House_price_prediction

December 18, 2025

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

[2]:
```
df1 = pd.read_csv("bengaluru_house_prices.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.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
```

```
[4]: df1.shape
```

```
[4]: (13320, 9)
```

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

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

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

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

```
[7]: df1.nunique()
```

```
[7]: area_type          4
     availability      81
     location        1305
     size              31
     society         2688
     total_sqft      2117
     bath              19
     balcony            4
     price           1994
     dtype: int64
```

```
[8]: df2= df1.drop(['area_type','society','balcony', 'availability' ],␣
      ↪axis='columns')
     df2.head()
```

```
[8]:                    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
```

[9]:
```
df2.isnull().sum()
```

[9]:
```
location       1
size          16
total_sqft     0
bath          73
price          0
dtype: int64
```

[10]:
```
df3 = df2.dropna()
df3.isnull().sum()
```

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

[11]:
```
df3.shape
```

[11]:
```
(13246, 5)
```

[12]:
```
df3.head()
```

[12]:
```
                   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
```

[13]:
```
df3['size'].unique()
```

[13]:
```
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)
```

[14]:
```
df3['bhk'] = df3['size'].apply(lambda x : int(x.split(' ')[0]))
```

```
C:\Users\dell\AppData\Local\Temp\ipykernel_14048\3847263516.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
```

```
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df3['bhk'] = df3['size'].apply(lambda x : int(x.split(' ')[0]))
```

[15]:
```
df3.head()
```

[15]:
```
                   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
```

[16]:
```
df3['bhk'].unique()
```

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

[17]:
```
df3[df3['bhk']>20]
```

[17]:
```
                      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
```

[18]:
```
df3.total_sqft.unique()
```

[18]:
```
array(['1056', '2600', '1440', …, '1133 - 1384', '774', '4689'],
      dtype=object)
```

[19]:
```
def is_float(x):
    try:
        float(x)
    except:
        return False
    return True
```

[20]:
```
df3[~df3['total_sqft'].apply(is_float)].head(10)
```

[20]:
```
               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
```

[21]:

```
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
```

[22]:

```
convert_sqft_to_num('2166')
```

[22]:

```
2166.0
```

[23]:

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

[23]:

```
2475.0
```

[24]:

```
convert_sqft_to_num('4125Perch')
```

[25]:

```
#applying this function into the total_sqft column to get the average
```

[26]:

```
df4 = df3.copy()
df4['total_sqft'] = df4['total_sqft'].apply(convert_sqft_to_num)
df4.head(3)
```

[26]:

```
                   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
```

[27]:

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

[27]:

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

[28]:

```
df4.head(3)
```

[28]:

```
                   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
```

```
[29]: #we cleaned the dataset
```

```
[30]: #now want to apply feature engineering
```

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

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

```
[31]:                  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
```

```
[32]: #check locations
```

```
[33]: df5['location'].value_counts()
```

```
[33]: location
      Whitefield           534
      Sarjapur  Road       392
      Electronic City      302
      Kanakpura Road       266
      Thanisandra          233
                          …
      Vidyapeeta             1
      Maruthi Extension      1
      Okalipura              1
      Old Town               1
      Abshot Layout          1
      Name: count, Length: 1304, dtype: int64
```

```
[34]: df5['location'].nunique()
```

```
[34]: 1304
```

```
[35]: df5.shape
```

```
[35]: (13246, 7)
```

```
[36]: # too many locations->cannot one-hot encoding
```

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

```
[37]: 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
```

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

```
[38]: 1052
```

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

```
[39]: 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
```

```
[40]: len(df5['location'].unique())
```

```
[40]: 1293
```

[41]:

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

[41]:

```
242
```

[42]:

```
df5.head()
```

[42]:

```
                  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
```

[43]:

```
 # outlier removal
```

[44]:

```
df5.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 13246 entries, 0 to 13319
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   location        13246 non-null  object
 1   size            13246 non-null  object
 2   total_sqft      13200 non-null  float64
 3   bath            13246 non-null  float64
 4   price           13246 non-null  float64
 5   bhk             13246 non-null  int64
 6   price_per_sqft  13200 non-null  float64
dtypes: float64(4), int64(1), object(2)
memory usage: 1.3+ MB
```

[45]:

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

[45]:

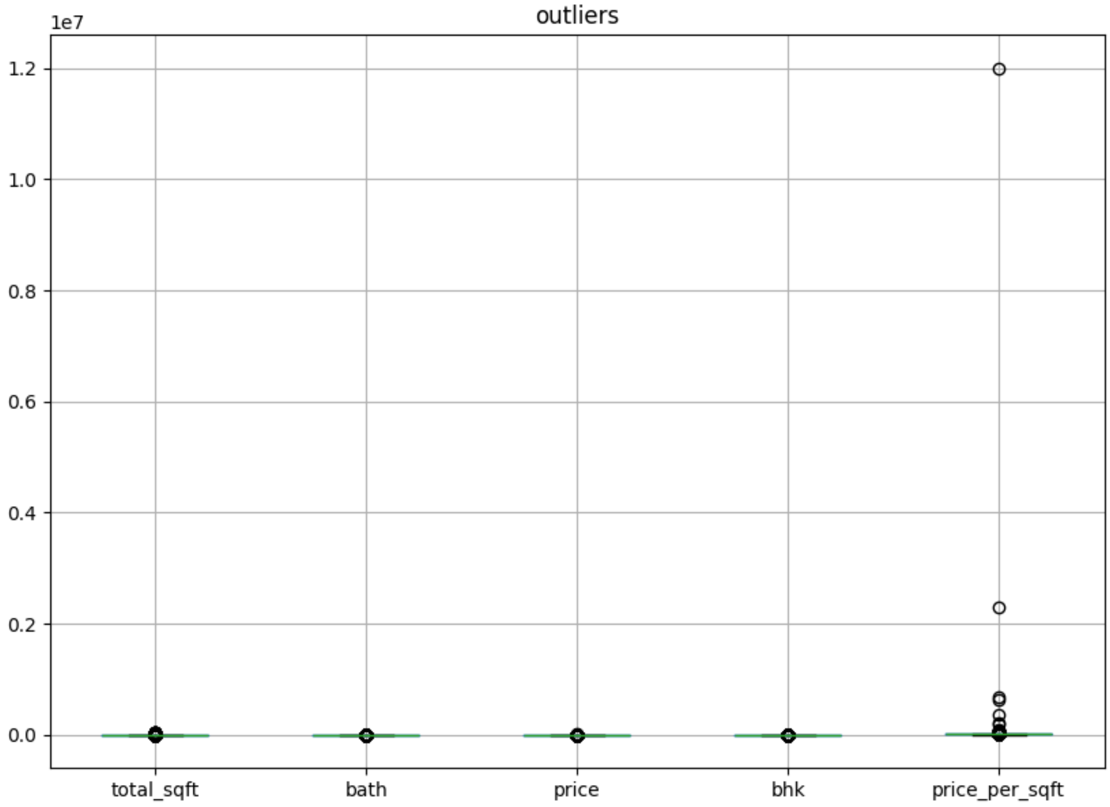
```
              location       size  total_sqft  bath  price  bhk  \
9                other  6 Bedroom      1020.0   6.0  370.0    6
45          HSR Layout  8 Bedroom       600.0   9.0  200.0    8
58       Murugeshpalya  6 Bedroom      1407.0   4.0  150.0    6
68  Devarachikkanahalli  8 Bedroom      1350.0   7.0   85.0    8
```

```
70                 other  3 Bedroom       500.0   3.0  100.0    3

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

[46]:
```
df5.shape
```

[46]:
```
(13246, 7)
```

[47]:
```
import matplotlib.pyplot as plt

df5.boxplot(figsize=(10,7), vert = True)
plt.title("outliers")
plt.show()
```



[48]:
```
df6 = df5[~(df5['total_sqft']/df5['bhk']<300)]
df6.shape
```

```
[48]: (12502, 7)
```

```
[49]: df6['price_per_sqft'].describe()
```

```
[49]: 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
```

```
[50]: df6.describe()
```

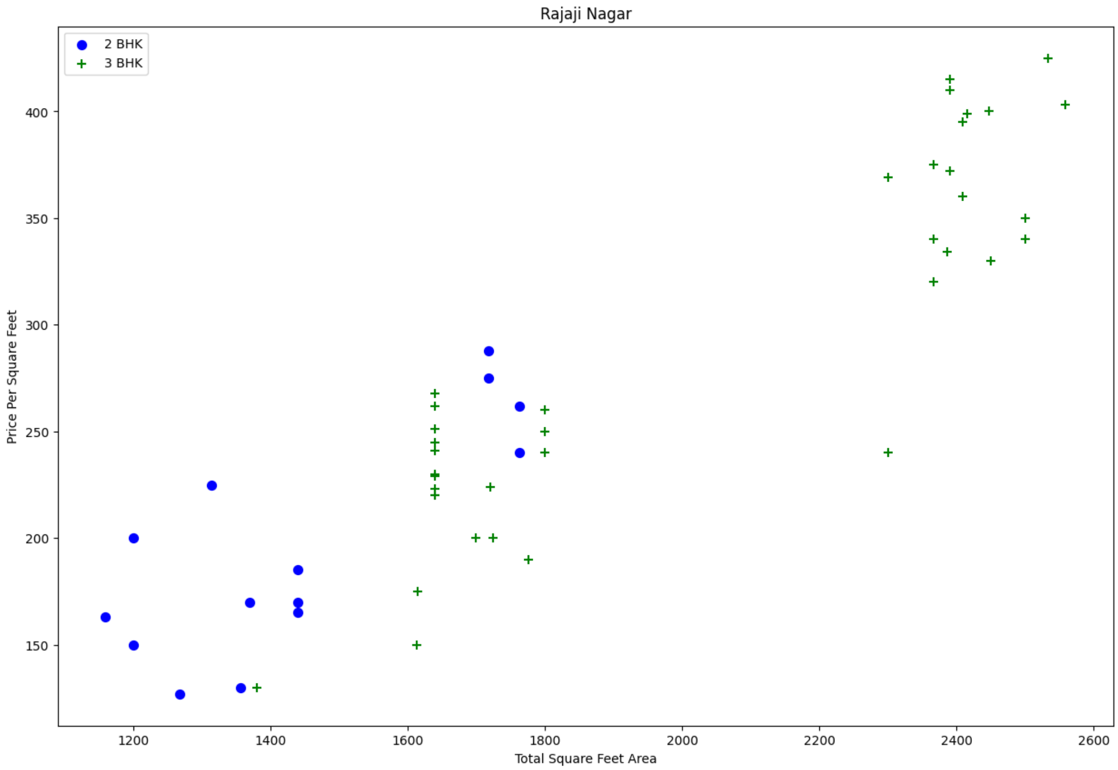
```
[50]:          total_sqft          bath         price           bhk  price_per_sqft
      count  12456.000000  12502.000000  12502.000000  12502.000000    12456.000000
      mean    1590.189927      2.564790    111.311915      2.650696     6308.502826
      std     1260.404795      1.084946    152.089966      0.981698     4168.127339
      min      300.000000      1.000000      9.000000      1.000000      267.829813
      25%     1115.000000      2.000000     49.000000      2.000000     4210.526316
      50%     1300.000000      2.000000     70.000000      3.000000     5294.117647
      75%     1700.000000      3.000000    115.000000      3.000000     6916.666667
      max    52272.000000     16.000000   3600.000000     16.000000   176470.588235
```

```
[51]: 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
```

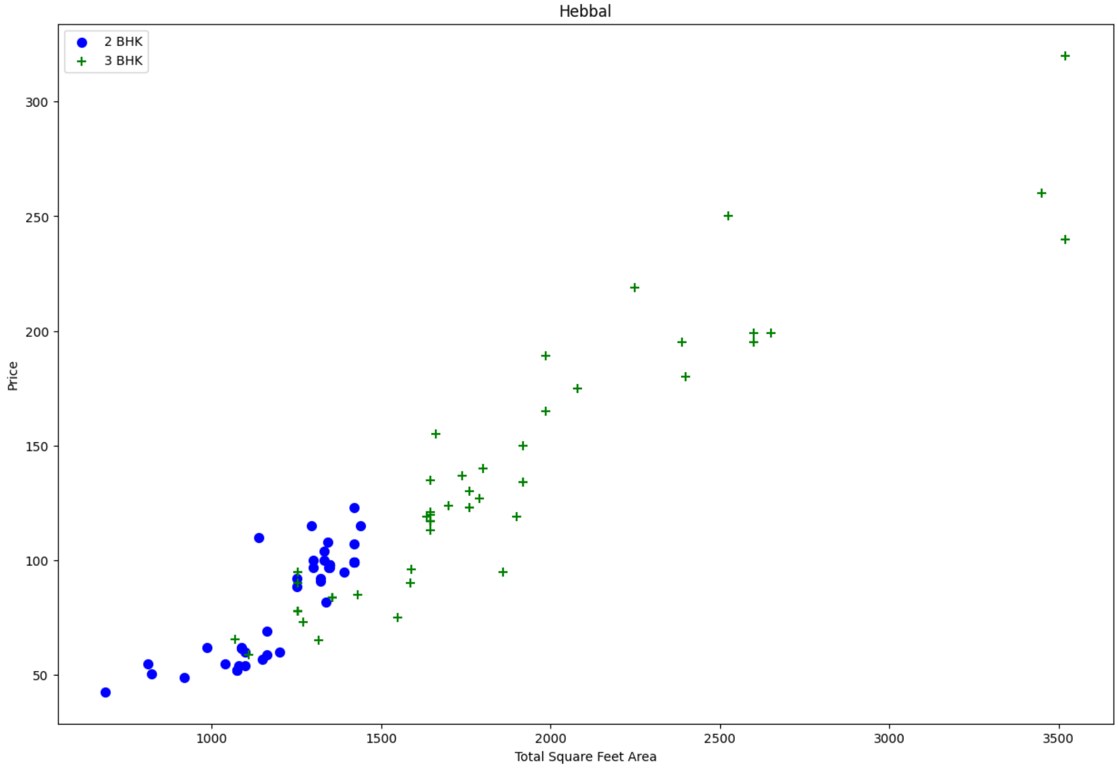
```
[51]: (10241, 7)
```

```
[52]: import matplotlib
      import matplotlib.pyplot as plt
      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 Per Square Feet")
    plt.title(location)
    plt.legend()


plot_scatter_chart(df7,"Rajaji Nagar")
plt.show()
```



[53]:
```
import matplotlib
import matplotlib.pyplot as plt
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")
    plt.title(location)
    plt.legend()


plot_scatter_chart(df7,"Hebbal")
plt.show()
```



[54]:
```
#remove those 2 BHK apartments whose price_per_sqft is less than mean␣
↪price_per_sqft of i BHK apartment
```

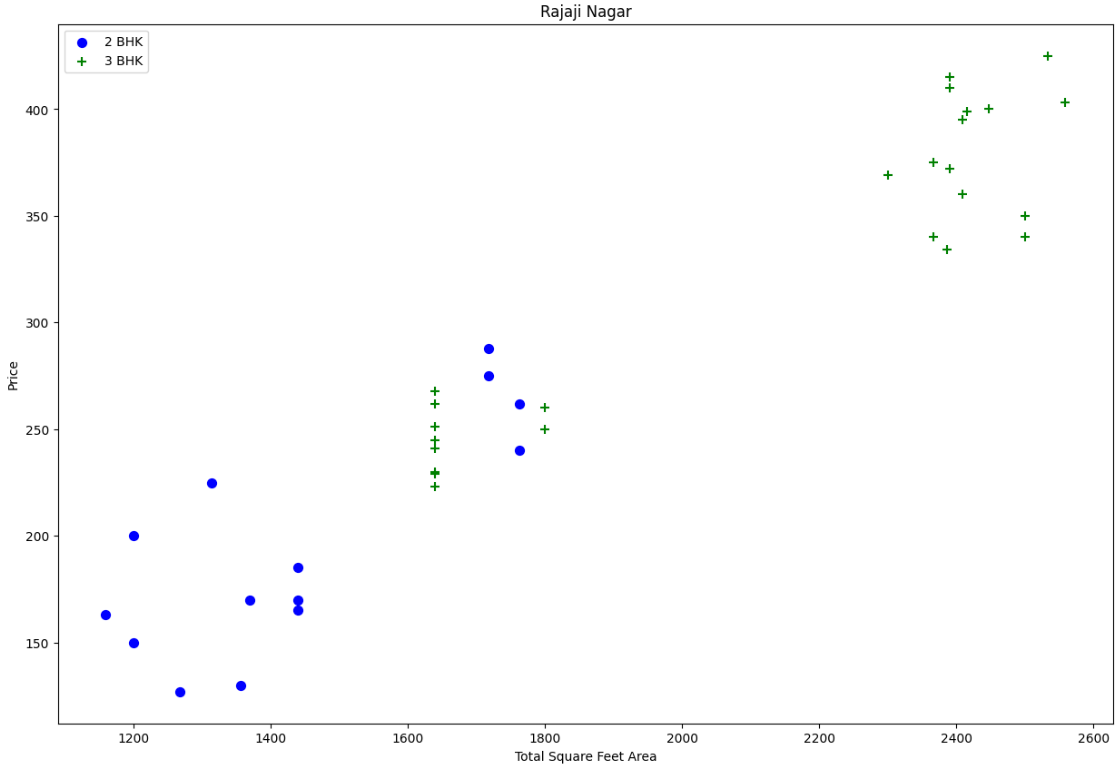
[55]:
```
def remove_bhk_outliers(df):
    exclude_indices = np.array([])
    for location, location_df in df.groupby('location'):
        bhk_stats = {}
        for bhk, bhk_df in location_df.groupby('bhk'):
            bhk_stats[bhk]={
                'mean': np.mean(bhk_df.price_per_sqft ),
                '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
```

[55]:

```
(7329, 7)
```

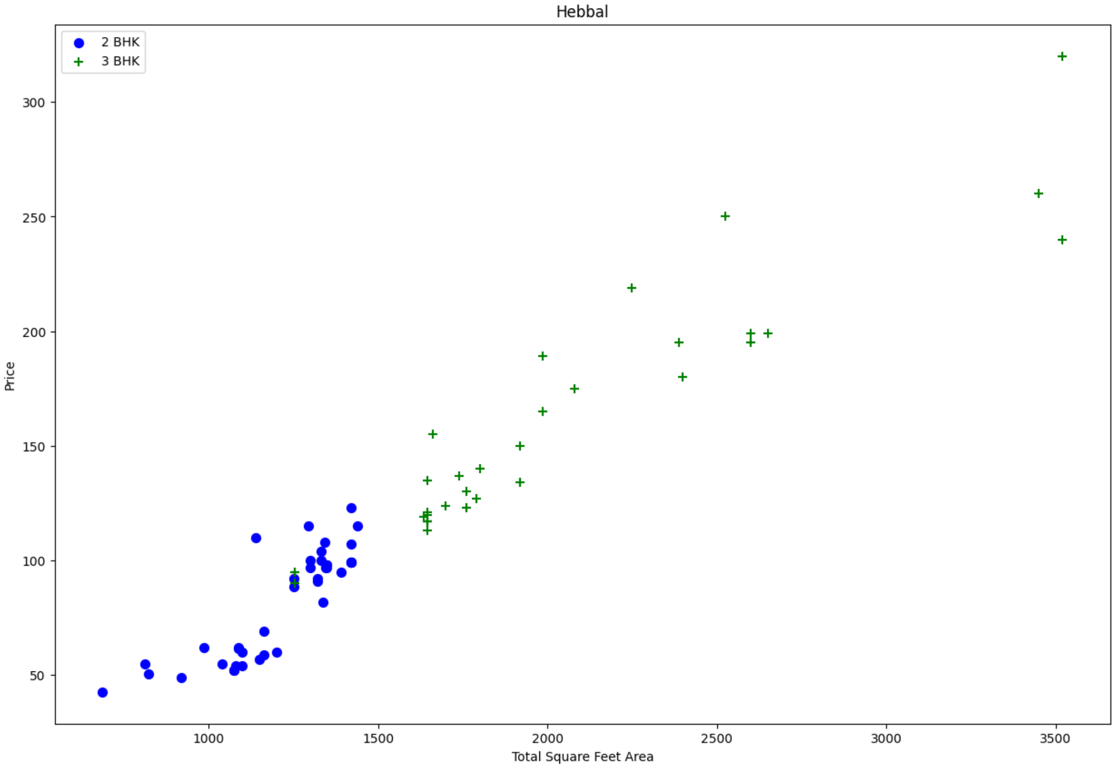
[56]:

```
plot_scatter_chart(df8, "Rajaji Nagar")
plt.show()
```

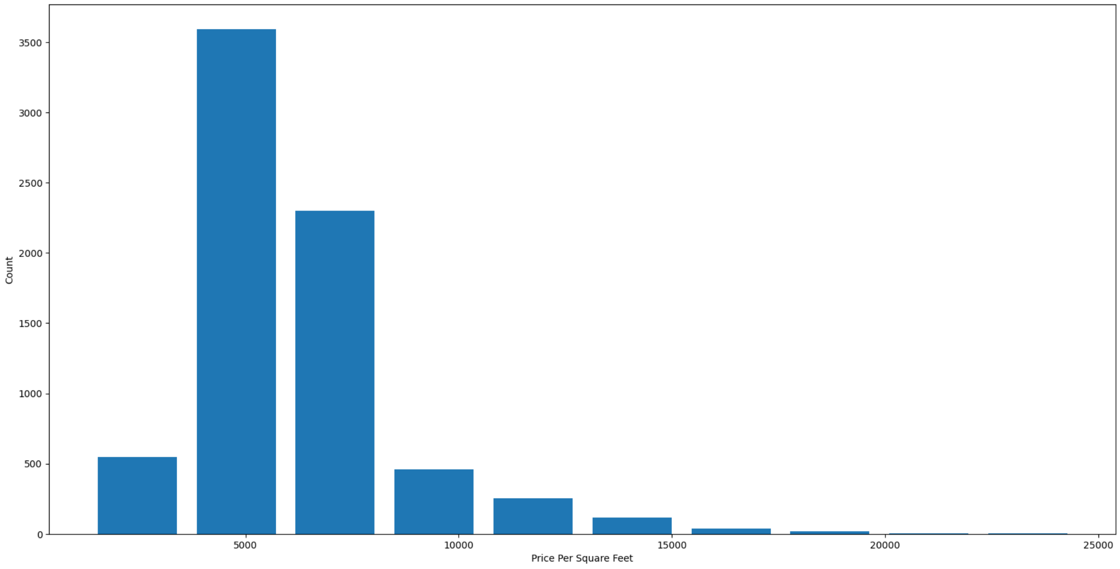


[57]:

```
plot_scatter_chart(df8, "Hebbal")
plt.show()
```



[58]:
```
matplotlib.rcParams["figure.figsize"]=(20,10)
plt.hist(df8.price_per_sqft, rwidth=0.8)
plt.xlabel("Price Per Square Feet")
plt.ylabel("Count")
plt.show()
```



[59]:
```
#dataset has normal distribution
```

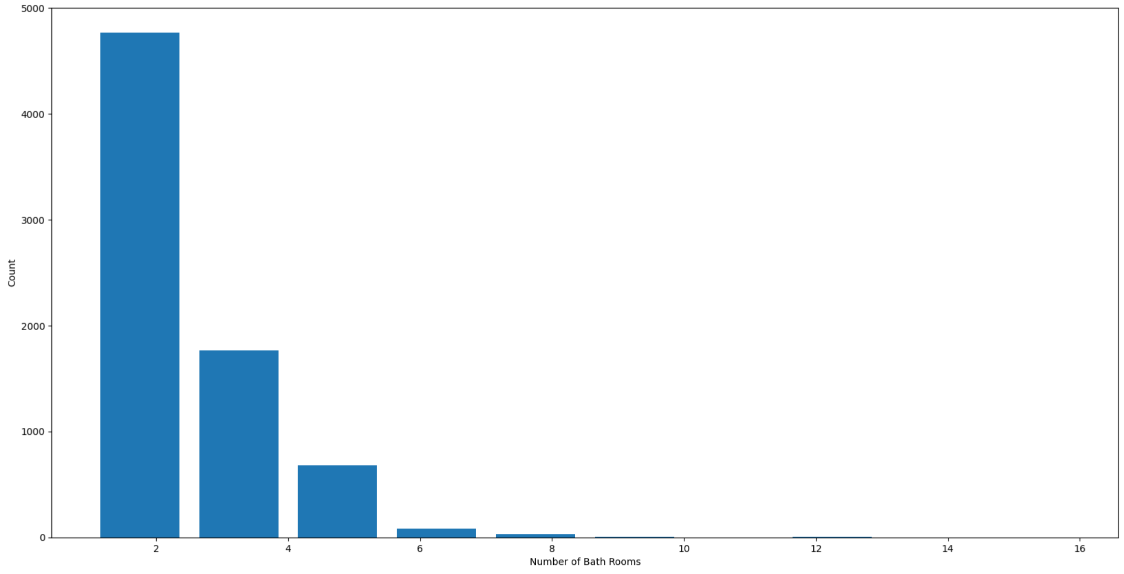
[60]:
```
df8.bath.unique()
```

[60]:
```
array([ 4.,  3.,  2.,  5.,  8.,  1.,  6.,  7.,  9., 12., 16., 13.])
```

[61]:
```
df8[df8['bath']>10]
```

[61]:
```
             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
```

[62]:
```
matplotlib.rcParams["figure.figsize"]=(20,10)
plt.hist(df8.bath, rwidth=0.8)
plt.xlabel("Number of Bath Rooms")
plt.ylabel("Count")
plt.show()
```



[63]:
```
df8[df8['bath']>df8.bhk+2]
```

[63]:
```
            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
```

```
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
```

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

```
[64]: (7251, 7)
```

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

```
[65]:              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
```

```
[66]: # create machine learning model
```

```
[67]: # to turn this categorical column into numbers we using one hot encoding
```

```
[73]: dummies = pd.get_dummies(df10['location'])
      dummies.head()
```

```
[73]:   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

         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

         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

         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

   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

   Yelenahalli  Yeshwanthpur  other
0        False         False  False
1        False         False  False
2        False         False  False
3        False         False  False
4        False         False  False

[5 rows x 242 columns]
```

[77]:

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

[77]:

```
              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  …  Vishveshwarya Layout  Vishwapriya Layout  \
0                 False  …                 False               False
1                 False  …                 False               False
2                 False  …                 False               False
3                 False  …                 False               False
4                 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

   Yelenahalli  Yeshwanthpur  other
0        False         False  False
1        False         False  False
2        False         False  False
3        False         False  False
4        False         False  False

[5 rows x 247 columns]
```

[81]:

```
df11= df11.drop('other', axis=1)
df11.head()
```

[81]:

```
              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]
```

[83]:
```
df12 = df11.drop('location',axis=1)
df12.head()
```

[83]:
```
   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]
```

[85]:

```python
df12.shape
```

[85]:

```
(7251, 245)
```

[87]:

```python
# now creating the independant variable
```

[89]:

```python
X = df12.drop('price', axis=1)
X.head()
```

[89]:

```
   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]
```

[91]:

```
y = df12['price']
y.head()
```

[91]:

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

[95]:

```
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)
```

[101]:

```
from sklearn.linear_model import LinearRegression
lr_clf =LinearRegression()
lr_clf.fit(X_train, y_train)
lr_clf.score(X_test,y_test )
```

[101]:

```
0.8452277697874369
```

[105]:

```
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)
```

[105]:

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

[107]:

```
# we getting scre more than 80 percent, now we will do hyper parameter tunning
```

[128]:

```
# from sklearn.model_selection import GridSearchCV
# from sklearn.linear_model import Lasso
# from sklearn.tree import DecisionTreeRegressor

# def find_best_model_using_grid_search_cv(X,y):
#     algos = {
#         'linear_regression' :{
#             'model' : LinearRegression(),
#             'params' :{
#                 'normalize' : [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])

# find_best_model_using_grid_search_cv(X,y)
```

[130]:

```
from sklearn.model_selection import GridSearchCV, ShuffleSplit
from sklearn.linear_model import LinearRegression, Lasso
from sklearn.tree import DecisionTreeRegressor
import pandas as pd

def find_best_model_using_grid_search_cv(X, y):
    algos = {
        'linear_regression': {
            'model': LinearRegression(),
            'params': {
                'fit_intercept': [True, False]   # normalize removed in sklearn
            }
        },
        'lasso': {
            'model': Lasso(),
            'params': {
                'alpha': [1, 2],
                'selection': ['random', 'cyclic']
            }
        },
        'decision_tree': {
            'model': DecisionTreeRegressor(),
            'params': {
                'criterion': ['squared_error', '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_grid_search_cv(X,y)
```

[130]:

```
               model  best_score  \
0  linear_regression    0.819001
1              lasso    0.687429
2      decision_tree    0.728380

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

[131]:

```
#so best one is linear regression
```

[136]:

```
X.columns
```

[136]:

```
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)
```

[138]:

```
np.where(X.columns=="2nd Phase Judicial Layout")[0][0]
```

[138]: 5

[140]:

```
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]
```

[142]:

```
predict_price("1st Phase JP Nagar", 1000,2,2)
```

```
C:\Users\dell\anaconda3\envs\tf2\lib\site-
packages\sklearn\utils\validation.py:2749: UserWarning: X does not have valid
feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

[142]: 83.49904677201745

[144]:

```
#we get 83 lacks
```

[150]:

```
predict_price("1st Phase JP Nagar", 1000,3,3)
```

```
C:\Users\dell\anaconda3\envs\tf2\lib\site-
packages\sklearn\utils\validation.py:2749: UserWarning: X does not have valid
feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

[150]: 86.80519395228475

[162]:

```
predict_price("Indira Nagar", 1000,3,4)
```

```
C:\Users\dell\anaconda3\envs\tf2\lib\site-
packages\sklearn\utils\validation.py:2749: UserWarning: X does not have valid
feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

[162]: 182.81142425609204

[164]:

```
df1.head()
```

[164]:

```
              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
```

[166]:

```
#now export this model into a pickle file
```

[170]:

```
import pickle
with open('bangaluru_home_price_model.pickle', 'wb') as f:
    pickle.dump(lr_clf, f)
```

[172]:

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

[ ]:

```

```