House Price Prediction

Project Report

Submitted in partial fulfilment of the requirement

For

MCA

Under the guidance

Of

Prof. Aashish Bhatnagar

Assistant Professor

INMANTEC, AKTU University

Ghaziabad, U.P

By

Abhishek

(2101130140003)

Integrated Academy of Management and Technology, Ghaziabad

MCA- 3rd Semester

INMANTEC Institutions, Delhi-Hapur Bypass
(NH-24 near Dasna Flyover, Udyog Kunj,
Ghaziabad, Uttar Pradesh 201009

INDEX

Integrated Academy of Management and Technology, Ghaziabad

Minor Project –KCA 353
MCA

ACKNOWLEDGEMENT

I am very grateful to my Minor Project (KCA 353) Prof. Aashish Bhatnagar for giving his valuable time and constructive guidance in preparing the Synopsis/Minor Project (KCA 353). It would not have been possible to complete this Minor Project (KCA 353) in short period of time without his/her kind encouragement and valuable guidance.

Date:

Signature

Abstract

- House Price forecasting is an important topic of real estate. The literature attempts to derive useful knowledge from historical data of property market.

- Machine learning techniques are applied to analyze historical property transaction in India (Bangaluru) to discover useful models for house buyers and seller.

- Revealed is the high discrepancy between house prices in the most expensive and most affordable suburbs in the city of Bangaluru.

- Moreover, experiments demonstrate that the Multiple Linear Regression that is based on mean squared error measurement is a competitive approach.

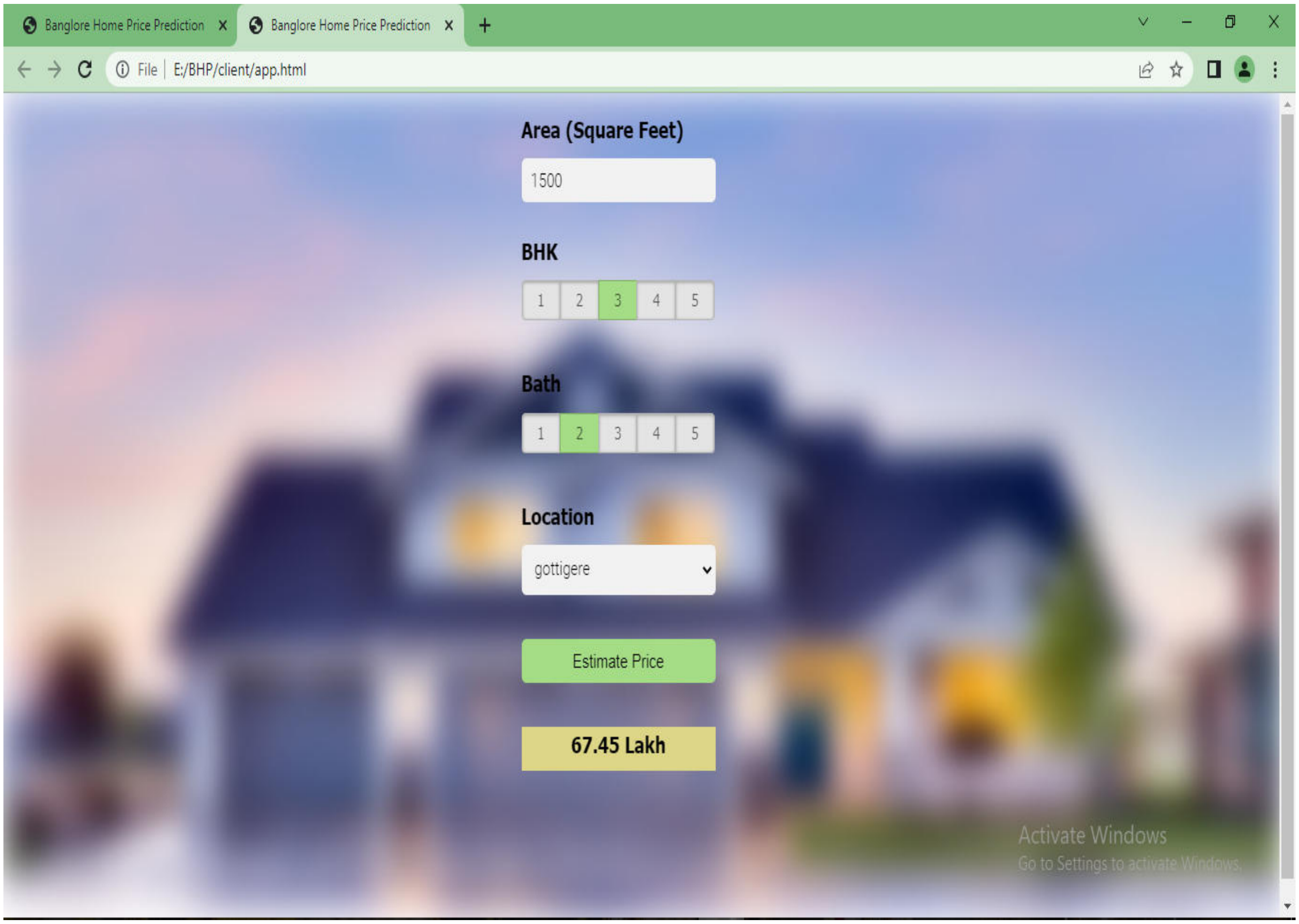
Intordunction

- House is one of the basic needs for a person and their prices vary from place to place depending on available amenities like parking place, locality, no_of_sqrft, bath, bhk etc.
- Buying a home is one of the biggest and most important choices for a family as they put all of their funds into investment and cover them over time with loans.
- In this project, we will implement a Bangalore House Price Prediction model using a Machine Learning Algorithm (Multiple Linear Regression) using Python Language.
- This model predicts the price of Bangalore's house with the help of few parameters like availability, size, total square feet, bath, location etc.
- Our model can be used by both house sellers and house buyers.
- For a user, employing a House Price system is one of the ways to reach the house price information that interests him.
- Bangaluru House price dataset is used to create the model. We are using Machine Learning Algorithm to create a predictive model.
- Multiple Linear Regression Algorithm is used to train and test the model in our project and predict the House price.
- Python is widely used for House Price Prediction.

Purpose of a recommendation system

There is a user viewpoint here: to easily and quickly find information of House Price in Bangulure save user’s time.

There is a viewpoint of the owner of the House : to add value to the *service, gain new users, increase sales of House though providing valuable price to user.*


Banglore Home Price Prediction
Banglore Home Price Prediction
File
E:/BHP/client/app.html
1
2
3
4
5
1
2
3
4
5
Estimate Price
Activate Windows
Go to Settings to activate Windows.

Area (Square Feet)

1500

BHK

3

Bath

2

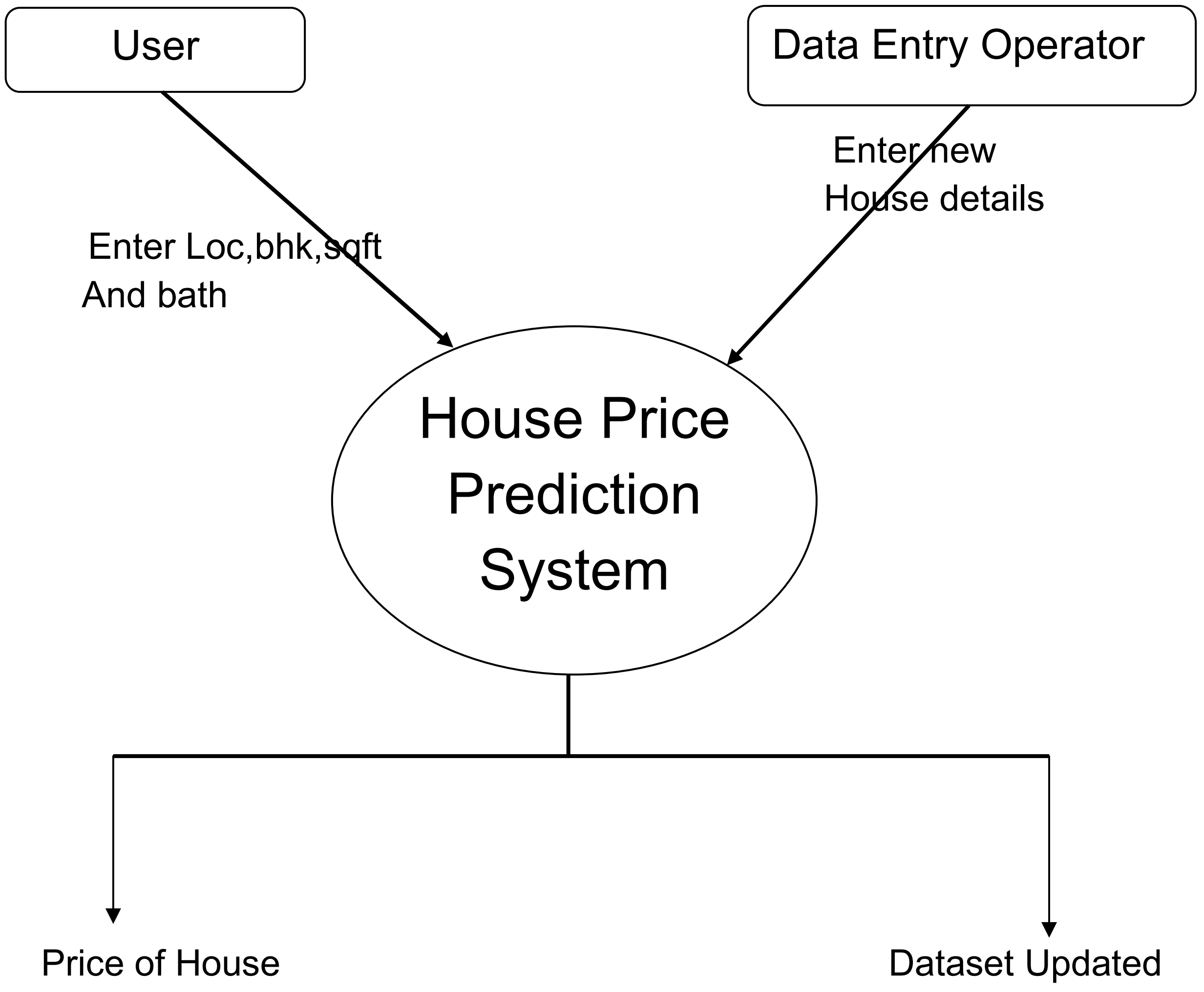
Location

gottigere

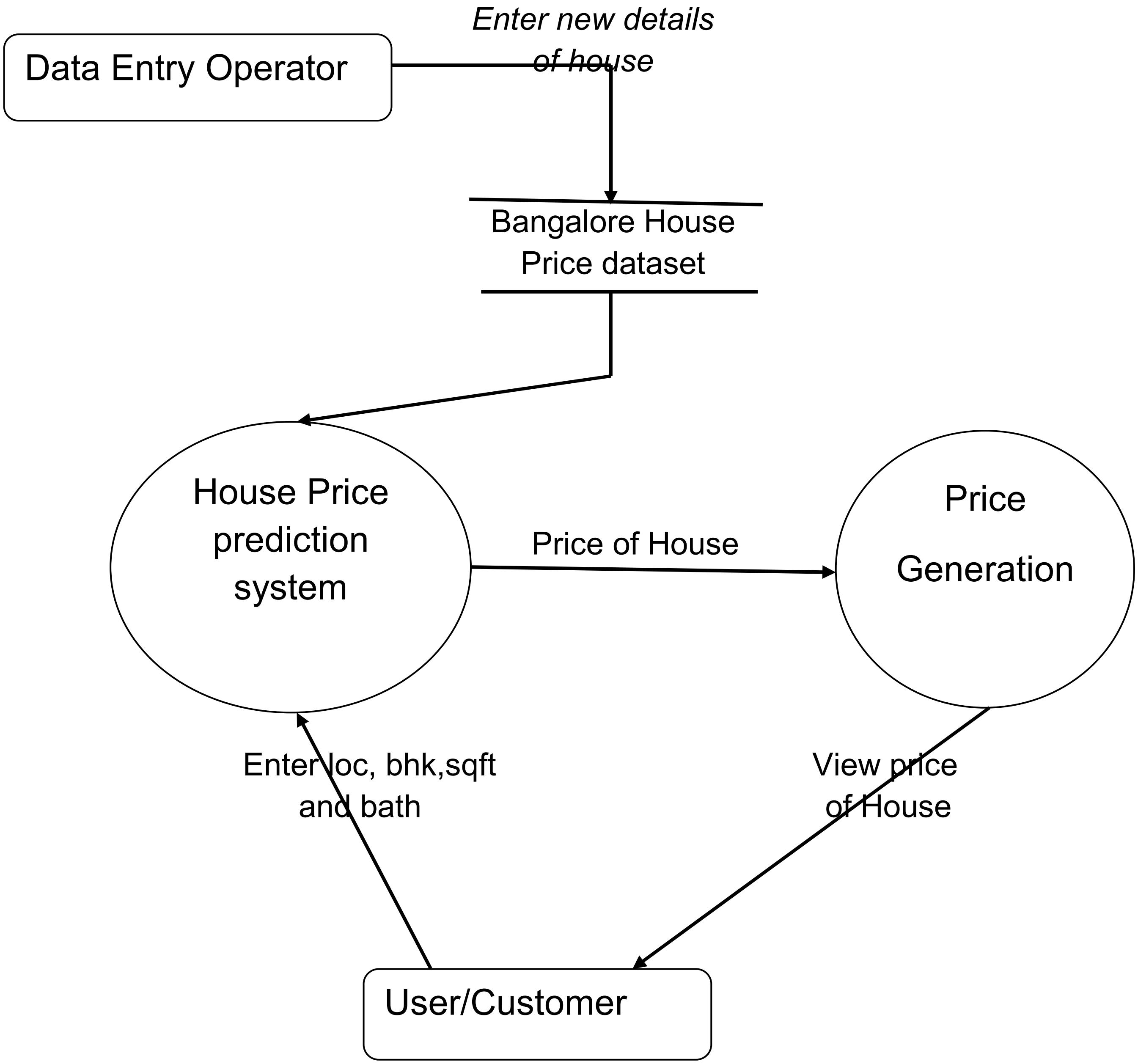
67.45 Lakh

Data Flow Diagram (DFD)

Level-0 DFD :-



Level-1 DFD :-


Data Entry Operator
Enter new details of house
Bangalore House Price dataset
House Price prediction system
Price of House
Price Generation
Enter loc, bhk,sqft and bath
View price of House
User/Customer

SYSTEM SOFTWARE REQUIREMENT SPECIFICATION (SRS)

Below are the requirements used for House Price prediction System

System Requirement

- **Jupyter Notebook:-** pip install jupyterlab
- **PyCharm:-** https://www.jetbrains.com/pycharm/download/#section=windows

Windows-Based Requirements

- Dual-core 64-bit processor
- 8 GB of memory
- Up to 24 GB of internal storage (Jupyter Notebook & PyCharm: 2.5GB+1GB for caches,)
- Windows 10, Windows 8.1 Update, Windows 8, and Windows 7.1

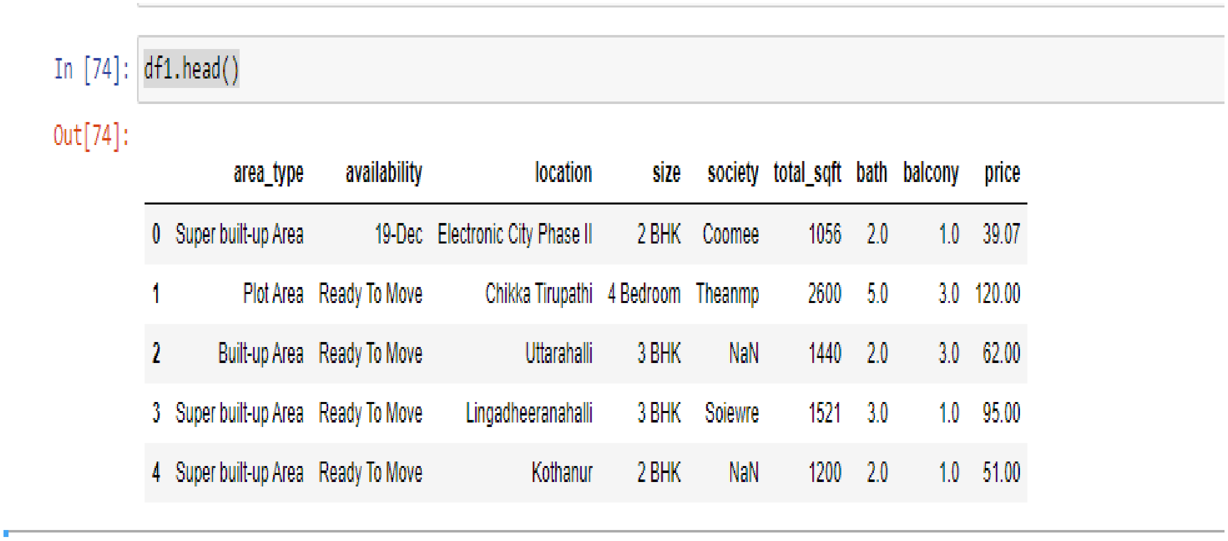
Library Requirements of PyCharm Platform

- Pandas (Accessing and modifying Datasets)
- Numpy (Creating Multidimensional array)
- Matplotlib(Data Visualization)
- Sklearn (Selecting, building and testing the Multiple Linear Regression model)
- Flask(For Server Side Programming)

Code

Model Building:-

```
import pandas as pd

import numpy as np

from matplotlib import pyplot as plt

%matplotlib inline

import matplotlib

matplotlib.rcParams["figure.figsize"]=(20,10)

df1=pd.read_csv('Bengaluru_House_Data.csv')

df1.head()
```

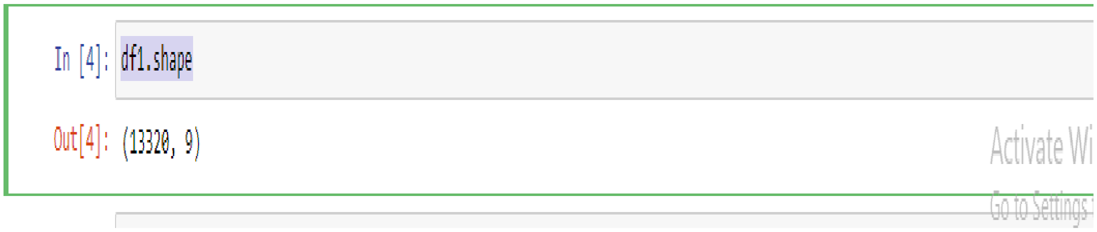


```
In [74]: df1.head()
```

Out[74]:

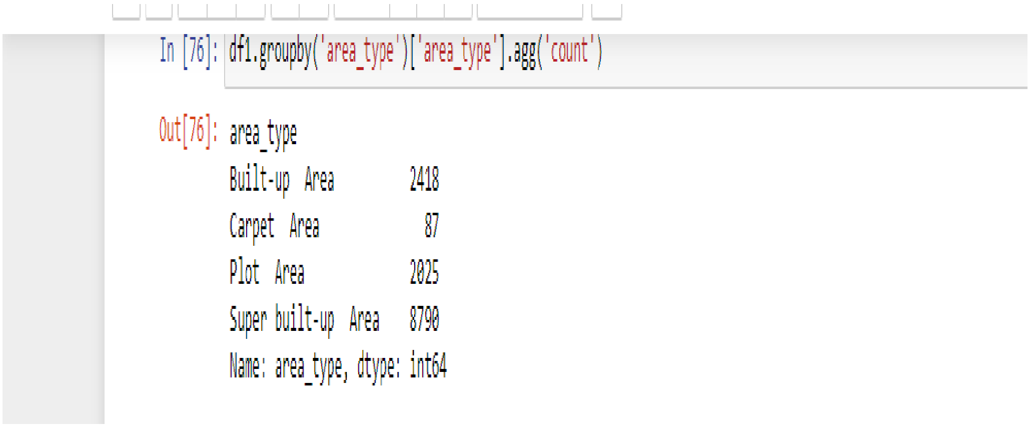
	area_type	availability	location	size	society	total_sqft	bath	balcony	price
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056	2.0	1.0	39.07
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600	5.0	3.0	120.00
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	NaN	1440	2.0	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
```



```
In [4]: df1.shape

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

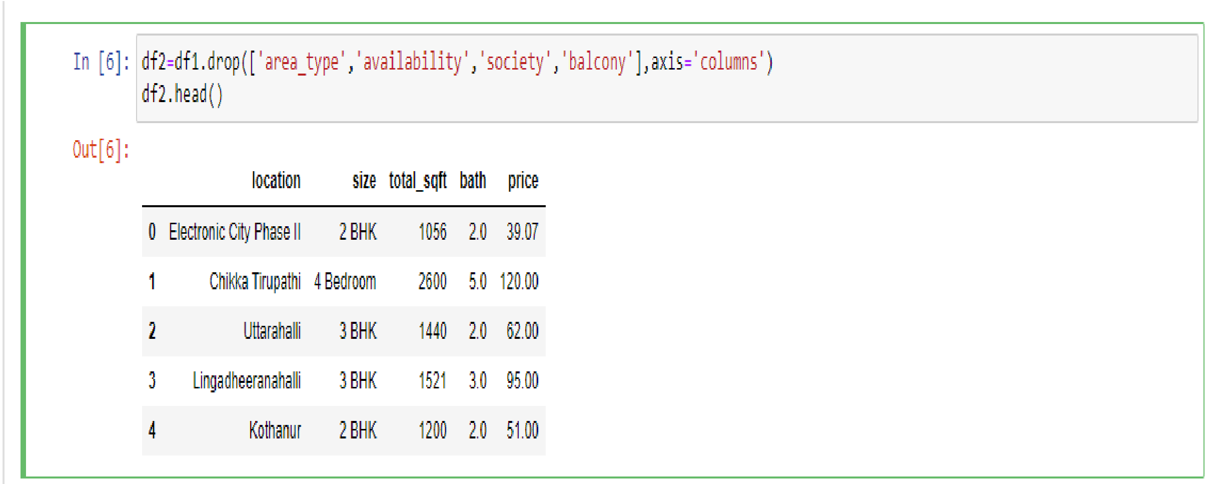
df1.groupby('area_type')['area_type'].agg('count')



```
In [76]: df1.groupby('area_type')['area_type'].agg('count')

Out[76]: area_type
         Built-up  Area        2418
         Carpet  Area            87
         Plot  Area            2025
         Super built-up  Area  8790
         Name: area_type, dtype: int64
```

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

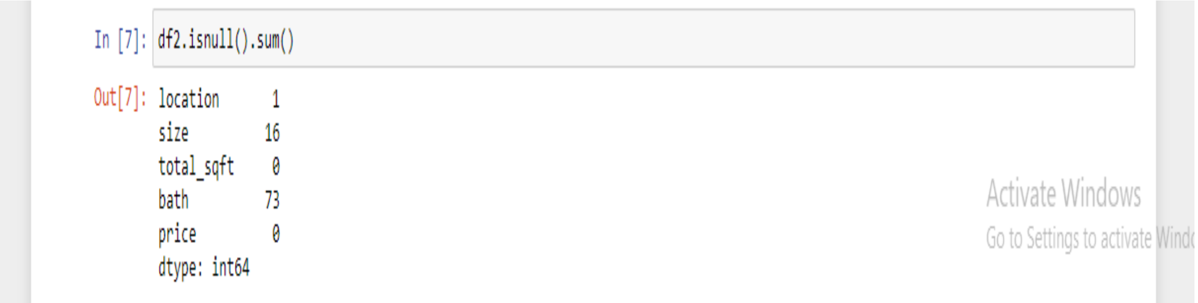
df2.head()



```
In [6]: df2=df1.drop(['area_type','availability','society','balcony'],axis='columns')
        df2.head()

Out[6]:
```

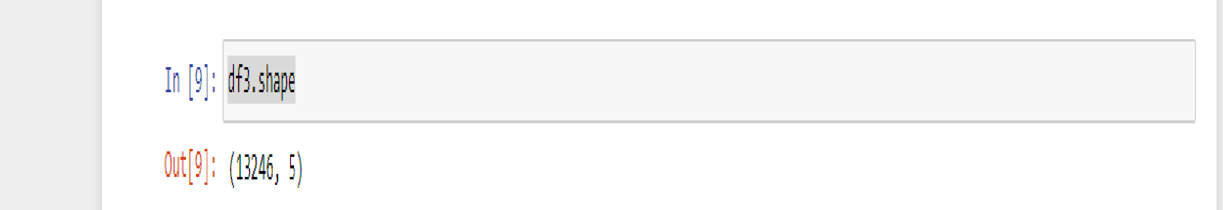
	location	size	total_sqft	bath	price
0	Electronic City Phase II	2 BHK	1056	2.0	39.07
1	Chikka Tirupathi	4 Bedroom	2600	5.0	120.00
2	Uttarahalli	3 BHK	1440	2.0	62.00
3	Lingadheeranahalli	3 BHK	1521	3.0	95.00
4	Kothanur	2 BHK	1200	2.0	51.00

df2.isnull().sum()



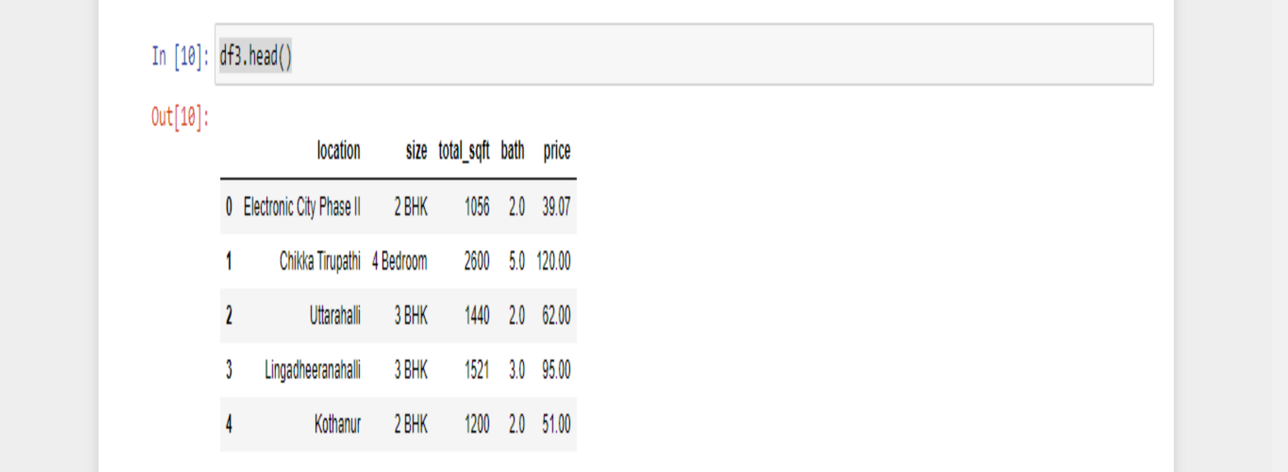
```
In [7]: df2.isnull().sum()

Out[7]: location      1
        size         16
        total_sqft    0
        bath         73
        price         0
        dtype: int64
```

df3.shape



```
In [9]: df3.shape

Out[9]: (13246, 5)
```

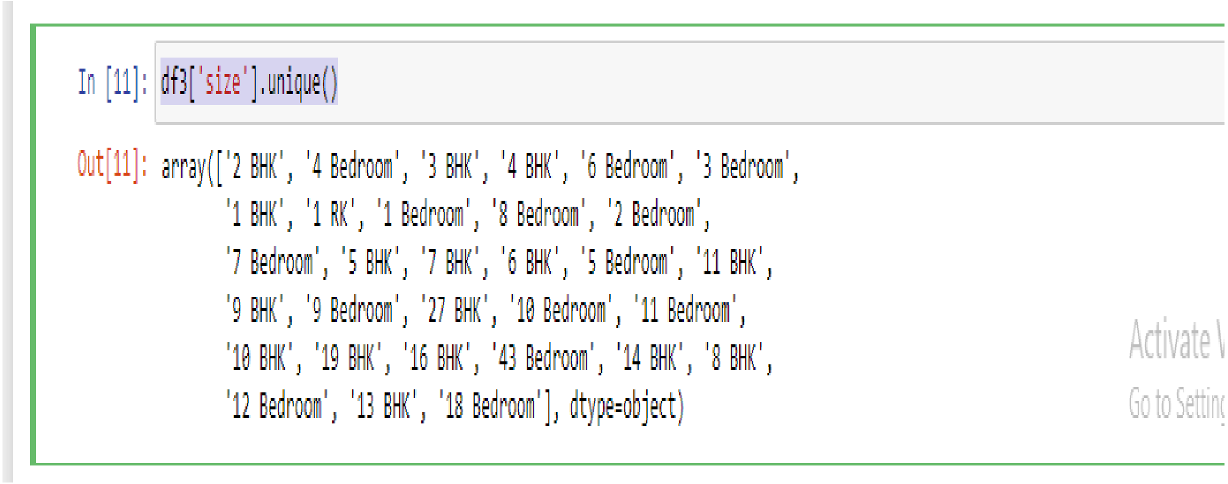
df3.head()



```
In [10]: df3.head()

Out[10]:
```

	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

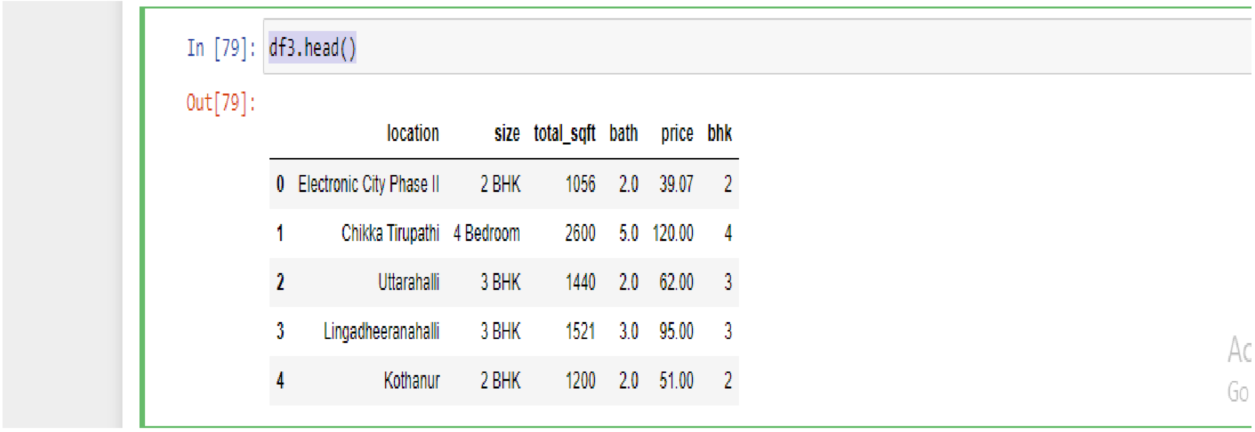
df3['size'].unique()



```
In [11]: df3['size'].unique()

Out[11]: 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)
```

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

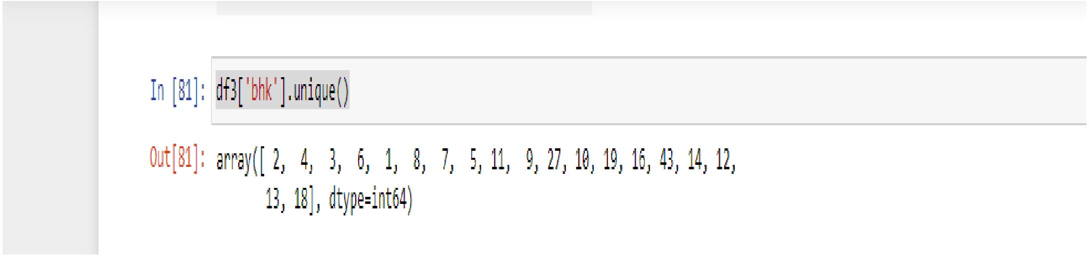
df3.head()



```
In [79]: df3.head()

Out[79]:
```

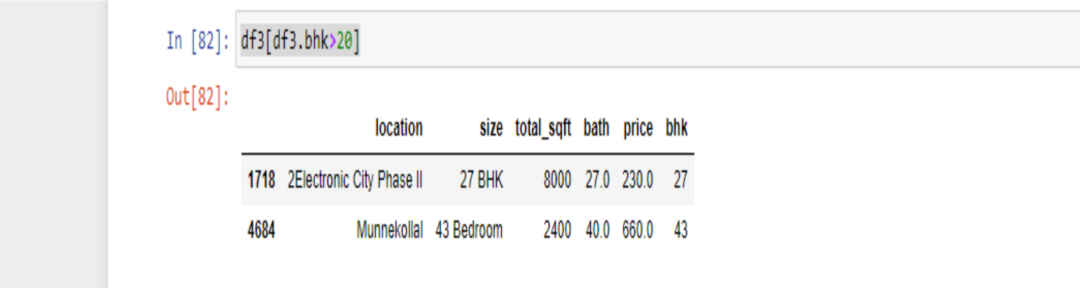
	location	size	total_sqft	bath	price	bhk
0	Electronic City Phase II	2 BHK	1056	2.0	39.07	2
1	Chikka Tirupathi	4 Bedroom	2600	5.0	120.00	4
2	Uttarahalli	3 BHK	1440	2.0	62.00	3
3	Lingadheeranahalli	3 BHK	1521	3.0	95.00	3
4	Kothanur	2 BHK	1200	2.0	51.00	2

df3['bhk'].unique()



```
In [81]: df3['bhk'].unique()

Out[81]: array([ 2,  4,  3,  6,  1,  8,  7,  5, 11,  9, 27, 10, 19, 16, 43, 14, 12,
                13, 18], dtype=int64)
```

df3[df3.bhk>20]

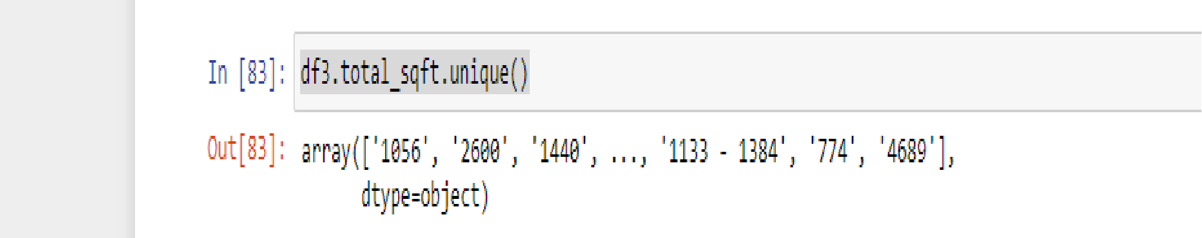


In [82]:
```
df3[df3.bhk>20]
```

Out[82]:

	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

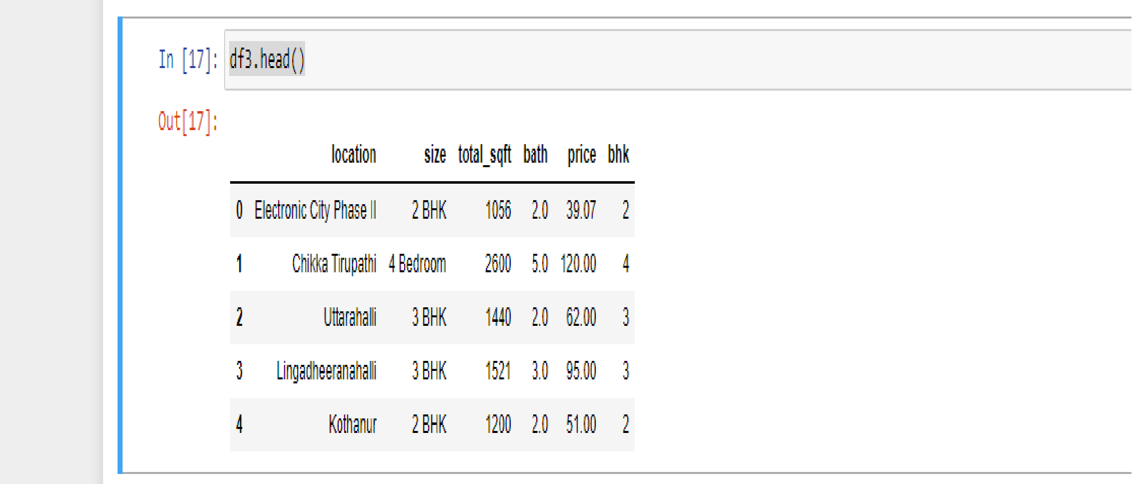
df3.total_sqft.unique()



In [83]:
```
df3.total_sqft.unique()
```

Out[83]:
```
array(['1056', '2600', '1440', ..., '1133 - 1384', '774', '4689'],
      dtype=object)
```

df3.head()



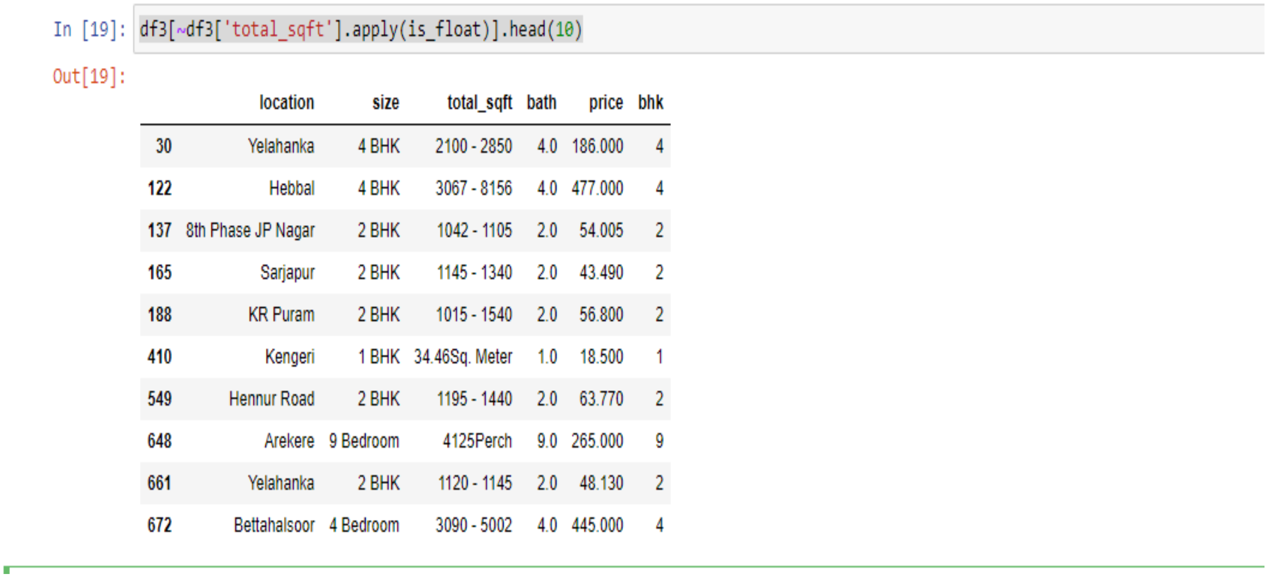
In [17]:
```
df3.head()
```

Out[17]:

	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

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

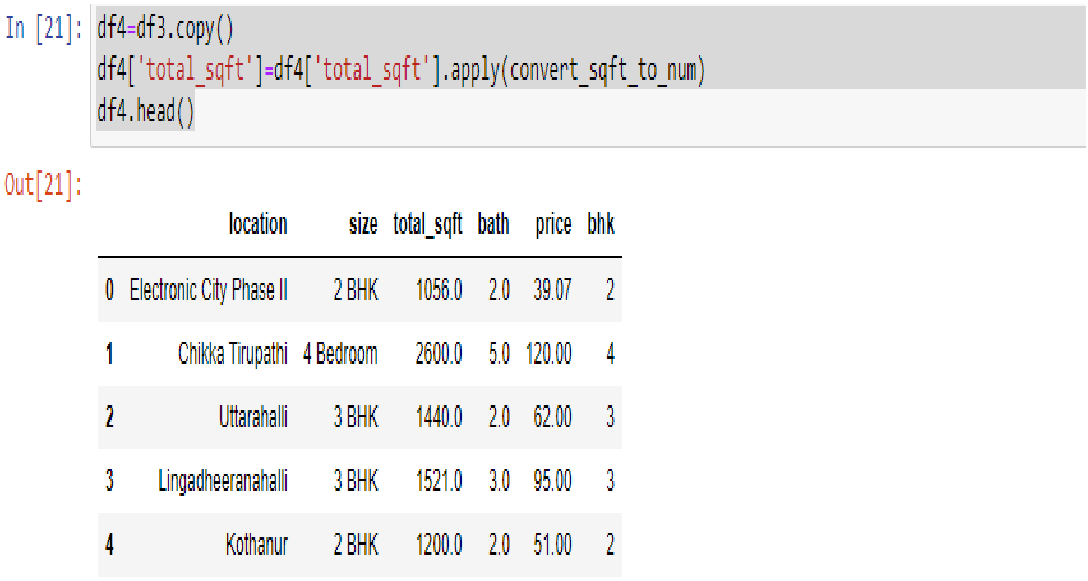
```
df3[~df3['total_sqft'].apply(is_float)].head(10)
```



```
In [19]: df3[~df3['total_sqft'].apply(is_float)].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

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

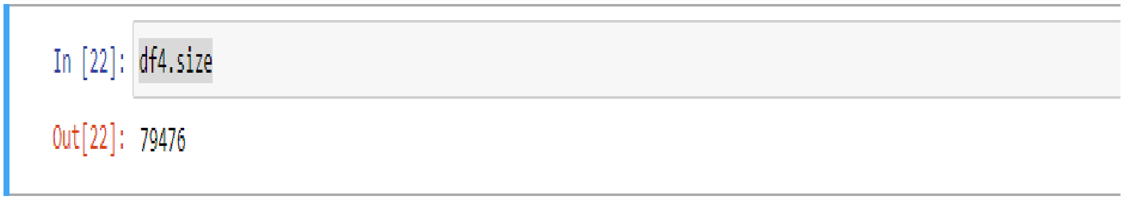


```
In [21]: df4=df3.copy()
         df4['total_sqft']=df4['total_sqft'].apply(convert_sqft_to_num)
         df4.head()
```

Out[21]:

	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

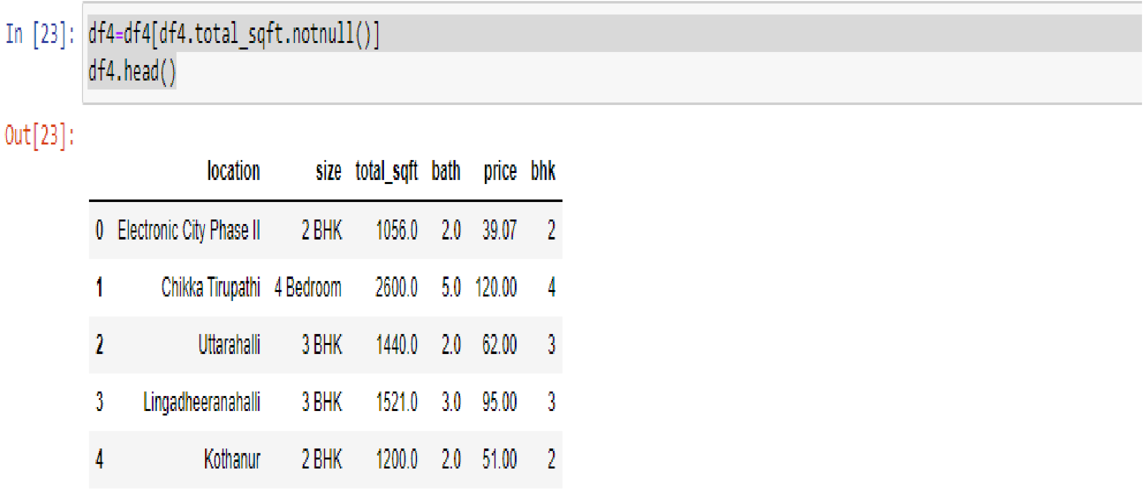
df4.size



```
In [22]: df4.size

Out[22]: 79476
```

df4=df4[df4.total_sqft.notnull()]

df4.head()

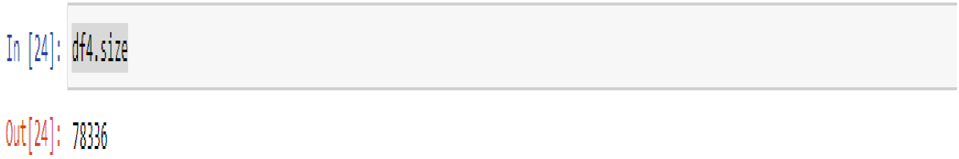


```
In [23]: df4=df4[df4.total_sqft.notnull()]
         df4.head()
```

Out[23]:

	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

df4.size

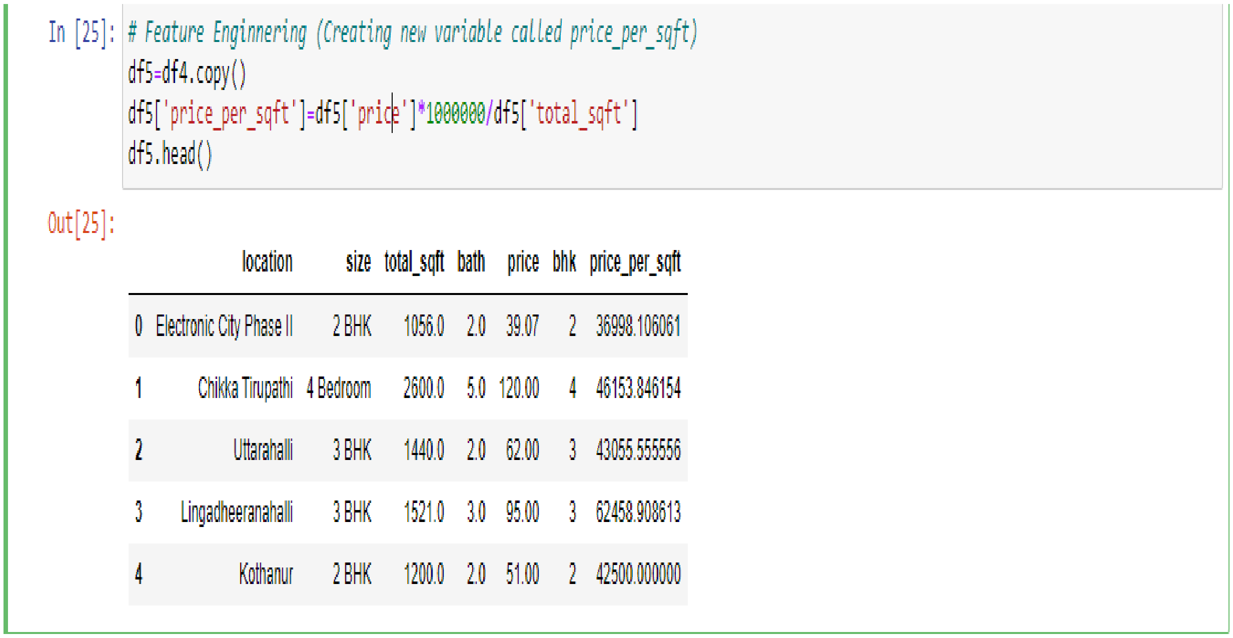


```
In [24]: df4.size

Out[24]: 78336
```

df5=df4.copy()

df5['price_per_sqft']=df5['price']*1000000/df5['total_sqft']

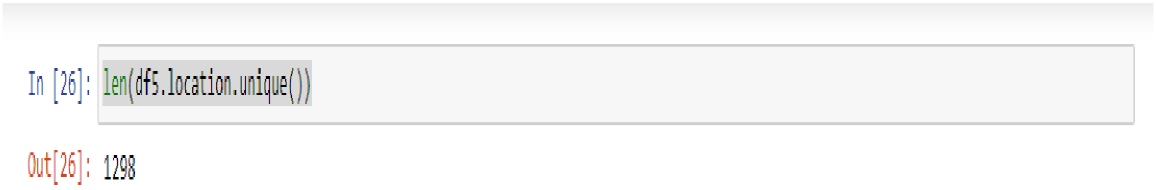
df5.head()



```
In [25]: # Feature Enginnering (Creating new variable called price_per_sqft)
         df5=df4.copy()
         df5['price_per_sqft']=df5['price']*1000000/df5['total_sqft']
         df5.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

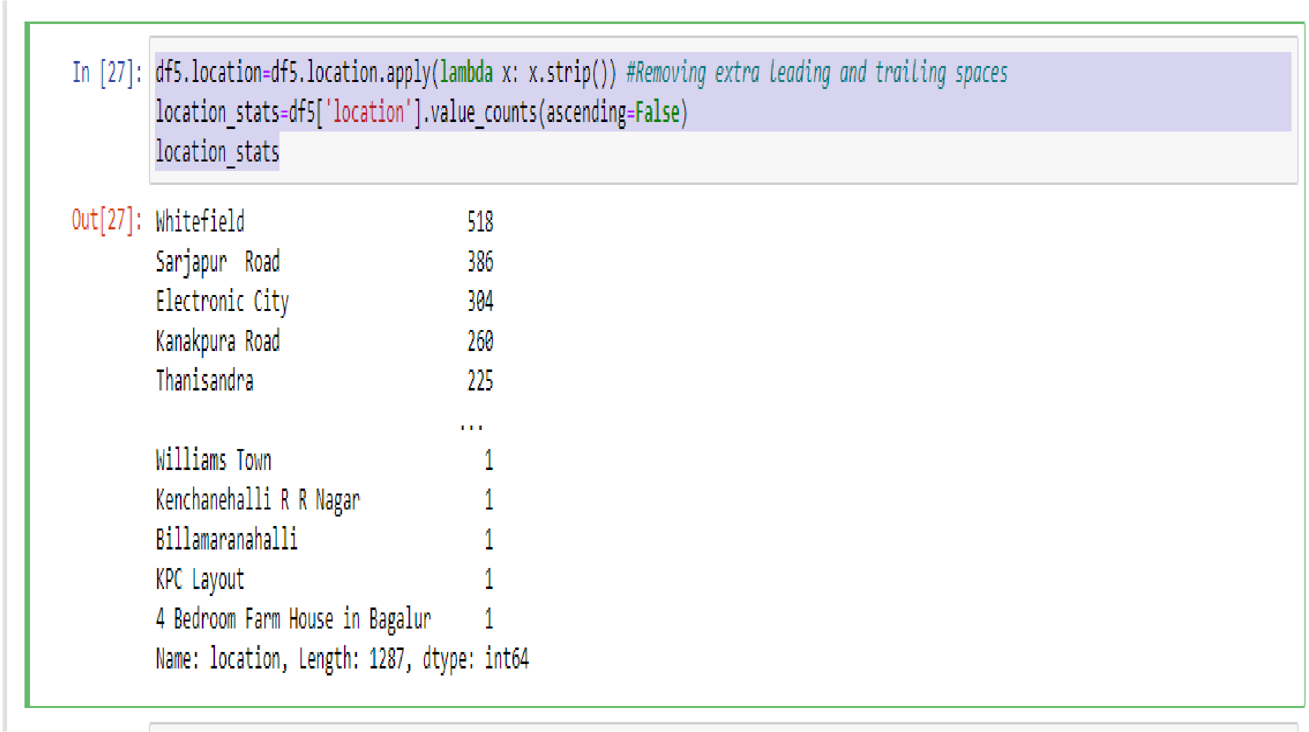
len(df5.location.unique())



```
In [26]: len(df5.location.unique())

Out[26]: 1298
```

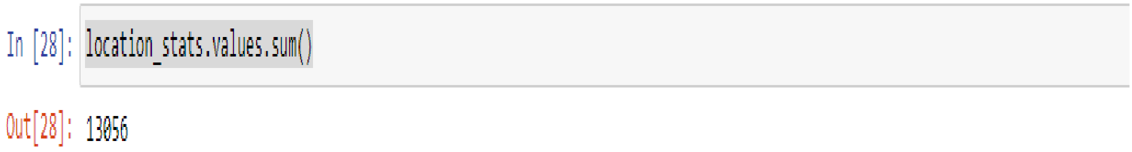
df5.location=df5.location.apply(lambda x: x.strip()) #Removing extra leading and trailing spaces

location_stats=df5['location'].value_counts(ascending=False) location_stats



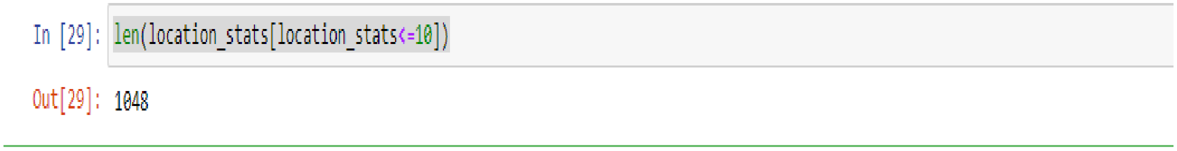
```
In [27]: df5.location=df5.location.apply(lambda x: x.strip()) #Removing extra leading and trailing spaces
         location_stats=df5['location'].value_counts(ascending=False)
         location_stats

Out[27]: Whitefield                     518
         Sarjapur  Road                 386
         Electronic City                304
         Kanakpura Road                 260
         Thanisandra                    225
                                       ...
         Williams Town                    1
         Kenchanehalli R R Nagar          1
         Billamaranahalli                 1
         KPC Layout                       1
         4 Bedroom Farm House in Bagalur  1
         Name: location, Length: 1287, dtype: int64
```

location_stats.values.sum()



```
In [28]: location_stats.values.sum()

Out[28]: 13056
```

len(location_stats[location_stats<=10])

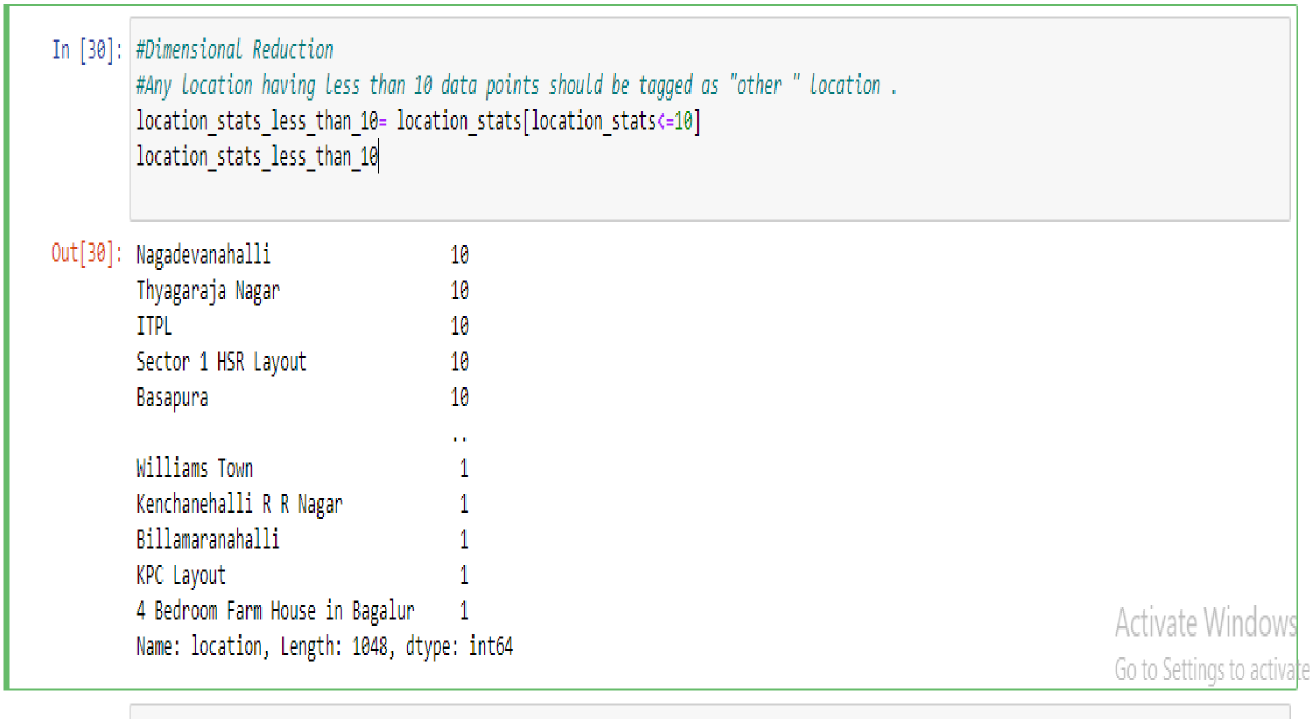


```
In [29]: len(location_stats[location_stats<=10])

Out[29]: 1048
```

#Dimensional Reduction

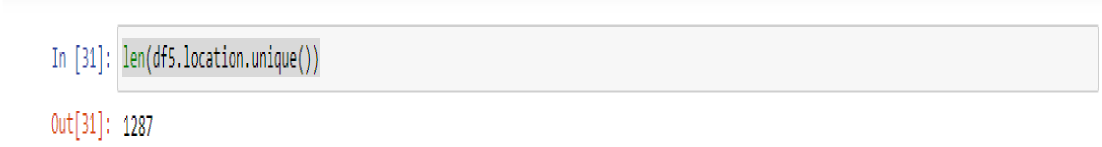
#Any location having less than 10 data points should be tagged as "other " location .

location_stats_less_than_10= location_stats[location_stats<=10] location_stats_less_than_10



```
In [30]: #Dimensional Reduction
         #Any location having less than 10 data points should be tagged as "other " location .
         location_stats_less_than_10= location_stats[location_stats<=10]
         location_stats_less_than_10

Out[30]: Nagadevanahalli              10
         Thyagaraja Nagar             10
         ITPL                         10
         Sector 1 HSR Layout          10
         Basapura                     10
                                      ..
         Williams Town                 1
         Kenchanehalli R R Nagar       1
         Billamaranahalli              1
         KPC Layout                    1
         4 Bedroom Farm House in Bagalur     1
         Name: location, Length: 1048, dtype: int64
```

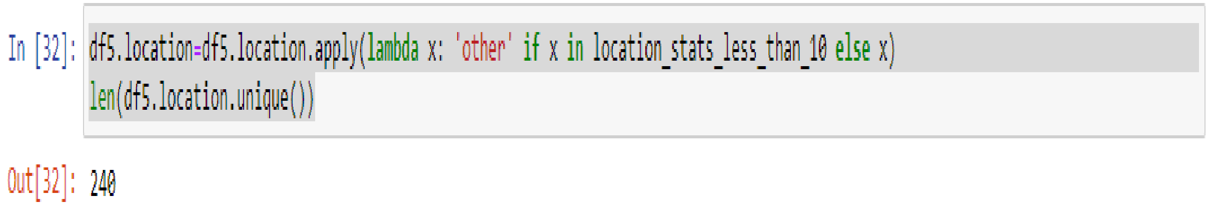
len(df5.location.unique())



```
In [31]: len(df5.location.unique())

Out[31]: 1287
```

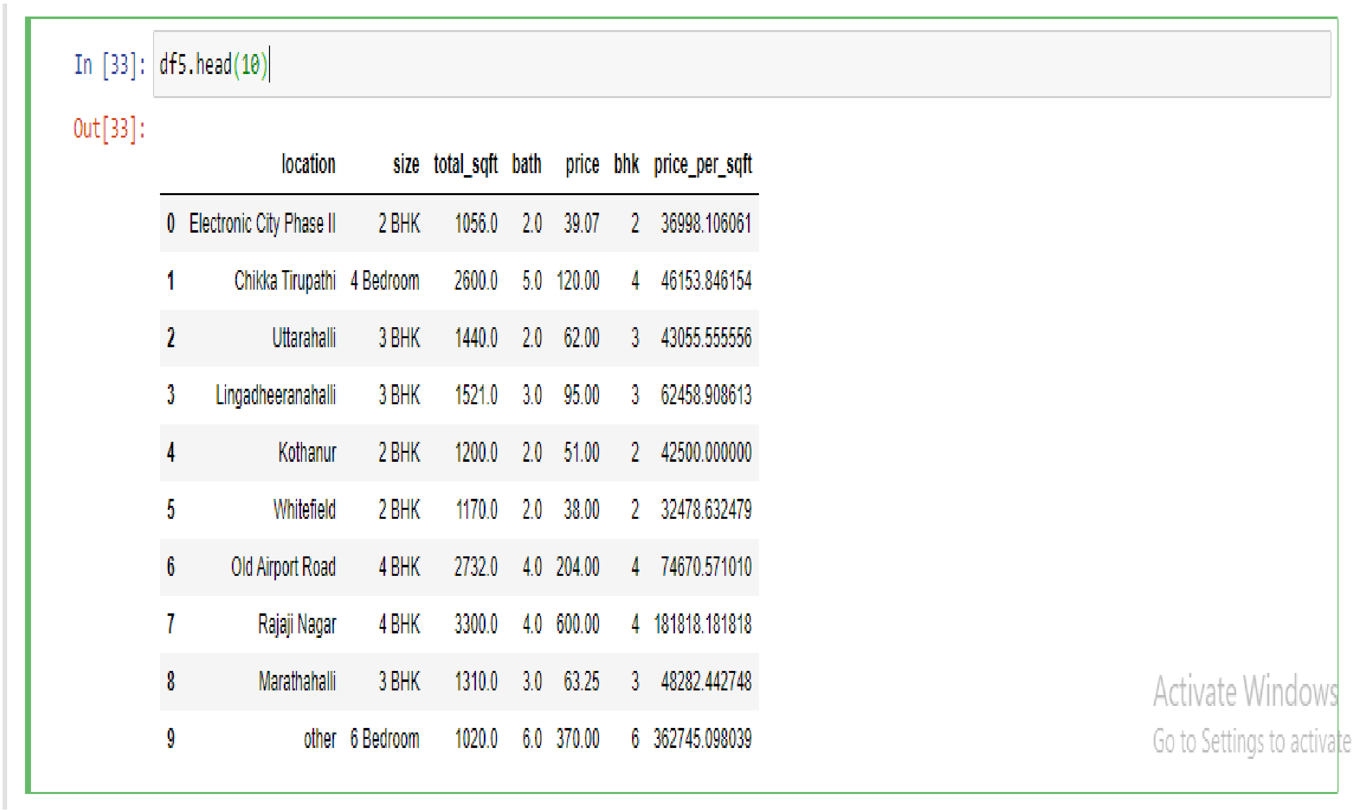
df5.location=df5.location.apply(lambda x: 'other' if x in location_stats_less_than_10 else x)

len(df5.location.unique())



```
In [32]: df5.location=df5.location.apply(lambda x: 'other' if x in location_stats_less_than_10 else x)
         len(df5.location.unique())

Out[32]: 240
```

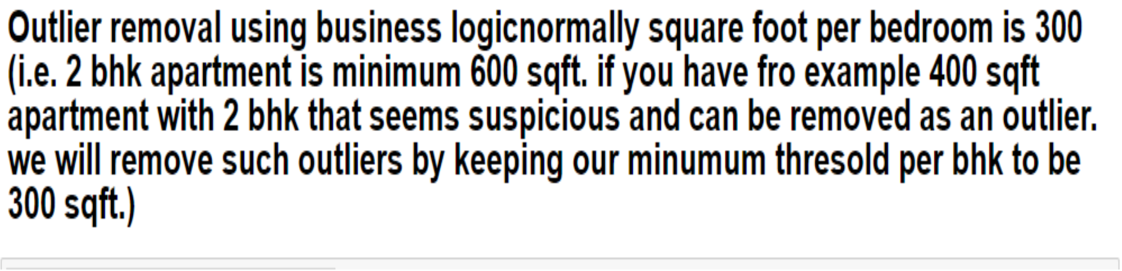
df5.head(10)



```
In [33]: df5.head(10)
```

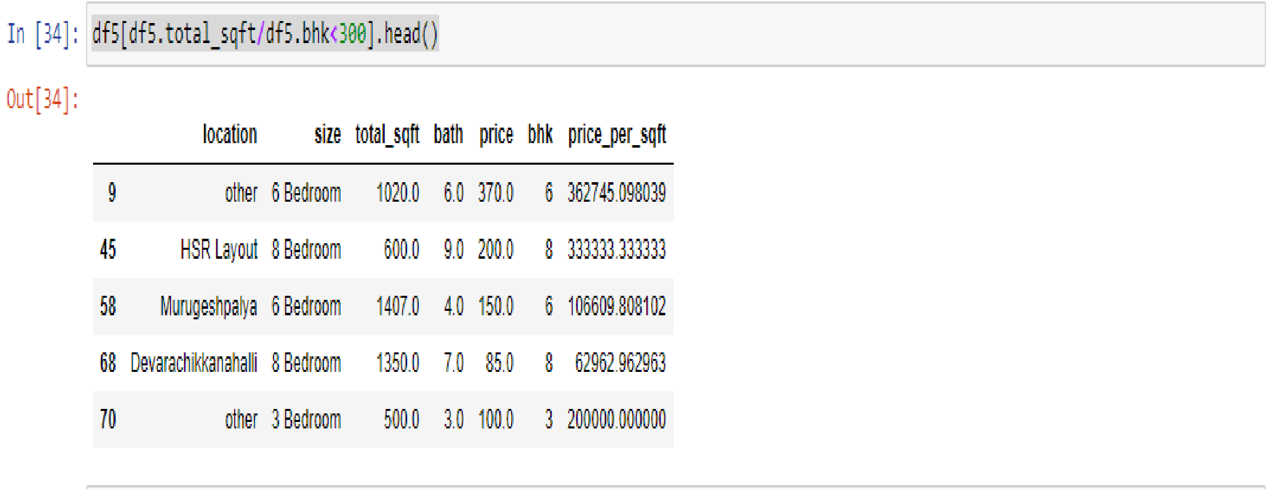
Out[33]:

	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



Outlier removal using business logicnormally square foot per bedroom is 300 (i.e. 2 bhk apartment is minimum 600 sqft. if you have fro example 400 sqft apartment with 2 bhk that seems suspicious and can be removed as an outlier. we will remove such outliers by keeping our minumum thresold per bhk to be 300 sqft.)

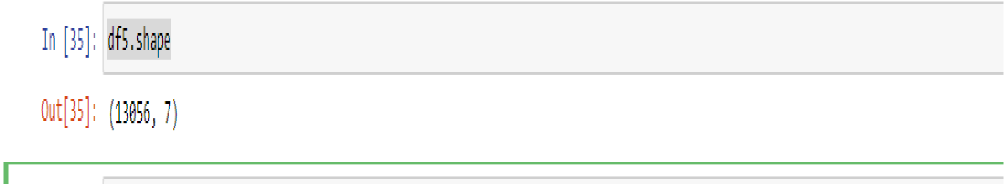
```
df5[df5.total_sqft/df5.bhk<300].head()
```



```
In [34]: df5[df5.total_sqft/df5.bhk<300].head()
Out[34]:
```

	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

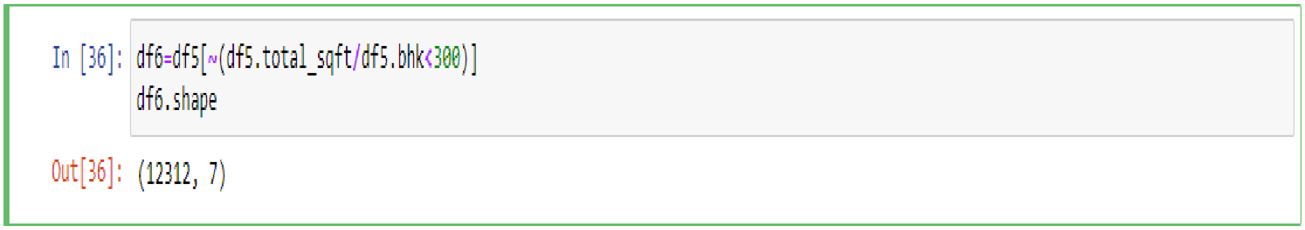
df5.shape



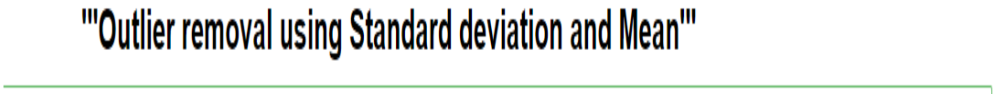
```
In [35]: df5.shape
Out[35]: (13056, 7)
```

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

df6.shape

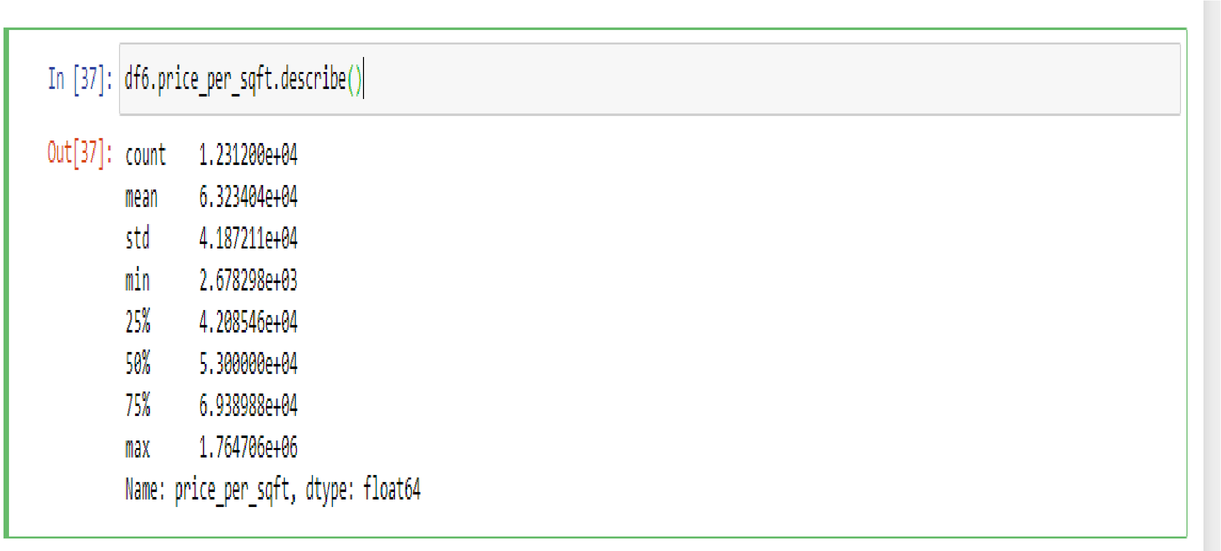


```
In [36]: df6=df5[~(df5.total_sqft/df5.bhk<300)]
         df6.shape
Out[36]: (12312, 7)
```



'''Outlier removal using Standard deviation and Mean'''

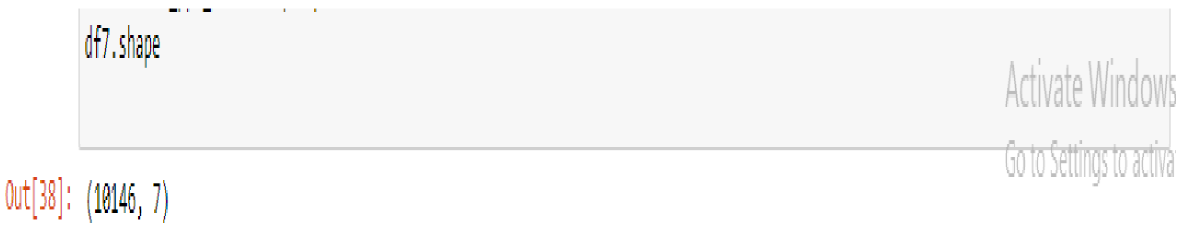
df6.price_per_sqft.describe()



```
In [37]: df6.price_per_sqft.describe()

Out[37]: count    1.231200e+04
         mean     6.323404e+04
         std      4.187211e+04
         min      2.678298e+03
         25%      4.208546e+04
         50%      5.300000e+04
         75%      6.938988e+04
         max      1.764706e+06
         Name: price_per_sqft, dtype: float64
```

# Here we find that min price per sqft is 267 rs/sqft and max is 176470 rs/sqft this shows 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
```

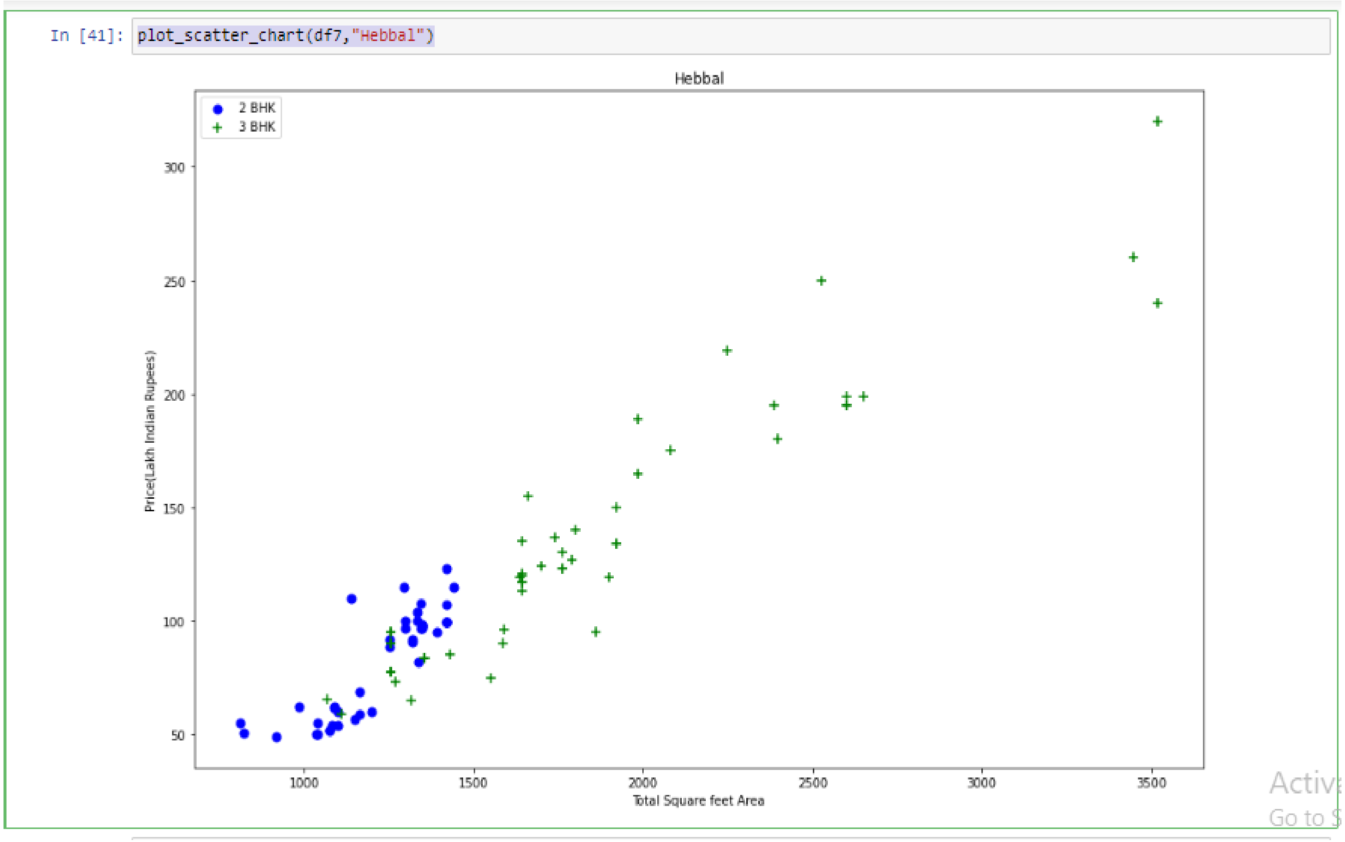


```
df7.shape
```

```
Out[38]: (10146, 7)
```

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")
```



```
In [41]: plot_scatter_chart(df7,"Hebbal")
```

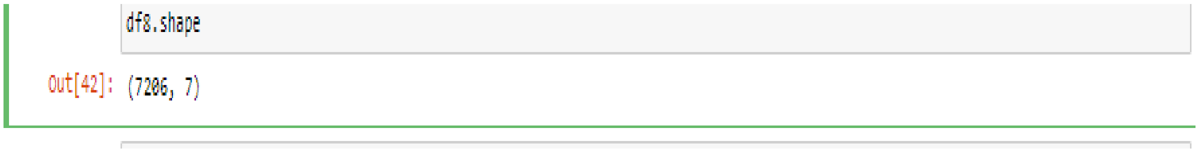
We should alos remove properties where for same location, the price of (for example) 3 bhk is less than 2 bhk (with same square ft area.). what we will do is for a given location, we will build a dictionary of stats per bhk, i.e.

```
# {
 '1':
    {
     'mean': 400,
     'std': 2000,
     'count': 34
    },
    '2':
     {
       'mean': 4300,
       'std': 2300,
       'count': 22
     },
}
```

now we can remove those 2 BHK apartments whose price_per_sqft is less than meaqn price_per_sqft of 1 BHK apartment

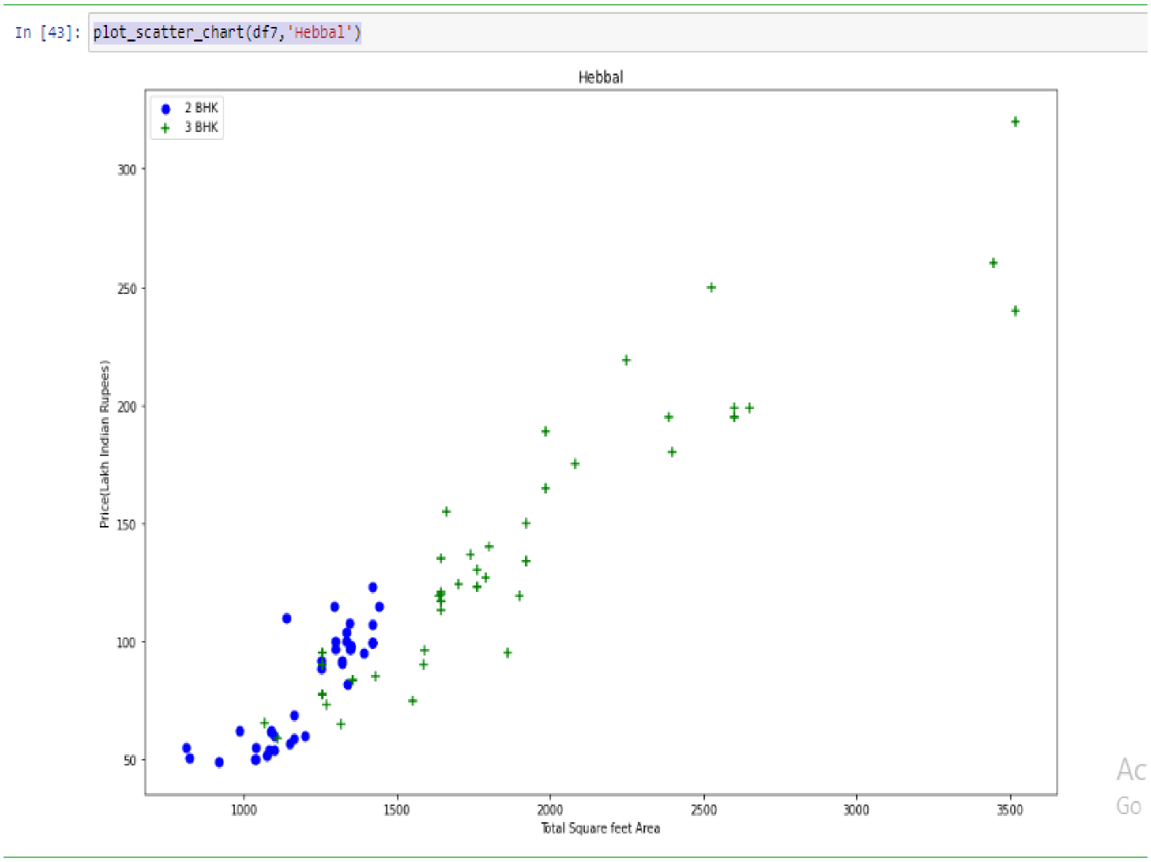
```
def remove_bhk_outliers(df):
    exclude_indices = np.array([])
    for location, location_df in df.groupby('location'):
        bhk_stats = {}
        for bhk, bhk_df in location_df.groupby('bhk'):
            bhk_stats[bhk] = {
                'mean': np.mean(bhk_df.price_per_sqft),
          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



```
df8.shape

Out[42]: (7206, 7)
```

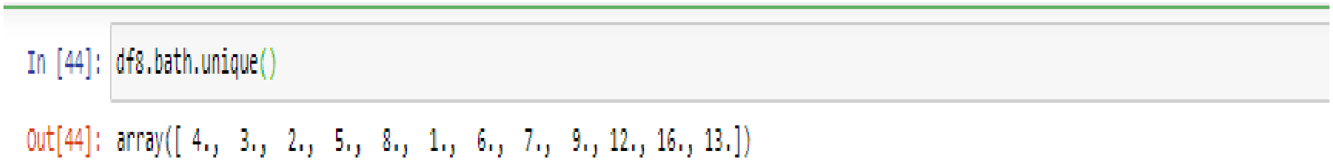
```
plot_scatter_chart(df7,'Hebbal')
```



```
In [43]: plot_scatter_chart(df7,'Hebbal')
```

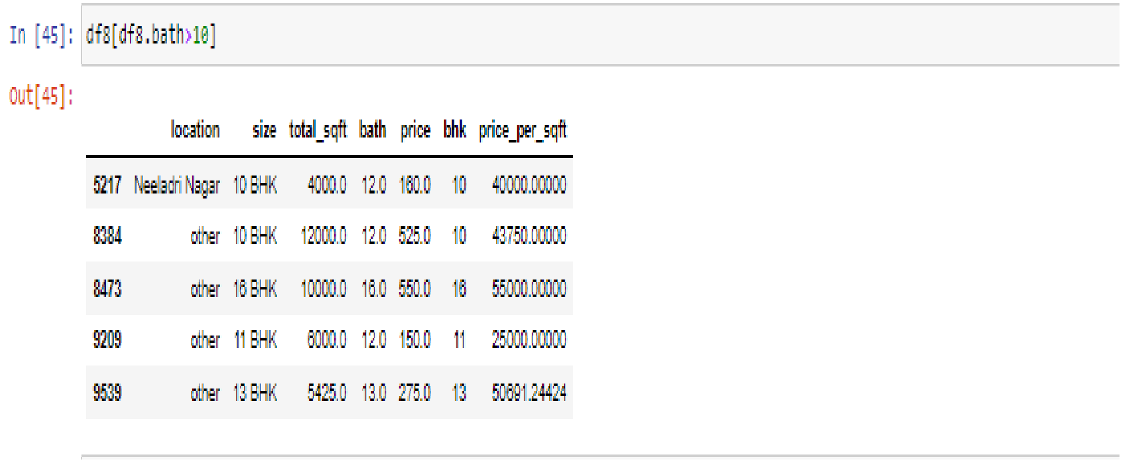
Outlier removal using Barhrooms Feature

df8.bath.unique()



```
In [44]: df8.bath.unique()

Out[44]: array([ 4.,  3.,  2.,  5.,  8.,  1.,  6.,  7.,  9., 12., 16., 13.])
```

df8[df8.bath>10]

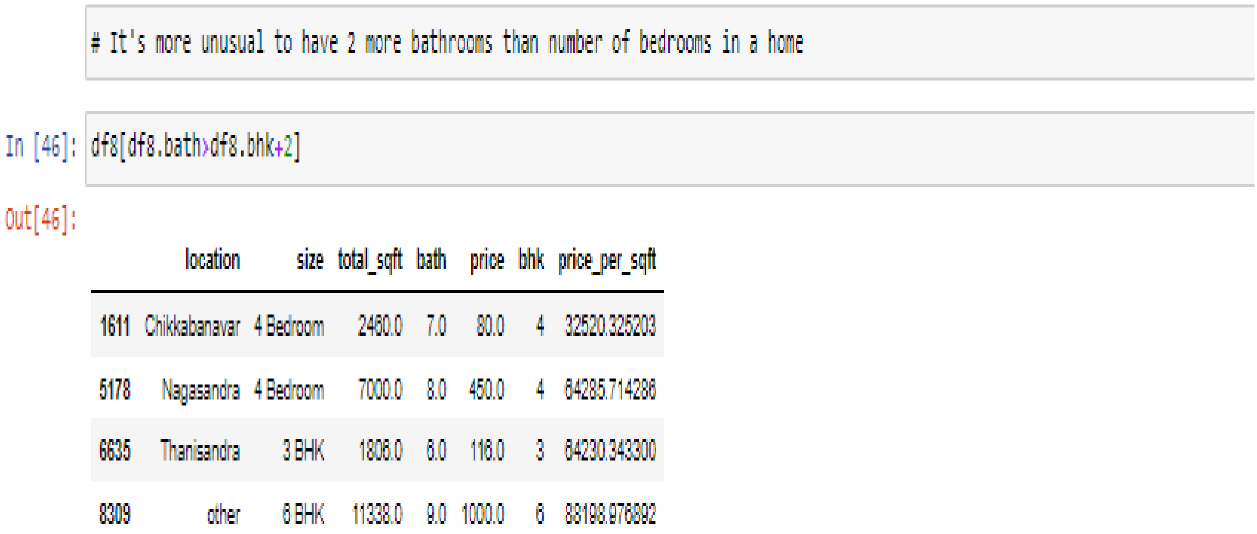


```
In [45]: df8[df8.bath>10]

Out[45]:
```

	location	size	total_sqft	bath	price	bhk	price_per_sqft
5217	Neeladri Nagar	10 BHK	4000.0	12.0	160.0	10	40000.00000
8384	other	10 BHK	12000.0	12.0	525.0	10	43750.00000
8473	other	16 BHK	10000.0	16.0	550.0	16	55000.00000
9209	other	11 BHK	6000.0	12.0	150.0	11	25000.00000
9539	other	13 BHK	5425.0	13.0	275.0	13	50691.24424

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

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

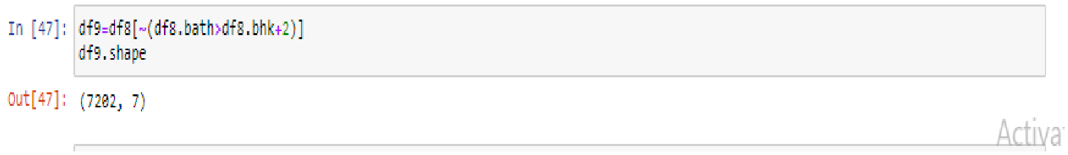


```
# It's more unusual to have 2 more bathrooms than number of bedrooms in a home

In [46]: df8[df8.bath>df8.bhk+2]

Out[46]:
```

	location	size	total_sqft	bath	price	bhk	price_per_sqft
1611	Chikkabanavar	4 Bedroom	2460.0	7.0	80.0	4	32520.325203
5178	Nagasandra	4 Bedroom	7000.0	8.0	450.0	4	64285.714286
6635	Thanisandra	3 BHK	1806.0	6.0	116.0	3	64230.343300
8309	other	6 BHK	11338.0	9.0	1000.0	6	88198.976892

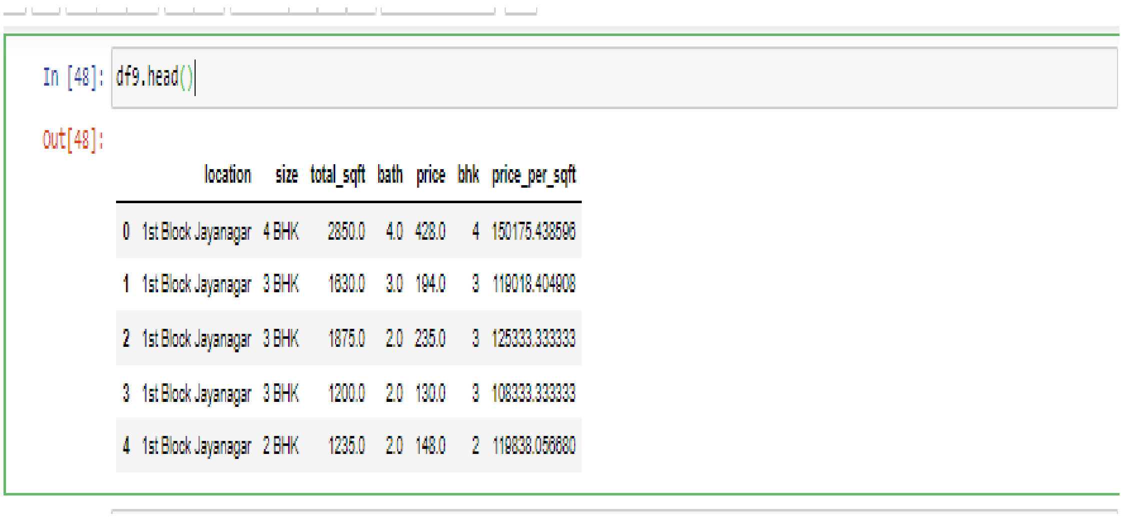
df9=df8[~(df8.bath>df8.bhk+2)]

df9.shape



```
In [47]: df9=df8[~(df8.bath>df8.bhk+2)]
         df9.shape

Out[47]: (7282, 7)
```

df9.head()



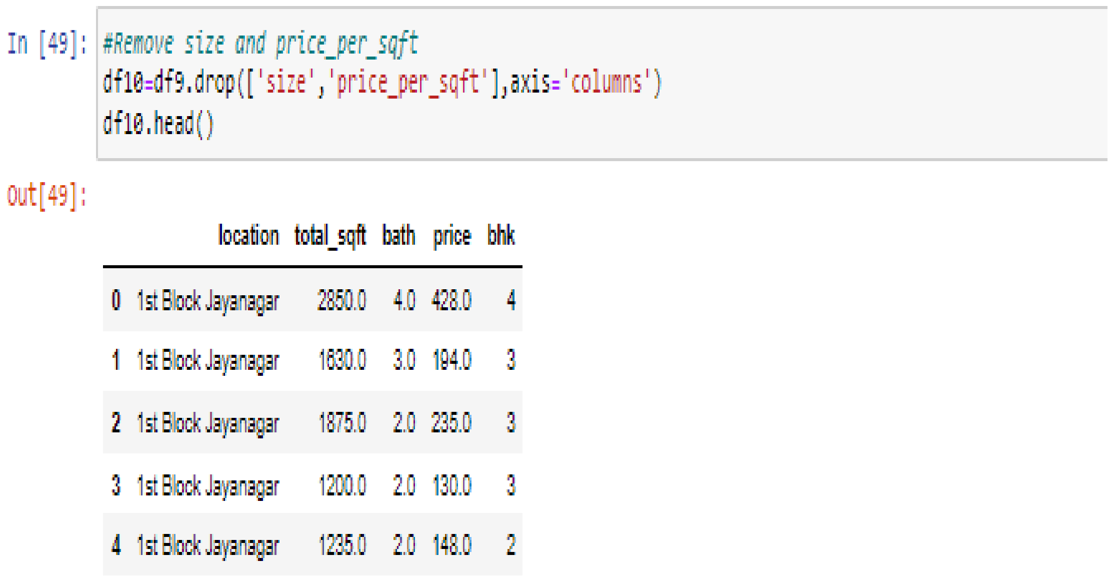
```
In [48]: df9.head()

Out[48]:
```

	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	1st Block Jayanagar	4 BHK	2850.0	4.0	428.0	4	150175.438596
1	1st Block Jayanagar	3 BHK	1630.0	3.0	194.0	3	119018.404908
2	1st Block Jayanagar	3 BHK	1875.0	2.0	235.0	3	125333.333333
3	1st Block Jayanagar	3 BHK	1200.0	2.0	130.0	3	108333.333333
4	1st Block Jayanagar	2 BHK	1235.0	2.0	148.0	2	119838.056680

#Remove size and price_per_sqft

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

df10.head()



```
In [49]: #Remove size and price_per_sqft
         df10=df9.drop(['size','price_per_sqft'],axis='columns')
         df10.head()
```

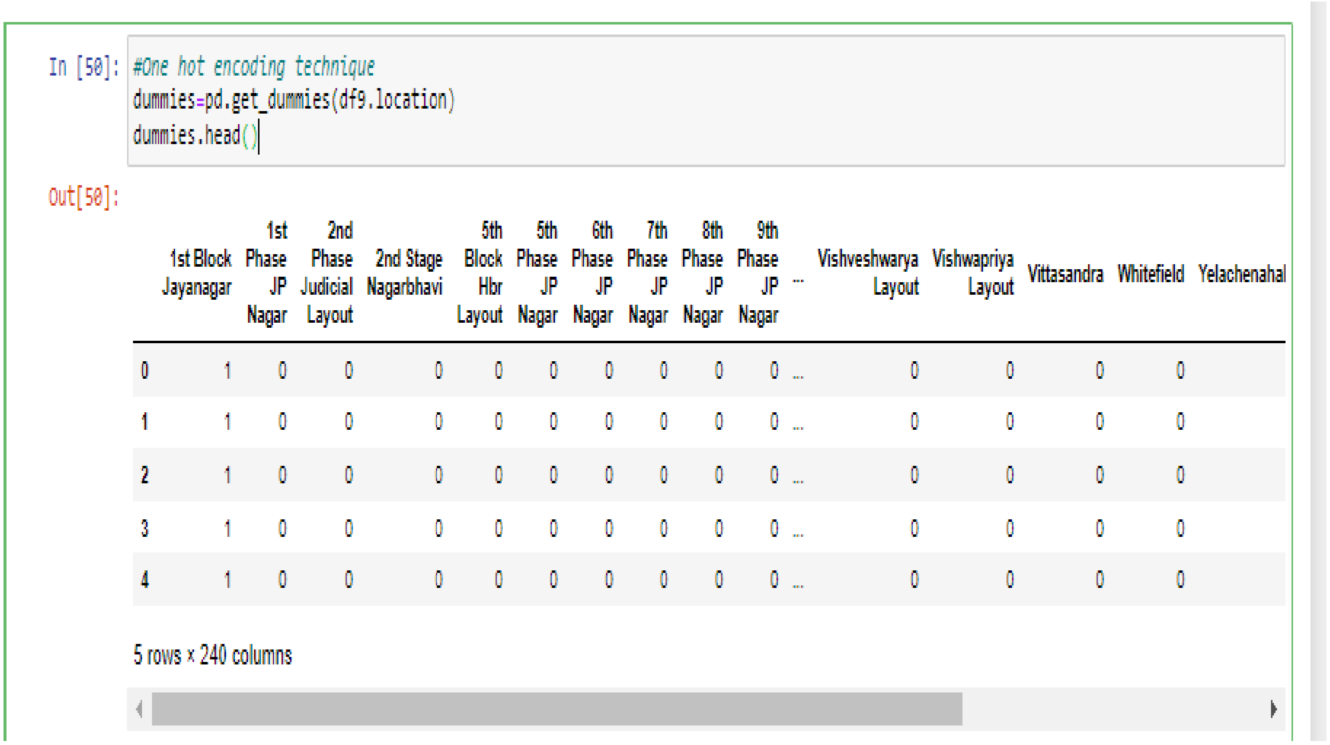
Out[49]:

	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

#One hot encoding technique

dummies=pd.get_dummies(df9.location)

dummies.head()



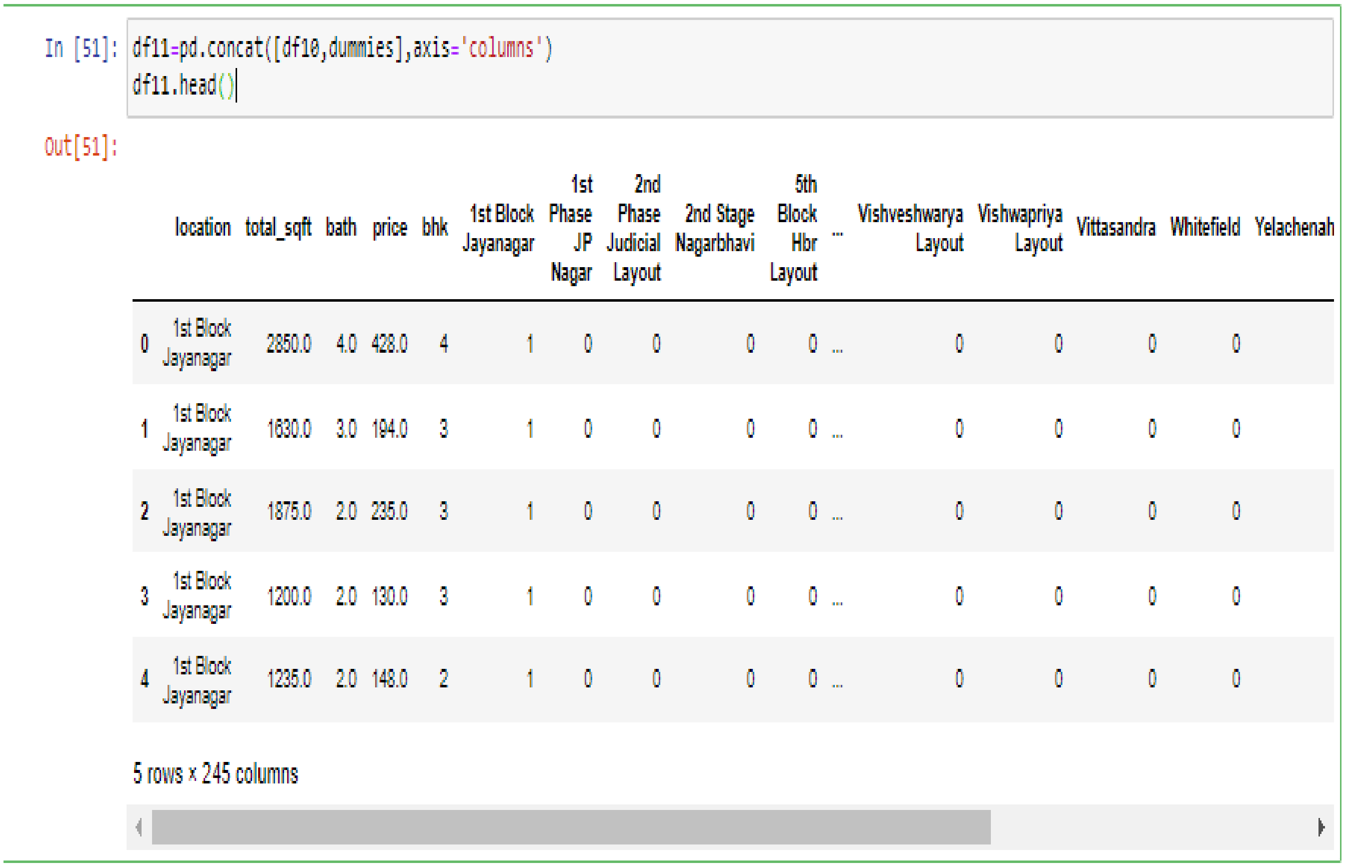
```
In [50]: #One hot encoding technique
         dummies=pd.get_dummies(df9.location)
         dummies.head()
```

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	...	Vishveshwarya Layout	Vishwapriya Layout	Vittasandra	Whitefield	Yelachenahal
0	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
1	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
2	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
3	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
4	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	

5 rows × 240 columns

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



```
In [51]: df11=pd.concat([df10,dummies],axis='columns')
         df11.head()
```

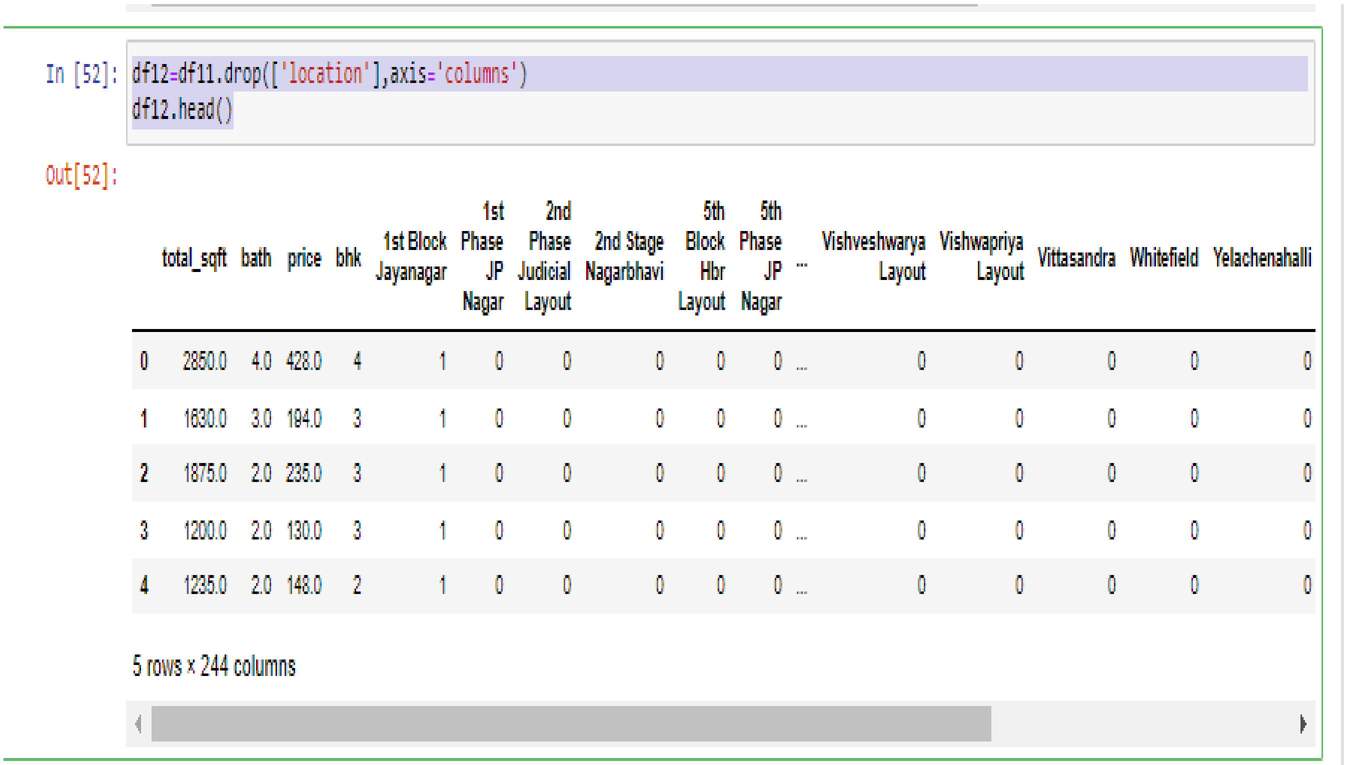
Out[51]:

	location	total_sqft	bath	price	bhk	1st Block Jayanagar	1st Phase JP Nagar	2nd Phase Judicial Layout	2nd Stage Nagarbhavi	5th Block Hbr Layout	...	Vishveshwarya Layout	Vishwapriya Layout	Vittasandra	Whitefield	Yelachenah
0	1st Block Jayanagar	2850.0	4.0	428.0	4	1	0	0	0	0	...	0	0	0	0	
1	1st Block Jayanagar	1630.0	3.0	194.0	3	1	0	0	0	0	...	0	0	0	0	
2	1st Block Jayanagar	1875.0	2.0	235.0	3	1	0	0	0	0	...	0	0	0	0	
3	1st Block Jayanagar	1200.0	2.0	130.0	3	1	0	0	0	0	...	0	0	0	0	
4	1st Block Jayanagar	1235.0	2.0	148.0	2	1	0	0	0	0	...	0	0	0	0	

5 rows × 245 columns

df12=df11.drop(['location'],axis='columns')

df12.head()



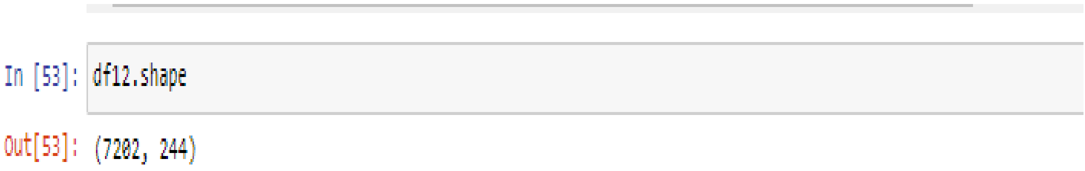
```
In [52]: df12=df11.drop(['location'],axis='columns')
         df12.head()
```

Out[52]:

	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	...	Vishveshwarya Layout	Vishwapriya Layout	Vittasandra	Whitefield	Yelachenahalli
0	2850.0	4.0	428.0	4	1	0	0	0	0	0	...	0	0	0	0	0
1	1630.0	3.0	194.0	3	1	0	0	0	0	0	...	0	0	0	0	0
2	1875.0	2.0	235.0	3	1	0	0	0	0	0	...	0	0	0	0	0
3	1200.0	2.0	130.0	3	1	0	0	0	0	0	...	0	0	0	0	0
4	1235.0	2.0	148.0	2	1	0	0	0	0	0	...	0	0	0	0	0

5 rows × 244 columns

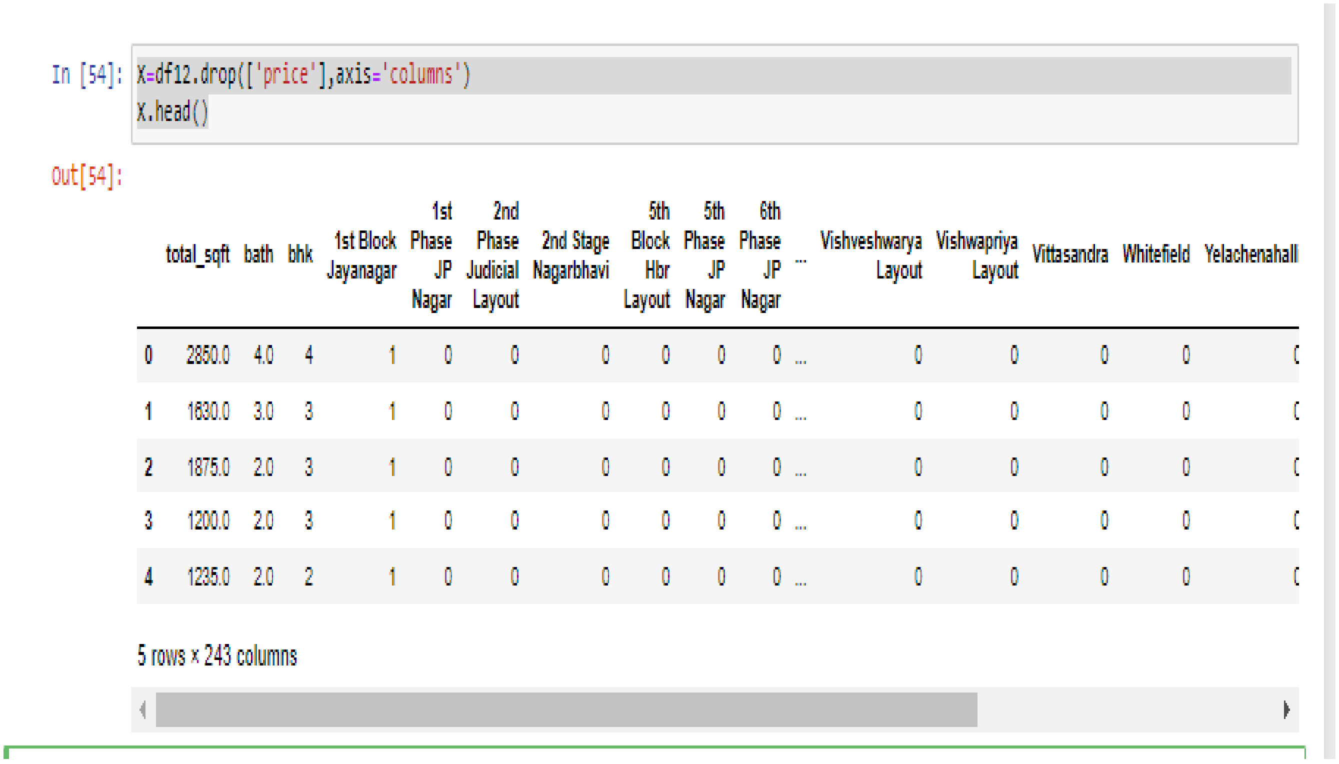
df12.shape



```
In [53]: df12.shape

Out[53]: (7202, 244)
```

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

X.head()



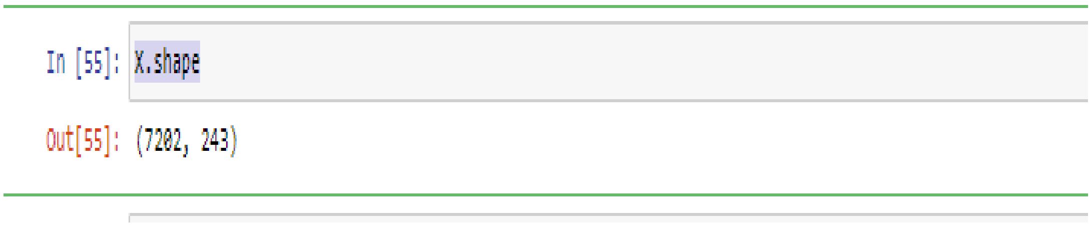
```
In [54]: X=df12.drop(['price'],axis='columns')
         X.head()
```

Out[54]:

	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	...	Vishveshwarya Layout	Vishwapriya Layout	Vittasandra	Whitefield	Yelachenahall
0	2850.0	4.0	4	1	0	0	0	0	0	0	...	0	0	0	0	0
1	1630.0	3.0	3	1	0	0	0	0	0	0	...	0	0	0	0	0
2	1875.0	2.0	3	1	0	0	0	0	0	0	...	0	0	0	0	0
3	1200.0	2.0	3	1	0	0	0	0	0	0	...	0	0	0	0	0
4	1235.0	2.0	2	1	0	0	0	0	0	0	...	0	0	0	0	0

5 rows × 243 columns

X.shape

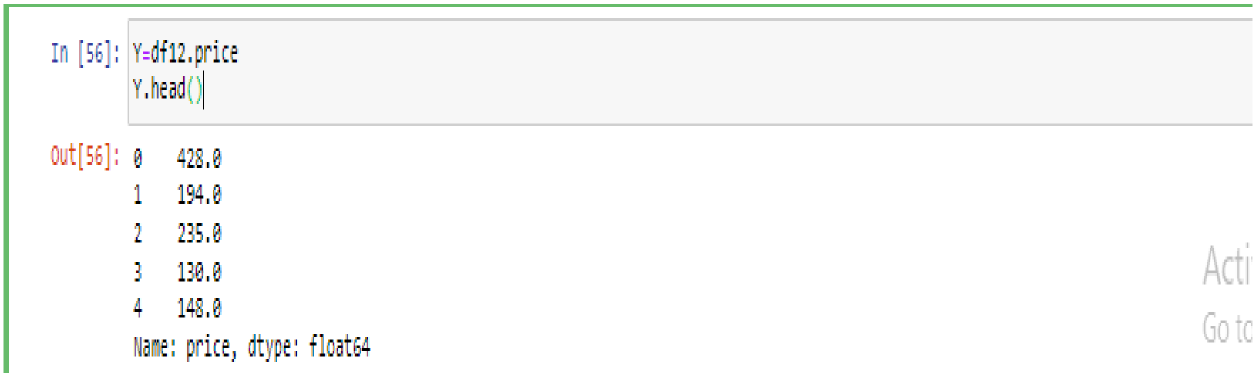


```
In [55]: X.shape

Out[55]: (7202, 243)
```

X.shape

Y=df12.price

Y.head()



```
In [56]: Y=df12.price
         Y.head()

Out[56]: 0    428.0
         1    194.0
         2    235.0
         3    130.0
         4    148.0
         Name: price, dtype: float64
```

from sklearn.model_selection import train_test_split

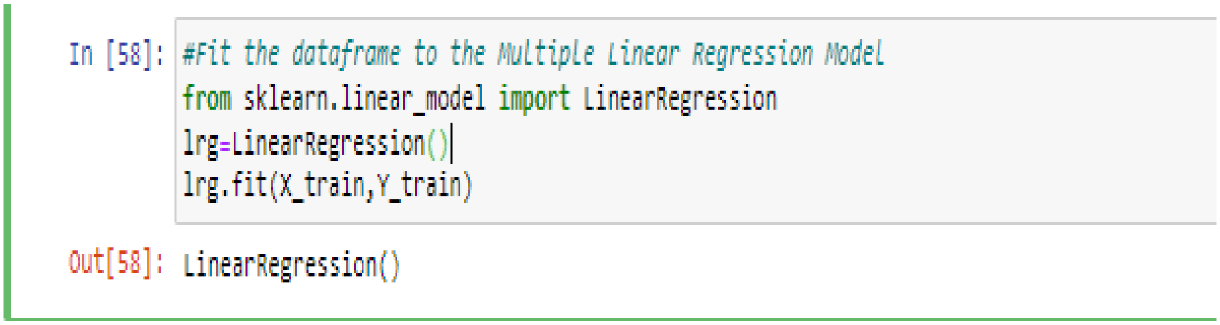
X_train, X_test, Y_train, Y_test= train_test_split(X,Y,test_size=0.2,random_state=10)

#Fit the dataframe to the Multiple Linear Regression Model

from sklearn.linear_model import LinearRegression

lrg=LinearRegression()

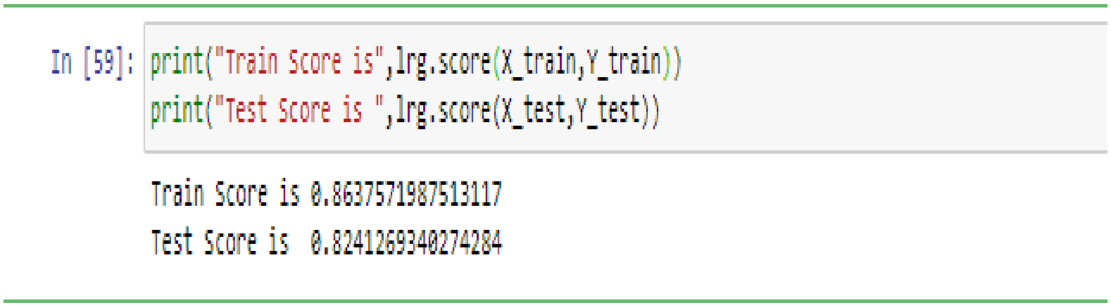
lrg.fit(X_train,Y_train)



```
In [58]: #Fit the dataframe to the Multiple Linear Regression Model
         from sklearn.linear_model import LinearRegression
         lrg=LinearRegression()
         lrg.fit(X_train,Y_train)

Out[58]: LinearRegression()
```

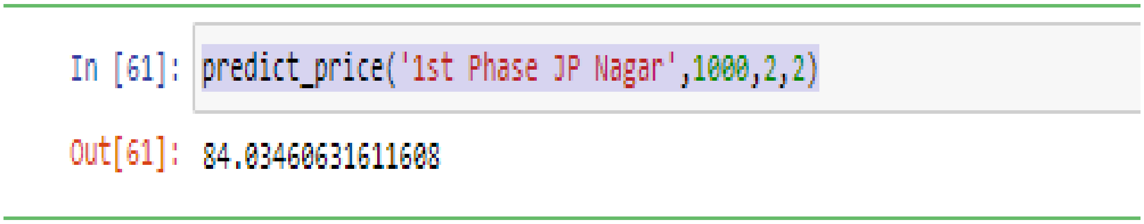
print("Train Score is",lrg.score(X_train,Y_train))

print("Test Score is ",lrg.score(X_test,Y_test))



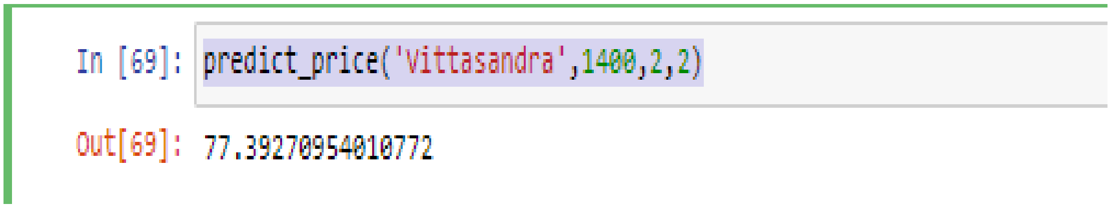
```
In [59]: print("Train Score is",lrg.score(X_train,Y_train))
         print("Test Score is ",lrg.score(X_test,Y_test))

         Train Score is 0.863757198751317
         Test Score is  0.8241269340274284
```

```
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 lrg.predict([x])[0]
predict_price('1st Phase JP Nagar',1000,2,2)
```



```
In [61]: predict_price('1st Phase JP Nagar',1000,2,2)

Out[61]: 84.03460631611608
```

```
predict_price('Vittasandra',1400,2,2)
```

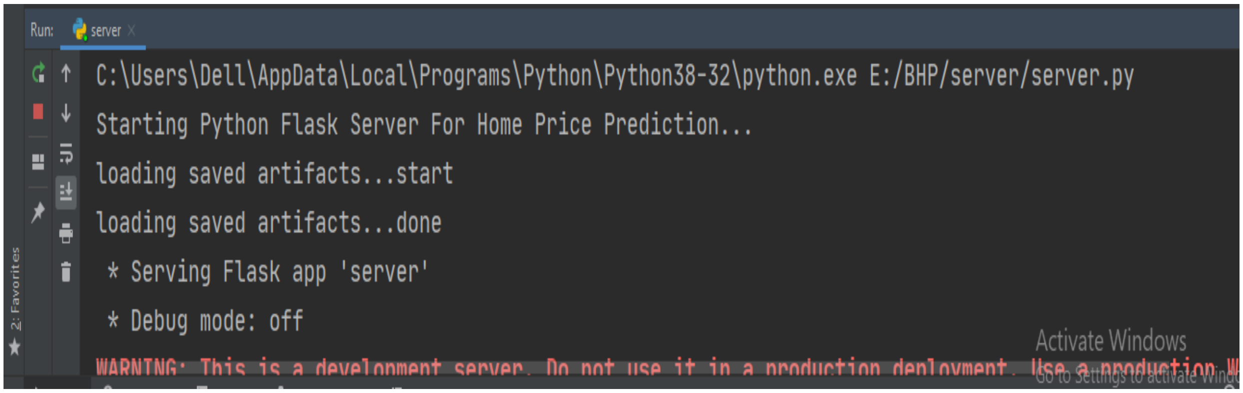


```
In [69]: predict_price('Vittasandra',1400,2,2)

Out[69]: 77.39270954010772
```

Server Side Programming

Server.py file:-

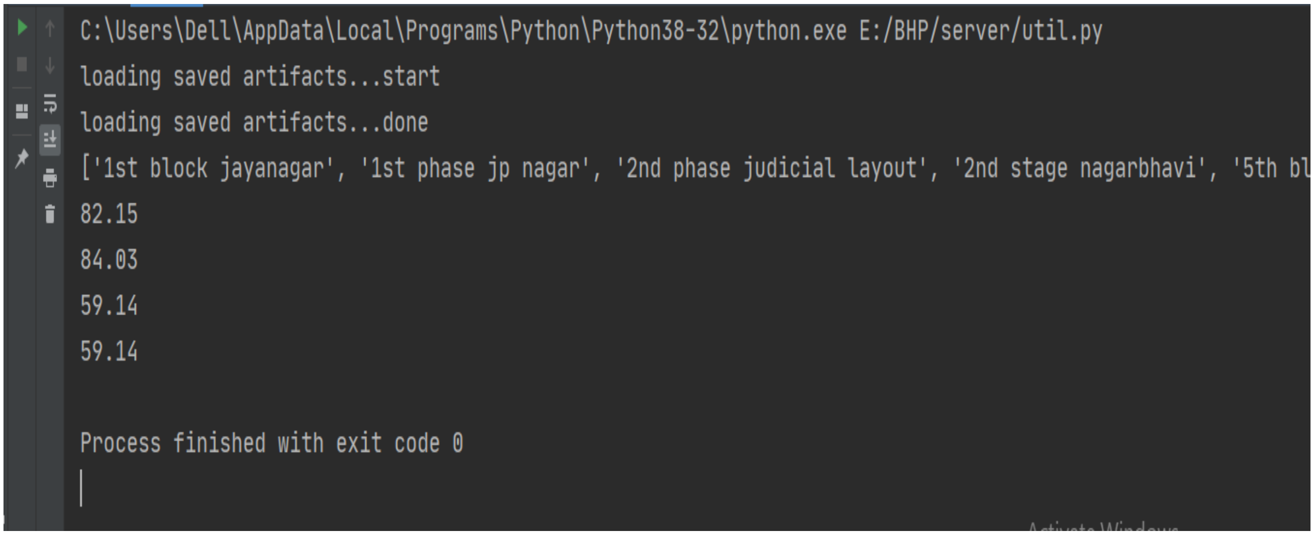
```
from flask import Flask, request, jsonify
import util

app = Flask(__name__)

@app.route('/get_location_names', methods=['GET'])
def get_location_names():
    response = jsonify({
        'locations': util.get_location_names()
    })
    response.headers.add('Access-Control-Allow-Origin', '*')

    return response

@app.route('/predict_home_price', methods=['GET', 'POST'])
def predict_home_price():
    total_sqft = float(request.form['total_sqft'])
    location = request.form['location']
    bhk = int(request.form['bhk'])
    bath = int(request.form['bath'])

    response = jsonify({
        'estimated_price':
util.get_estimated_price(location,total_sqft,bhk,bath)
    })
    response.headers.add('Access-Control-Allow-Origin', '*')

    return response

if __name__ == "__main__":
    print("Starting Python Flask Server For Home Price Prediction...")
    util.load_saved_artifacts()
    app.run()
```



```
C:\Users\Dell\AppData\Local\Programs\Python\Python38-32\python.exe E:/BHP/server/server.py
Starting Python Flask Server For Home Price Prediction...
loading saved artifacts...start
loading saved artifacts...done
 * Serving Flask app 'server'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production W
```

Util.py file:-

```
import pickle
import json
import numpy as np

__locations = None
__data_columns = None
__model = None

def get_estimated_price(location,sqft,bhk,bath):
    try:
        loc_index = __data_columns.index(location.lower())
    except:
        loc_index = -1

    x = np.zeros(len(__data_columns))
    x[0] = sqft
    x[1] = bath
    x[2] = bhk
    if loc_index>=0:
        x[loc_index] = 1

    return round(__model.predict([x])[0],2)


def load_saved_artifacts():
    print("loading saved artifacts...start")
    global  __data_columns
    global __locations

    with open("E:\BHP\server\columns.json", "r") as f:
        __data_columns = json.load(f)['data_columns']
        __locations = __data_columns[3:]  # first 3 columns are sqft,
bath, bhk

    global __model
    if __model is None:
        with open('E:\BHP\server\House_Price_Prediction.pickle', 'rb') as
f:
            __model = pickle.load(f)
    print("loading saved artifacts...done")

def get_location_names():
    return __locations

def get_data_columns():
    return __data_columns

if __name__ == '__main__':
    load_saved_artifacts()
    print(get_location_names())
    print(get_estimated_price('1st Phase JP Nagar',1000, 3, 3))
    print(get_estimated_price('1st Phase JP Nagar', 1000, 2, 2))
```



```
C:\Users\Dell\AppData\Local\Programs\Python\Python38-32\python.exe E:/BHP/server/util.py
loading saved artifacts...start
loading saved artifacts...done
['1st block jayanagar', '1st phase jp nagar', '2nd phase judicial layout', '2nd stage nagarbhavi', '5th bl
82.15
84.03
59.14
59.14

Process finished with exit code 0
```

Client Side Programming

App.html file:-

```
<!DOCTYPE html>
<html>
<head>
    <title>Banglore Home Price Prediction</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"><
/script>
    <script src="app.js"></script>
        <link rel="stylesheet" href="app.css">
</head>
<body>
<div class="img"></div>
<form class="form">
        <h2>Area (Square Feet)</h2>
    <input class="area"  type="text" id="uiSqft" class="floatLabel"
name="Squareft" value="1000">
        <h2>BHK</h2>
        <div class="switch-field">
        <input type="radio" id="radio-bhk-1" name="uiBHK"
value="1"/>
        <label for="radio-bhk-1">1</label>
        <input type="radio" id="radio-bhk-2" name="uiBHK"
value="2" checked/>
        <label for="radio-bhk-2">2</label>
        <input type="radio" id="radio-bhk-3" name="uiBHK"
value="3"/>
        <label for="radio-bhk-3">3</label>
        <input type="radio" id="radio-bhk-4" name="uiBHK"
value="4"/>
        <label for="radio-bhk-4">4</label>
        <input type="radio" id="radio-bhk-5" name="uiBHK"
value="5"/>
        <label for="radio-bhk-5">5</label>
    </div>
    </form>
<form class="form">
    <h2>Bath</h2>
    <div class="switch-field">
        <input type="radio" id="radio-bath-1" name="uiBathrooms"
value="1"/>
        <label for="radio-bath-1">1</label>
        <input type="radio" id="radio-bath-2" name="uiBathrooms"
value="2" checked/>
        <label for="radio-bath-2">2</label>
        <input type="radio" id="radio-bath-3" name="uiBathrooms"
value="3"/>
        <label for="radio-bath-3">3</label>
        <input type="radio" id="radio-bath-4" name="uiBathrooms"
value="4"/>
        <label for="radio-bath-4">4</label>
```

```
                <input type="radio" id="radio-bath-5" name="uiBathrooms"
value="5"/>
                <label for="radio-bath-5">5</label>
        </div>
                <h2>Location</h2>
        <div>
  <select class="location" name="" id="uiLocations">
    <option value="" disabled="disabled" selected="selected">Choose
a Location</option>
                <option>Electronic City</option>
       <option>Rajaji Nagar</option>
  </select>
</div>
        <button class="submit" onclick="onClickedEstimatePrice()"
type="button">Estimate Price</button>
        <div id="uiEstimatedPrice" class="result">  <h2></h2> </div>
</body>
</html>
```

App.css file:-

```
.switch-field {
        display: flex;
        margin-bottom: 36px;
        overflow: hidden;
}

.switch-field input {
        position: absolute !important;
        clip: rect(0, 0, 0, 0);
        height: 1px;
        width: 1px;
        border: 0;
        overflow: hidden;
}

.switch-field label {
        background-color: #e4e4e4;
        color: rgba(0, 0, 0, 0.6);
        font-size: 14px;
        line-height: 1;
        text-align: center;
        padding: 8px 16px;
        margin-right: -1px;
        border: 1px solid rgba(0, 0, 0, 0.2);
        box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.3), 0 1px rgba(255,
255, 255, 0.1);
        transition: all 0.1s ease-in-out;
}

.switch-field label:hover {
        cursor: pointer;
}

.switch-field input:checked + label {
        background-color: #a5dc86;
        box-shadow: none;
}

.switch-field label:first-of-type {
        border-radius: 4px 0 0 4px;
}

.switch-field label:last-of-type {
        border-radius: 0 4px 4px 0;
```

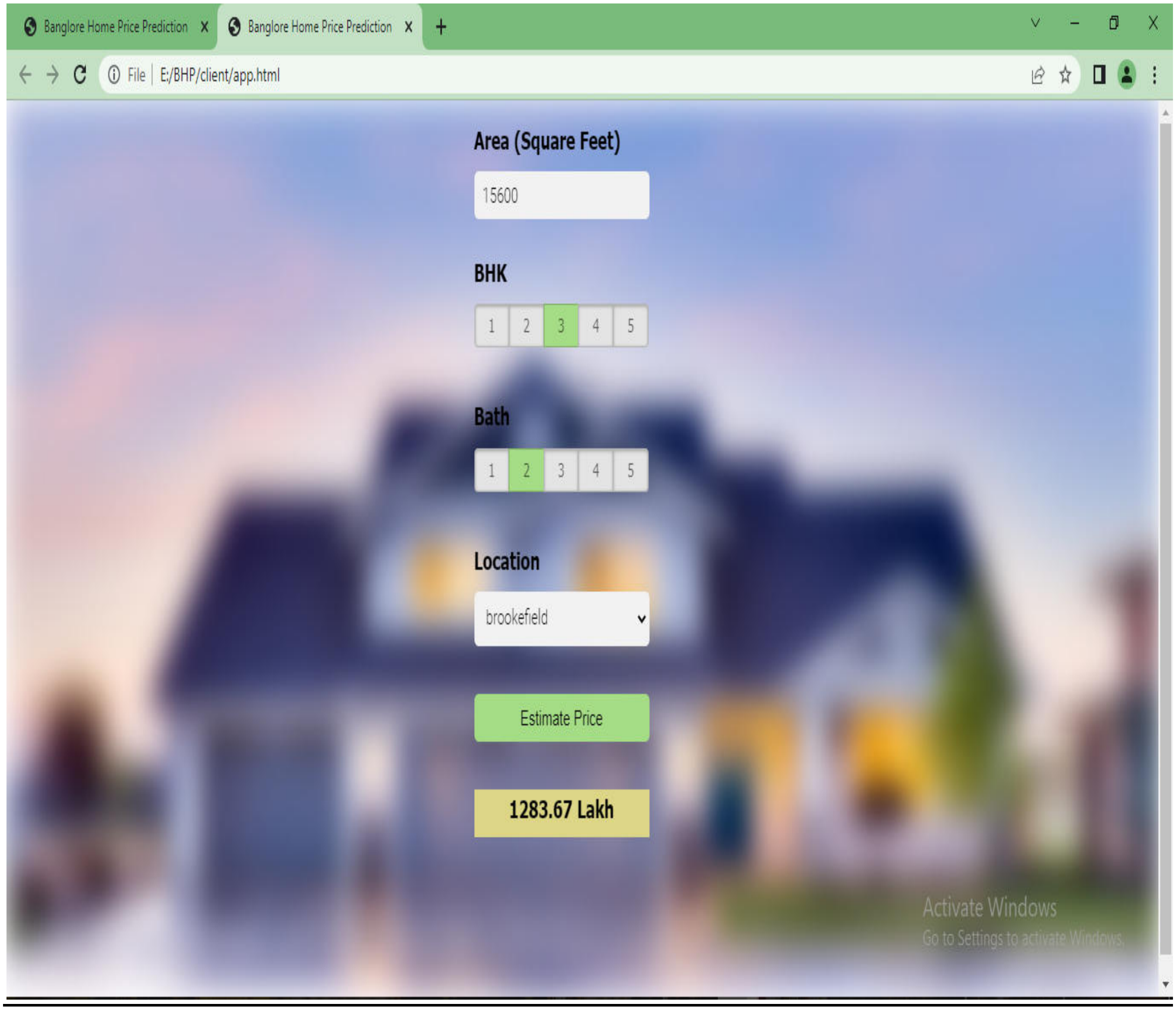
```
}

.form {
        max-width: 270px;
        font-family: "Lucida Grande", Tahoma, Verdana, sans-serif;
        font-weight: normal;
        line-height: 1.625;
        margin: 8px auto;
        padding-left: 16px;
        z-index: 2;
}

h2 {
        font-size: 18px;
        margin-bottom: 8px;
}
.area{
  font-family: "Roboto", sans-serif;
  outline: 0;
  background: #f2f2f2;
  width: 76%;
  border: 0;
  margin: 0 0 10px;
  padding: 10px;
  box-sizing: border-box;
  font-size: 15px;
  height: 35px;
  border-radius: 5px;
}

.location{
  font-family: "Roboto", sans-serif;
  outline: 0;
  background: #f2f2f2;
  width: 76%;
  border: 0;
  margin: 0 0 10px;
  padding: 10px;
  box-sizing: border-box;
  font-size: 15px;
  height: 40px;
  border-radius: 5px;
}

.submit{
  background: #a5dc86;
  width: 76%;
  border: 0;
  margin: 25px 0 10px;
  box-sizing: border-box;
  font-size: 15px;
         height: 35px;
         text-align: center;
         border-radius: 5px;
}

.result{
                  background: #dcd686;
                  width: 76%;
                  border: 0;
                  margin: 25px 0 10px;
                  box-sizing: border-box;
                  font-size: 15px;
                  height: 35px;
```

```
                 text-align: center;
}

.img {
  background: url('bhp.jpg');
        background-repeat: no-repeat;
  background-size: auto;
  background-size:100% 100%;
  -webkit-filter: blur(5px);
  -moz-filter: blur(5px);
  -o-filter: blur(5px);
  -ms-filter: blur(5px);
  filter: blur(15px);
  position: fixed;
  width: 100%;
  height: 100%;
  top: 0;
  left: 0;
  z-index: -1;
}

body, html {
  height: 100%;
}
```

App.js file

```
function getBathValue() {
  var uiBathrooms = document.getElementsByName("uiBathrooms");
  for(var i in uiBathrooms) {
    if(uiBathrooms[i].checked) {
        return parseInt(i)+1;
    }
  }
  return -1; // Invalid Value
}

function getBHKValue() {
  var uiBHK = document.getElementsByName("uiBHK");
  for(var i in uiBHK) {
    if(uiBHK[i].checked) {
        return parseInt(i)+1;
    }
  }
  return -1; // Invalid Value
}

function onClickedEstimatePrice() {
  console.log("Estimate price button clicked");
  var sqft = document.getElementById("uiSqft");
  var bhk = getBHKValue();
  var bathrooms = getBathValue();
  var location = document.getElementById("uiLocations");
  var estPrice = document.getElementById("uiEstimatedPrice");

   var url = "http://127.0.0.1:5000/predict_home_price";

  $.post(url, {
      total_sqft: parseFloat(sqft.value),
      bhk: bhk,
      bath: bathrooms,
      location: location.value
  },function(data, status) {
      console.log(data.estimated_price);
      estPrice.innerHTML = "<h2>" + data.estimated_price.toString() + "
Lakh</h2>";
```

```
      console.log(status);
  });
}

function onPageLoad() {
  console.log( "document loaded" );
   var url = "http://127.0.0.1:5000//get_location_names";
  $.get(url,function(data, status) {
      console.log("got response for get_location_names request");
      if(data) {
          var locations = data.locations;
          var uiLocations = document.getElementById("uiLocations");
          $('#uiLocations').empty();
          for(var i in locations) {
              var opt = new Option(locations[i]);
              $('#uiLocations').append(opt);
          }
      }
  });
}

window.onload = onPageLoad;
```


Banglore Home Price Prediction
Banglore Home Price Prediction
File | E:/BHP/client/app.html
Estimate Price
Activate Windows
Go to Settings to activate Windows.

Area (Square Feet)

15600

BHK

1 2 [✓ 3] 4 5

Bath

1 [✓ 2] 3 4 5

Location

brookefield

1283.67 Lakh

Feasibility Study

Feasibility study can help you determine whether or not you should proceed with your project. It is essential to evaluate cost and benefit. It is essential to evaluate cost and benefit of the proposed system. Five types of feasibility study are taken into consideration.

1) **Technical Feasibility**
   It includes finding out technologies for the project, both hardware and software. User must have PC their inputs and for house price output. Minimum hardware requirements: Dual-core 64-bit processor, 8 GB of memory, Up to 24 GB of internal. These are very cheap now a days and everyone generally possess them.

2) **Operational Feasibility**
   It is the ease and simplicity of operation of proposed system. System does not require any special skill set for users to operate it. In fact, it is designed to be used by almost everyone. Kids who still don't know to write can read out problems for system and get answers.

3) **Economical Feasibility**.
   Here, we find the total cost and benefit of the proposed system over current system. Once the hardware and software package needs get consummated, there is no want for the user of our system to pay for any further overhead.
   User also would have to pay for PC. Our application can scale back the time that's wasted in manual processes. The storage and handling issues of the registers are resolved.

Survey

1) Lu et.al proposed a hybrid prediction model; the study looked at the impact of land financing and household spending on real estate prices in 33 major Chinese cities. The implementation of Panel data validation of fixedeffects model regression findings our proposition After establishing control of the city's local people, the rate of growth, per capita GDP, and the number of students enrolled in regular classrooms are all things to think about. Institutions of higher education, gender ratio, and consumer pricing Higher education institutions, gender ratios, and consumer pricing urban population density, land finance, and urban development are all indices to look at. People's consumption levels will have a positive impact on real estate. It can formulate policies for the government, provide constructive opinions when planning to sell land, and prevent the local government from relying excessively on land revenue while attempting to expand by confirming that land transfer has a significant impact on the real estate price and the promotion mode of the factor and the house price. Land finance encourages economic growth, which leads to skyrocketing real estate values. This article indicates that citizens' consumption levels are a significant element influencing real estate price fluctuations, allowing the government to employ various information channels and data to forecast the real estate market's prospects and design suitable policies.

2) Lim et.al purposed useful models for predicting property prices. It also provides details on the Melbourne housing market. To begin, the raw data is cleaned and transformed into a readable data-set. The data is then reduced and transformed using Stepwise and PCA techniques. Following that, a variety of tactics are implemented and evaluated in order to arrive at the optimal solution. According to the evaluation phase, combining Step-wise and SVM models is a competitive strategy. As a result, future deployments may include it. This research can also be extended to transitional datasets from other sections of the Australian property market. The studies were run on a Windows system using the R programming language. Both the train and assessment datasets Mean Squared Error (MSE) are shown. The baseline for model comparison will be linear regression, as discussed previously. Each model's evaluation ratio is equal to its evaluation MSE divided by Linear regression's evaluation MSE. The higher the accuracy of the model's forecast, the lower the evaluation ratio[5].

Future Scope and Challenges

- One of the challenges while proposing an algorithm for House Price Prediction is to determine the attributes affecting the price.

- As different factors affect RS differently, therefore, how to assign appropriate weights to the attributes is a major task when designing an algorithm.

- In Future, the models can be upgraded with some better techniques in terms of getting higher and better accuracy.

- It is, therefore, challenging to cope with the problem of changing users requirements.

- Inferences obtained w.r.t data from a large urban area like Bengaluru may not directly represent the exact same correlation of the same features when data is gathered from suburban area close to Bengaluru.

- It’s found that images of house interiors have major impacts on pricing. However, retrieving interior design of the houses is not always possible.

References/Glossary

Content

https://www.ijraset.com/research-paper/house-price-prediction-using-ml

https://www.slideshare.net/AdityaKumar1505/house-price-prediction-235540647

https://www.researchgate.net/publication/349477129_House_Price_Prediction

Dataset->

https://www.kaggle.com/datasets/anmolkumar/house-price-prediction-challenge

Code->

https://www.kaggle.com/code/ameythakur20/bangalore-house-price-prediction-model