MINI PROJECT
(2020-21)

House Price Prediction using Machine Learning tools

MID-TERM REPORT

Institute of Engineering & Technology

Submitted by-
Shivam Tripathi
(181500698)
Shubham Singh
(181500698)

Supervised By: -
Mr. Vinay Agrawal
Assistant Professor
Department of Computer Engineering & Applications

Contents

Introduction

1.1 General Introduction to the topic

House price forecasting is an important topic of real estate. The literature attempts to derive useful knowledge from historical data of property markets. Machine learning techniques are applied to analyze historical property transactions in Banglore to discover useful models for house buyers and sellers. Revealed is the high discrepancy between house prices in the most expensive and most affordable suburbs in the city of Banglore. Moreover, experiments demonstrate that the Multiple Linear Regression that is based on mean squared error measurement is a competitive approach.

Python Programming:

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.
Python is a programming language that lets you work quickly and integrate systems more efficiently.

Numpy and Pandas:

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. This tutorial explains the basics of NumPy such as its architecture and environment. It also discusses the various array functions, types of indexing, etc.
Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib:

Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays. Matplotlib is written in Python and makes use of NumPy, the numerical mathematics extension of Python.

Sklearn:

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

1.2 Hardware and Software Requirements:

Hardware Requirements:

- 4/8 GB RAM
- i5 processor

Software Requirements:

- Jupyter Notebook
- Windows 10
- Visual Studio Code

Problem Statement:

Housing prices are an important reflection of the economy, and housing
price ranges are of great interest for both buyers and sellers. In this project. house prices will be predicted given explanatory variables that cover many aspects of residential houses. As continuous house prices, they will be predicted with various regression techniques as individual price ranges,they will be predicted with classification methods.
The goal of this project is to create a regression model and a classification model that are able to accurately estimate the price of the house.

Implementation

This data science project walks through the step by step process of how to build a real estate price prediction website. We will first build a model using sklearn and linear regression using Banglore home prices dataset from kaggle.com.
Second step would be to write a python flask server that uses the saved model to serve HTTP requests.
Third component is the website built in HTML, CSS and JavaScript that allows users to enter home square ft area, bedrooms etc and it will call a python flask server to retrieve the predicted price.
During model building we will cover almost all data science concepts such as data load and cleaning, outlier detection and removal, feature engineering, dimensionality reduction, grid searchcv for hyper parameter tuning, k fold cross validation etc.

Progress

In [1]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline
import matplotlib
matplotlib.rcParams["figure.figsize"] = (20,10)
```

In [2]:
```
housing1 = pd.read_csv("Real_Estate_Price_Prediction_Dataset.csv")
```

In [3]:
```
housing1.head()
```

Out[3]:

	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

In [4]:
```
housing1.shape
```

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

In [5]:
```
housing1.info()
```

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

In [6]:
```
housing1.groupby('area_type')['area_type'].agg('count')
```

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

In [7]:
```
housing2 = housing1.drop(['area_type','society','balcony','availability'],axis='columns')
```

In [8]:
```
housing2.head()
```

Out[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

In [9]:
```
housing2.isnull().sum()
```

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

In [10]:
```
housing3 = housing2.dropna()
```

In [11]:
```
housing3.isnull().sum()
```

Out[11]:
```
location      0
size          0
total_sqft    0
bath          0
price         0
dtype: int64
```

In [12]:
```
housing3.shape
```

Out[12]:
```
(13246, 5)
```

In [13]:
```
housing3['size'].unique()
```

Out[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)
```

Feature Engineering

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

In [14]:
```
housing3['bhk']=housing3['size'].apply(lambda x: int(x.split(' ')[0]))
```

```
<ipython-input-14-4955814f9b3c>: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
  housing3['bhk']=housing3['size'].apply(lambda x: int(x.split(' ')[0]))
```

In [15]:
```
housing3.head()
```

Out[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

In [16]:
```
housing3['bhk'].unique()
```

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

In [17]:
```
housing3[housing3.bhk>20]
```

Out[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

In [18]:
```
housing3.total_sqft.unique()
```

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

Explore total_sqft feature

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

In [20]:
```
housing3[~housing3['total_sqft'].apply(is_float)].head(10)
```

Out[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

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

In [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
```

In [22]:
```
housing4 = housing3.copy()
housing4.total_sqft = housing4.total_sqft.apply(convert_sqft_to_num)
housing4 = housing4[housing4.total_sqft.notnull()]
housing4 = housing4[housing4.total_sqft.notnull()]
housing4.head(3)
```

Out[22]:

	location	size	total_sqft	bath	price	bhk
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.07	2
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3

For below row, it shows total_sqft as 2475 which is an average of the range 2100-2850

```
In [23]: housing4.loc[30]
```

```
Out[23]: location       Yelahanka
         size             4 BHK
         total_sqft        2475
         bath                 4
         price              186
         bhk                  4
         Name: 30, dtype: object
```

```
In [24]: (2100+2850)/2
```

```
Out[24]: 2475.0
```

```
In [25]: housing4.head()
```

Out[25]:

	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

```
In [26]: housing5 = housing4.copy()
         housing5['price_per_sqft'] = housing5['price']*100000/housing5['total_sqft']
         housing5.head()
```

Out[26]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.07	2	3699.810606
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4	4615.384615
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3	4305.555556
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.00	3	6245.890861
4	Kothanur	2 BHK	1200.0	2.0	51.00	2	4250.000000

Add new feature called price per square feet

```
In [27]: housing5_stats = housing5['price_per_sqft'].describe()
         housing5_stats
```

```
Out[27]: count    1.320000e+04
         mean     7.920759e+03
         std      1.067272e+05
         min      2.678298e+02
         25%      4.267701e+03
         50%      5.438331e+03
         75%      7.317073e+03
         max      1.200000e+07
         Name: price_per_sqft, dtype: float64
```

```
In [28]: housing5.to_csv("bhp.csv",index=False)
```

```
In [29]: housing5.location = housing5.location.apply(lambda x: x.strip())
         location_stats = housing5['location'].value_counts(ascending=False)
         location_stats
```

```
Out[29]: Whitefield                                 533
         Sarjapur  Road                             392
         Electronic City                           304
         Kanakpura Road                            264
         Thanisandra                               235
                                                  ...
         Lakshminarayanapura, Electronic City Phase 2      1
         basaveshwarnagar                          1
         Goraguntepalya                            1
         BAGUR ROAD                                1
         Michael Palaya                            1
         Name: location, Length: 1287, dtype: int64
```

```
In [30]: location_stats.values.sum()
```

```
Out[30]: 13200
```

```
In [31]: len(location_stats[location_stats>10])
```

```
Out[31]: 240
```

```
In [32]: len(location_stats)
```

```
Out[32]: 1287
```

```
In [33]: len(location_stats[location_stats<=10])
```

```
Out[33]: 1047
```

Dimensionality Reduction

In [34]:
```
location_stats_less_than_10 = location_stats[location_stats<=10]
location_stats_less_than_10
```

Out[34]:
```
Thyagaraja Nagar                          10
Ganga Nagar                               10
Naganathapura                             10
Basapura                                  10
Nagappa Reddy Layout                      10
                                          ..
Lakshminarayanapura, Electronic City Phase 2     1
basaveshwarnagar                          1
Goraguntepalya                            1
BAGUR ROAD                                1
Michael Palaya                            1
Name: location, Length: 1047, dtype: int64
```

In [35]:
```
len(housing5.location.unique())
```

Out[35]: 1287

In [36]:
```
housing5.location = housing5.location.apply(lambda x: 'other' if x in location_stats_less_than_10 else x)
len(housing5.location.unique())
```

Out[36]: 241

In [37]:
```
housing5.head(10)
```

Out[37]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.07	2	3699.810606
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4	4615.384615
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3	4305.555556
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.00	3	6245.890861
4	Kothanur	2 BHK	1200.0	2.0	51.00	2	4250.000000
5	Whitefield	2 BHK	1170.0	2.0	38.00	2	3247.863248
6	Old Airport Road	4 BHK	2732.0	4.0	204.00	4	7467.057101
7	Rajaji Nagar	4 BHK	3300.0	4.0	600.00	4	18181.818182
8	Marathahalli	3 BHK	1310.0	3.0	63.25	3	4828.244275
9	other	6 Bedroom	1020.0	6.0	370.00	6	36274.509804

Outlier Removal Using Business Logic

In [38]:
```
housing5[housing5.total_sqft/housing5.bhk<300].head()
```

Out[38]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
9	other	6 Bedroom	1020.0	6.0	370.0	6	36274.509804
45	HSR Layout	8 Bedroom	600.0	9.0	200.0	8	33333.333333
58	Murugeshpalya	6 Bedroom	1407.0	4.0	150.0	6	10660.980810
68	Devarachikkanahalli	8 Bedroom	1350.0	7.0	85.0	8	6296.296296
70	other	3 Bedroom	500.0	3.0	100.0	3	20000.000000

In [39]:
```
housing5.shape
```

Out[39]: (13200, 7)

In [40]:
```
housing6 = housing5[~(housing5.total_sqft/housing5.bhk<300)]
housing6.shape
```

Out[40]: (12456, 7)

Outlier Removal Using Standard Deviation and Mean

In [41]:
```
housing6.price_per_sqft.describe()
```

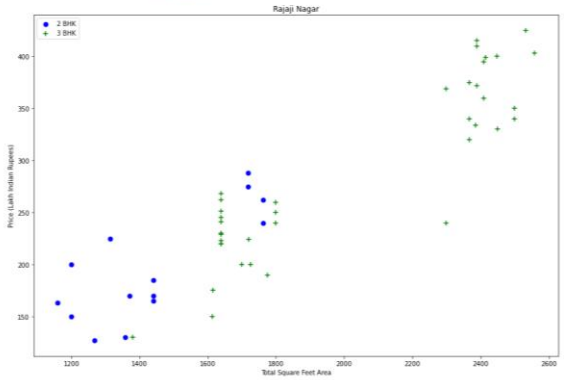
Out[41]:
```
count     12456.000000
mean       6308.502826
std        4168.127339
min         267.829813
25%        4210.526316
50%        5294.117647
75%        6916.666667
max      176470.588235
Name: price_per_sqft, dtype: float64
```

In [42]:
```
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
housing7 = remove_pps_outliers(housing6)
housing7.shape
```

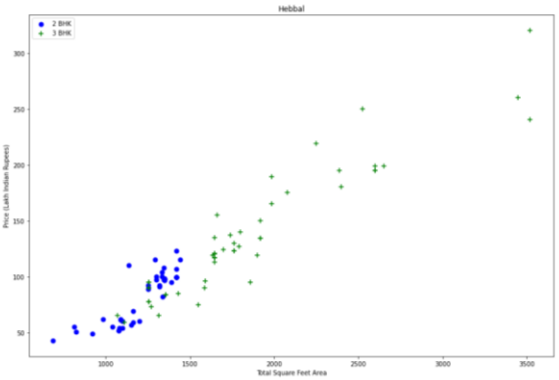
Out[42]: (10242, 7)

In [43]:
```
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(housing7,"Rajaji Nagar")
```



In [44]:
```
plot_scatter_chart(housing7,"Hebbal")
```

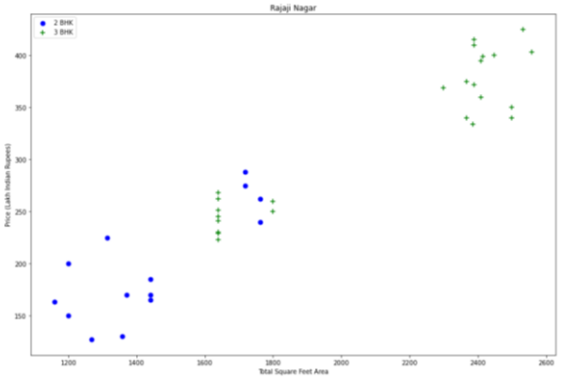


In [45]:
```
def remove_bhk_outliers(df):
    exclude_indices = np.array([])
    for location, location_df in df.groupby('location'):
        bhk_stats = {}
        for bhk, bhk_df in location_df.groupby('bhk'):
            bhk_stats[bhk] = {
                'mean': np.mean(bhk_df.price_per_sqft),
                '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')
housing8 = remove_bhk_outliers(housing7)
# df8 = df7.copy()
housing8.shape
```

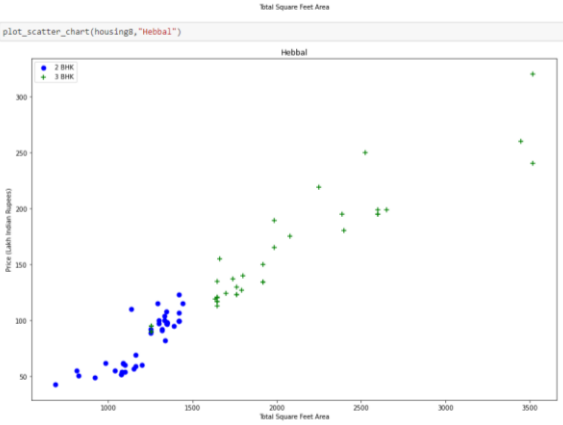
Out[45]: (7317, 7)

In [46]:
```
plot_scatter_chart(housing8,"Rajaji Nagar")
```

In [46]:
```
plot_scatter_chart(housing8,"Rajaji Nagar")
```

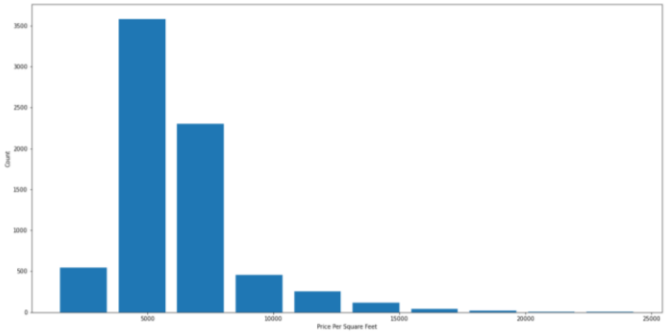


In [47]:
```
plot_scatter_chart(housing8,"Hebbal")
```



In [48]:
```
import matplotlib
matplotlib.rcParams["figure.figsize"] = (20,10)
plt.hist(housing8.price_per_sqft,rwidth=0.8)
plt.xlabel("Price Per Square Feet")
plt.ylabel("Count")
```

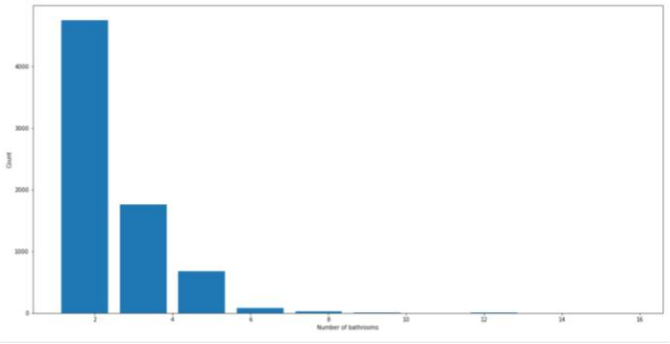
Out[48]:
```
Text(0, 0.5, 'Count')
```



Outlier Removal Using Bathrooms Feature

```
In [49]: housing8.bath.unique()
Out[49]: array([ 4.,  3.,  2.,  5.,  8.,  1.,  6.,  7.,  9., 12., 16., 13.])
```

```
In [50]: plt.hist(housing8.bath,rwidth=0.8)
         plt.xlabel("Number of bathrooms")
         plt.ylabel("Count")
Out[50]: Text(0, 0.5, 'Count')
```




```
In [51]: housing8[housing8.bath>10]
```

Out[51]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
5277	Neeladri Nagar	10 BHK	4000.0	12.0	160.0	10	4000.000000
8483	other	10 BHK	12000.0	12.0	525.0	10	4375.000000
8572	other	16 BHK	10000.0	16.0	550.0	16	5500.000000
9306	other	11 BHK	6000.0	12.0	150.0	11	2500.000000
9637	other	13 BHK	5425.0	13.0	275.0	13	5069.124424

```
In [52]: housing8[housing8.bath>housing8.bhk+2]
```

Out[52]:

	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
8408	other	6 BHK	11338.0	9.0	1000.0	6	8819.897689

```
In [53]: housing9 = housing8[housing8.bath<housing8.bhk+2]
         housing9.shape
Out[53]: (7239, 7)
```

```
In [54]: housing9.head(2)
```

Out[54]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	1st Block Jayanagar	4 BHK	2850.0	4.0	428.0	4	15017.543860
1	1st Block Jayanagar	3 BHK	1630.0	3.0	194.0	3	11901.840491

```
In [55]: housing10 = housing9.drop(['size','price_per_sqft'],axis='columns')
         housing10.head(3)
```

Out[55]:

	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

Use One Hot Encoding For Location

```
In [56]: dummies = pd.get_dummies(housing10.location)
         dummies.head(3)
```

Out[56]:

	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	Yelachenal
0	1	0	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	0
2	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0

3 rows × 241 columns

In [57]:
```
housing11 = pd.concat([housing10,dummies.drop('other',axis='columns')],axis='columns')
housing11.head()
```

Out[57]:

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

5 rows × 245 columns

In [58]:
```
housing12 = housing11.drop('location',axis='columns')
housing12.head(2)
```

Out[58]:

	total_sqft	bath	price	bhk	1st Block Jayanagar	1st Phase JP Nagar	2nd Phase Judicial Layout	2nd Stage Nagarbhavi	5th Block Hbr Layout	5th Phase JP Nagar	...	Vijayanagar	Vishveshwarya Layout	Vishwapriya Layout	Vittasandra	Whitefield
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 rows × 244 columns

References

www.geeksforgeeks.com

www.youtube.com

www.jupyter.org