	area_typeavailabilitylocationsizesocietytotal_sqftbathbalconyprice0 Super built-up Area19-DecElectronic City Phase II2 BHKCoomee10562.01.039.071 Plot AreaReady To MoveChikka Tirupathi4 BedroomTheanmp26005.03.0120.002 Built-up AreaReady To MoveUttarahalli3 BHKNaN14402.03.062.003 Super built-up AreaReady To MoveLingadheeranahalli3 BHKSoiewre15213.01.095.004 Super built-up AreaReady To MoveKothanur2 BHKNaN12002.01.051.00
In [19]:	4 Super built-up Area Ready To Move Kothanur 2 BHK NaN 1200 2.0 1.0 51.00
	# Column Non-Null Count Dtype
	8 price 13320 non-null float64 dtypes: float64(3), object(6) memory usage: 936.7+ KB Cleaning data df.isnull().sum() # counting the number of null value in each column
[-+]:	area_type 0 availability 0 location 1 size 16 society 5502 total_sqft 0 bath 73 balcony 609 price 0 dtype: int64
In [23]: In [25]:	<pre># assumption since availability does not affect the price of the house in any meaning wa y we will drop it # society has a lot of null values therefore we will drop it df.drop(['availability', 'society'], axis = 1, inplace = True) df.isnull().sum() area_type</pre>
In [27]: Out[27]: In [28]: In [34]:	<pre># imputing the values in balcony column balcony_median = df['balcony'].median() balcony_median</pre>
In [33]:	<pre>location 1 size 16 total_sqft 0 bath 0 balcony 0 price 0 dtype: int64 # same imputing teh values in bath df.bath = df.bath.fillna(df['bath'].median())</pre>
Out[38]: In [42]: In [43]:	<pre>df['size'].dtype dtype('0') df.dropna(inplace = True) df.shape (13303, 7)</pre>
Out[44]:	<pre>df['size'].unique() array(['2 BHK', '4 Bedroom', '3 BHK', '4 BHK', '6 Bedroom', '3 Bedroom',</pre>
In [46]: In [47]: Out[47]:	# so that we can convert this feild into int df['BHK']= df['size'].apply(lambda x: int(x.split(" ")[0])) df area_type
In [48]: Out[48]:	
In [50]: Out[50]:	array(['Electronic City Phase II', 'Chikka Tirupathi', 'Uttarahalli',,
In [58]: Out[58]:	Whitefield 539 Sarjapur Road 397 Electronic City 302 Kanakpura Road 273 Thanisandra 234 Kumbhena Agrahara 1 Pai layout , Mahadevapura 1 RMC YARD 1 Panathur Road, 1
In [59]: Out[59]:	Shakthi Nagar 1 Name: location, Length: 1304, dtype: int64
	Banaswadi 1
In [60]: In [62]:	<pre>white field,kadugodi</pre>
In [62]: Out[62]:	area_type location bath balcony price BHK total_sqft 1 1 1 1 1 1 1.25Acres 1 1 1 1 1 1.26Acres 1 1 1 1 1 1000 172 172 172 172 172
	1000 - 1285 1 <td< td=""></td<>
[n [78]:	<pre>2110 rows × 6 columns # coverting total_sqft into float values def convert_float(x): tokens = x.split('-') if len(tokens)==2: return (float(tokens[1])+float(tokens[0]))/2 try: return float(x)</pre>
<pre>In [79]: In [80]: In [81]: Out[81]:</pre>	<pre>except: return None df1 = df.copy() df1['total_sqft1'] = df1['total_sqft'].apply(convert_float) df1</pre>
¥[area_type location total_sqft bath balcony price BHK total_sqft1 0 Super built-up Area Electronic City Phase II 1056 2.0 1.0 39.07 2 1056.0 1 Plot Area Chikka Tirupathi 2600 5.0 3.0 120.00 4 2600.0 2 Built-up Area Uttarahalli 1440 2.0 3.0 62.00 3 1440.0 3 Super built-up Area Lingadheeranahalli 1521 3.0 1.0 95.00 3 1521.0 4 Super built-up Area Kothanur 1200 2.0 1.0 51.00 2 1200.0
In [83]: Out[83]:	<pre>dtype('float64') df1['total_sqft1'].isnull().sum() 46</pre>
<pre>In [84]: In [85]: In [86]: Out[86]:</pre>	<pre>df1.drop('total_sqft', axis = 1, inplace = True)</pre>
	1 Plot Area Chikka Tirupathi 5.0 3.0 120.00 4 2600.0 2 Built-up Area Uttarahalli 2.0 3.0 62.00 3 1440.0 3 Super built-up Area Lingadheeranahalli 3.0 1.0 95.00 3 1521.0 4 Super built-up Area Kothanur 2.0 1.0 51.00 2 1200.0 13315 Built-up Area Whitefield 4.0 0.0 231.00 5 3453.0 13316 Super built-up Area Richards Town 5.0 2.0 400.00 4 3600.0 13317 Built-up Area Raja Rajeshwari Nagar 2.0 1.0 60.00 2 1141.0
In [87]:	13318 Super built-up Area Padmanabhanagar 4.0 1.0 488.00 4 4689.0 13319 Super built-up Area Doddathoguru 1.0 17.00 1 550.0 13257 rows × 7 columns
	<pre>df1.info() <class 'pandas.core.frame.dataframe'=""> Int64Index: 13257 entries, 0 to 13319</class></pre>
	<pre><class 'pandas.core.frame.dataframe'=""> Int64Index: 13257 entries, 0 to 13319 Data columns (total 7 columns): # Column</class></pre>
	<pre><class 'pandas.core.frame.dataframe'=""> Int64Index: 13257 entries, 0 to 13319 Data columns (total 7 columns): # Column Non-Null Count Dtype</class></pre>
	<pre>cclass 'pandas.core.frame.DataFrame'> Int64Index: 13257 entries, 0 to 13319 Data columns (total 7 columns): # Column</pre>
Out[88]:	<pre>cclass 'pandas.core.frame.DataFrame'> Int64Index: 13257 entries, 0 to 13319 Data columns (total 7 columns):</pre>
Out[88]: Out[90]: In []: In [98]:	<pre>cclass 'pandas.core.frame.DataFrame'> Int64Index: 13257 entries, 0 to 13319 Data columns (total 7 columns):</pre>
Out[88]: Out[90]: In [98]: In [104]:	<pre><class 'pandas.core.frame.dataframe'=""> IntGAIndex: 13257 entries, 8 to 13319 Data columns (Sortal 7 columns):</class></pre>
Out[88]: Out[90]: In [98]: In [104]:	<pre>cclass 'pandas.core.frame.DataFrame'> Int64Index: 13287 entries, 0 to 13319 Data column</pre>
In [90]: In [90]: In [98]: In [104]: In [105]: In [106]: In [107]:	Class Sandas Core Trage Data Frame Data
Out[88]: In [90]: In [98]: In [104]: In [105]: It [106]: It [107]: It [107]: It [109]:	Calaba Sanda Care Frame DataFrame Sanda
Out[88]: In [90]: In [98]: In [104]: In [105]: It [106]: It [107]: It [107]: It [109]:	Column
Dut[88]: In [90]: In [98]: In [104]: In [105]: In [107]: In [109]: In [109]: In [109]:	### Special Conference Secretary Secre
Dut[88]: In [90]: In [90]: In [104]: In [105]: In [107]: In [107]: In [109]: In [109]: In [109]:	### STATE OF CONTROLLED TO STATE OF THE PROPERTY OF CONTROLLED TO STATE OF CONTROLLED TO ST
Out [88]: In [90]: In [98]: In [104]: In [105]: In [107]: In [109]: In [109]: In [109]:	Calcase (Section Control From Marketone) 1
Dut[88]: In [90]: In [90]: In [104]: In [105]: In [107]: In [109]: In [109]: In [110]: In [111]: In [111]: In [112]: In [114]: In [115]:	Calcally (1976) 100
Dut[88]: In [90]: In [90]: In [104]: In [105]: In [107]: In [109]: In [109]: In [110]: In [111]: In [111]: In [112]: In [114]: In [115]:	### Application
Dut[88]: In [90]: In [90]: In [98]: In [104]: In [105]: In [107]: In [109]: In [110]: In [111]: In [111]: In [111]: In [112]: In [114]: In [115]:	Section Commonweal Common
In [90]: Out [88]: Out [90]: In []: In [98]: In [104]: In [105]: In [107]: In [109]: In [111]: In [111]: In [111]: In [111]: In [112]: In [114]: In [115]: I	### Color Section Process Color Process Color Colo
In [90]: In [90]: In [90]: In [98]: In [104]: In [105]: In [107]: In [109]: In [111]: In [111]: In [111]: In [111]: In [115]: In [116]: In [117]: In [118]:	### STATE OF THE PARTY OF THE ADMINISTRATION OF THE PARTY OF THE ADMINISTRATION OF THE A
Dut[88]: In [90]: In [90]: In [98]: In [105]: In [106]: In [107]: In [107]: In [109]: In [110]: In [111]: In [111]: In [112]: In [### Company Section Se
Dut[88]: Out[88]: Out[90]: Out[90]: In [90]: In [105]: In [106]: In [107]: In [107]: In [109]: In [110]: In [111]: In [111]: In [112]: In [1	### PATE OF THE PA
Dut[88]: In [90]: In [90]: In [90]: In [105]: In [105]: In [106]: In [107]: In [109]: In [110]: In [121]: In [122]: In [123]: In [126]: In [127]: In [128]: In [Column C
Dut[88]: In [90]: In [90]: In [90]: In [105]: In [105]: In [106]: In [107]: In [109]: In [110]: In [121]: In [122]: In [123]: In [126]: In [127]: In [128]: In [Column
Dut[88]: In [99]: In [99]: In [98]: In [104]: In [105]: In [107]: In [109]: In [110]: In [111]: In [112]: In [112]: In [112]: In [112]: In [112]: In [12]: In [1	### Company of the Co
Dut [88]: Out [88]: Out [90]: Out [90]: Out [90]: Out [105]: Out [105]: Out [106]:	## Company of the Com
Dut[88]: In [99]: In [99]: In [98]: In [105]: In [106]: In [107]: In [109]: In [110]: In [110]: In [111]: In [112]: In [112]: In [12]: In [12]	March Marc
In [90]: In [90]: In [90]: In [98]: In [104]: In [105]: In [106]: In [107]: In [109]: In [110]: In [111]: In [112]: In [112]: In [112]: In [112]: In [123]: In [124]: In [127]: In [128]: In [128]: In [127]: In [128]:	Section Comparison Compar
In [90]: In [90]: In [90]: In [90]: In [90]: In [105]: In [105]: In [106]: In [106]: In [107]: In [110]: In [111]: In [112]: In [112]: In [121]: In [122]: In [123]: In [126]: In [126]: In [127]: In [128]: In [1	Section Comparison Compar
Dut[90]: Dut[90]: In [9]: In [98]: In [104]: In [105]: In [106]: In [106]: In [107]: In [109]: In [110]: In [111]: In [112]: In [121]: In [121]: In [122]: In [123]: In [124]: In [126]: In [127]: In [128]: In [129]: In [120]: In [1	Compared
th [90]: th [90]: th [90]: th [10]: th [11]: th [12]: th [12]: th [12]: th [13]:	March Marc
In [90]: In [90]: In [90]: In [90]: In [90]: In [100]: I	Company Comp
In [109]: In [100]:	March Marc

Bengaluru House Price

The Data has the following columns

5. Society: type of locality

1. Area type: Built up area, carpet area rtc

I have used linear regression the predict the price of the houses.

2. Avalaibility: from when can the tenants take the house

3. Location: Name of the area where the house is located

4. Size: in the format of BHK(bedroom, hall, kitchen)

6. total_sqft: total floor area of the house in square foot

7. bath: Number of bathrooms in the house

8. balcony: Number of balconies in the house

9. price: price of the house in lakhs