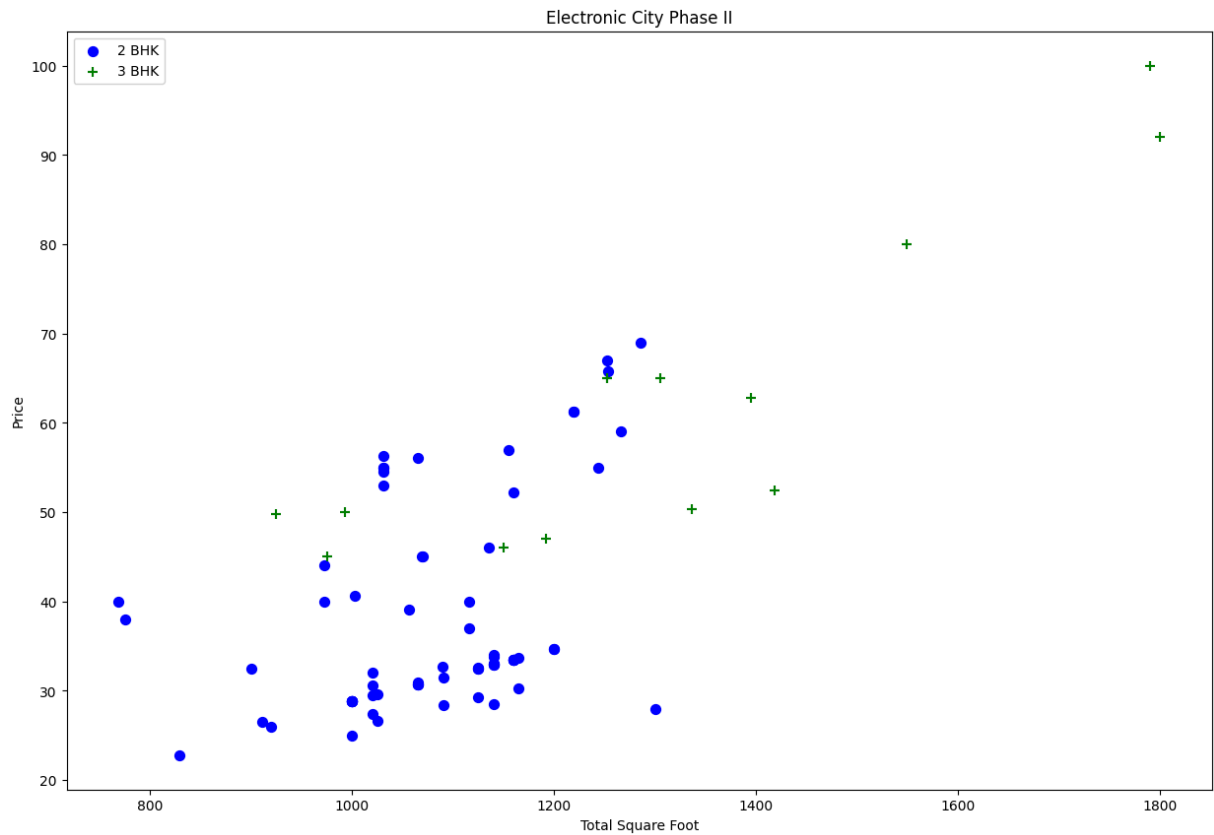


```
In [44]: def remove_bhk_outliers(df):
exclude_indices=np.array([])
for location, location_df in df.groupby('location'):
    bhk_sats={}
    for BHK,BHK_df in location_df.groupby('BHK'):
        bhk_sats[BHK]={
            'mean':np.mean(BHK_df.price_per_sqft),
            'std':np.std(BHK_df.price_per_sqft),
            'count':BHK_df.shape[0]
        }
    for BHK,BHK_df in location_df.groupby('BHK'):
        stats=bhk_sats.get(BHK-1)
        if stats and stats['count']>5:
            exclude_indices=np.append(exclude_indices,BHK_df[BHK_df.price_per_s
return df.drop(exclude_indices,axis='index')

data4=remove_bhk_outliers(data3)
data4.shape
```

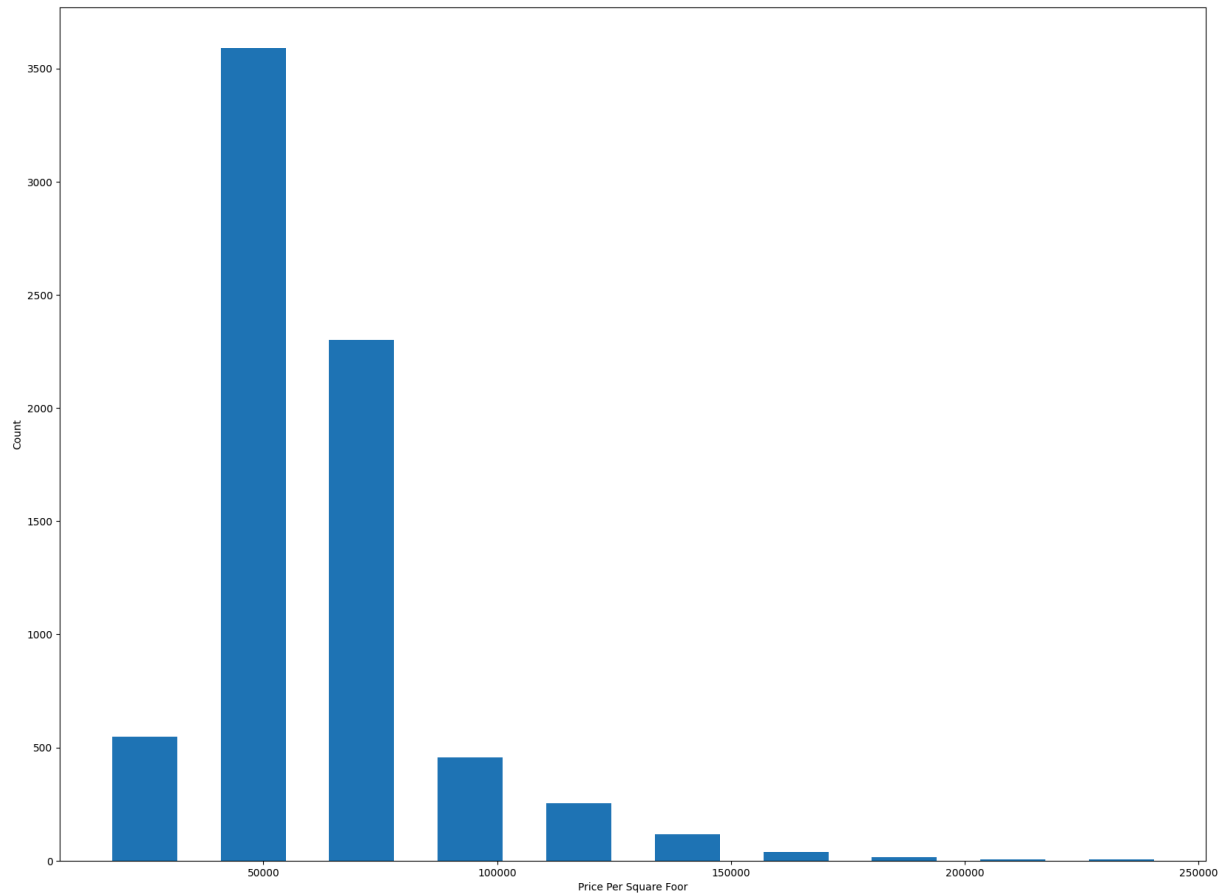
Out[44]: (7329, 7)

```
In [45]: plot_scatter_chart(data4,"Electronic City Phase II")
```



```
In [46]: plt.rcParams['figure.figsize']=(20,15)
plt.hist(data4.price_per_sqft,rwidth=0.6)
plt.xlabel("Price Per Square Foot")
plt.ylabel("Count")
```

```
Out[46]: Text(0, 0.5, 'Count')
```



```
In [48]: data5 = data4.drop(['size','price_per_sqft'], axis='columns')
data5
```

Out[48]:

| | 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 |

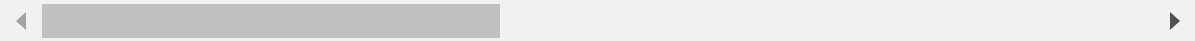
7329 rows × 5 columns

```
In [50]: dummies = pd.get_dummies(data5.location)
dummies.head(10)
```

Out[50]:

| | 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 |
|----|------------------------|--------------------------|---------------------------------|-------------------------|----------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| 0 | True | False | False | False | False | False | False | False | False | False |
| 1 | True | False | False | False | False | False | False | False | False | False |
| 2 | True | False | False | False | False | False | False | False | False | False |
| 3 | True | False | False | False | False | False | False | False | False | False |
| 4 | True | False | False | False | False | False | False | False | False | False |
| 5 | True | False | False | False | False | False | False | False | False | False |
| 6 | True | False | False | False | False | False | False | False | False | False |
| 8 | False | True | False | False | False | False | False | False | False | False |
| 9 | False | True | False | False | False | False | False | False | False | False |
| 10 | False | True | False | False | False | False | False | False | False | False |

10 rows × 242 columns



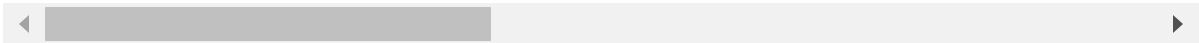
```
In [52]: data6 = pd.concat([data5, dummies.drop('other', axis='columns')], axis='columns')
```

```
In [53]: data6.head()
```

Out[53]:

| | location | total_sqft | bath | price | BHK | 1st Block Jayanagar | 1st Phase JP Nagar | 2nd Phase Judicial Layout | 2nd Stage Nagarbhavi | 5th Block Hbr Layout |
|---|------------------------|------------|------|-------|-----|------------------------|-----------------------------|------------------------------------|-------------------------|-------------------------------|
| 0 | 1st Block Jayanagar | 2850.0 | 4.0 | 428.0 | 4 | True | False | False | False | False |
| 1 | 1st Block Jayanagar | 1630.0 | 3.0 | 194.0 | 3 | True | False | False | False | False |
| 2 | 1st Block Jayanagar | 1875.0 | 2.0 | 235.0 | 3 | True | False | False | False | False |
| 3 | 1st Block Jayanagar | 1200.0 | 2.0 | 130.0 | 3 | True | False | False | False | False |
| 4 | 1st Block Jayanagar | 1235.0 | 2.0 | 148.0 | 2 | True | False | False | False | False |

5 rows × 246 columns

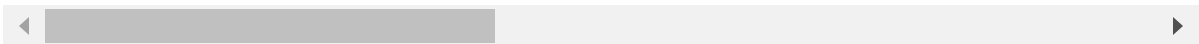


```
In [54]: data7 = data6.drop('location', axis='columns')
data7.head()
```

Out[54]:

| | 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 |
|---|------------|------|-------|-----|------------------------|-----------------------------|------------------------------------|-------------------------|-------------------------------|-----------------------------|
| 0 | 2850.0 | 4.0 | 428.0 | 4 | True | False | False | False | False | False |
| 1 | 1630.0 | 3.0 | 194.0 | 3 | True | False | False | False | False | False |
| 2 | 1875.0 | 2.0 | 235.0 | 3 | True | False | False | False | False | False |
| 3 | 1200.0 | 2.0 | 130.0 | 3 | True | False | False | False | False | False |
| 4 | 1235.0 | 2.0 | 148.0 | 2 | True | False | False | False | False | False |

5 rows × 245 columns



Build a New Model

```
In [56]: data7.shape
```

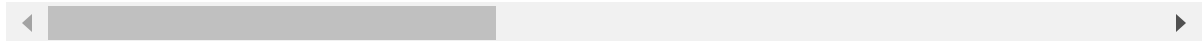
Out[56]: (7329, 245)

```
In [57]: X= data7.drop('price', axis='columns')
X.head()
```

Out[57]:

| | 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 |
|---|------------|------|-----|------------------------|-----------------------------|------------------------------------|-------------------------|-------------------------------|-----------------------------|-----------------------------|
| 0 | 2850.0 | 4.0 | 4 | True | False | False | False | False | False | False |
| 1 | 1630.0 | 3.0 | 3 | True | False | False | False | False | False | False |
| 2 | 1875.0 | 2.0 | 3 | True | False | False | False | False | False | False |
| 3 | 1200.0 | 2.0 | 3 | True | False | False | False | False | False | False |
| 4 | 1235.0 | 2.0 | 2 | True | False | False | False | False | False | False |

5 rows × 244 columns

In [58]: `y = data7.price`In [60]: `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)`In [62]: `from sklearn.linear_model import LinearRegression
model=LinearRegression()
model.fit(X_train,y_train)
model.score(X_test,y_test)`

Out[62]: 0.856884774614467

In [63]: `## Using K Fold cross validation to measure accuracy of our LinearRegression model`In [64]: `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)`

Out[64]: array([0.81924325, 0.86698523, 0.86457433, 0.8857566 , 0.86309462])

Testing the model

In [71]: `def price_predict(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 model.predict([x])[0]`

```
In [72]: price_predict('1st Phase JP Nagar',1000,2,2)
```

```
D:\Python\Lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid  
feature names, but LinearRegression was fitted with feature names  
warnings.warn(
```

```
Out[72]: 89.14476596048132
```

```
In [73]: price_predict('1st Phase JP Nagar',1000,2,3)
```

```
D:\Python\Lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid  
feature names, but LinearRegression was fitted with feature names  
warnings.warn(
```

```
Out[73]: 86.10727992159647
```

```
In [74]: price_predict('5th Phase JP Nagar',1000,2,2)
```

```
D:\Python\Lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid  
feature names, but LinearRegression was fitted with feature names  
warnings.warn(
```

```
Out[74]: 45.23283232918844
```

```
In [75]: price_predict('Indira Nagar',1000,2,2)
```

```
D:\Python\Lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid  
feature names, but LinearRegression was fitted with feature names  
warnings.warn(
```

```
Out[75]: 178.8296668379752
```