UNIVERSITY OF MUMBAI

PROJECT REPORT ON

REAL ESTATE PRICE PREDICTION PROJECT

SUBMITTED BY

ASHISH RAMJANAM MALLAH

UNDER THE GUIDANCE OF

PROF.BHANUDAS SATAM & AQUILA SHAIKH

LATE BHAUSAHEB HIRAY SMARNIKA SAMITI TRUST

HIRAY GROUP OF INSTITUTES

MUMBAI - 400051

MAHARASHTRA

MCA SEM I [2020-2021]

LATE BHAUSAHEB HIRAY S.S. TRUST'S
INSTITUTE OF COMPUTER APPLICATION

ISO 90012008 CERTIFIED

S.N. 341, Next to New English School, Govt. Colony,

Bandra (East), Mumbai – 400051,

Tel: 91-22-26570892/3181

Date:

CERTIFICATE

This is to certify that Mr. ASHISH RAMJANAM MALLAH

---Roll No. 202114

is a student of MCA of 1th year Semester-I has completed successfully full-semester Mini-Project of subject "REAL STATE PRICE PREDICTION" for the academic year 2020 – 21.

----------------- **---------------**

Subject In-Charge **Director**

External Examiner

PROFORMA FOR THE APPROVAL PROJECT PROPOSAL

PNR No.:- 2017016400250142 **SEAT No.:-**202178

Name of the Student:- ASHISH RAMJANAM MALLAH

Title of the Project:- REAL ESTATE PRICE PREDICTION

Name of the Guide:- Prof. AQUILA SHAIKH & BHANUDAS SATAM

Teaching experience of the Guide:

Is this your first submission? Yes No

______________________ ___________________

Signature of the Student **Signature of the Guide**

Date: ________________ Date:_______________

Signature of the Coordinator: __________________

Date: ____________________

ACKNOWLEDGEMENT

I extend my deepest appreciation to my esteemed guide, **Prof. AQUILA SHAIKH & BHANUDAS SATAM** for providing me with the possibility to complete this project with the right guidance and advice.

Special gratitude I give to my respected head of the division **PROF. VIKRAM PATALBANSI**, for allowing me to use the facilities available and also help me to coordinate my project Furthermore, I would also like to acknowledge with much appreciation the crucial role of faculty members on this occasion.

Last but not least, I would like to thank friends who help me to assemble the parts and gave a suggestion about the project.

Abstract:-

Real estate is the least transparent industry in our ecosystem. Housing prices keep changing day in and day out and sometimes are hyped rather than being based on valuation.

Predicting housing prices with real factors is the main crux of our research project. Here we aim to make our evaluations based on every basic parameter that is considered while determining the price.

We use various regression techniques in this pathway, and our results are not sole determination of one technique rather it is the weighted mean of various techniques to give most accurate results.

The results proved that this approach yields minimum error and maximum accuracy than individual algorithms applied. We also propose to use real-time neighborhood details using Google maps to get exact real-world valuations.

TABLE OF CONTENTS

5) ABBREVIATIONS		
6) CONCLUSION		
7) REFERENCE		

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Real Estate Price Prediction Project

Investment is a business activity on which most people are interested in this globalization era. There are several objects that are often used for investment, for example, gold, stocks and property. In particular, property investment has increased significantly.

Housing price trends are not only the concern of buyers and sellers, but it also indicates the current economic situation. There are many factors which has impact on house prices, such as numbers of bedrooms and bathrooms. Even the nearby location, a location with a great accessibility to highways, expressways, schools, shopping malls and local employment opportunities contributes to the rise in house price.

Manual house predication becomes difficult, hence there are many systems developed for house price prediction. We have proposed an advanced house prediction system using linear regression. This system aim is to make a model which can give us a good house pricing prediction based on other variables. We are going to use Linear Regression for this dataset and hence it gives a good accuracy.

This house price prediction project has two modules namely, Admin and User. Admin can add location and view the location. Admin has authority to add density on the basis of per unit area. User can view the location and see the predicted housing price for the particular location.

1.2.Need and Motivation:-

Having lived in India for so many years if there is one thing that I had been taking for granted, it’s that housing and rental prices continue to rise. Since the housing crisis of 2008, housing prices have recovered remarkably well, especially in major housing markets. However, in the 4th quarter of 2016, I was surprised to read that Bombay housing prices had fallen the most in the last 4 years. In fact, median resale prices for condos and coops fell 6.3%, marking the first time there was a decline since Q1 of 2017.

The decline has been partly attributed to political uncertainty domestically and abroad and the 2014 election. So, to maintain the transparency among customers and also the comparison can be made easy through this model. If customer finds the price of house at some given website higher than the price predicted by the model, so he can reject that house.

REQUIREMENTS

❖ Hardware Requirement:

- Processor –Core i3
- Hard Disk – 160 GB
- Memory – 1GB RAM

❖ Software Requirement:

- Windows 7 or higher
- Python
- python flask server

❖ Advantages

- Saves time
- Easy to access the system anywhere and anytime.

❖ Limitation

- Requires an active internet connection.

❖ Application

- This system can be used by the multiple peoples to get the counselling sessions online.

❖ Modules:

The system comprises of 3 major modules with their sub-modules as follows:

1. **Admin:**
   - **Add Location:** Admin can add locations.
   - **View Location:** Admin can View the added location.
   - **Add Density:** Admin can add density of the houses by per unit area.

2. **User:**
   - **View Location:** User can view the location.
   - **View Predicted housing price:** User can view the predicted price of house.

2.DATA SET

Steps in Preparing Data for Model

❖ STEPS:

This data science project walks through 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 user to enter home square ft area, bedrooms etc and it will call python flask server to retrieve the predicted price.

During model building we will cover almost all **data science concepts** such as :-

2.1. Data load and cleaning,

2.2. Feature engineering,

2.3. Outlier detection and removal,

2.4. Dimensionality Reduction

2.5. Grid Search Cv

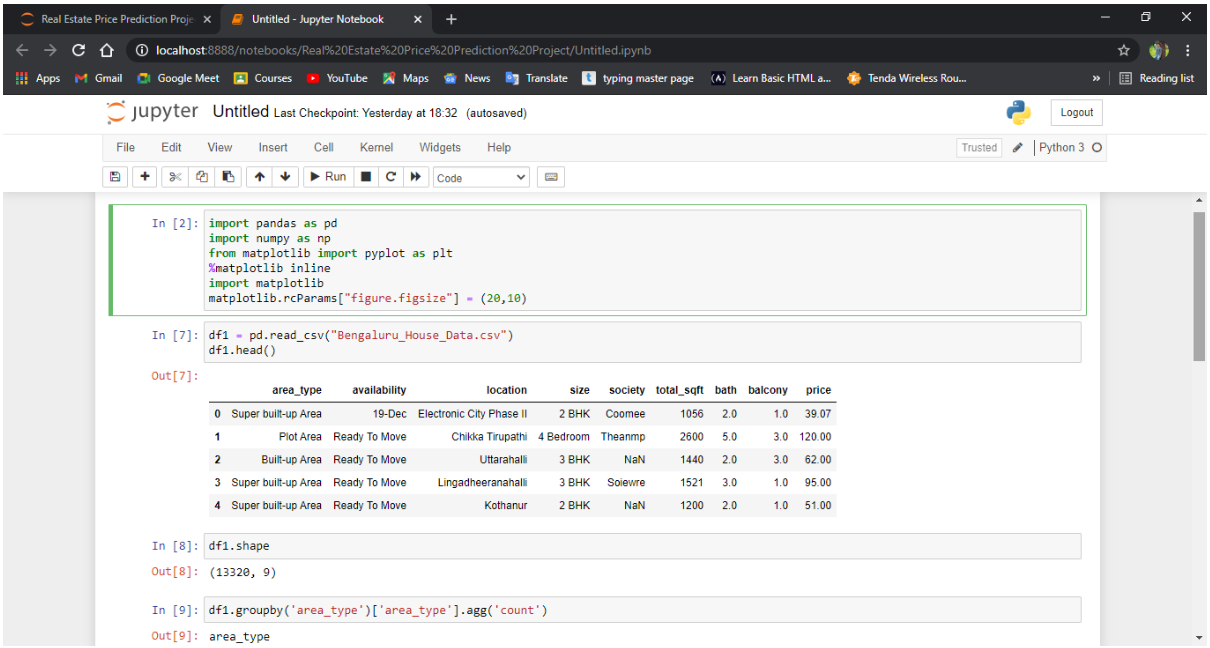
2.6. K-Fold Cross Validation

2.1)Data Load and Cleaning:-

Data cleansing or data cleaning is the process of detecting and correcting corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data.

Data cleansing may be performed interactively with data wrangling tools, or as batch processing through scripting.

Data Cleaning Screen Shot :-

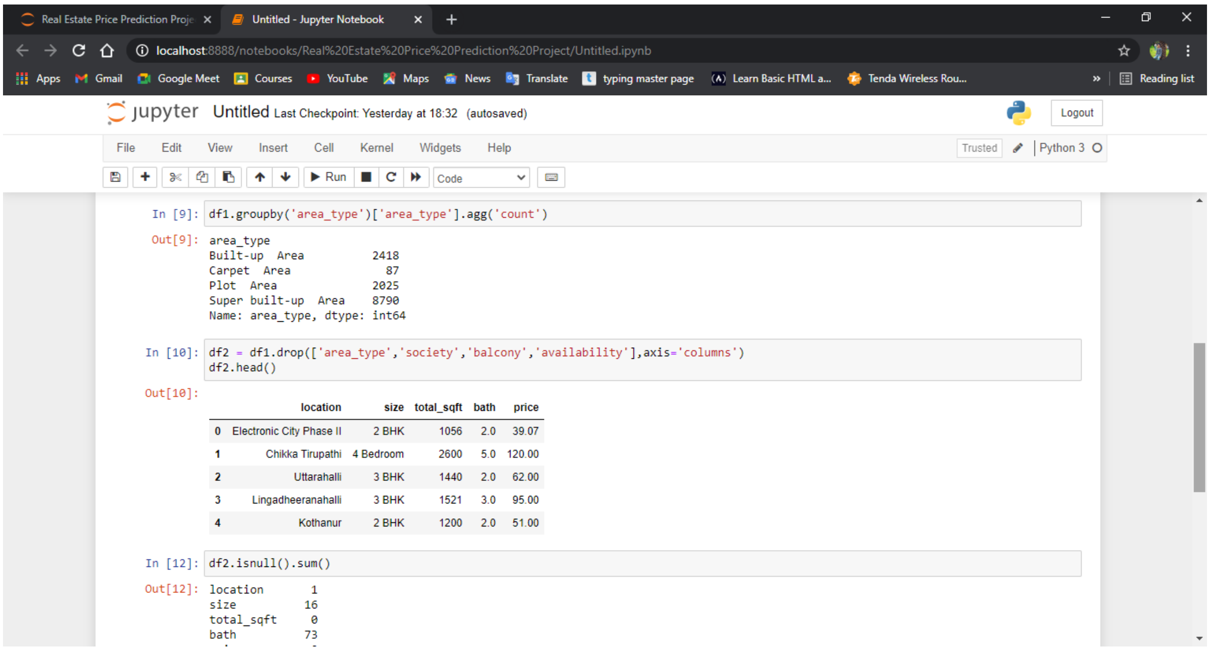


Real Estate Price Prediction Proje
Untitled - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/Untitled.ipynb
Apps
Gmail
Google Meet
Courses
YouTube
Maps
News
Translate
typing master page
Learn Basic HTML a...
Tenda Wireless Rou...
Reading list
jupyter
Untitled Last Checkpoint: Yesterday at 18:32 (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

```
In [2]: 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 [7]: df1 = pd.read_csv("Bengaluru_House_Data.csv")
        df1.head()

Out[7]:
```

	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

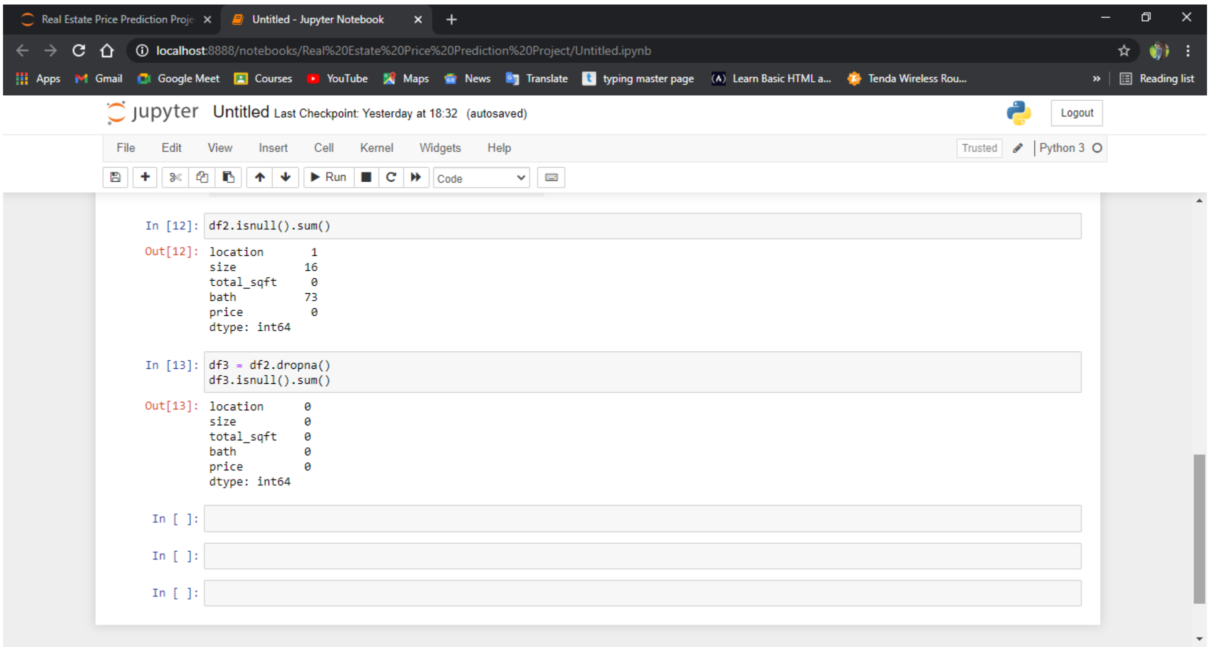
```
In [8]: df1.shape

Out[8]: (13320, 9)

In [9]: df1.groupby('area_type')['area_type'].agg('count')

Out[9]: area_type
```


Real Estate Price Prediction Proje
Untitled - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/Untitled.ipynb
Apps
Gmail
Google Meet
Courses
YouTube
Maps
News
Translate
typing master page
Learn Basic HTML a...
Tenda Wireless Rou...
Reading list
jupyter Untitled Last Checkpoint: Yesterday at 18:32 (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

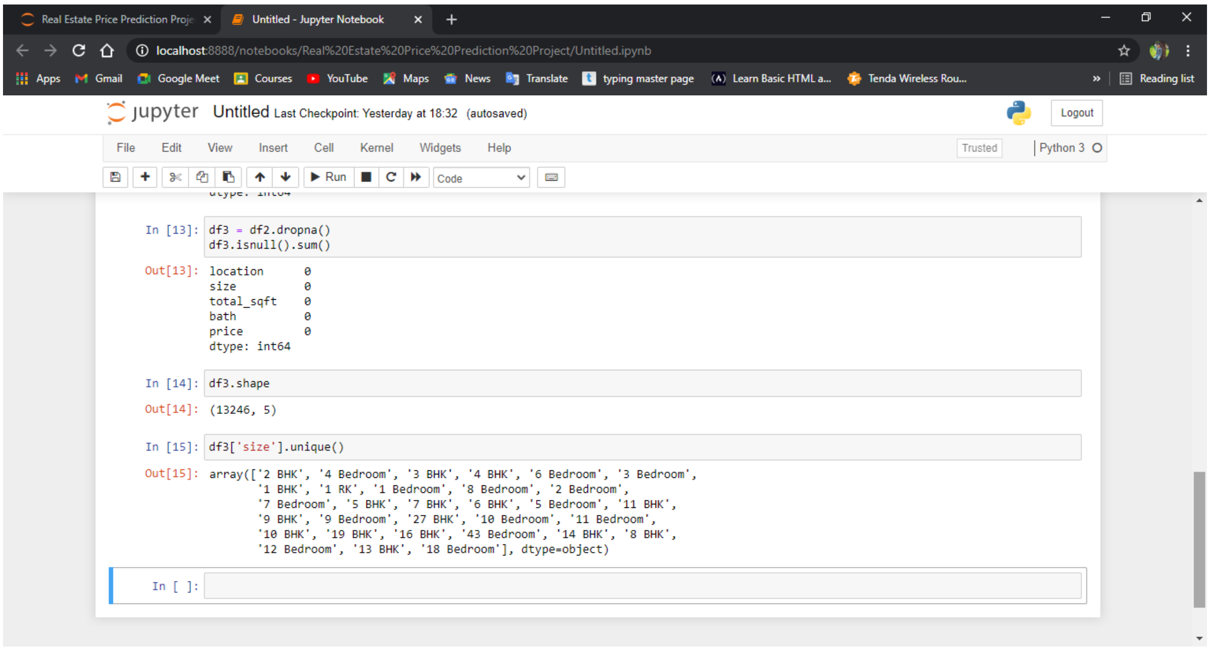
```
In [9]: df1.groupby('area_type')['area_type'].agg('count')

Out[9]: area_type
        Built-up  Area          2418
        Carpet  Area              87
        Plot  Area              2025
        Super built-up  Area    8790
        Name: area_type, dtype: int64

In [10]: df2 = df1.drop(['area_type','society','balcony','availability'],axis='columns')
         df2.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

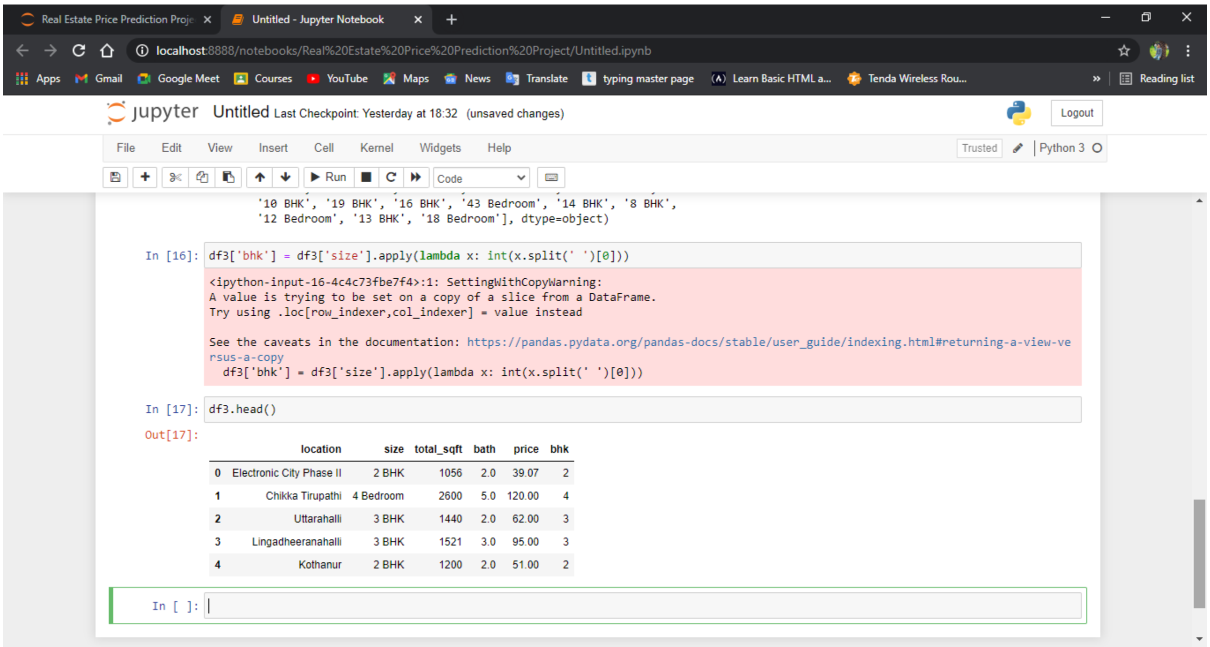
```
In [12]: df2.isnull().sum()

Out[12]: location       1
         size          16
         total_sqft     0
         bath          73
```


Real Estate Price Prediction Proje
Untitled - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/Untitled.ipynb
Apps
Gmail
Google Meet
Courses
YouTube
Maps
News
Translate
typing master page
Learn Basic HTML a...
Tenda Wireless Rou...
Reading list
jupyter Untitled Last Checkpoint: Yesterday at 18:32 (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

```
In [12]: df2.isnull().sum()

Out[12]: location       1
         size          16
         total_sqft     0
         bath          73
         price          0
         dtype: int64

In [13]: df3 = df2.dropna()
         df3.isnull().sum()

Out[13]: location      0
         size          0
         total_sqft    0
         bath          0
         price         0
         dtype: int64

In [ ]:

In [ ]:

In [ ]:
```


Real Estate Price Prediction Proje
Untitled - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/Untitled.ipynb
Apps
Gmail
Google Meet
Courses
YouTube
Maps
News
Translate
typing master page
Learn Basic HTML a...
Tenda Wireless Rou...
Reading list
jupyter
Untitled Last Checkpoint: Yesterday at 18:32 (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

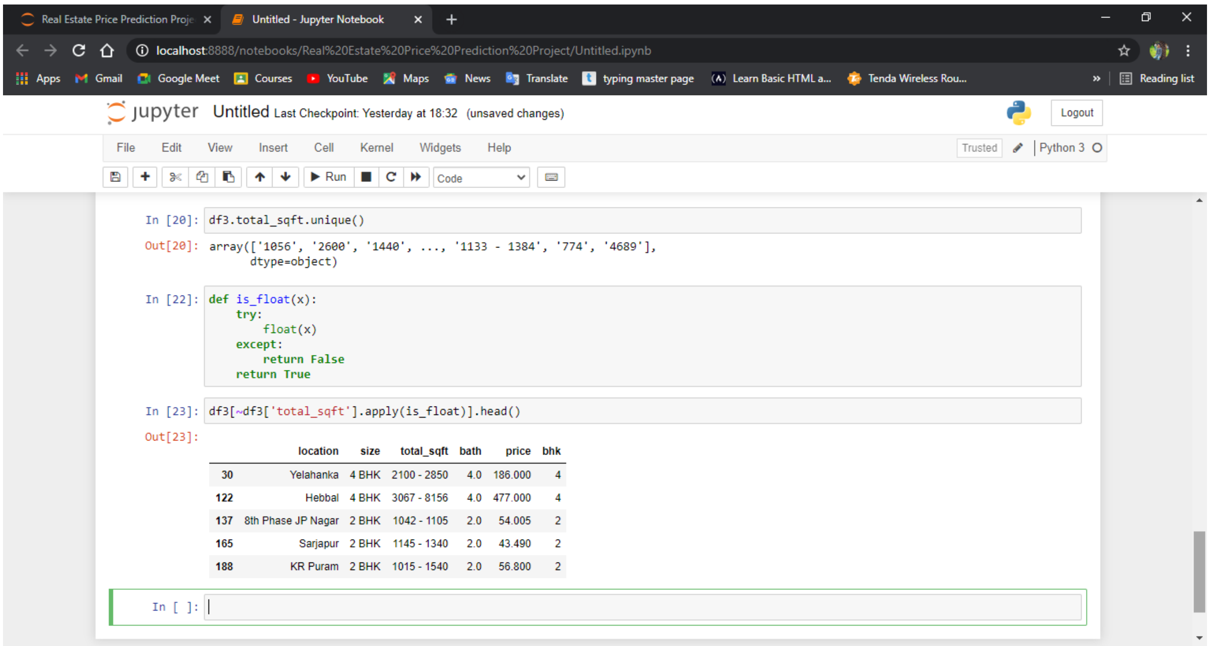
```
In [13]: df3 = df2.dropna()
         df3.isnull().sum()

Out[13]: location      0
         size          0
         total_sqft    0
         bath          0
         price         0
         dtype: int64

In [14]: df3.shape

Out[14]: (13246, 5)

In [15]: df3['size'].unique()

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

In [ ]:
```


Real Estate Price Prediction Proje
Untitled - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/Untitled.ipynb
Apps
Gmail
Google Meet
Courses
YouTube
Maps
News
Translate
typing master page
Learn Basic HTML a...
Tenda Wireless Rou...
Reading list
jupyter
Untitled Last Checkpoint: Yesterday at 18:32 (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

```
                '10 BHK', '19 BHK', '16 BHK', '43 Bedroom', '14 BHK', '8 BHK',
                '12 Bedroom', '13 BHK', '18 Bedroom'], dtype=object)

In [16]: df3['bhk'] = df3['size'].apply(lambda x: int(x.split(' ')[0]))

         <ipython-input-16-4c4c73fbe7f4>: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-ve
         rsus-a-copy
           df3['bhk'] = df3['size'].apply(lambda x: int(x.split(' ')[0]))

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

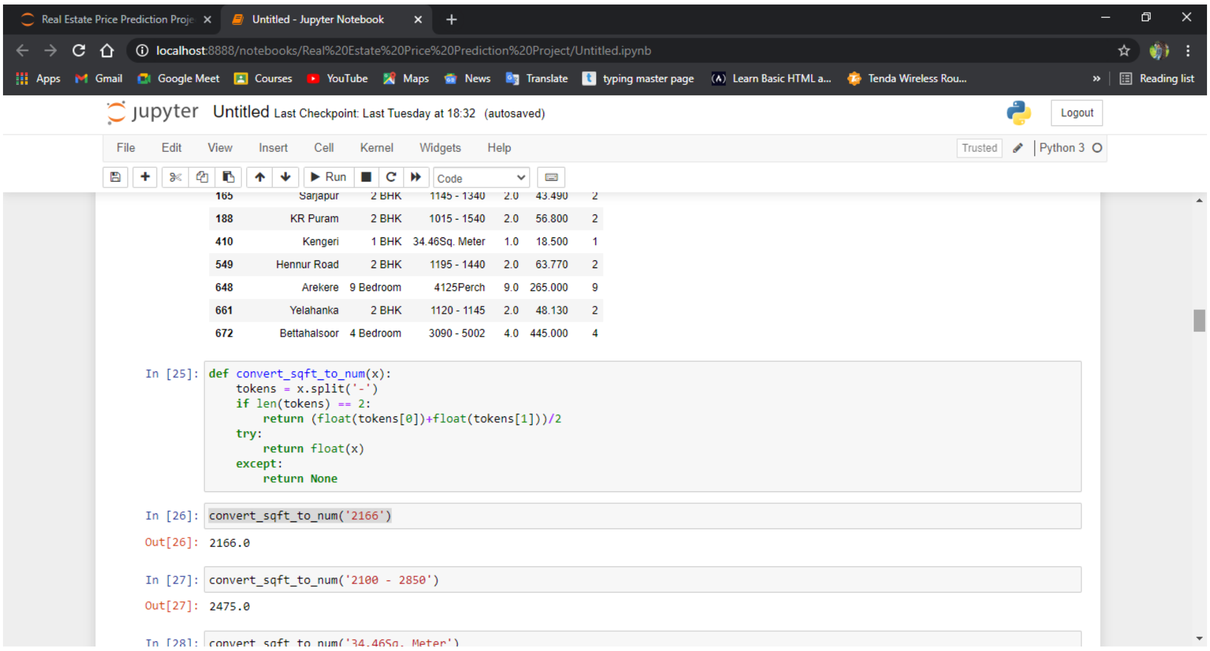
```
In [ ]:
```


Real Estate Price Prediction Proje
Untitled - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/Untitled.ipynb
Apps Gmail Google Meet Courses YouTube Maps News Translate typing master page Learn Basic HTML a... Tenda Wireless Rou... Reading list
jupyter Untitled Last Checkpoint: Yesterday at 18:32 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3
Run Code

```
In [20]: df3.total_sqft.unique()

Out[20]: array(['1056', '2600', '1440', ..., '1133 - 1384', '774', '4689'],
               dtype=object)

In [22]: def is_float(x):
             try:
                 float(x)
             except:
                 return False
             return True

In [23]: df3[~df3['total_sqft'].apply(is_float)].head()

Out[23]:
```

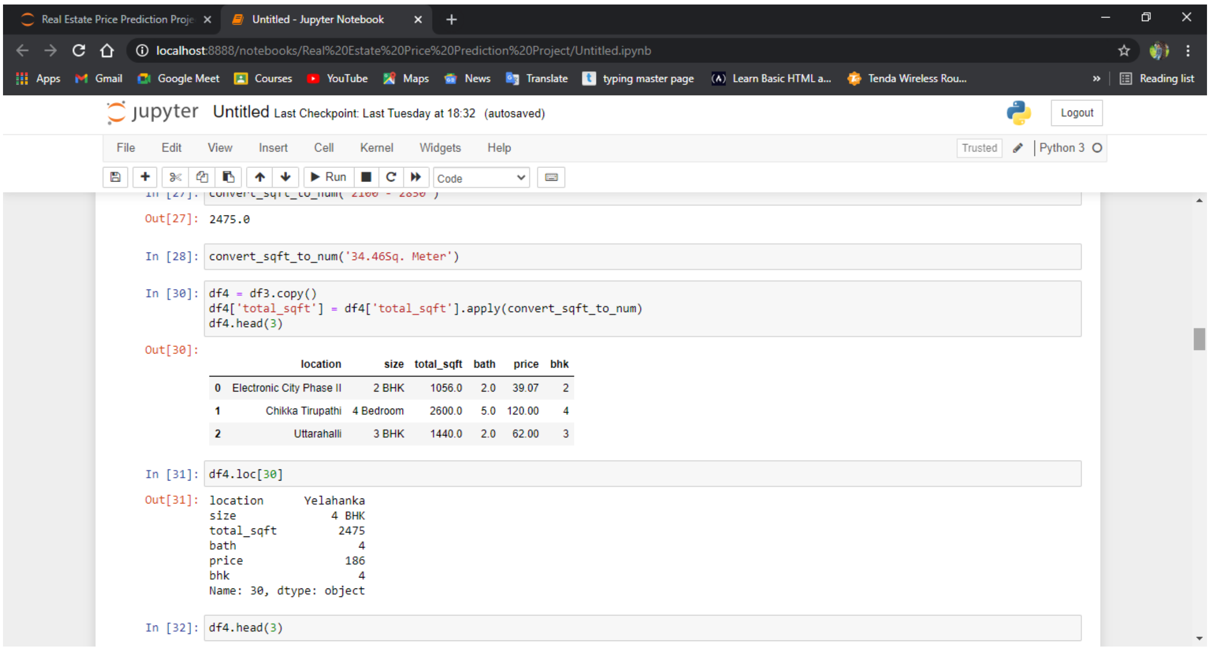
	location	size	total_sqft	bath	price	bhk
30	Yelahanka	4 BHK	2100 - 2850	4.0	186.000	4
122	Hebbal	4 BHK	3067 - 8156	4.0	477.000	4
137	8th Phase JP Nagar	2 BHK	1042 - 1105	2.0	54.005	2
165	Sarjapur	2 BHK	1145 - 1340	2.0	43.490	2
188	KR Puram	2 BHK	1015 - 1540	2.0	56.800	2

```
In [ ]:
```


Real Estate Price Prediction Proje
Untitled - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/Untitled.ipynb
Apps Gmail Google Meet Courses YouTube Maps News Translate typing master page Learn Basic HTML a... Tenda Wireless Rou... Reading list
jupyter Untitled Last Checkpoint: Last Tuesday at 18:32 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3
Run Code

165	Sarjapur	2 BHK	1145 - 1340	2.0	43.490	2
188	KR Puram	2 BHK	1015 - 1540	2.0	56.800	2
410	Kengeri	1 BHK	34.46Sq. Meter	1.0	18.500	1
549	Hennur Road	2 BHK	1195 - 1440	2.0	63.770	2
648	Arekere	9 Bedroom	4125Perch	9.0	265.000	9
661	Yelahanka	2 BHK	1120 - 1145	2.0	48.130	2
672	Bettahalsoor	4 Bedroom	3090 - 5002	4.0	445.000	4

```
In [25]: 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 [26]: convert_sqft_to_num('2166')

Out[26]: 2166.0

In [27]: convert_sqft_to_num('2100 - 2850')

Out[27]: 2475.0

In [28]: convert_sqft_to_num('34.46Sq. Meter')
```


Real Estate Price Prediction Proje
Untitled - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/Untitled.ipynb
Apps
Gmail
Google Meet
Courses
YouTube
Maps
News
Translate
typing master page
Learn Basic HTML a...
Tenda Wireless Rou...
Reading list
jupyter Untitled Last Checkpoint: Last Tuesday at 18:32 (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

```
In [27]: convert_sqft_to_num('2100 - 2850')
Out[27]: 2475.0

In [28]: convert_sqft_to_num('34.46Sq. Meter')

In [30]: df4 = df3.copy()
         df4['total_sqft'] = df4['total_sqft'].apply(convert_sqft_to_num)
         df4.head(3)
Out[30]:
```

	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

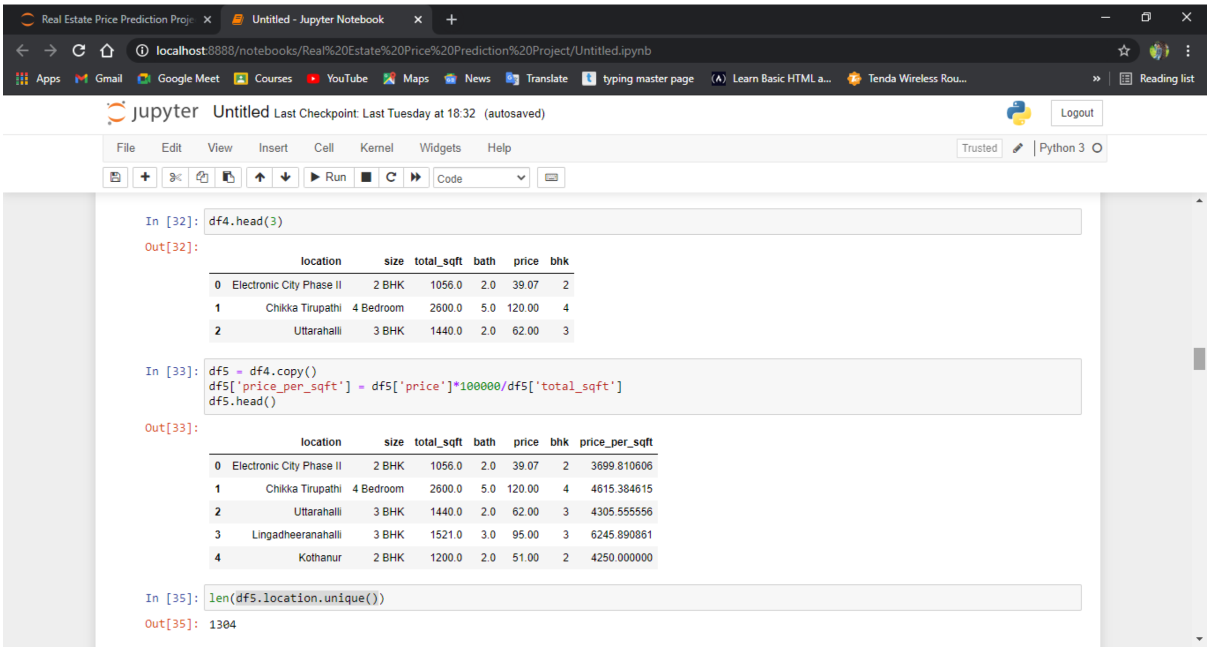
```
In [31]: df4.loc[30]
Out[31]: location       Yelahanka
         size               4 BHK
         total_sqft          2475
         bath                   4
         price                186
         bhk                    4
         Name: 30, dtype: object

In [32]: df4.head(3)
```

2.2) Feature Engineering:-

Feature engineering is **the process of using domain knowledge to extract features from raw data.**

These features can be used to improve the performance of machine learning algorithms. Feature engineering can be considered as applied machine learning itself.

ScreenShot:-



Real Estate Price Prediction Proje
Untitled - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/Untitled.ipynb
Apps Gmail Google Meet Courses YouTube Maps News Translate typing master page Learn Basic HTML a... Tenda Wireless Rou... Reading list
jupyter Untitled Last Checkpoint: Last Tuesday at 18:32 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3
Run Code

```
In [32]: df4.head(3)
```

Out[32]:

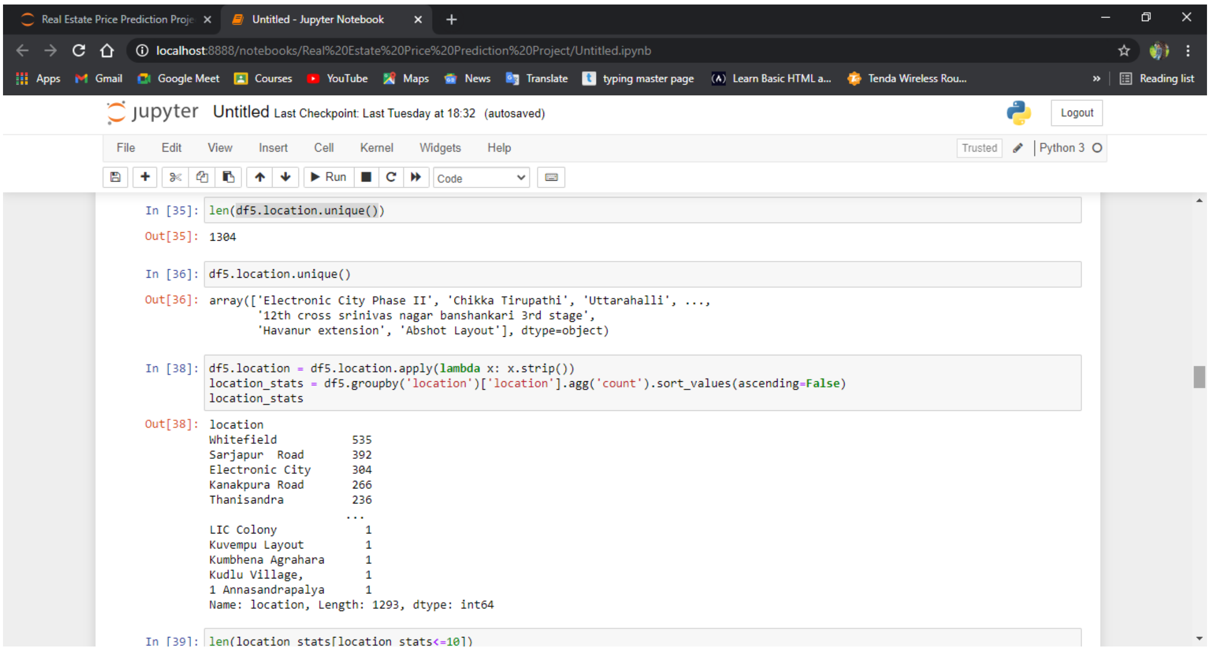
	location	size	total_sqft	bath	price	bhk
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.07	2
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3

```
In [33]: df5 = df4.copy()
         df5['price_per_sqft'] = df5['price']*100000/df5['total_sqft']
         df5.head()
```

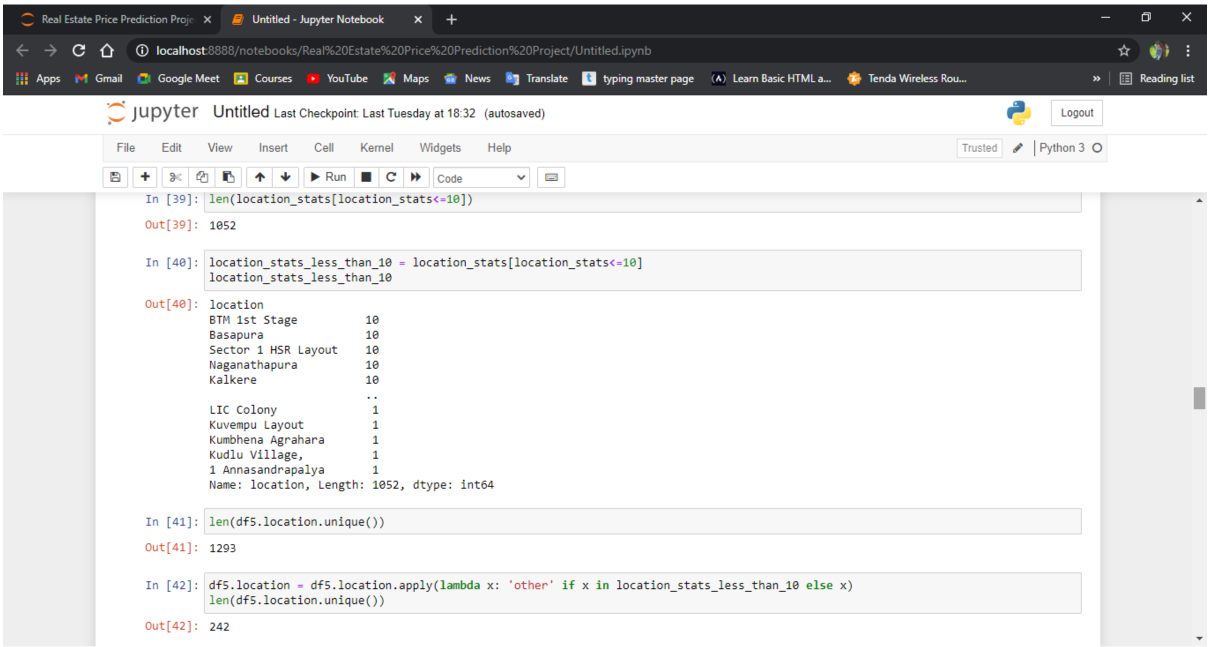
Out[33]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.07	2	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

```
In [35]: len(df5.location.unique())

Out[35]: 1304
```


Real Estate Price Prediction Proje
Untitled - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/Untitled.ipynb
Apps Gmail Google Meet Courses YouTube Maps News Translate typing master page Learn Basic HTML a... Tenda Wireless Rou... Reading list
jupyter Untitled Last Checkpoint: Last Tuesday at 18:32 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
In [35]: len(df5.location.unique())

Out[35]: 1304

In [36]: df5.location.unique()

Out[36]: array(['Electronic City Phase II', 'Chikka Tirupathi', 'Uttarahalli', ...,
                '12th cross srinivas nagar banshankari 3rd stage',
                'Havanur extension', 'Abshot Layout'], dtype=object)

In [38]: df5.location = df5.location.apply(lambda x: x.strip())
         location_stats = df5.groupby('location')['location'].agg('count').sort_values(ascending=False)
         location_stats

Out[38]: location
         Whitefield            535
         Sarjapur  Road        392
         Electronic City       304
         Kanakpura Road        266
         Thanisandra           236
                              ...
         LIC Colony              1
         Kuvempu Layout          1
         Kumbhena Agrahara       1
         Kudlu Village,          1
         1 Annasandrapalya       1
         Name: location, Length: 1293, dtype: int64

In [39]: len(location stats[location stats<=10])
```


Real Estate Price Prediction Proje
Untitled - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/Untitled.ipynb
Apps Gmail Google Meet Courses YouTube Maps News Translate typing master page Learn Basic HTML a... Tenda Wireless Rou... Reading list
jupyter Untitled Last Checkpoint: Last Tuesday at 18:32 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
In [39]: len(location_stats[location_stats<=10])

Out[39]: 1052

In [40]: location_stats_less_than_10 = location_stats[location_stats<=10]
         location_stats_less_than_10

Out[40]: location
         BTM 1st Stage          10
         Basapura               10
         Sector 1 HSR Layout    10
         Naganathapura          10
         Kalkere                10
                                ..
         LIC Colony              1
         Kuvempu Layout          1
         Kumbhena Agrahara       1
         Kudlu Village,          1
         1 Annasandrapalya       1
         Name: location, Length: 1052, dtype: int64

In [41]: len(df5.location.unique())

Out[41]: 1293

In [42]: df5.location = df5.location.apply(lambda x: 'other' if x in location_stats_less_than_10 else x)
         len(df5.location.unique())

Out[42]: 242
```

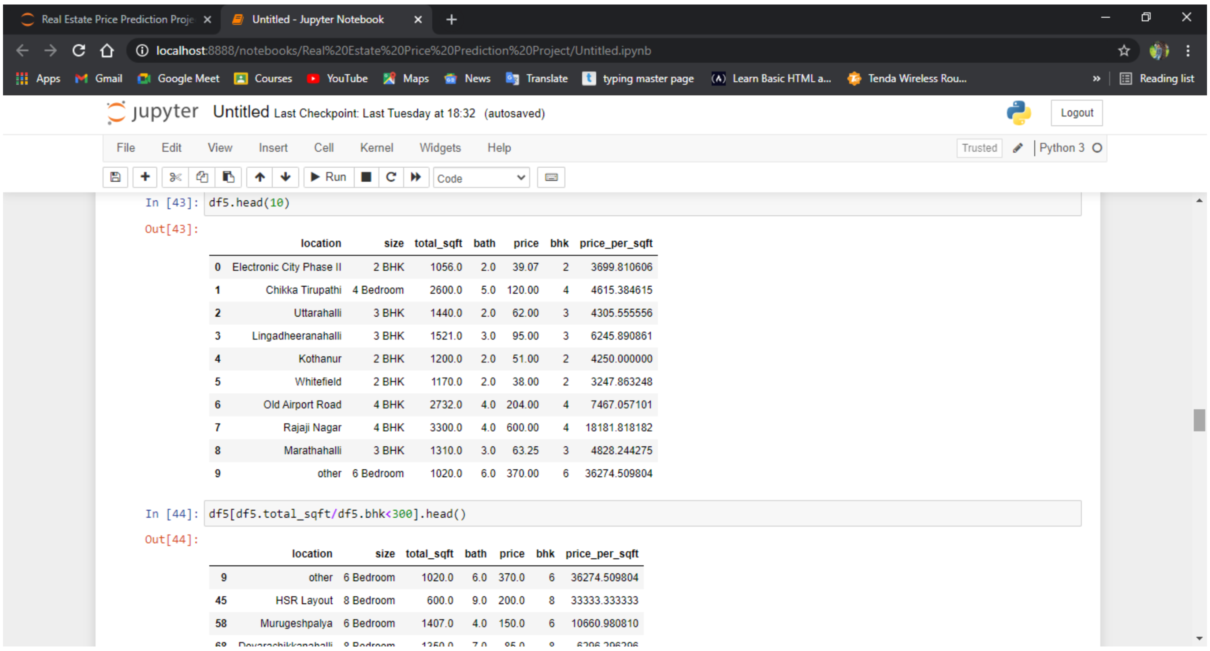
2.3) Outlier detection and removal:-

In statistics, an outlier is a data point that differs significantly from other observations.

An outlier may be due to variability in the measurement or it may indicate experimental error; the latter are sometimes excluded from the data set.

An outlier can cause serious problems in statistical analyses.

ScreenShot:-


Real Estate Price Prediction Proje
Untitled - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/Untitled.ipynb
Apps
Gmail
Google Meet
Courses
YouTube
Maps
News
Translate
typing master page
Learn Basic HTML a...
Tenda Wireless Rou...
Reading list
jupyter
Untitled Last Checkpoint: Last Tuesday at 18:32 (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

In [43]:
```
df5.head(10)
```

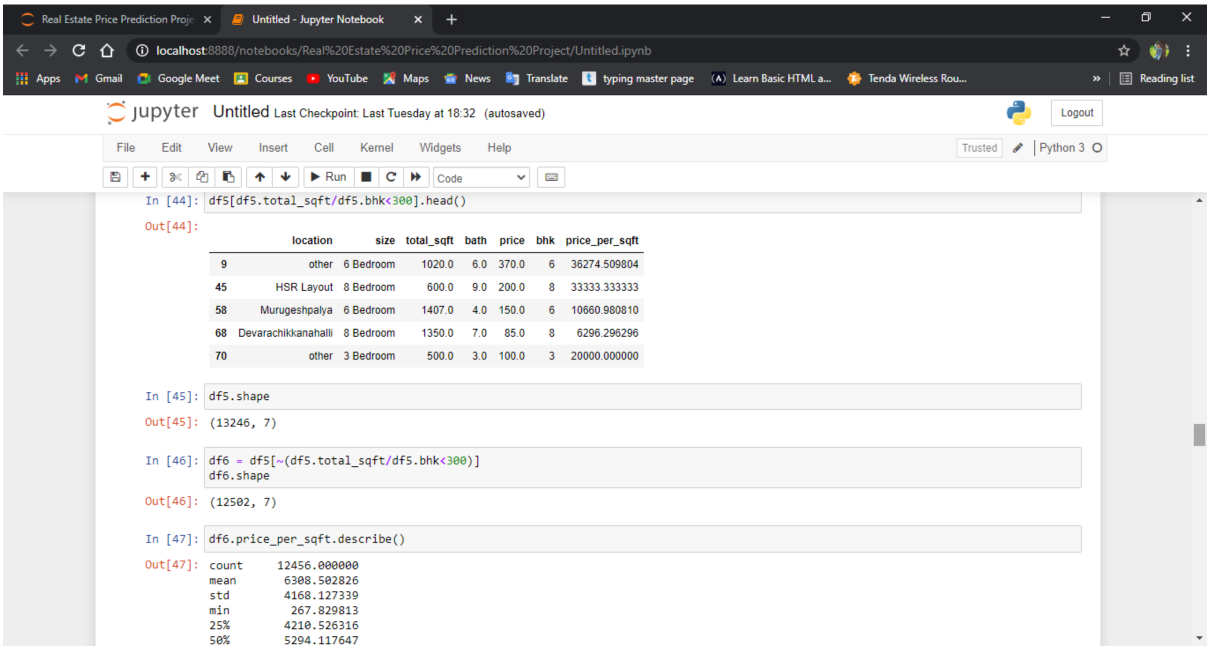
Out[43]:

	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

In [44]:
```
df5[df5.total_sqft/df5.bhk<300].head()
```

Out[44]:

	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

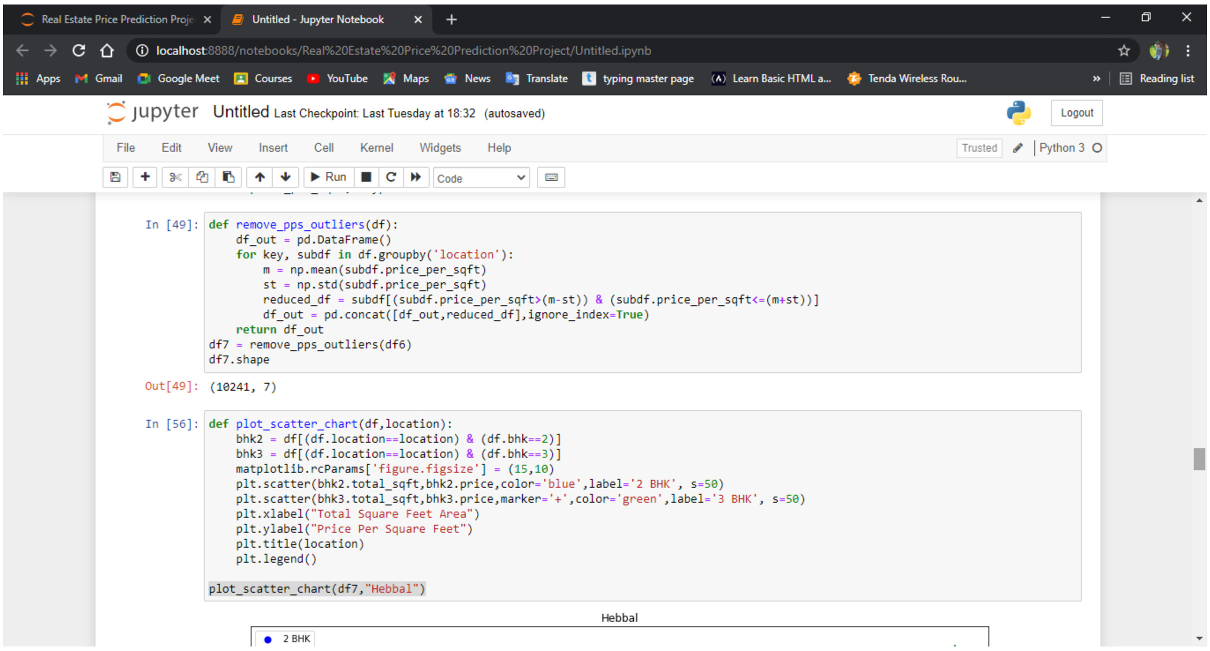

Real Estate Price Prediction Proje
Untitled - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/Untitled.ipynb
Apps
Gmail
Google Meet
Courses
YouTube
Maps
News
Translate
typing master page
Learn Basic HTML a...
Tenda Wireless Rou...
Reading list
jupyter
Untitled Last Checkpoint: Last Tuesday at 18:32 (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

```
In [44]: df5[df5.total_sqft/df5.bhk<300].head()
```

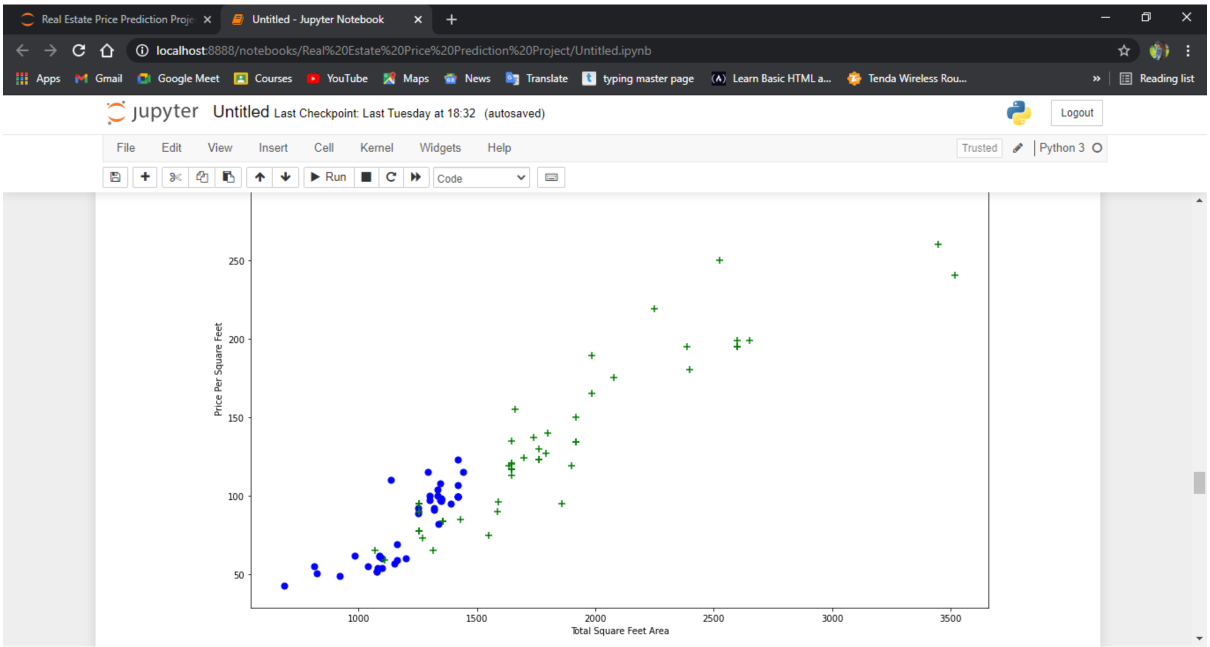
Out[44]:

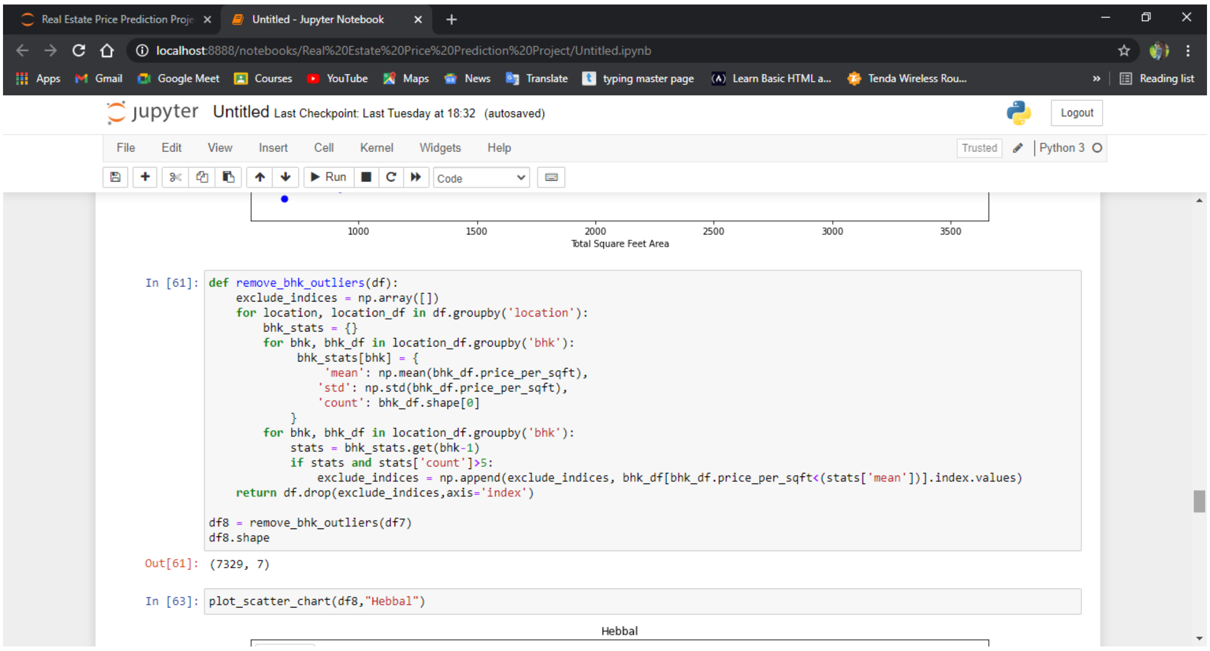
	location	size	total_sqft	bath	price	bhk	price_per_sqft
9	other	6 Bedroom	1020.0	6.0	370.0	6	36274.509804
45	HSR Layout	8 Bedroom	600.0	9.0	200.0	8	33333.333333
58	Murugeshpalya	6 Bedroom	1407.0	4.0	150.0	6	10660.980810
68	Devarachikkanahalli	8 Bedroom	1350.0	7.0	85.0	8	6296.296296
70	other	3 Bedroom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

```
In [45]: df5.shape

Out[45]: (13246, 7)

In [46]: df6 = df5[~(df5.total_sqft/df5.bhk<300)]
         df6.shape

Out[46]: (12502, 7)

In [47]: df6.price_per_sqft.describe()

Out[47]: count    12456.000000
         mean      6308.502826
         std       4168.127339
         min        267.829813
         25%       4210.526316
         50%       5294.117647
```



```
In [49]: 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

Out[49]: (10241, 7)

In [56]: 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,"Hebbal")
```





In [61]:

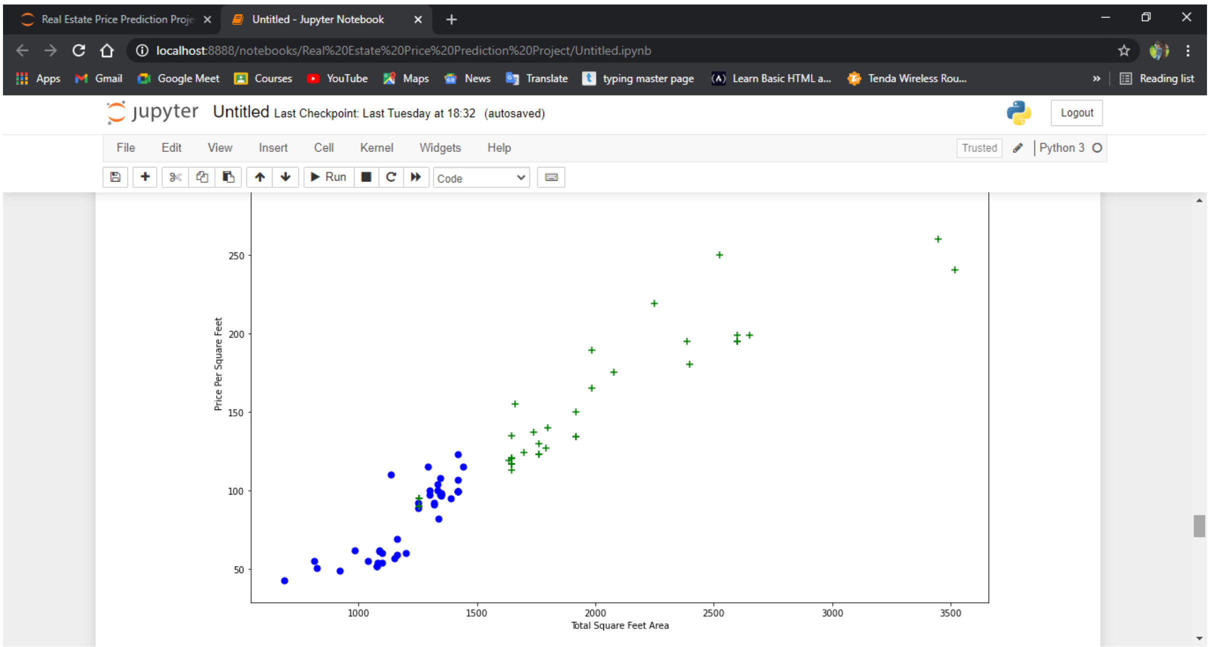
```
def remove_bhk_outliers(df):
    exclude_indices = np.array([])
    for location, location_df in df.groupby('location'):
        bhk_stats = {}
        for bhk, bhk_df in location_df.groupby('bhk'):
             bhk_stats[bhk] = {
                 'mean': np.mean(bhk_df.price_per_sqft),
        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

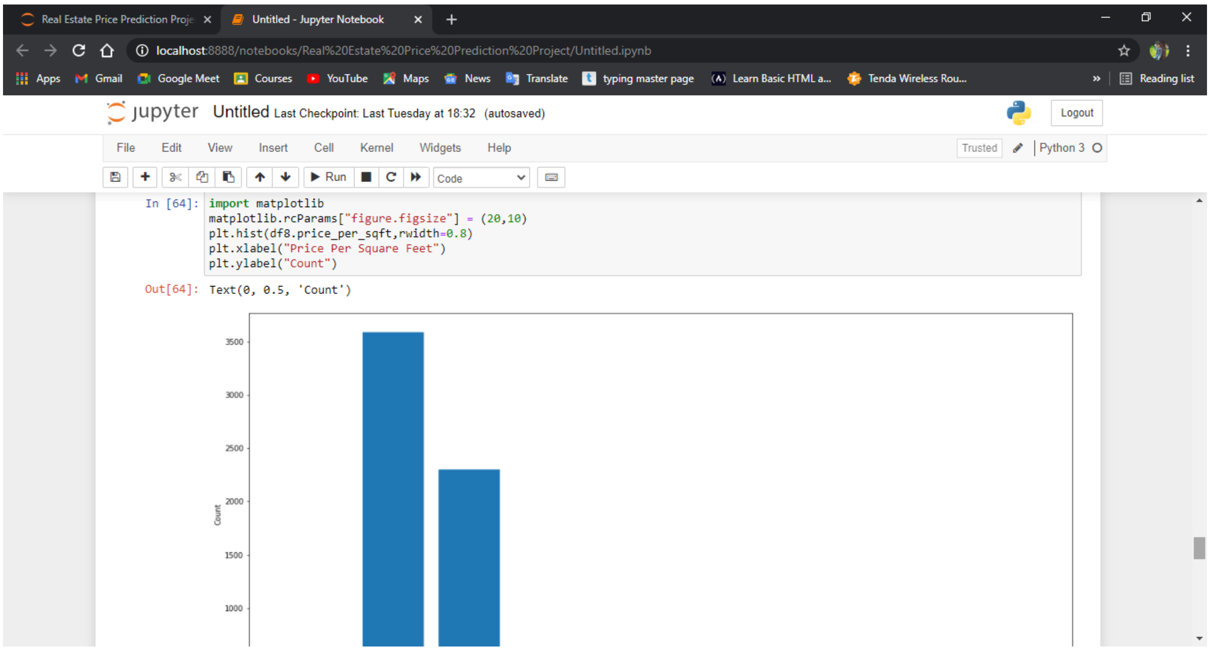
Out[61]:

```
(7329, 7)
```

In [63]:

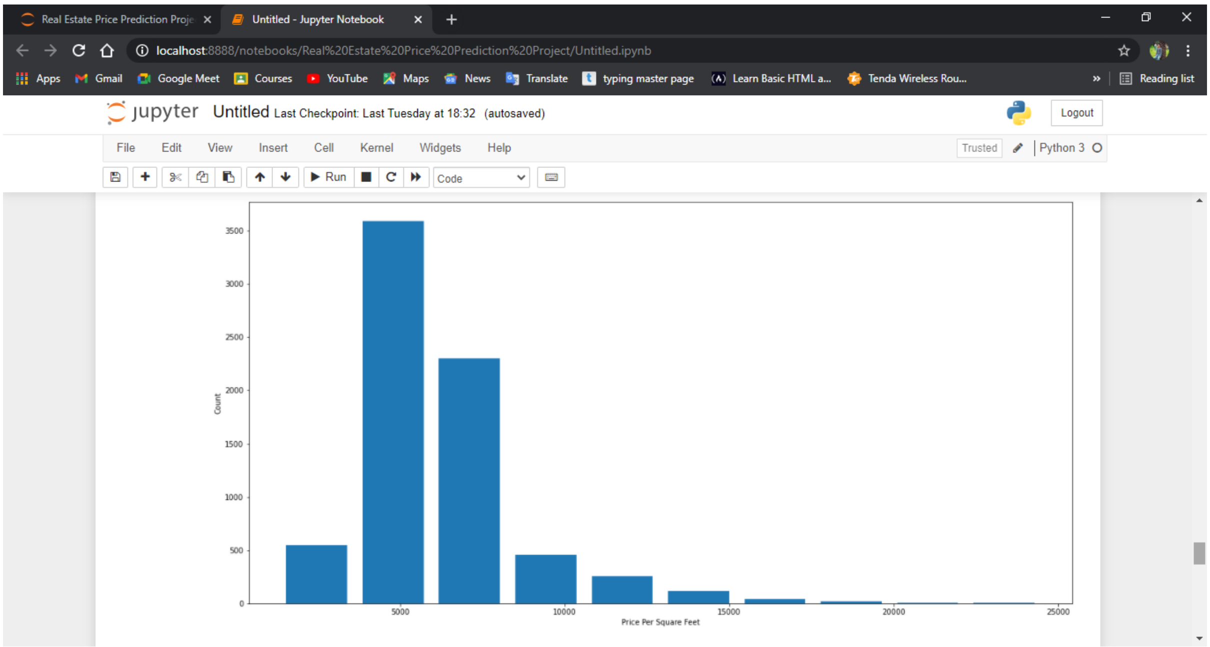
```
plot_scatter_chart(df8,"Hebbal")
```

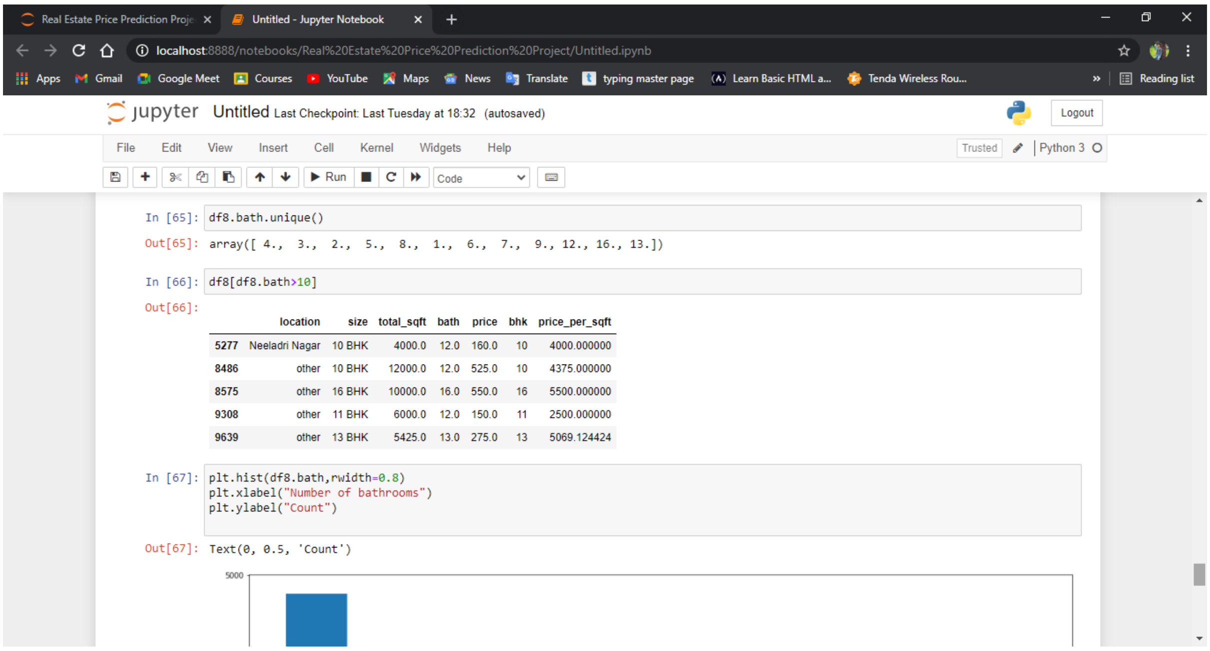

Real Estate Price Prediction Proje
Untitled - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/Untitled.ipynb
Apps Gmail Google Meet Courses YouTube Maps News Translate typing master page Learn Basic HTML a... Tenda Wireless Rou... Reading list
jupyter Untitled Last Checkpoint: Last Tuesday at 18:32 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3
Run Code
250
200
150
100
50
Price Per Square Feet
1000
1500
2000
2500
3000
3500
Total Square Feet Area


Real Estate Price Prediction Proje
Untitled - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/Untitled.ipynb
Apps Gmail Google Meet Courses YouTube Maps News Translate typing master page Learn Basic HTML a... Tenda Wireless Rou... Reading list
jupyter Untitled Last Checkpoint: Last Tuesday at 18:32 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3
Run Code

```
In [64]: import matplotlib
         matplotlib.rcParams["figure.figsize"] = (20,10)
         plt.hist(df8.price_per_sqft,rwidth=0.8)
         plt.xlabel("Price Per Square Feet")
         plt.ylabel("Count")

Out[64]: Text(0, 0.5, 'Count')
```

3500
3000
2500
2000
1500
1000
Count

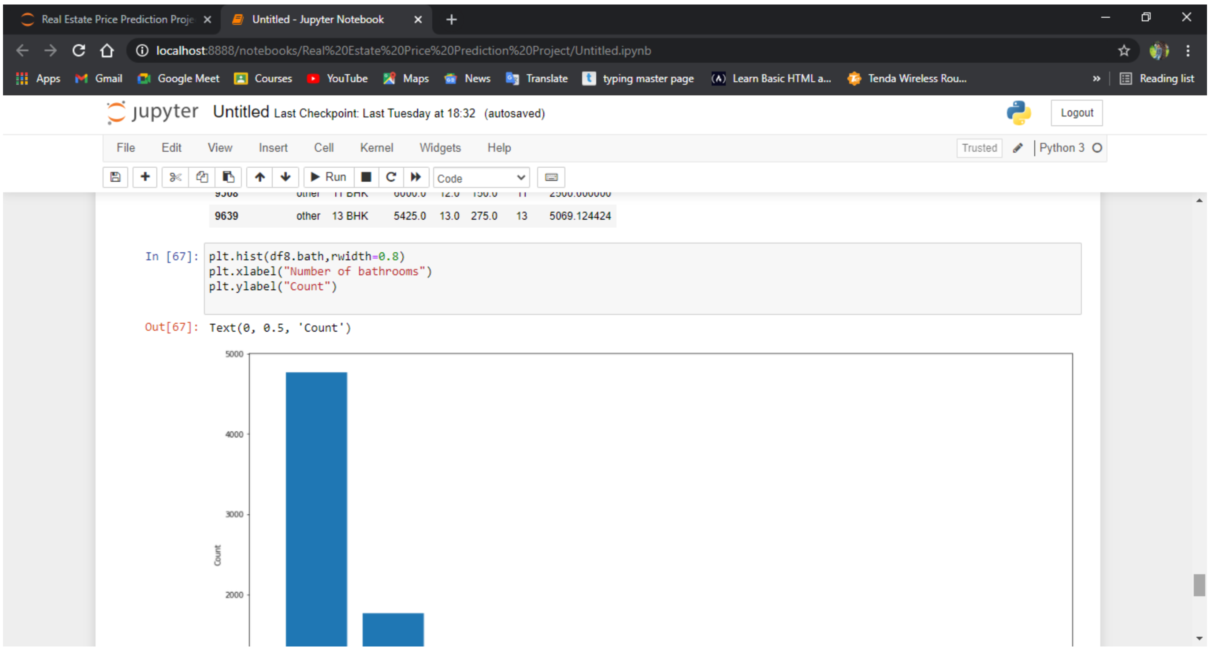

Real Estate Price Prediction Proje
Untitled - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/Untitled.ipynb
jupyter
Untitled Last Checkpoint: Last Tuesday at 18:32 (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
3500
3000
2500
2000
1500
1000
500
0
Count
5000
10000
15000
20000
25000
Price Per Square Feet


Real Estate Price Prediction Proje
Untitled - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/Untitled.ipynb
jupyter
Untitled Last Checkpoint: Last Tuesday at 18:32 (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

```
In [65]: df8.bath.unique()
Out[65]: array([ 4.,  3.,  2.,  5.,  8.,  1.,  6.,  7.,  9., 12., 16., 13.])

In [66]: df8[df8.bath>10]
Out[66]:
```

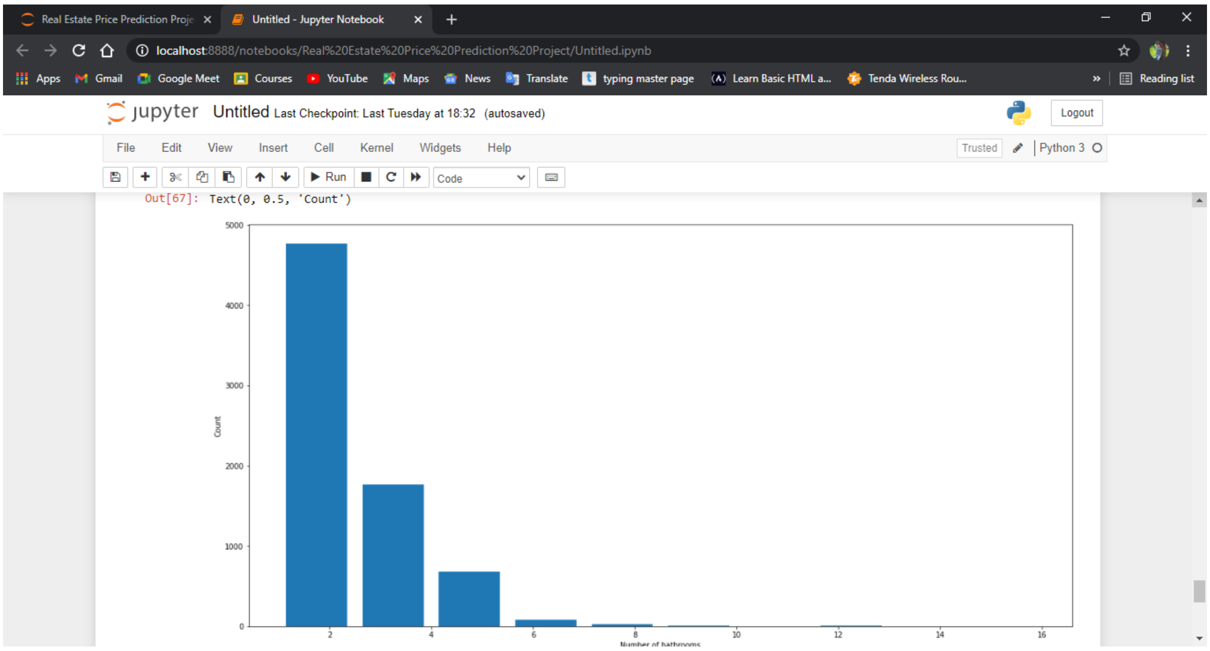
	location	size	total_sqft	bath	price	bhk	price_per_sqft
5277	Neeladri Nagar	10 BHK	4000.0	12.0	160.0	10	4000.000000
8486	other	10 BHK	12000.0	12.0	525.0	10	4375.000000
8575	other	16 BHK	10000.0	16.0	550.0	16	5500.000000
9308	other	11 BHK	6000.0	12.0	150.0	11	2500.000000
9639	other	13 BHK	5425.0	13.0	275.0	13	5069.124424

```
In [67]: plt.hist(df8.bath,rwidth=0.8)
         plt.xlabel("Number of bathrooms")
         plt.ylabel("Count")

Out[67]: Text(0, 0.5, 'Count')
```

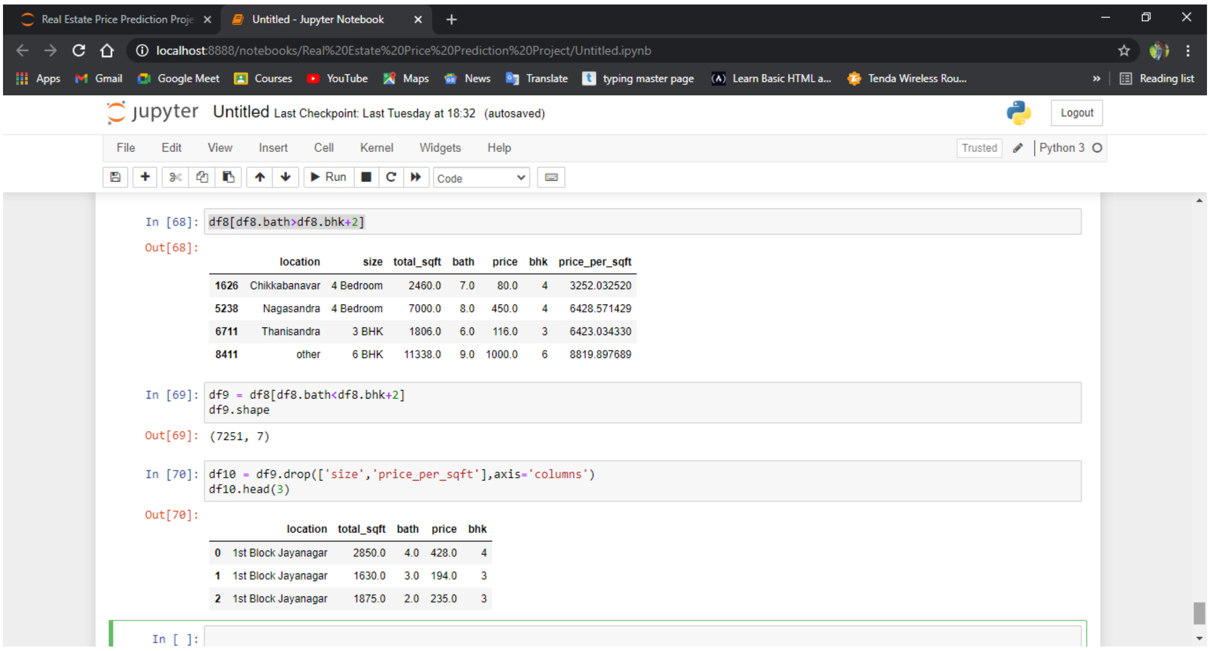


```
9508     other  11 BHK  6000.0  12.0  150.0  11  2500.000000
9639     other  13 BHK  5425.0  13.0  275.0  13  5069.124424
```

```
In [67]: plt.hist(df8.bath,rwidth=0.8)
         plt.xlabel("Number of bathrooms")
         plt.ylabel("Count")

Out[67]: Text(0, 0.5, 'Count')
```



```
Out[67]: Text(0, 0.5, 'Count')
```



Real Estate Price Prediction Proje
Untitled - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/Untitled.ipynb
Apps
Gmail
Google Meet
Courses
YouTube
Maps
News
Translate
typing master page
Learn Basic HTML a...
Tenda Wireless Rou...
Reading list
jupyter
Untitled Last Checkpoint: Last Tuesday at 18:32 (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

```
In [68]: df8[df8.bath>df8.bhk+2]
Out[68]:
```

	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

```
In [69]: df9 = df8[df8.bath<df8.bhk+2]
         df9.shape
Out[69]: (7251, 7)

In [70]: df10 = df9.drop(['size','price_per_sqft'],axis='columns')
         df10.head(3)
Out[70]:
```

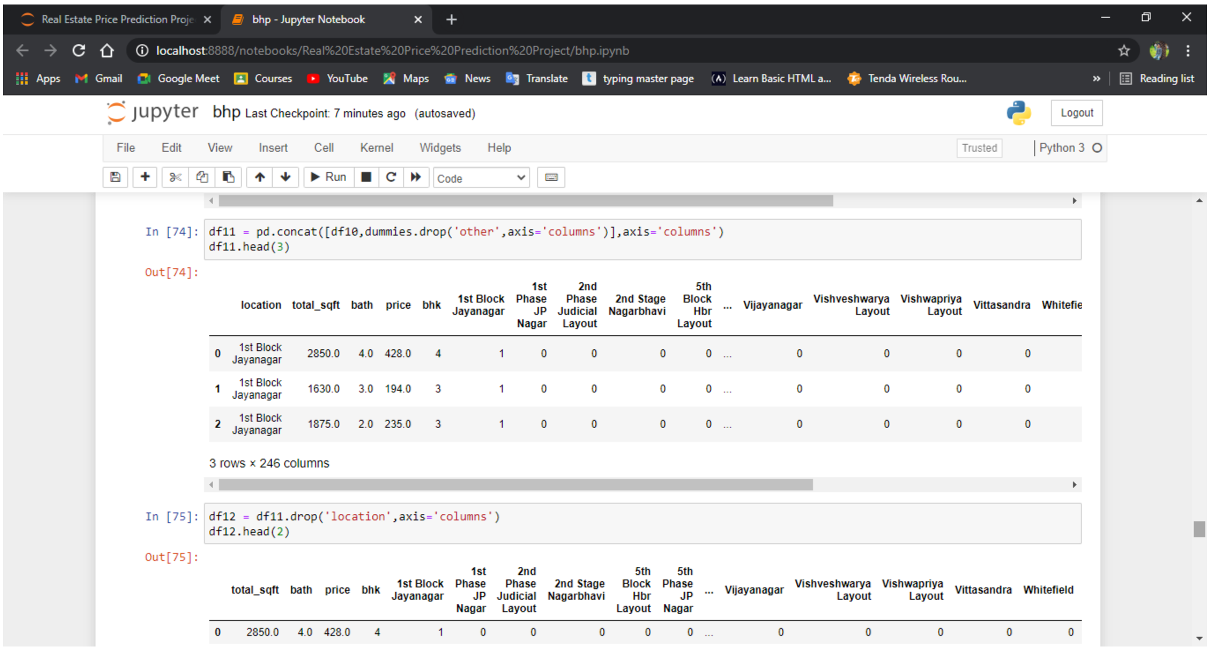
	location	total_sqft	bath	price	bhk
0	1st Block Jayanagar	2850.0	4.0	428.0	4
1	1st Block Jayanagar	1630.0	3.0	194.0	3
2	1st Block Jayanagar	1875.0	2.0	235.0	3

```
In [ ]:
```

2.4Dimensionality reduction:-

Dimensionality reduction, or dimension reduction, is the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data, ideally close to its intrinsic dimension. Working in high-dimensional spaces can be undesirable for many reasons; raw data are often sparse as a consequence of the curse of dimensionality, and analyzing the data is usually computationally intractable. Dimensionality reduction is common in fields that deal with large numbers of observations and/or large numbers of variables, such as signal processing, speech recognition, neuroinformatics, and bioinformatics.

<u>ScreenShot:-</u>



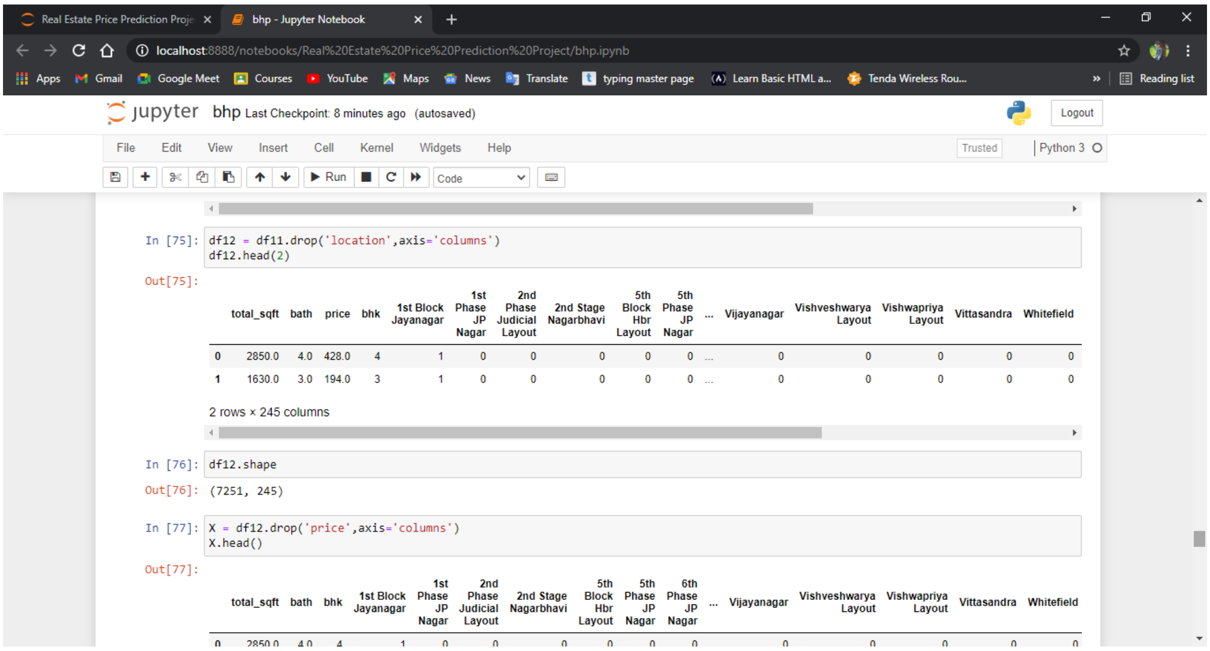
```
In [74]: df11 = pd.concat([df10,dummies.drop('other',axis='columns')],axis='columns')
         df11.head(3)
Out[74]:
```

	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	Whitefie
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 rows × 246 columns

```
In [75]: df12 = df11.drop('location',axis='columns')
         df12.head(2)
Out[75]:
```

	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


Real Estate Price Prediction Proje
bhp - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/bhp.ipynb
Apps Gmail Google Meet Courses YouTube Maps News Translate typing master page Learn Basic HTML a... Tenda Wireless Rou... Reading list
jupyter bhp Last Checkpoint: 8 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

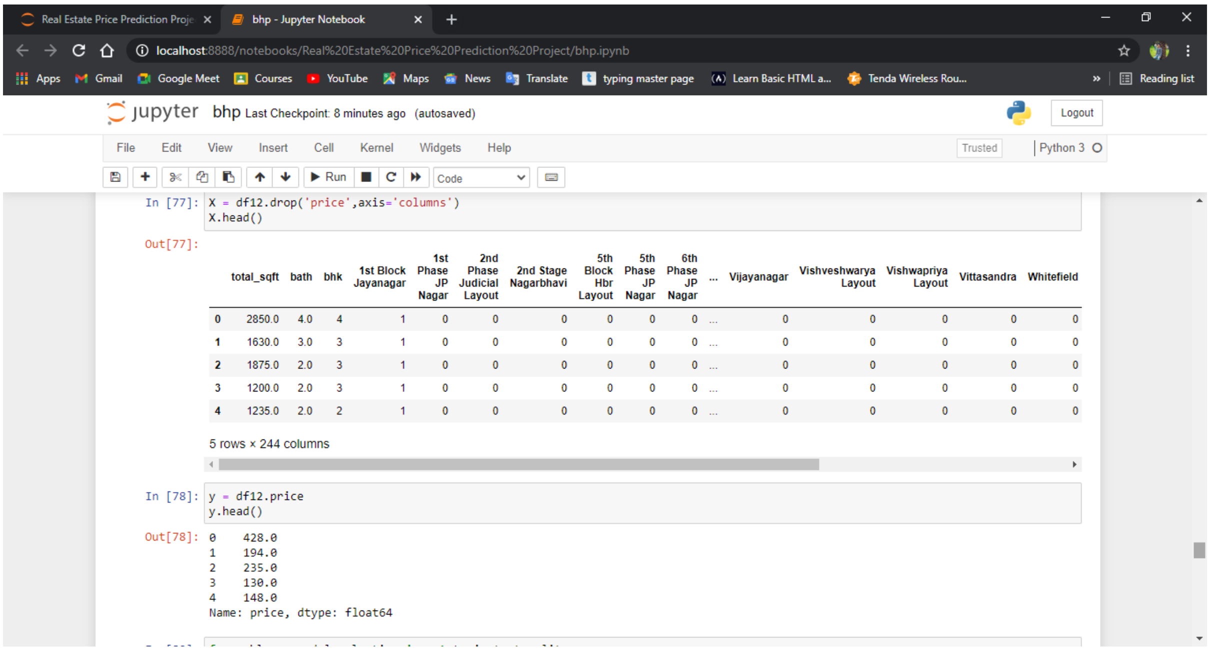
```
In [75]: df12 = df11.drop('location',axis='columns')
         df12.head(2)
Out[75]:
```

	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 × 245 columns

```
In [76]: df12.shape
Out[76]: (7251, 245)

In [77]: X = df12.drop('price',axis='columns')
         X.head()
Out[77]:
```

	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
0	2850.0	4.0	4	1	0	0	0	0	0	0		0	0	0	0	0


Real Estate Price Prediction Proje
bhp - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/bhp.ipynb
Apps Gmail Google Meet Courses YouTube Maps News Translate typing master page Learn Basic HTML a... Tenda Wireless Rou... Reading list
jupyter bhp Last Checkpoint: 8 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
In [77]: X = df12.drop('price',axis='columns')
         X.head()
Out[77]:
```

	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
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

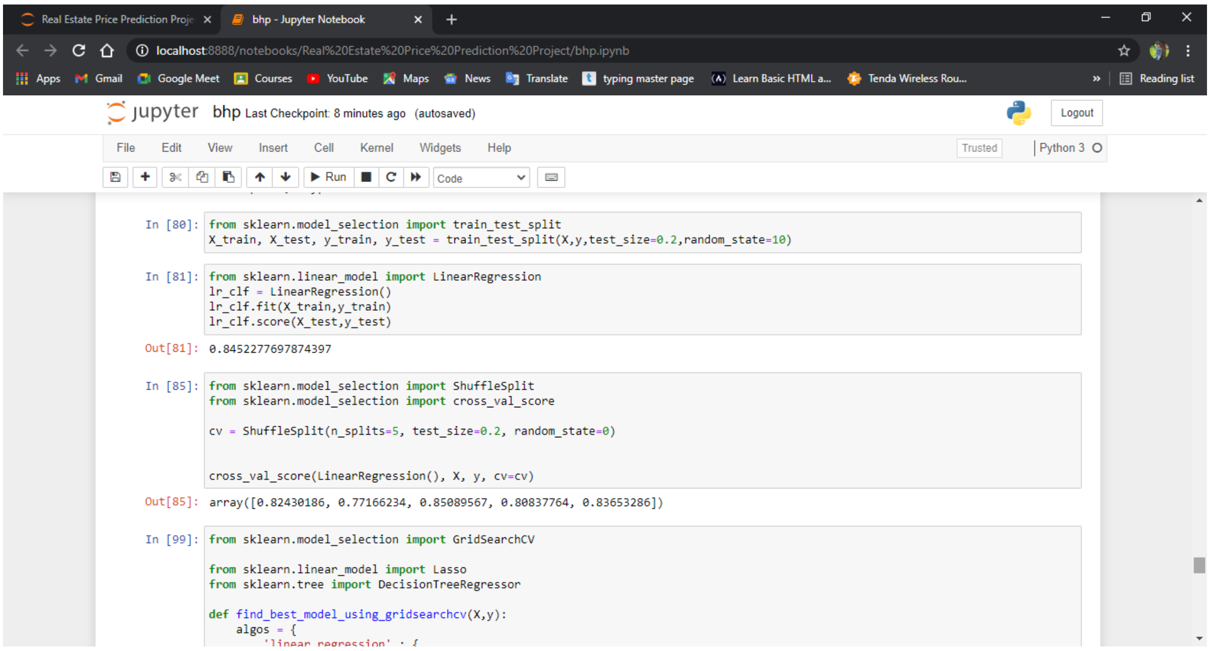
5 rows × 244 columns

```
In [78]: y = df12.price
         y.head()
Out[78]: 0    428.0
         1    194.0
         2    235.0
         3    130.0
         4    148.0
         Name: price, dtype: float64
```

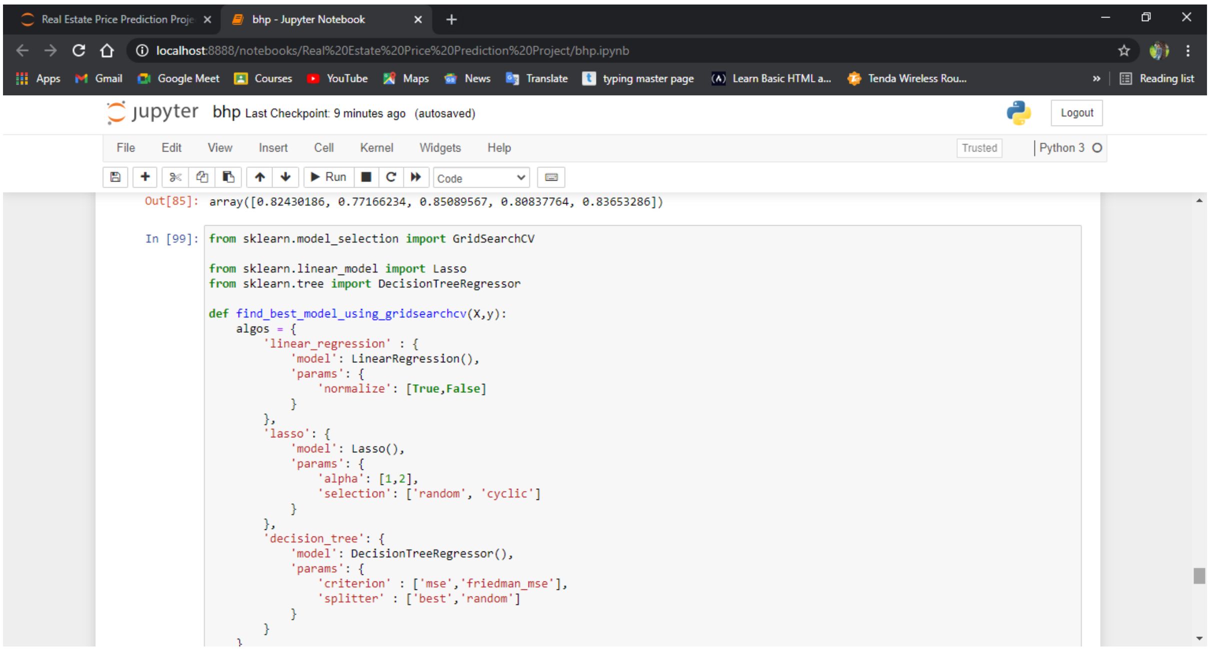
2.5 Grid Search Cv:-

GridSearchCV is a method to search the candidate best parameters exhaustively from the grid of given parameters. Target estimator (model) and parameters for search need to be provided for this cross-validation search method.

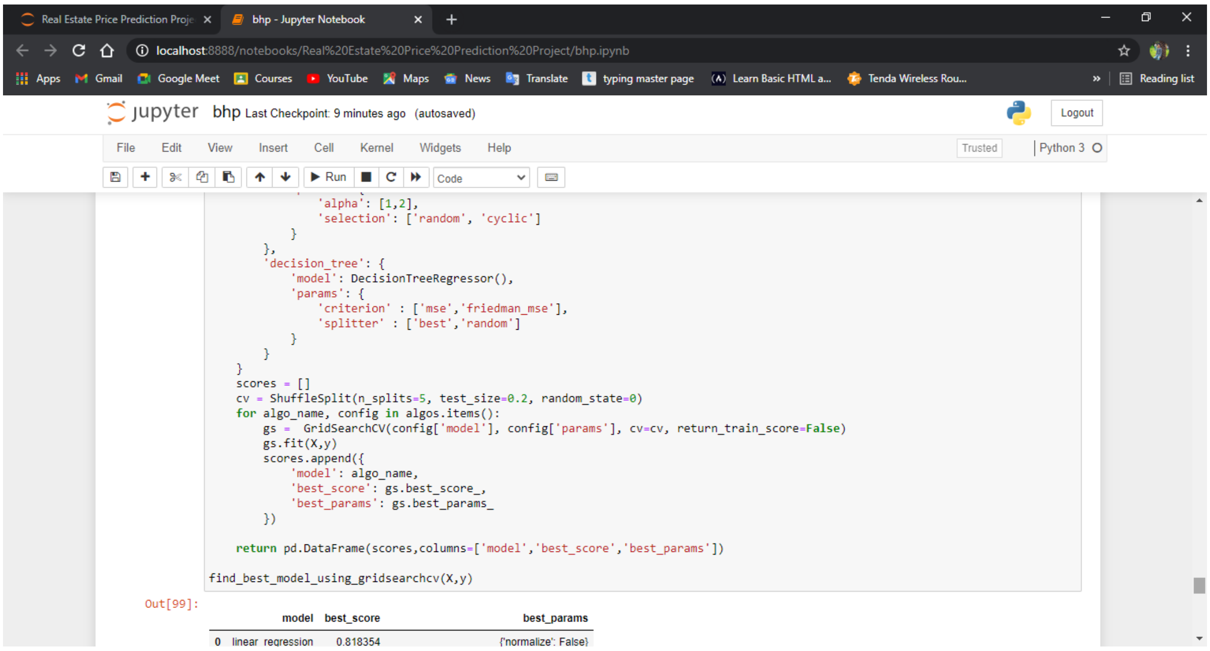
ScreenShot:-


Real Estate Price Prediction Proje
bhp - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/bhp.ipynb
jupyter bhp Last Checkpoint: 8 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
In [80]: 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)

In [81]: from sklearn.linear_model import LinearRegression
         lr_clf = LinearRegression()
         lr_clf.fit(X_train,y_train)
         lr_clf.score(X_test,y_test)

Out[81]: 0.8452277697874397

In [85]: from sklearn.model_selection import ShuffleSplit
         from sklearn.model_selection import cross_val_score

         cv = ShuffleSplit(n_splits=5, test_size=0.2, random_state=0)


         cross_val_score(LinearRegression(), X, y, cv=cv)

Out[85]: array([0.82430186, 0.77166234, 0.85089567, 0.80837764, 0.83653286])

In [99]: from sklearn.model_selection import GridSearchCV

         from sklearn.linear_model import Lasso
         from sklearn.tree import DecisionTreeRegressor

         def find_best_model_using_gridsearchcv(X,y):
             algos = {
                 'linear_regression' : {
```


Real Estate Price Prediction Proje
bhp - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/bhp.ipynb
jupyter bhp Last Checkpoint: 9 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

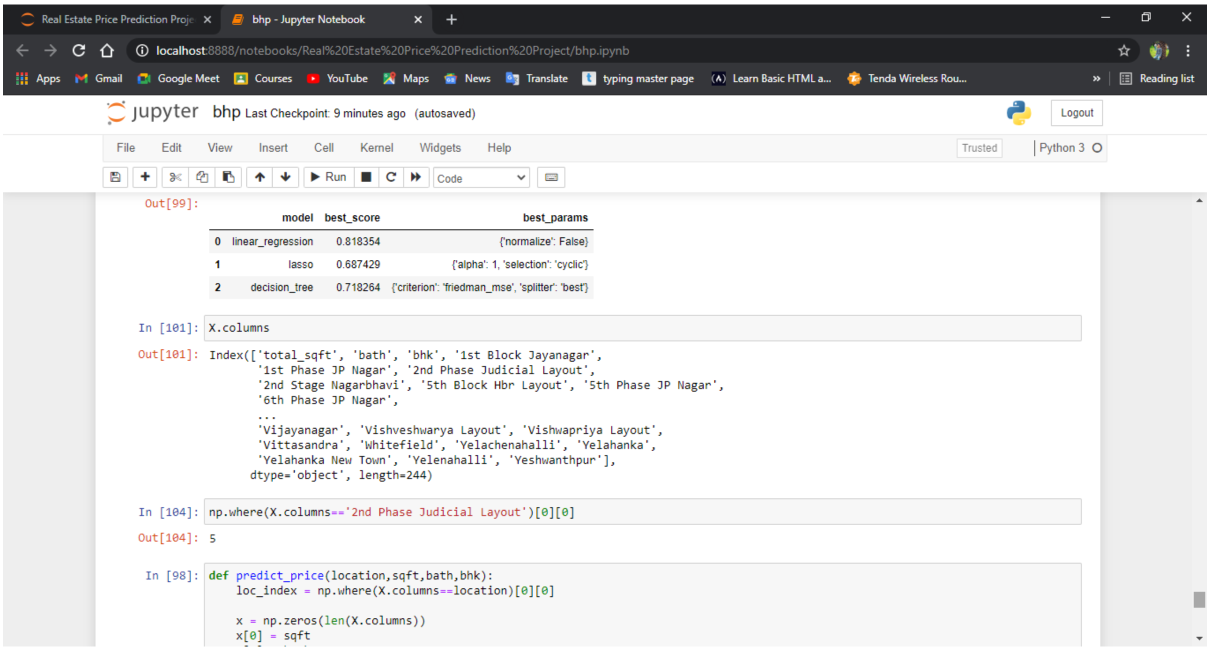
```
Out[85]: array([0.82430186, 0.77166234, 0.85089567, 0.80837764, 0.83653286])

In [99]: from sklearn.model_selection import GridSearchCV

         from sklearn.linear_model import Lasso
         from sklearn.tree import DecisionTreeRegressor

         def find_best_model_using_gridsearchcv(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']
                     }
                 }
             }
```


Real Estate Price Prediction Proje
bhp - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/bhp.ipynb
Apps Gmail Google Meet Courses YouTube Maps News Translate typing master page Learn Basic HTML a... Tenda Wireless Rou... Reading list
jupyter bhp Last Checkpoint: 9 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3
Run Code

```
                '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_gridsearchcv(X,y)
```

Out[99]:

	model	best_score	best_params
0	linear regression	0.818354	{'normalize': False}


Real Estate Price Prediction Proje
bhp - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/bhp.ipynb
Apps Gmail Google Meet Courses YouTube Maps News Translate typing master page Learn Basic HTML a... Tenda Wireless Rou... Reading list
jupyter bhp Last Checkpoint: 9 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3
Run Code

Out[99]:

	model	best_score	best_params
0	linear_regression	0.818354	{'normalize': False}
1	lasso	0.687429	{'alpha': 1, 'selection': 'cyclic'}
2	decision_tree	0.718264	{'criterion': 'friedman_mse', 'splitter': 'best'}

```
In [101]: X.columns

Out[101]: 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)

In [104]: np.where(X.columns=='2nd Phase Judicial Layout')[0][0]

Out[104]: 5

In [98]: 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
```

K-Fold Cross Validation:-

Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.

The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into.

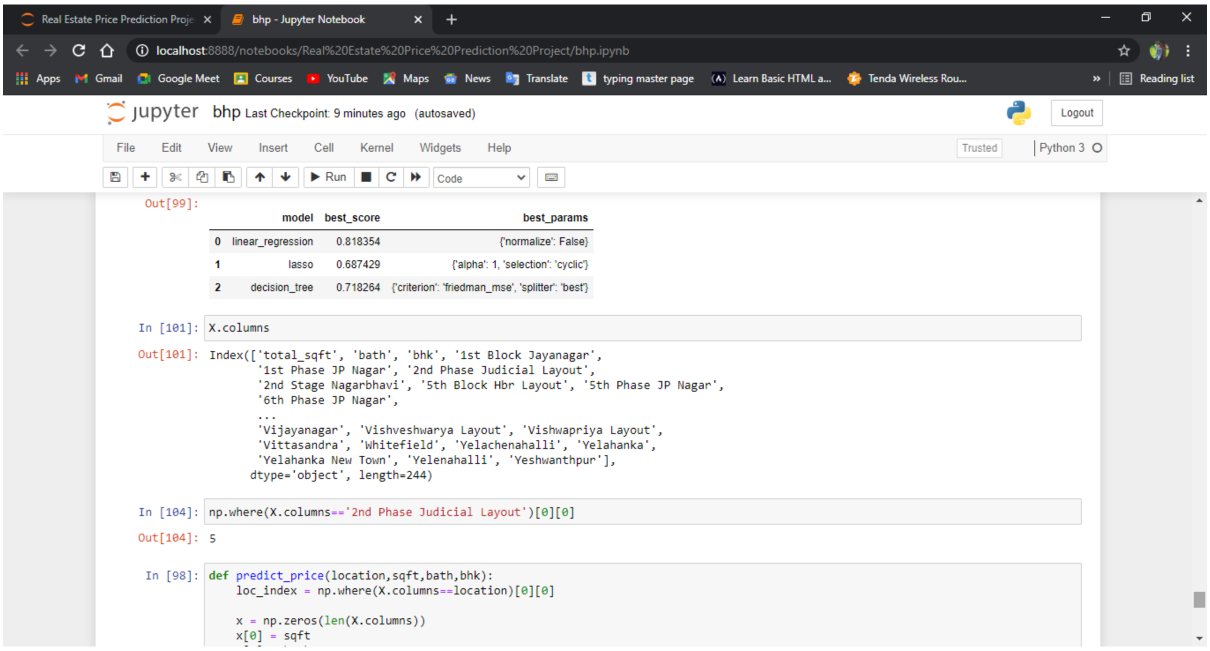
As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation.

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data.

That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.

It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split.

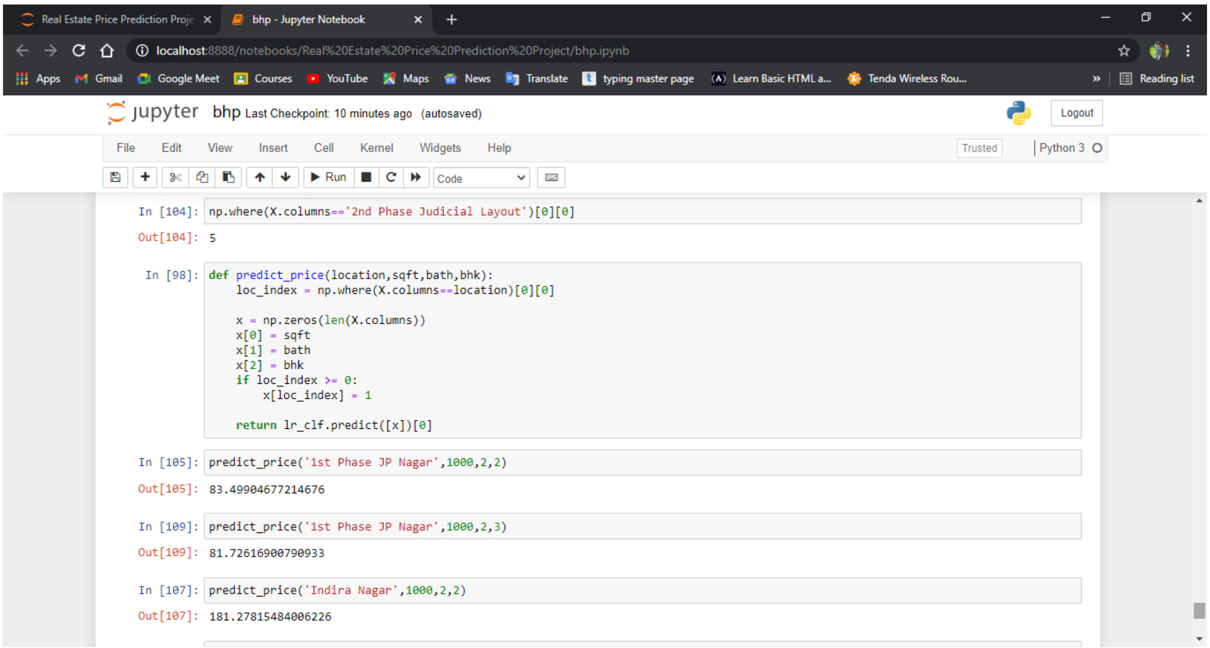
ScreenShot:-


Real Estate Price Prediction Proje
bhp - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/bhp.ipynb
jupyter bhp Last Checkpoint: 9 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

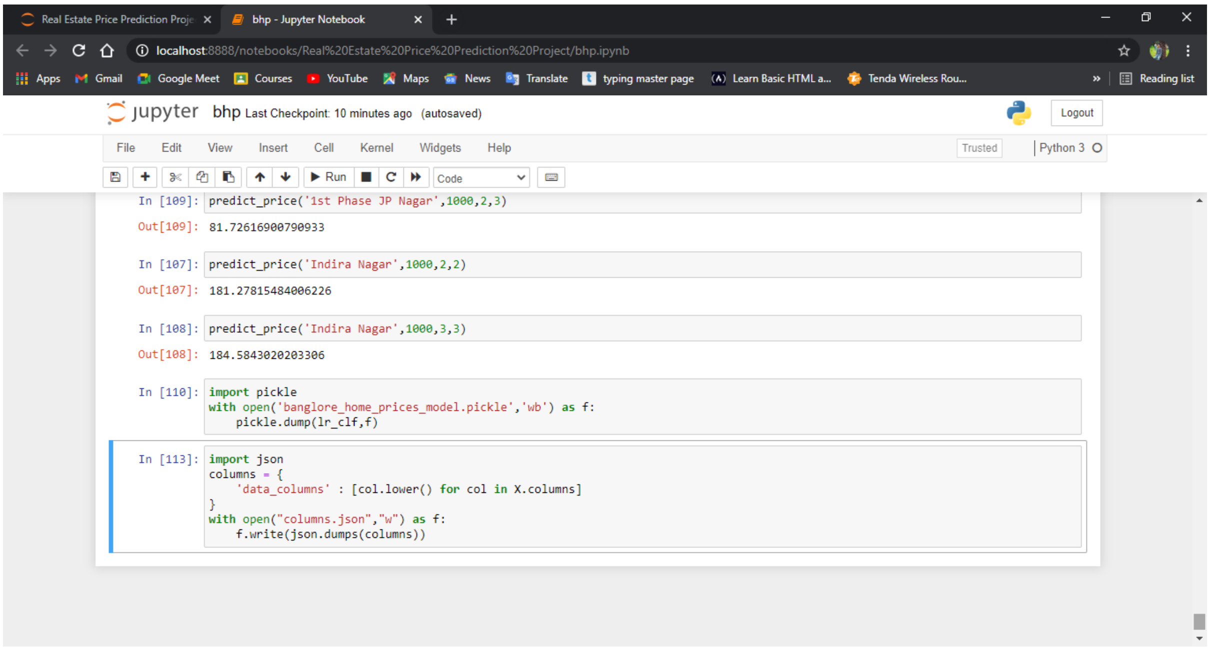
Out[99]:

	model	best_score	best_params
0	linear_regression	0.818354	{'normalize': False}
1	lasso	0.687429	{'alpha': 1, 'selection': 'cyclic'}
2	decision_tree	0.718264	{'criterion': 'friedman_mse', 'splitter': 'best'}

```
In [101]: X.columns

Out[101]: 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)

In [104]: np.where(X.columns=='2nd Phase Judicial Layout')[0][0]

Out[104]: 5

In [98]: 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
```


Real Estate Price Prediction Proje
bhp - Jupyter Notebook
localhost:8888/notebooks/Real%20Estate%20Price%20Prediction%20Project/bhp.ipynb
jupyter bhp Last Checkpoint: 10 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
In [104]: np.where(X.columns=='2nd Phase Judicial Layout')[0][0]

Out[104]: 5

In [98]: 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]

In [105]: predict_price('1st Phase JP Nagar',1000,2,2)

Out[105]: 83.49904677214676

In [109]: predict_price('1st Phase JP Nagar',1000,2,3)

Out[109]: 81.72616900790933

In [107]: predict_price('Indira Nagar',1000,2,2)

Out[107]: 181.27815484006226
```



```
In [109]: predict_price('1st Phase JP Nagar',1000,2,3)

Out[109]: 81.72616900790933

In [107]: predict_price('Indira Nagar',1000,2,2)

Out[107]: 181.27815484006226

In [108]: predict_price('Indira Nagar',1000,3,3)

Out[108]: 184.5843020203306

In [110]: import pickle
          with open('banglore_home_prices_model.pickle','wb') as f:
              pickle.dump(lr_clf,f)

In [113]: import json
          columns = {
              'data_columns' : [col.lower() for col in X.columns]
          }
          with open("columns.json","w") as f:
              f.write(json.dumps(columns))
```

3.LANGUAGE & MODELS USED:-

Technology and tools wise this project covers:-

1) Python
2) Numpy and Pandas for data cleaning
3) Matplotlib for data visualization
4) Sklearn for model building
5) Jupyter notebook, visual studio code and pycharm as IDE
6) Python flask for http server
7) HTML/CSS/Javascript for UI

Python

Python is widely used in scientific and numeric computing:

- **SciPy is a collection of packages for mathematics, science, and engineering.**
- **Pandas is a data analysis and modelling library.**
- **IPython is a powerful interactive shell that features easy editing and recording of a work session, and supports visualizations and parallel computing.**
- **The Software Carpentry Course teaches basic skills for scientific**

Libraries Used for this Project include –

- Pandas
- NumPy
- Matplotlib
- Seaborn
- Scikit Learn

Juypter Notebook:-

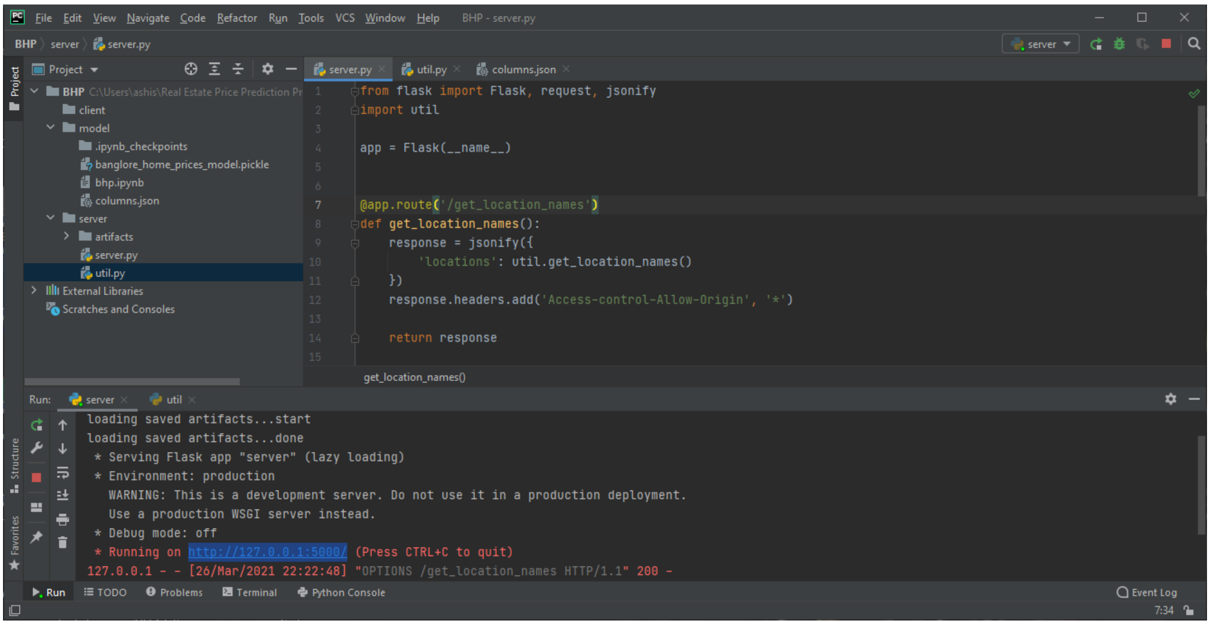
Project Jupyter is a project and community whose goal is to "develop open-source software, open-standards, and services for interactive computing across dozens of programming languages". It was spun off from IPython in 2014 by Fernando Pérez. Project Jupyter's name is a reference to the three core programming languages supported by Jupyter, which are Julia, Python and R, and also a homage to Galileo's notebooks recording the discovery of the moons of Jupiter. Project Jupyter has developed and supported the interactive computing products Jupyter Notebook, JupyterHub, and JupyterLab.

Python Flask Server:-

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

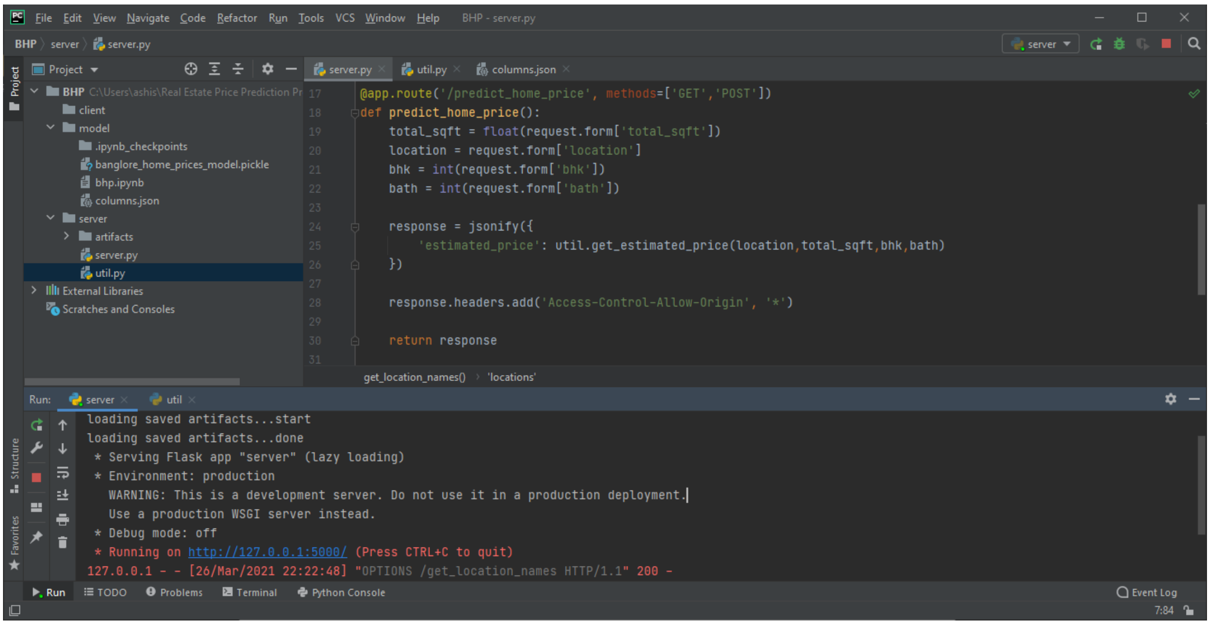
Python Flask Server:-

Server.py


File Edit View Navigate Code Refactor Run Tools VCS Window Help BHP - server.py
BHP server server.py
Project
server.py util.py columns.json
BHP C:\Users\ashis\Real Estate Price Prediction Pr
client
model
.ipynb_checkpoints
banglore_home_prices_model.pickle
bhp.ipynb
columns.json
server
artifacts
server.py
util.py
External Libraries
Scratches and Consoles
get_location_names()
Run: server util
Run TODO Problems Terminal Python Console Event Log
7:34

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

app = Flask(__name__)


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

    return response

```

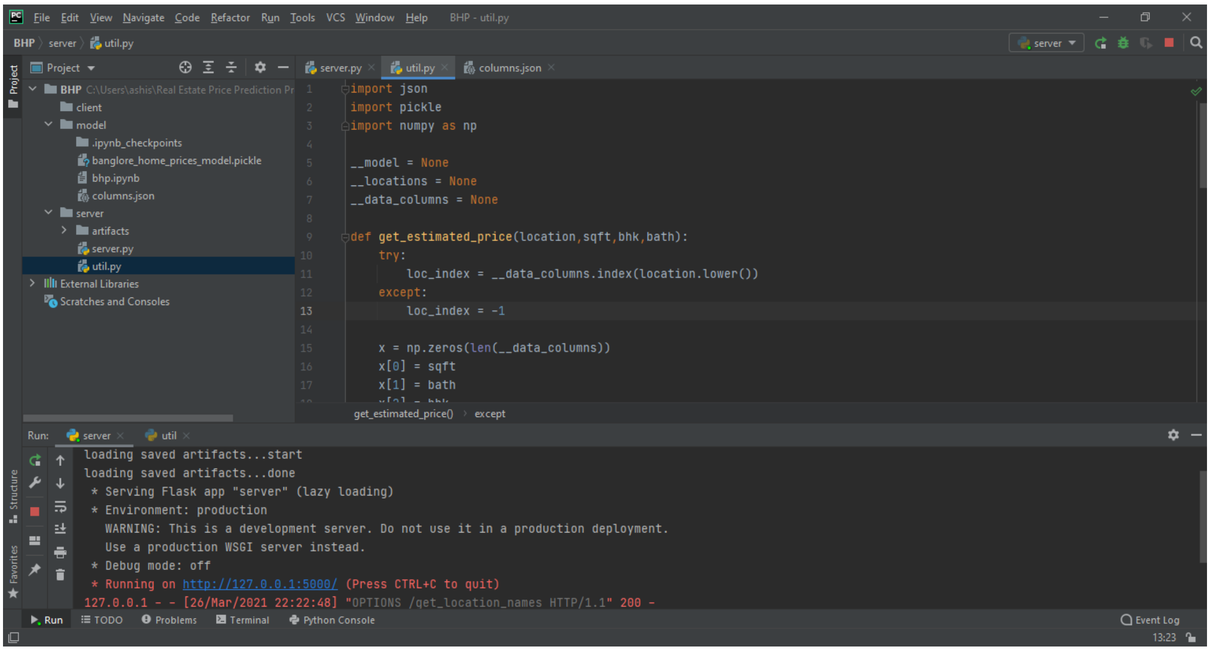
```
loading saved artifacts...start
loading saved artifacts...done
 * Serving Flask app "server" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [26/Mar/2021 22:22:48] "OPTIONS /get_location_names HTTP/1.1" 200 -
```


File Edit View Navigate Code Refactor Run Tools VCS Window Help BHP - server.py
BHP server server.py
Project
server.py util.py columns.json
BHP C:\Users\ashis\Real Estate Price Prediction Pr
client
model
.ipynb_checkpoints
banglore_home_prices_model.pickle
bhp.ipynb
columns.json
server
artifacts
server.py
util.py
External Libraries
Scratches and Consoles
get_location_names() 'locations'
Run: server util
Run TODO Problems Terminal Python Console Event Log
7:84

```python
@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

```

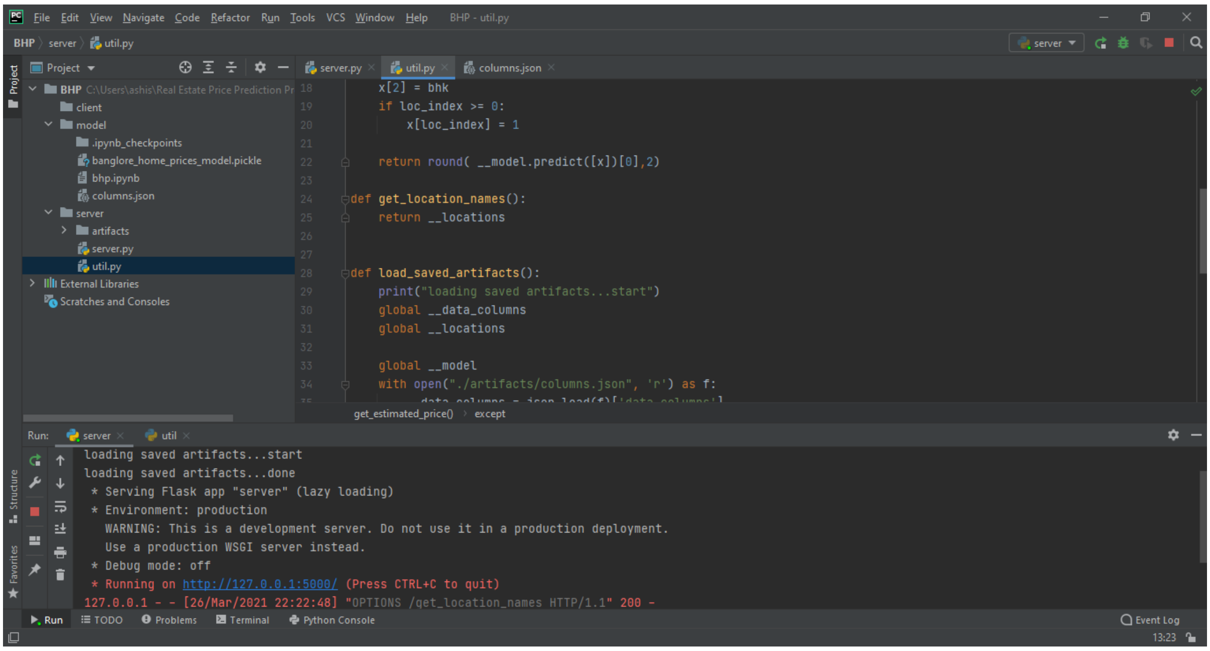
```
loading saved artifacts...start
loading saved artifacts...done
 * Serving Flask app "server" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [26/Mar/2021 22:22:48] "OPTIONS /get_location_names HTTP/1.1" 200 -
```

util.py


File Edit View Navigate Code Refactor Run Tools VCS Window Help BHP - util.py
BHP server util.py
Project
BHP C:\Users\ashis\Real Estate Price Prediction Pr
client
model
.ipynb_checkpoints
banglore_home_prices_model.pickle
bhp.ipynb
columns.json
server
artifacts
server.py
util.py
External Libraries
Scratches and Consoles
server.py
util.py
columns.json
get_estimated_price() › except
Run: server util
Run TODO Problems Terminal Python Console
Event Log
13:23

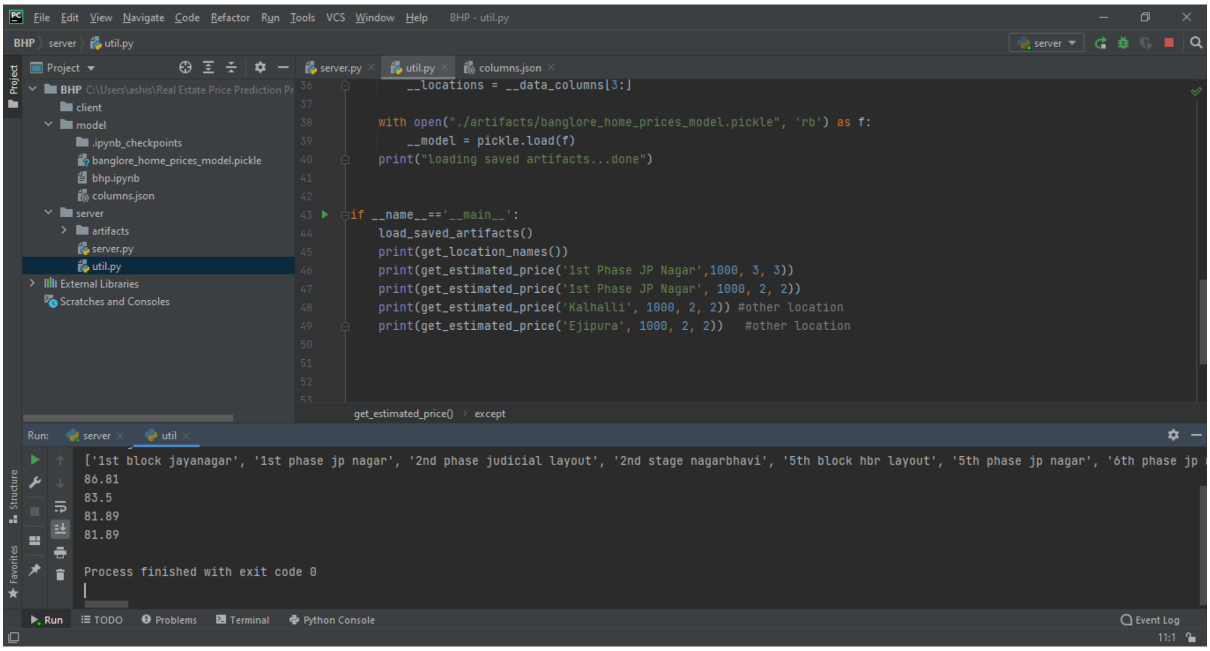
```python
import json
import pickle
import numpy as np

__model = None
__locations = None
__data_columns = 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
```

```
loading saved artifacts...start
loading saved artifacts...done
 * Serving Flask app "server" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [26/Mar/2021 22:22:48] "OPTIONS /get_location_names HTTP/1.1" 200 -
```


File Edit View Navigate Code Refactor Run Tools VCS Window Help BHP - util.py
BHP server util.py
Project
BHP C:\Users\ashis\Real Estate Price Prediction Pr
client
model
.ipynb_checkpoints
banglore_home_prices_model.pickle
bhp.ipynb
columns.json
server
artifacts
server.py
util.py
External Libraries
Scratches and Consoles
server.py
util.py
columns.json
get_estimated_price() › except
Run: server util
Run TODO Problems Terminal Python Console
Event Log
13:23

```python
    x[2] = bhk
    if loc_index >= 0:
        x[loc_index] = 1

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

def get_location_names():
    return __locations


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

    global __model
    with open("./artifacts/columns.json", 'r') as f:
```

```
loading saved artifacts...start
loading saved artifacts...done
 * Serving Flask app "server" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [26/Mar/2021 22:22:48] "OPTIONS /get_location_names HTTP/1.1" 200 -
```



```
        __locations = __data_columns[3:]

    with open("./artifacts/banglore_home_prices_model.pickle", 'rb') as f:
        __model = pickle.load(f)
    print("loading saved artifacts...done")


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))
    print(get_estimated_price('Kalhalli', 1000, 2, 2)) #other location
    print(get_estimated_price('Ejipura', 1000, 2, 2))   #other location
```

```
['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
86.81
83.5
81.89
81.89

Process finished with exit code 0
```

Python Flask Server Code:-

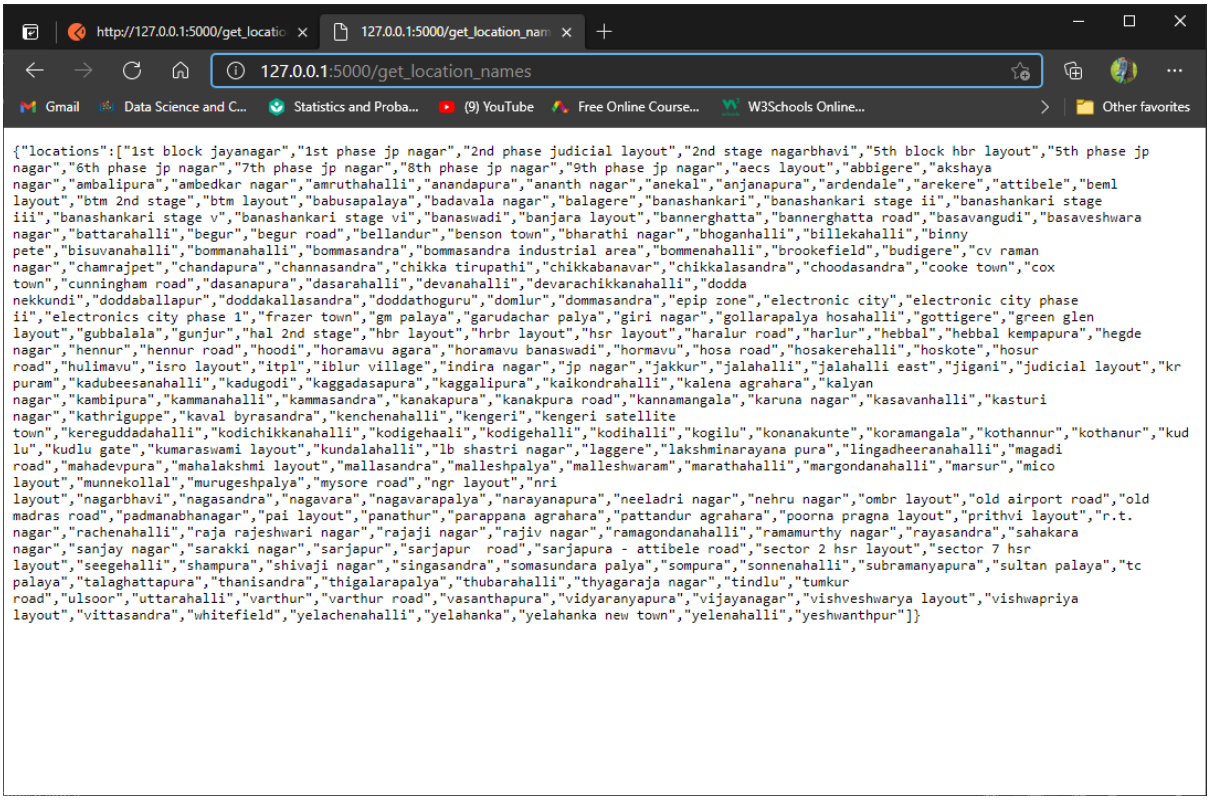
Server.py code:-

```
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()
```

util.py :-

```
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("./artifacts/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('./artifacts/banglore_home_prices_model.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))
    print(get_estimated_price('Kalhalli', 1000, 2, 2)) # other location
    print(get_estimated_price('Ejipura', 1000, 2, 2))  # other location
```

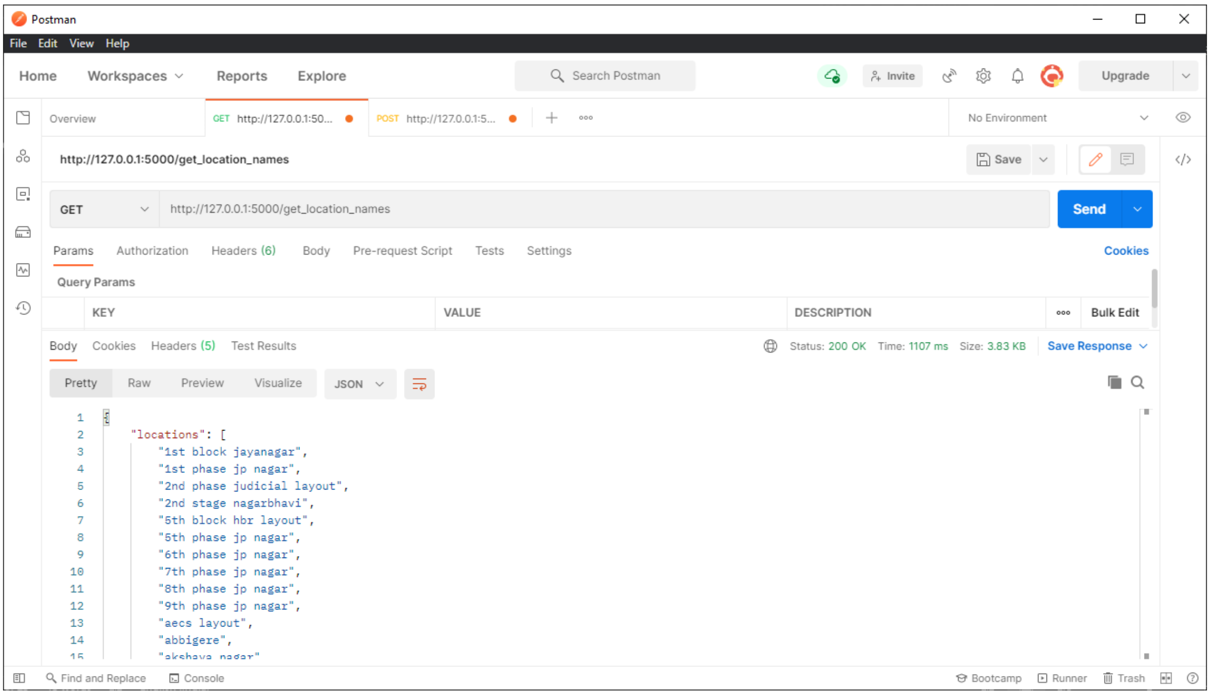
Server.py output:-



```
{"locations":["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","aecs layout","abbigere","akshaya
nagar","ambalipura","ambedkar nagar","amruthahalli","anandapura","ananth nagar","anekal","anjanapura","ardendale","arekere","attibele","beml
layout","btm 2nd stage","btm layout","babusapalaya","badavala nagar","balagere","banashankari","banashankari stage ii","banashankari stage
iii","banashankari stage v","banashankari stage vi","banaswadi","banjara layout","bannerghatta","bannerghatta road","basavangudi","basaveshwara
nagar","battarahalli","begur","begur road","bellandur","benson town","bharathi nagar","bhoganhalli","billekahalli","binny
pete","bisuvanahalli","bommanahalli","bommasandra","bommasandra industrial area","bommenahalli","brookefield","budigere","cv raman
nagar","chamrajpet","chandapura","channasandra","chikka tirupathi","chikkabanavar","chikkalasandra","choodasandra","cooke town","cox
town","cunningham road","dasanapura","dasarahalli","devanahalli","devarachikkanahalli","dodda
nekkundi","doddaballapur","doddakallasandra","doddathoguru","domlur","dommasandra","epip zone","electronic city","electronic city phase
ii","electronics city phase 1","frazer town","gm palaya","garudachar palya","giri nagar","gollarapalya hosahalli","gottigere","green glen
layout","gubbalala","gunjur","hal 2nd stage","hbr layout","hrbr layout","hsr layout","haralur road","harlur","hebbal","hebbal kempapura","hegde
nagar","hennur","hennur road","hoodi","horamavu agara","horamavu banaswadi","hormavu","hosa road","hosakerehalli","hoskote","hosur
road","hulimavu","isro layout","itpl","iblur village","indira nagar","jp nagar","jakkur","jalahalli","jalahalli east","jigani","judicial layout","kr
puram","kadubeesanahalli","kadugodi","kaggadasapura","kaggalipura","kaikondrahalli","kalena agrahara","kalyan
nagar","kambipura","kammanahalli","kammasandra","kanakapura","kanakpura road","kannamangala","karuna nagar","kasavanhalli","kasturi
nagar","kathriguppe","kaval byrasandra","kenchenahalli","kengeri","kengeri satellite
town","kereguddadahalli","kodichikkanahalli","kodigehaali","kodigehalli","kodihalli","kogilu","konanakunte","koramangala","kothannur","kothanur","kud
lu","kudlu gate","kumaraswami layout","kundalahalli","lb shastri nagar","laggere","lakshminarayana pura","lingadheeranahalli","magadi
road","mahadevpura","mahalakshmi layout","mallasandra","malleshpalya","malleshwaram","marathahalli","margondanahalli","marsur","mico
layout","munnekollal","murugeshpalya","mysore road","ngr layout","nri
layout","nagarbhavi","nagasandra","nagavara","nagavarapalya","narayanapura","neeladri nagar","nehru nagar","ombr layout","old airport road","old
madras road","padmanabhanagar","pai layout","panathur","parappana agrahara","pattandur agrahara","poorna pragna layout","prithvi layout","r.t.
nagar","rachenahalli","raja rajeshwari nagar","rajaji nagar","rajiv nagar","ramagondanahalli","ramamurthy nagar","rayasandra","sahakara
nagar","sanjay nagar","sarakki nagar","sarjapur","sarjapur  road","sarjapura - attibele road","sector 2 hsr layout","sector 7 hsr
layout","seegehalli","shampura","shivaji nagar","singasandra","somasundara palya","sompura","sonnenahalli","subramanyapura","sultan palaya","tc
palaya","talaghattapura","thanisandra","thigalarapalya","thubarahalli","thyagaraja nagar","tindlu","tumkur
road","ulsoor","uttarahalli","varthur","varthur road","vasanthapura","vidyaranyapura","vijayanagar","vishveshwarya layout","vishwapriya
layout","vittasandra","whitefield","yelachenahalli","yelahanka","yelahanka new town","yelenahalli","yeshwanthpur"]}
```

Using PostMan Application for GET & POST:-

Postman is a scalable API testing tool that quickly integrates into CI/CD pipeline. It started in 2012 as a side project by Abhinav Asthana to simplify API workflow in testing and development. API stands for Application Programming Interface which allows software applications to communicate with each other via API calls.


Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Invite
Upgrade
Overview
No Environment
Save
Send
Params Authorization Headers (6) Body Pre-request Script Tests Settings
Cookies
Query Params
Bulk Edit
Body Cookies Headers (5) Test Results
Save Response
Pretty Raw Preview Visualize JSON
Find and Replace Console
Bootcamp Runner Trash

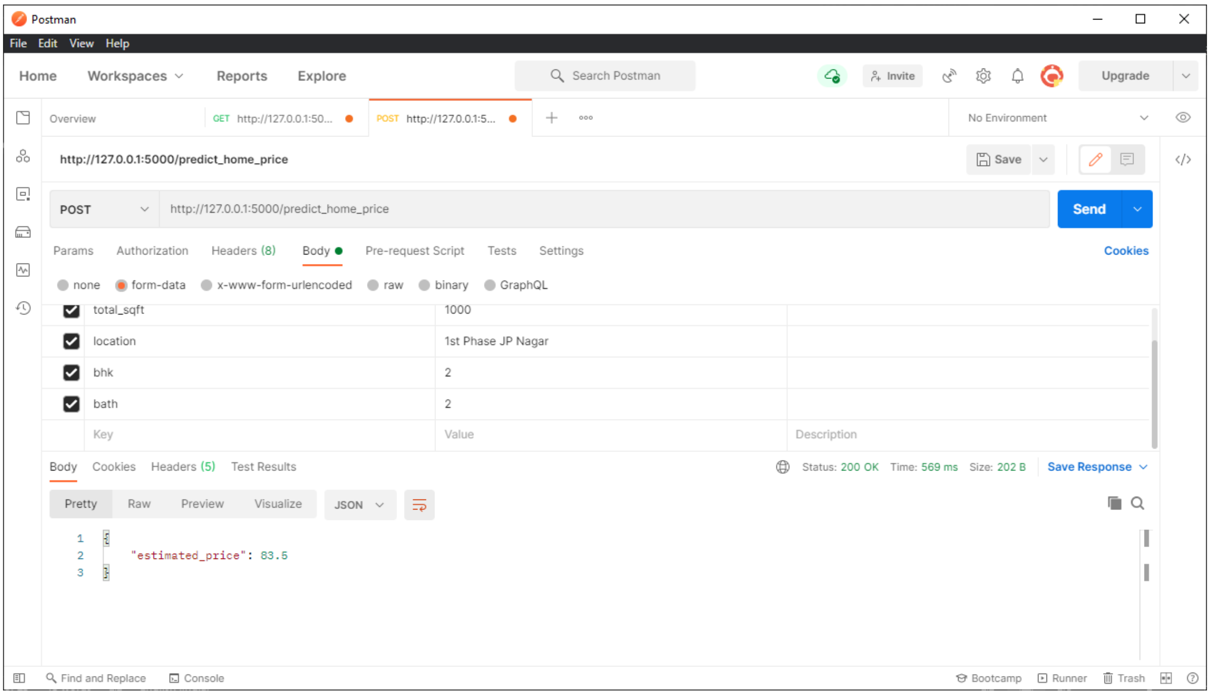
http://127.0.0.1:5000/get_location_names

GET http://127.0.0.1:5000/get_location_names

KEY	VALUE	DESCRIPTION

Status: 200 OK Time: 1107 ms Size: 3.83 KB

```
{
    "locations": [
        "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",
        "aecs layout",
        "abbigere",
        "akshaya nagar"
```


Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Invite
Upgrade
Overview
No Environment
Save
Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings
Cookies
Body Cookies Headers (5) Test Results
Save Response
Pretty Raw Preview Visualize JSON
Find and Replace Console
Bootcamp Runner Trash

http://127.0.0.1:5000/predict_home_price

POST http://127.0.0.1:5000/predict_home_price

○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	Key	Value	Description
☑	total_sqft	1000	
☑	location	1st Phase JP Nagar	
☑	bhk	2	
☑	bath	2	

Status: 200 OK Time: 569 ms Size: 202 B

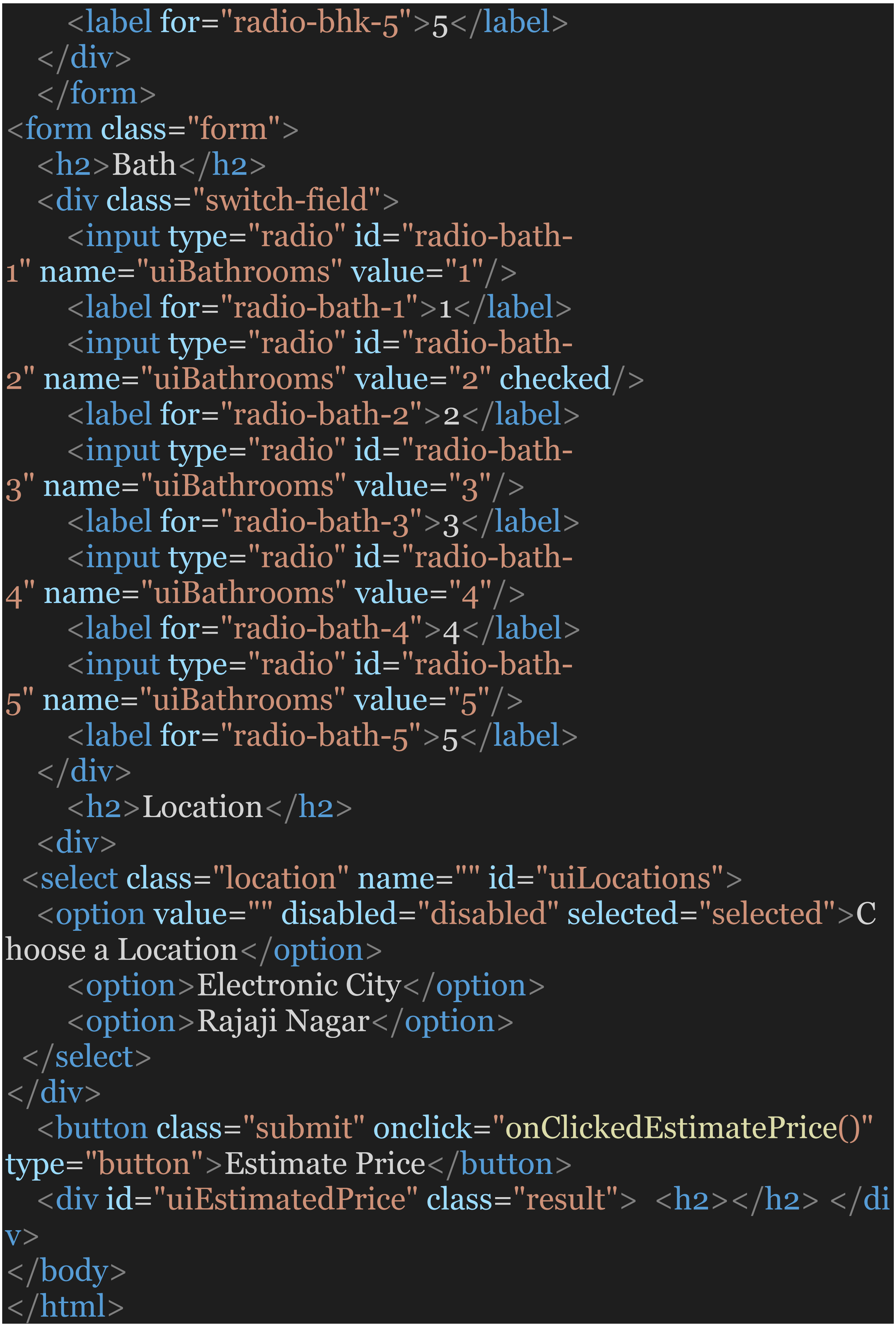
```
{
    "estimated_price": 83.5
}
```

3.7 HTML/CSS/JAVASCRIPT FOR USER INTERFACE:-

Visual Studio Code:-

App.html :-

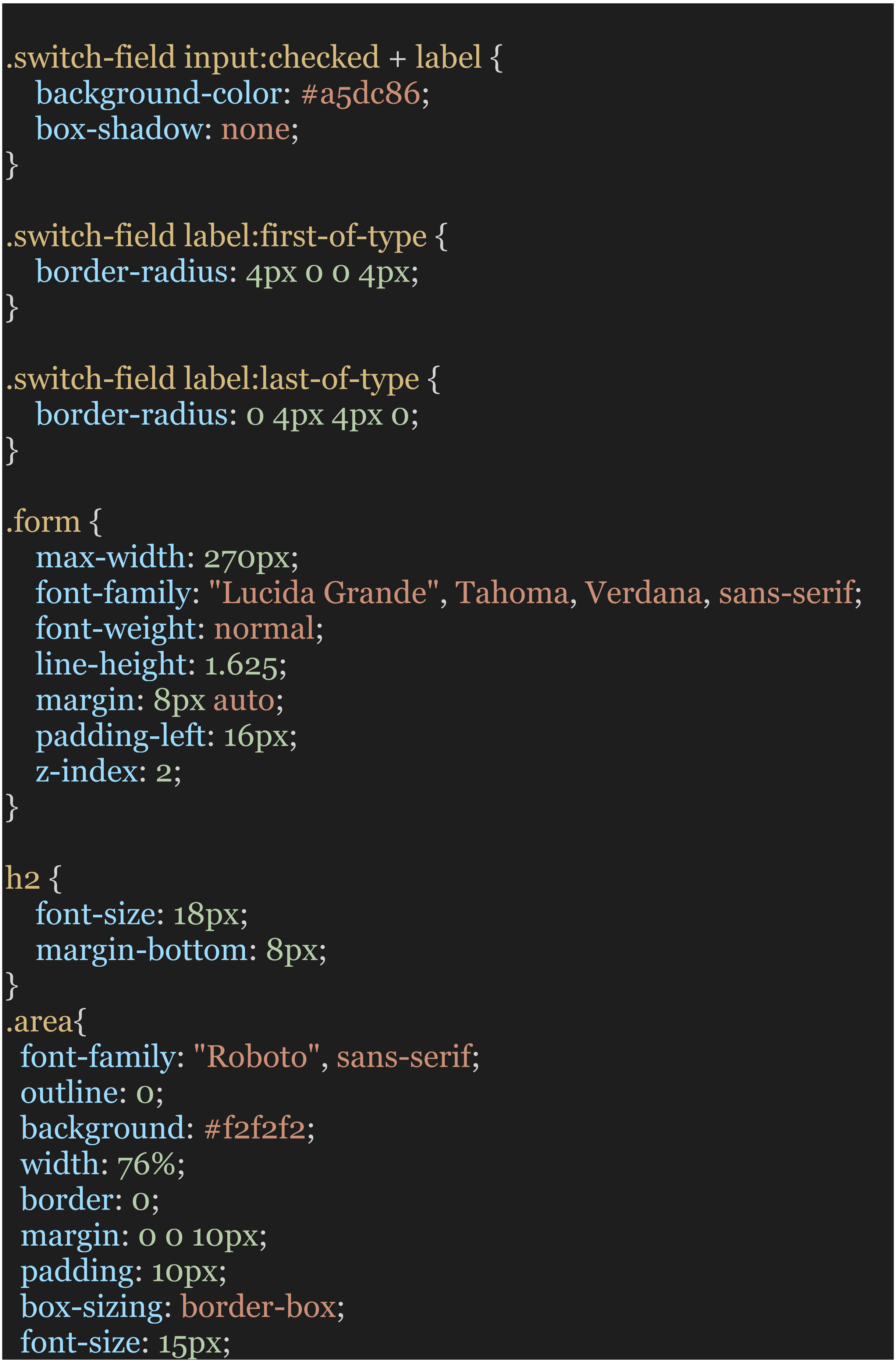
```
<!DOCTYPE html>

<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">C
hoose 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> </di
v>
</body>
</html>
```

app.css :-

```
@import url(https://fonts.googleapis.com/css?family=Roboto:300);

.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;
}
```

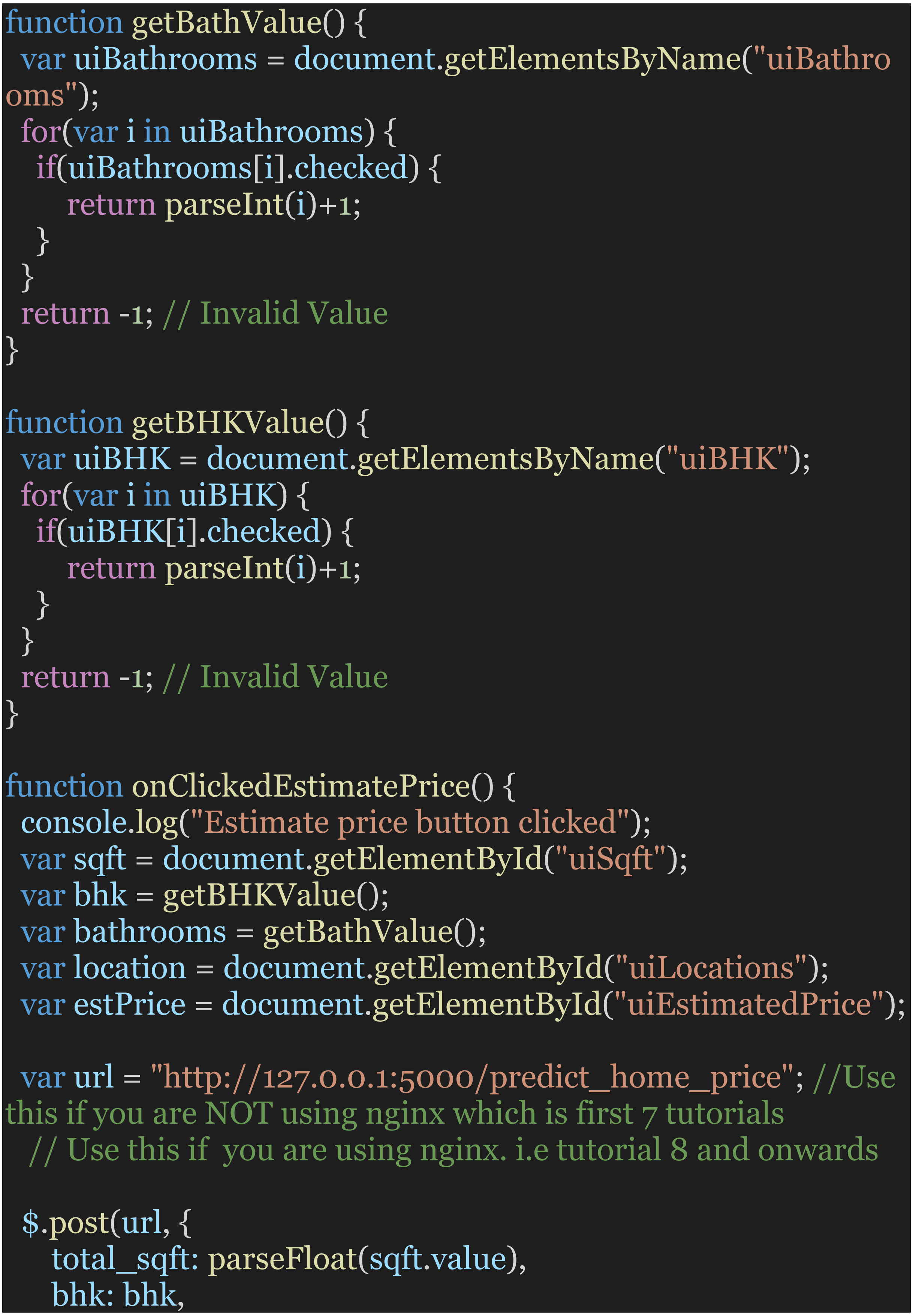


```
.switch-field input:checked + label {
  background-color: #a5dc86;
  box-shadow: none;
}

.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('https://images.unsplash.com/photo-
1564013799919-ab600027ffc6?ixlib=rb-
1.2.1&auto=format&fit=crop&w=1350&q=80');
   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 :-



```
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"; //Use this if you are NOT using nginx which is first 7 tutorials
   // Use this if  you are using nginx. i.e tutorial 8 and onwards

  $.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.toStr
ing() + " Lakh</h2>";
    console.log(status);
  });
}
function onPageLoad() {
  console.log( "document loaded" );
  var url = "http://127.0.0.1:5000/get_location_names"; // Use
 this if you are NOT using nginx which is first 7 tutorials
  //var url = "/api/get_location_names"; // Use this if  you are
using nginx. i.e tutorial 8 and onwards
  $.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;
```

MODELS USED

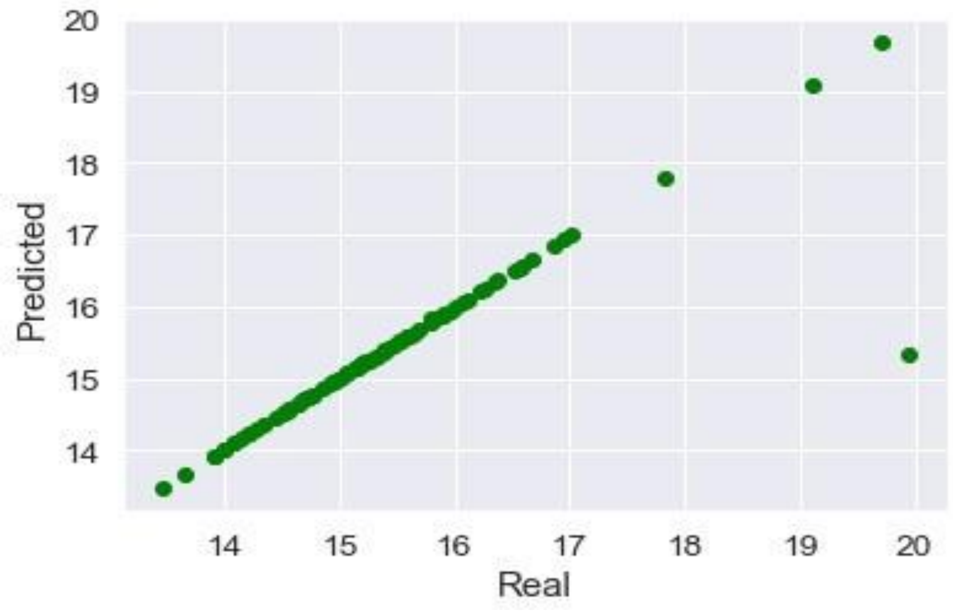
Regression Model:-

Linear Regression is a machine learning algorithm based on supervised learning.

It performs a regression task. Regression models a target prediction value based on independent variables.

It is mostly used for finding out the relationship between variables and forecasting.

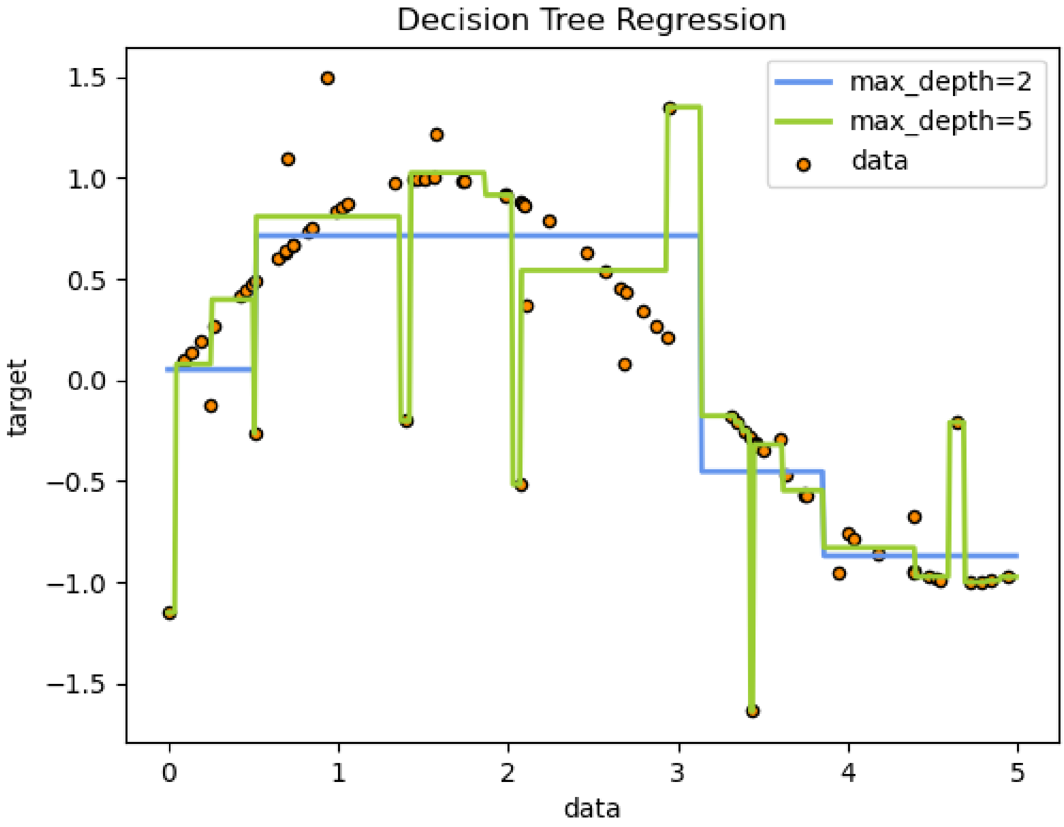
Real Vs Predicted



Decision Tree Regressor:-

A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility.

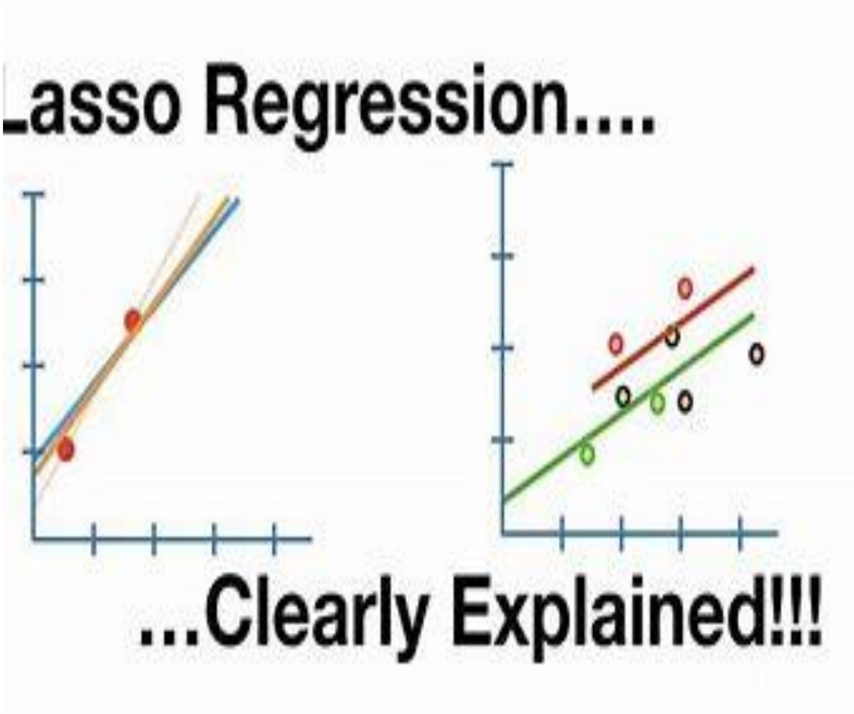
It is one way to display an algorithm that only contains conditional control statements.



Lasso Regression:-

In statistics and machine learning, lasso is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the resulting statistical model.

It was originally introduced in geophysics, and later by Robert Tibshirani, who coined the term.



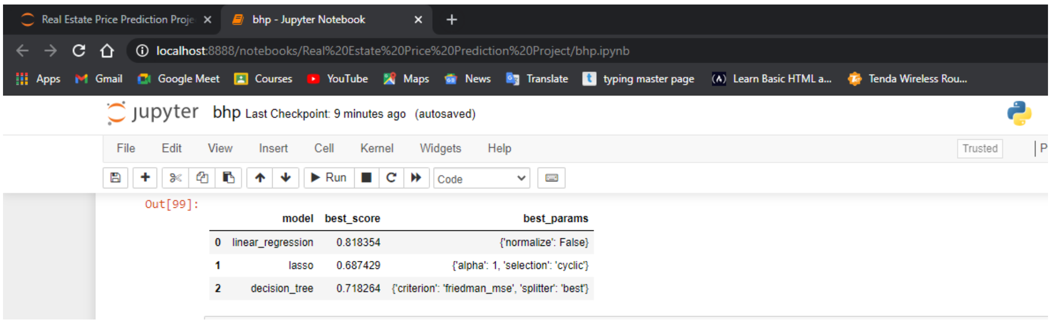
RESULTS AND DISCUSSIONS

Best Suited Model

So, our study showed that……..

Linear Regression displayed the best performance for this Dataset and can be used for deploying purposes.

Decision Tree Regressor and Lasso Regressor are far behind, so can't be recommended for further deployment purposes.



Out[99]:

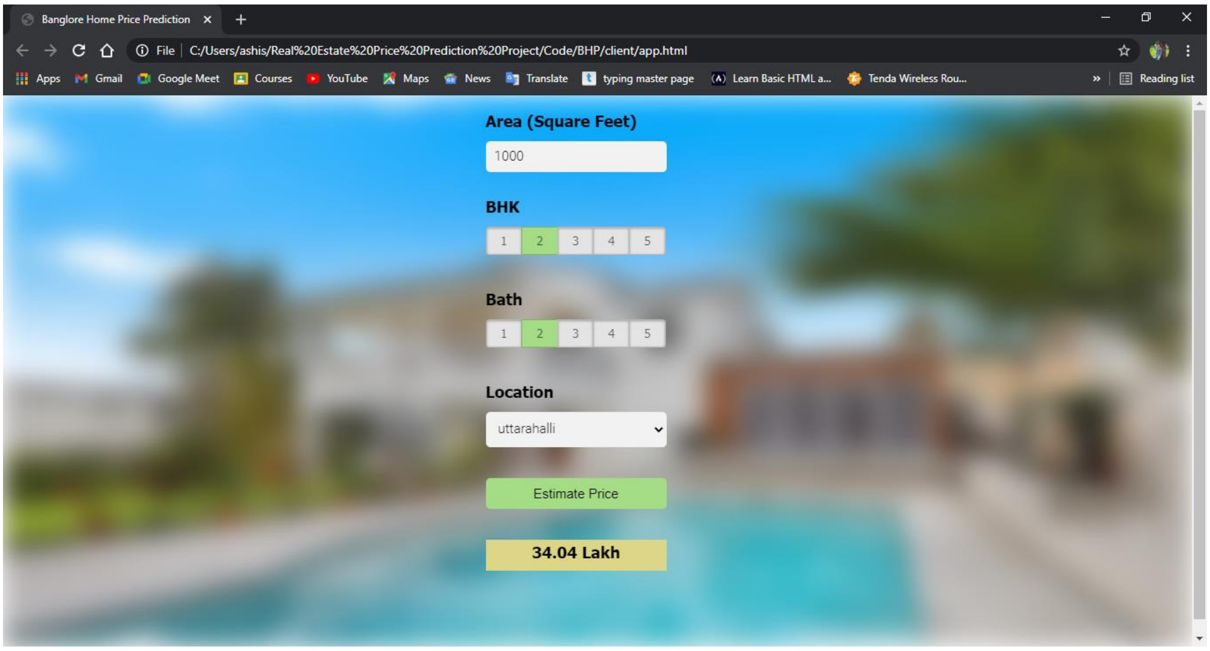
	model	best_score	best_params
0	linear_regression	0.818354	{'normalize': False}
1	lasso	0.687429	{'alpha': 1, 'selection': 'cyclic'}
2	decision_tree	0.718264	{'criterion': 'friedman_mse', 'splitter': 'best'}

Deployment App

The Model is deployed through Python Web App Flask in collaboration with HTML and CSS.

Output screenshoot:-

1)web Page:-



Area (Square Feet)
1000

BHK
1 [✓ 2] 3 4 5

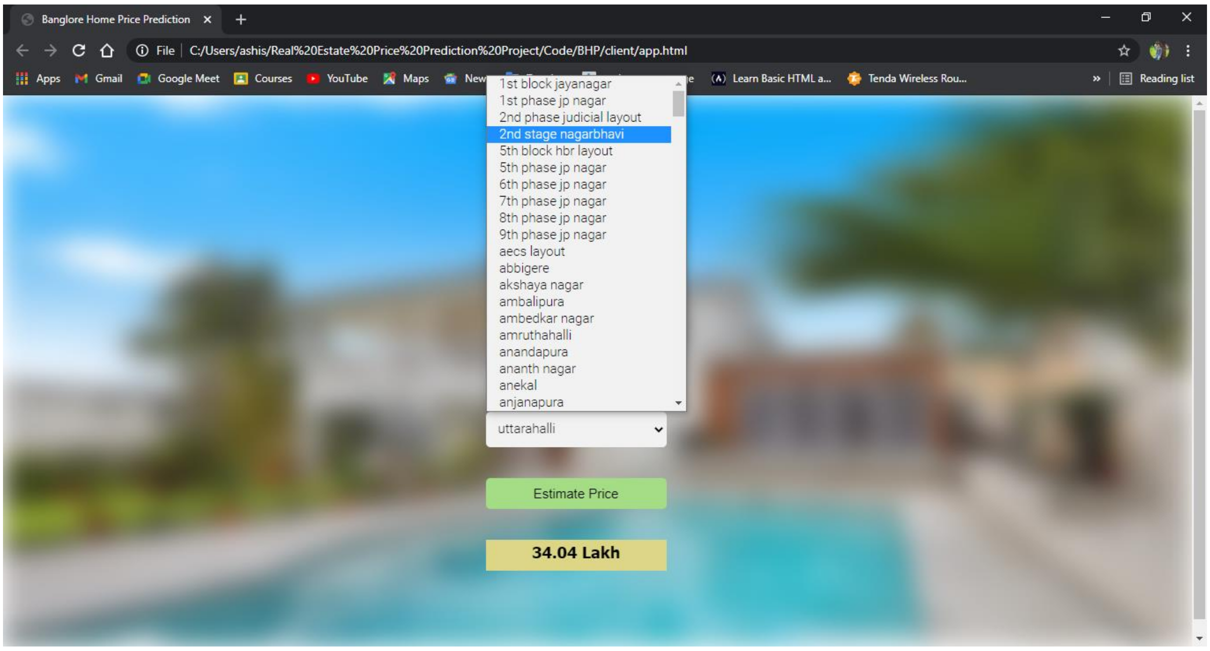
Bath
1 [✓ 2] 3 4 5

Location
uttarahalli

Estimate Price

34.04 Lakh

2) All Location scroll:-

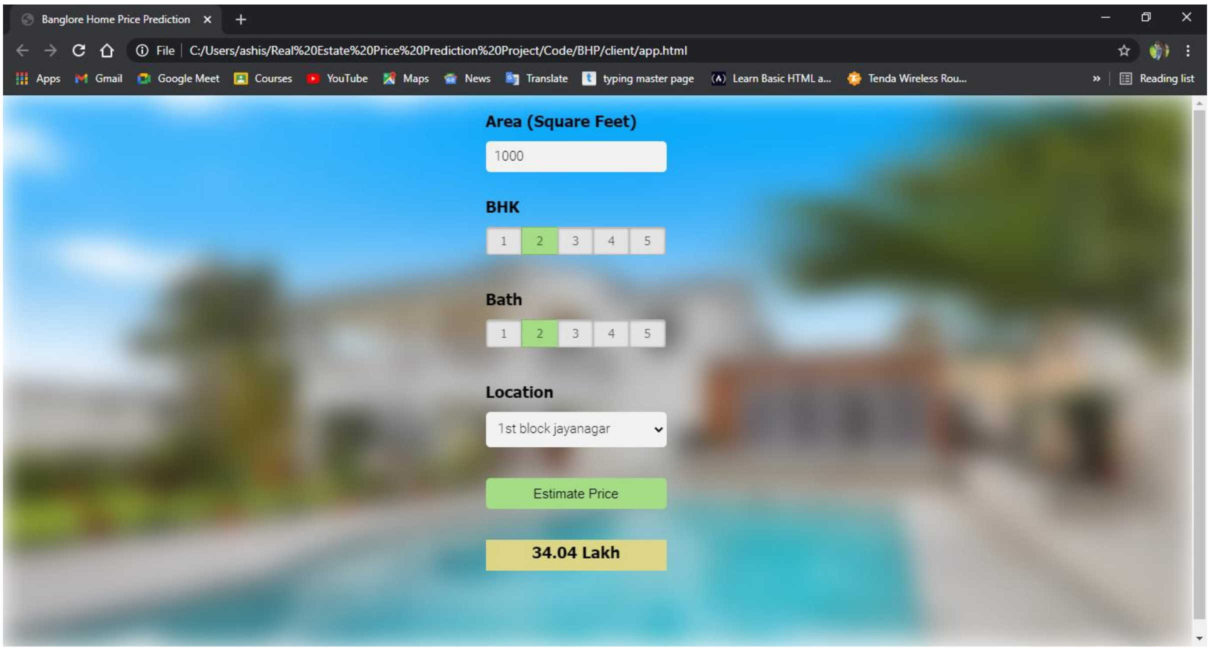


uttarahalli

Estimate Price

34.04 Lakh

3)Selection of sqft, bhk, bath:-



Area (Square Feet)
1000

BHK
1 [✓ 2] 3 4 5

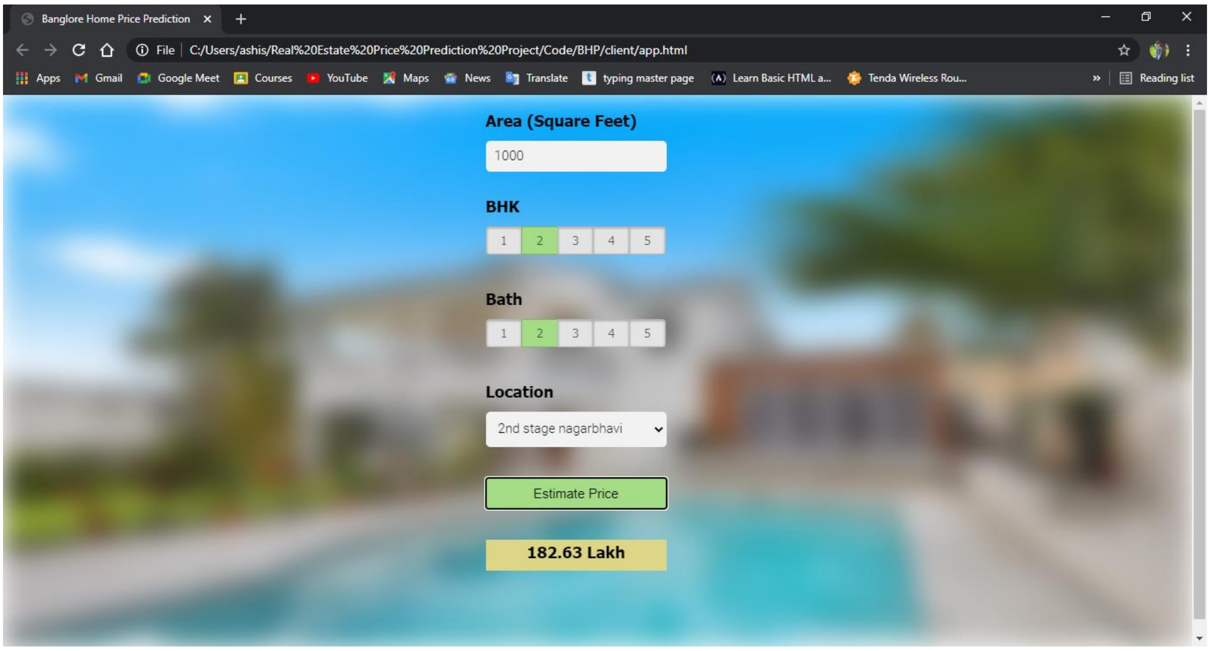
Bath
1 [✓ 2] 3 4 5

Location
1st block jayanagar

Estimate Price

34.04 Lakh

4)Price Prediction Depends upon bhk, bath,sqft, different location:-



Area (Square Feet)
1000

BHK
1 [✓ 2] 3 4 5

Bath
1 [✓ 2] 3 4 5

Location
2nd stage nagarbhavi

Estimate Price

182.63 Lakh

CONCLUSION

So, our Aim is achieved as we have successfully ticked all our parameters as mentioned in our CONTEXT Column.

We use different model to predict best price for home in their particular location.

It is seen that circle rate is the most effective attribute in predicting the house price and that the Linear Regression is the most effective model for our Dataset with BEST score of 0.818354.

❖ Bibliography

- ✓ Find Open Datasets and Machine Learning Projects | Kaggle
- ✓ Towards Data Science
- ✓ RxJS, ggplot2, Python Data Persistence, Caffe2, PyBrain, Python Data Access, H2O, Colab, Theano, Flutter, KNime, Mean.js, Weka, Solidity (tutorialspoint.com)
- ✓ Learn R, Python & Data Science Online | DataCamp
- ✓ YouTube
- ✓ https://www.khanacademy.org
- ✓ Analyticsvidhya.com
- ✓ Machinelearningplus.com
- ✓ Simplilearn.com
- ✓ www.tutorialpoint.com