B. Tech CSE III year

SEMESTER: 6th

Mini Project –II

Final Report

Department of Computer Science and Application

Institute of Engineering and Technology

Real Estate Price Prediction

Submitted To: | **Submitted By:**

Mr. Ankit Arora

Ayush Kumar

(201500179)

Acknowledgement

I thank the almighty for giving us the courage and perseverance in completing the project. This project itself is an acknowledgement for all those people who have given us their heartfelt co-operation in making this project a grand success. I extend our sincere thanks to Mr Ankit Arora, Technical Trainer at “GLA University, Mathura” for providing his valuable guidance at every stage of this project work. We are profoundly grateful towards the unmatched services rendered by him. And last but not least, I would like to express our deep sense of gratitude and earnest thanks giving to our dear parents for their moral support and heartfelt cooperation in doing the mini project.

Declaration

We hereby declare that the work which is being presented in the MINI Project “**Real Estate Price Prediction”,** in partial fulfilment of the requirements for MINI Project viva voce, is an authentic record of our own work carried by me under the supervision of my mentor Mr Ankit Arora.

Course: B. Tech (Computer Science and Engineering)
Year: 3rd
Semester: 6th
Supervised By: Mr. Mandeep Singh, Technical Trainer

Introduction

Real estate is a highly dynamic and complex industry that is impacted by a multitude of factors, ranging from macroeconomic conditions to local demographics. Accurately predicting real estate prices is a critical task for investors, homeowners, and real estate agents. With the availability of large amounts of data and advancements in machine learning techniques, it is now possible to develop predictive models that can forecast real estate prices with a high degree of accuracy. In this project, we aim to build a real estate price prediction model that leverages historical data on property sales, location, and other relevant features. By using advanced algorithms and techniques, our model will be able to forecast future property values, which will be invaluable to real estate professionals and investors alike.

What is in need right now?

There can be several reasons why someone might need a real estate price prediction project, depending on their specific context and goals. Here are a few possible examples:

1. Real estate investors: Investors may use real estate price prediction models to estimate the future value of properties they are considering purchasing. This can help them make informed decisions about whether a property is likely to appreciate in value and generate a good return on investment.

2. Real estate agents: Real estate agents may use price prediction models to provide guidance to their clients on what price to list their property at or what price to offer when making a purchase. This can help ensure that properties are priced competitively and sell quickly.

3. Policy makers: Governments or other organizations may use real estate price prediction models to monitor housing markets and identify areas where intervention may be needed. For example, if a model predicts that prices in a certain area are likely to skyrocket in the near future, policymakers may want to take steps to ensure that housing remains affordable for low- and middle-income residents.

Overall, real estate price prediction models can be useful in a variety of contexts where understanding future property values is important.

Along with the rapid development, the need for effectiveness and efficiency is prioritized in various fields. The purpose of this project is to design an automatic door that only detects an authorized Radio Frequency Identification (RFID) card to open.

The use of RFID systems can strengthen the security level of building access. This study uses a data processing method in the form of an ID number generated from a tag.

How does our mini project do what it is supposed to?

The goal of such a project is to predict the price of a property based on various factors such as location, size, age, number of rooms, amenities, and other relevant factors. This is typically done using machine learning algorithms.

Here's a high-level overview of how a real estate price prediction project works:

1. Data collection
2. Data cleaning and pre-processing
3. Feature engineering
4. Model selection and training
5. Model Evaluation
6. Deployment

Overall, a real estate price prediction project aims to use historical data to predict the sale price of a new property based on various factors. The accuracy of the prediction depends on the quality of the data, the effectiveness of the machine learning algorithm, and the features used to train the model.

What profits do we gain from it?

A real estate price prediction project can provide various types of profiles depending on the specific goals and methods used in the project. Here are some examples:

1. Property value trends

2. Demographic analysis

3. Property characteristics

4. Market Analysis

Overall, a real estate price prediction project can provide a comprehensive profile of the local real estate market, which can be useful for a variety of purposes, such as investment decisions, marketing strategies, and policy development.

Resources used:

Software/Platforms used:

1. **Kaggle:** Kaggle is an online platform that hosts as well as provides a community for data scientists and machine learning enthusiasts to collaborate, learn and share knowledge. Kaggle offers a wide range of datasets for users to work on and compete with, as well as resources like notebooks, forums, and tutorials to help users improve their skills.

2. **Google colab:** Google Colab is a cloud-based platform that provides a free Jupyter notebook environment for machine learning and data analysis. Colab is designed to make it easy to write and run Python code using Google's cloud infrastructure. Colab integrates with other Google services like Drive and GitHub, making it easy to import and export data and code. Colab is popular among data scientists, researchers, and students who want to experiment with machine learning algorithms and explore data in a collaborative and accessible environment.

Software/Technologies used:

1. Python
2. NumPy/Pandas
3. Flask
4. Tableau
5. Html
6. CSS
7. JavaScript

Python Flask Server

Flask is a micro web framework written in Python. It is designed to be lightweight and flexible, allowing developers to quickly and easily create web applications with minimal overhead. Flask provides tools and libraries for handling HTTP requests and responses, managing sessions, and rendering templates. It also supports database integration, authentication, and security features.

Why did we used Flask in our project?

1. Flask is known for its simplicity and ease of use.
2. It does not come with built-in features like ORM or database administration, but it allows developers to easily add third-party extensions and libraries to their projects.
3. Flask is widely used for building RESTful APIs, web applications, and prototypes.
4. Its simplicity and flexibility make it a popular choice for developers who want to create web applications quickly and with minimal setup.

Pandas and NumPy

Pandas and NumPy are two popular Python libraries used for data manipulation, analysis, and computation.

NumPy, short for Numerical Python, is a library that provides efficient multi-dimensional array operations, as well as mathematical functions for working with these arrays. It is a fundamental library for scientific computing in Python and is widely used in fields such as machine learning, data science, and engineering.

Pandas, on the other hand, is a library built on top of NumPy that provides high-performance data manipulation and analysis tools. It offers data structures for effectively handling and analyzing large and complex datasets, such as Series (for 1-dimensional data) and Data Frames (for 2-dimensional data). Pandas provides powerful data indexing, filtering, merging, and aggregation capabilities, and is a popular tool for data wrangling and preparation in data science and machine learning projects.

Together, Pandas and NumPy form a powerful combination for data analysis and manipulation in Python.

Tableau

Tableau is a powerful data visualization and business intelligence software. Tableau allows users to connect to a wide variety of data sources, from spreadsheets and databases to cloud-based applications and big data platforms. With Tableau, users can quickly create interactive dashboards, reports, and charts that help them understand and communicate insights from their data. Tableau provides a drag-and-drop interface that allows users to build visualizations without writing any code, and it also offers a range of advanced features for data preparation, statistical analysis, and machine learning.

Source Code:

server.py

```
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'])
```

```python
    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

```python
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)
```

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

    with open("./artifacts/columnss.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/bhpmini2.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))
```

Screenshots:

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

```
[2] df1=pd.read_csv("bhp.csv")
    df1.head()
```

	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

```
[3] df1.shape

(13320, 9)
```

area type feature

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

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

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

	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

data cleaning

```
[6] df2.isnull().sum()

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

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

location      0
size          0
total_sqft    0
bath          0
price         0
dtype: int64
```

```
[8] df3.shape

(13246, 5)
```

```
[9] df3['size'].unique()

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',
```

[12]
```
df3['bhk'].unique()
```

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

[13]
```
df3[df3.bhk>20]
```

	location	size	total_sqft	bath	price	bhk
1718	2Electronic City Phase II	27 BHK	8000	27.0	230.0	27
4684	Munnekollal	43 Bedroom	2400	40.0	660.0	43

```
df3.total_sqft.unique()
```

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

[15]
```
def is_float(x):
  try:
    float(x)
  except:
    return False
  return True
```

[16]
```
df3[~df3['total_sqft'].apply(is_float)]
```


[17]
```
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
```

[18]
```
convert_sqft_to_num('850 - 1060')
```

```
955.0
```

[19]
```
convert_sqft_to_num('2334')
```

```
2334.0
```

[20]
```
df4=df3.copy()
df4['total_sqft']=df4['total_sqft'].apply(convert_sqft_to_num)
df4.head()
```

	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

[21]
```
df4.iloc[30]
```

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

[22]
```
df5=df4.copy()
df5['price_per_sqft']=df5['price']*100000/df5['total_sqft']
df5.head()
```

	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

[23]
```
len(df5.location.unique())
```

```
1304
```

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

```
location
Whitefield               535
Sarjapur  Road           392
Electronic City          304
Kanakpura Road           266
Thanisandra              236
                        ...
1 Giri Nagar               1
Kanakapura Road,           1
Kanakapura main  Road      1
Karnataka Shabarimala      1
whitefiled                 1
Name: location, Length: 1293, dtype: int64
```

```
[25] len(location_stats[location_stats<=10])
```

```
1052
```

```
[26] location_stats_less_than_10 =location_stats[location_stats<=10]
     location_stats_less_than_10
```

```
location
Basapura                10
1st Block Koramangala   10
Gunjur Palya            10
```



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

```
242
```

```
[28] df5.head(10)
```

	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

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



```
[30] df5.shape
```

```
(13246, 7)
```

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

```
(12502, 7)
```

```
[32] df6.price_per_sqft.describe()
```

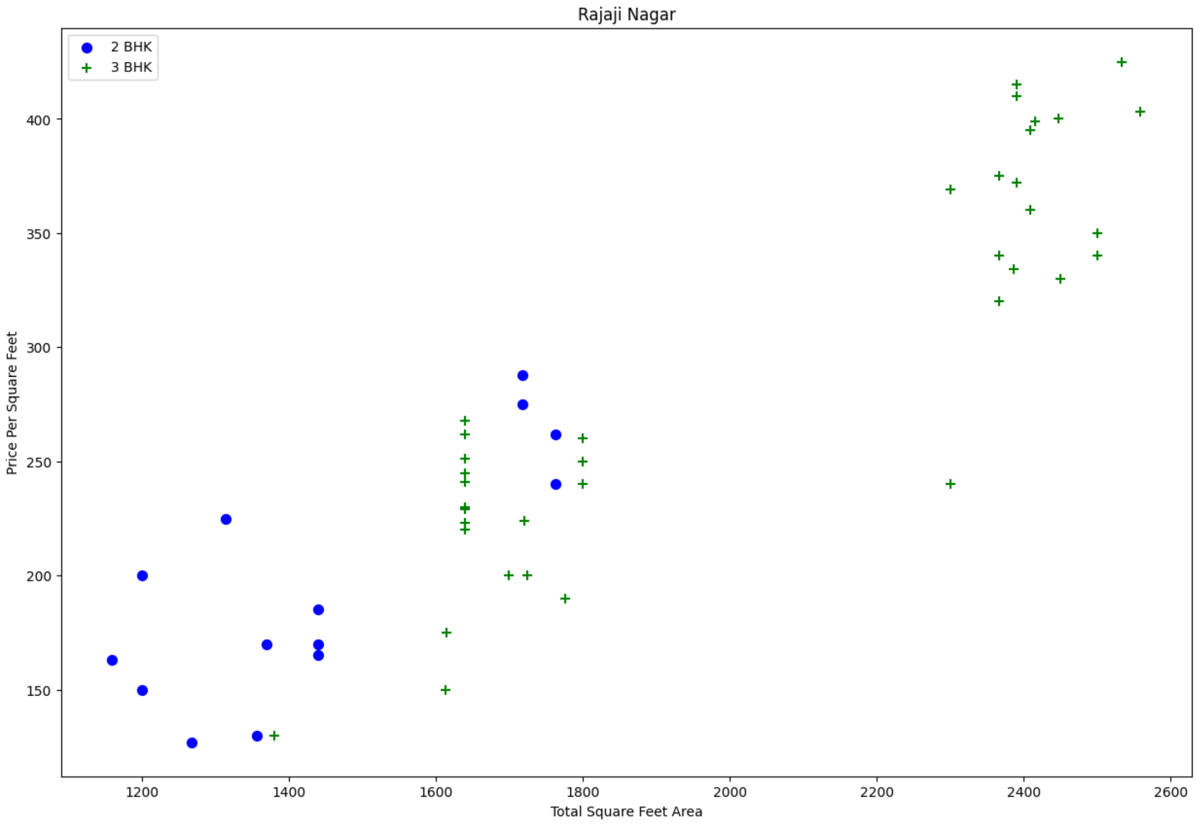
```
count     12456.000000
mean       6308.502826
std        4168.127339
min         267.829813
25%        4210.526316
50%        5294.117647
75%        6916.666667
max      176470.588235
Name: price_per_sqft, dtype: float64
```

```
[33] 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
```



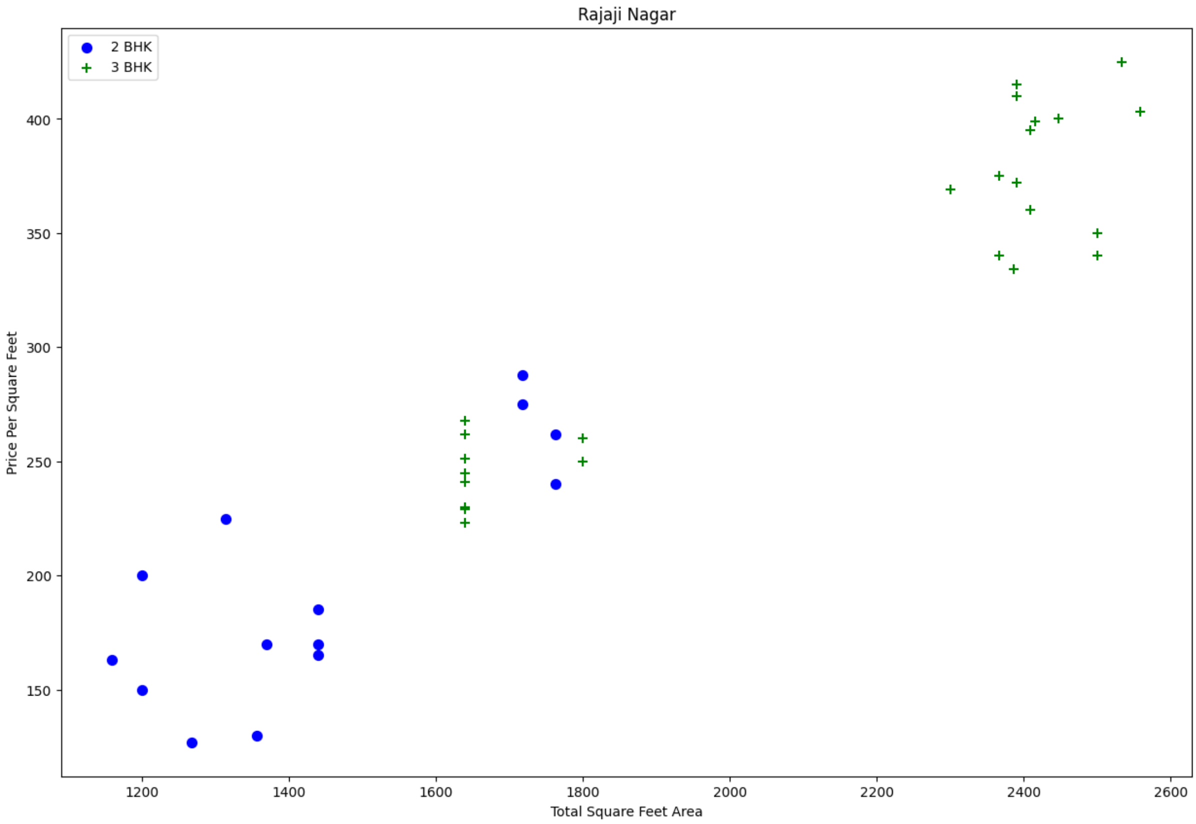
```
:4] def plot_scatter_chart(df,location):
      bhk2=df[(df.location==location)&(df.bhk==2)]
      bhk3=df[(df.location==location)&(df.bhk==3)]
      matplotlib.rcParams['figure.figsize']=(15,10)
      plt.scatter(bhk2.total_sqft,bhk2.price,color='blue',label='2 BHK',s=50)
      plt.scatter(bhk3.total_sqft,bhk3.price,marker="+",color='green',label='3 BHK',s=50)
      plt.xlabel("Total Square Feet Area")
      plt.ylabel("Price Per Square Feet")
      plt.title(location)
      plt.legend()

    plot_scatter_chart(df7,"Rajaji Nagar")
```

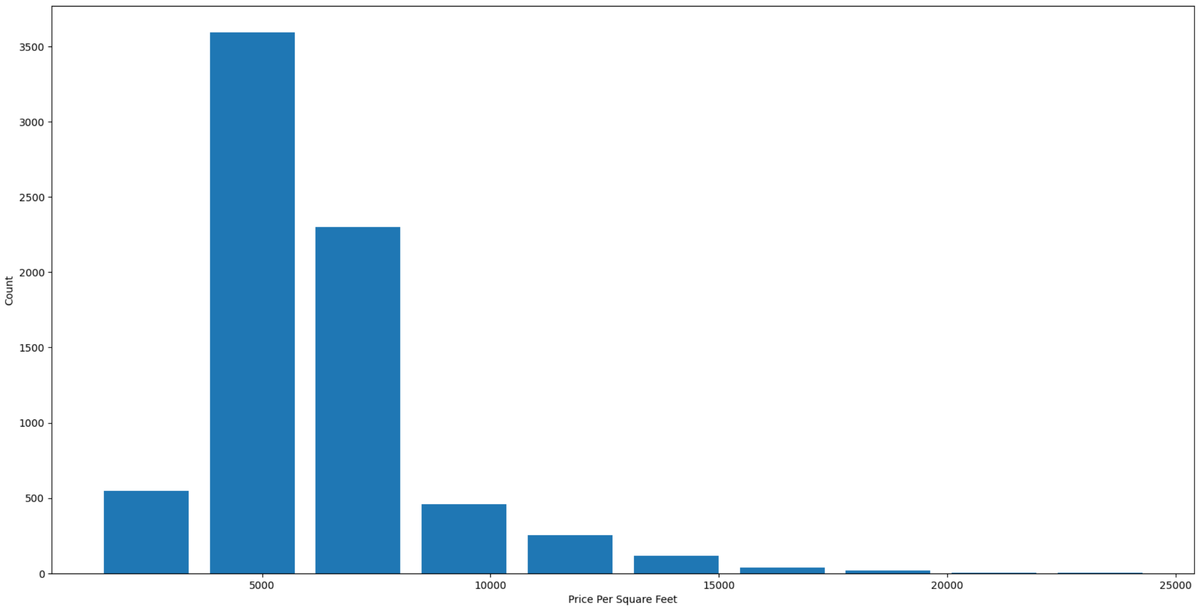



```
5]  def remove_bhk_outliers(df):
      exclude_indices=np.array([])
      for location,location_df in df.groupby('location'):
        bhk_stats={}
        for bhk,bhk_df in location_df.groupby('bhk'):
          bhk_stats[bhk]={
              'mean': np.mean(bhk_df.price_per_sqft),
              'std':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




```
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")
```



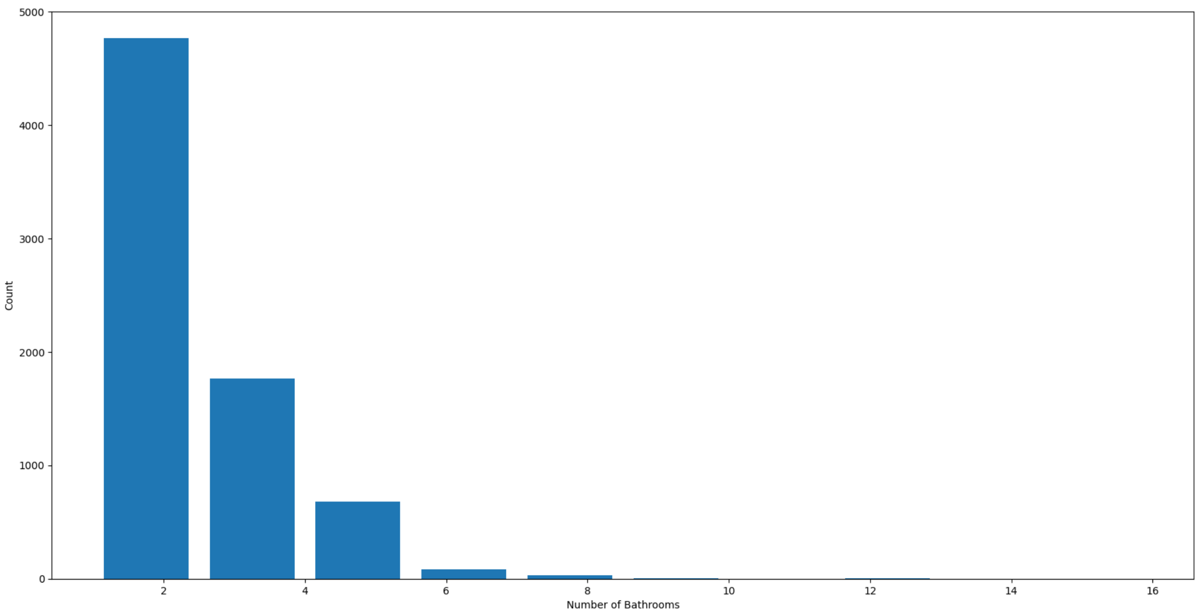
[38] df8.bath.unique()

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

[39] df8[df8.bath>10]

	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

[40]
```
plt.hist(df8.bath,rwidth=0.8)
plt.xlabel("Number of Bathrooms")
plt.ylabel("Count")
```



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

	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

42]
```
df9=df8[df8.bath<df8.bhk+2]
df9.shape
```

```
(7251, 7)
```

43]
```
df10=df9.drop(['size','price_per_sqft'],axis='columns')
df10.head()
```

	location	total_sqft	bath	price	bhk
0	1st Block Jayanagar	2850.0	4.0	428.0	4
1	1st Block Jayanagar	1630.0	3.0	194.0	3
2	1st Block Jayanagar	1875.0	2.0	235.0	3
3	1st Block Jayanagar	1200.0	2.0	130.0	3
4	1st Block Jayanagar	1235.0	2.0	148.0	2

45]
```
dummies=pd.get_dummies(df10.location)
dummies.head()
```

	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	Yelachenahalli
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	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0

5 rows × 242 columns

46]
```
df11=pd.concat([df10,dummies.drop('other',axis='columns')],axis='columns')
df11.head()
```

47]
```
df12=df11.drop('location',axis='columns')
df12.head()
```

	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	1875.0	2.0	235.0	3	1	0	0	0	0	0	...	0	0	0	0	0
3	1200.0	2.0	130.0	3	1	0	0	0	0	0	...	0	0	0	0	0
4	1235.0	2.0	148.0	2	1	0	0	0	0	0	...	0	0	0	0	0

5 rows × 245 columns

48]
```
X=df12.drop('price',axis='columns')
X.head()
```

49]
```
y=df12.price
y.head()
```

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

50]
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train,y_test=train_test_split(X.values,y,test_size=0.2,random_state=10)
```

51]
```
from sklearn.linear_model import LinearRegression
lr_clf=LinearRegression()
lr_clf.fit(X_train,y_train)
lr_clf.score(X_test,y_test)
```

```
0.8452277697874376
```

52]
```
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)
```

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


53]
```
from sklearn.model selection import GridSearchCV
```

```
[53] 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']
                 }
             }
         }
         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)
```



```
[54] 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]
```

```
[55] predict_price('1st Phase JP Nagar',1000, 2, 2)

     83.49904677206221
```

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

     86.80519395233001
```

+ Code   + Text

```
[57] predict_price('Indira Nagar',1000, 2, 2)

     181.2781548400639
```

```
     predict_price('Indira Nagar',1000, 3, 3)

     184.5843020203317
```


predict_price('Indira Nagar',1000, 3, 3)

184.5843020203317


```
[ ] import pickle
    with open('bhpmini2.pickle','wb') as f:
        pickle.dump(lr_clf,f)
```

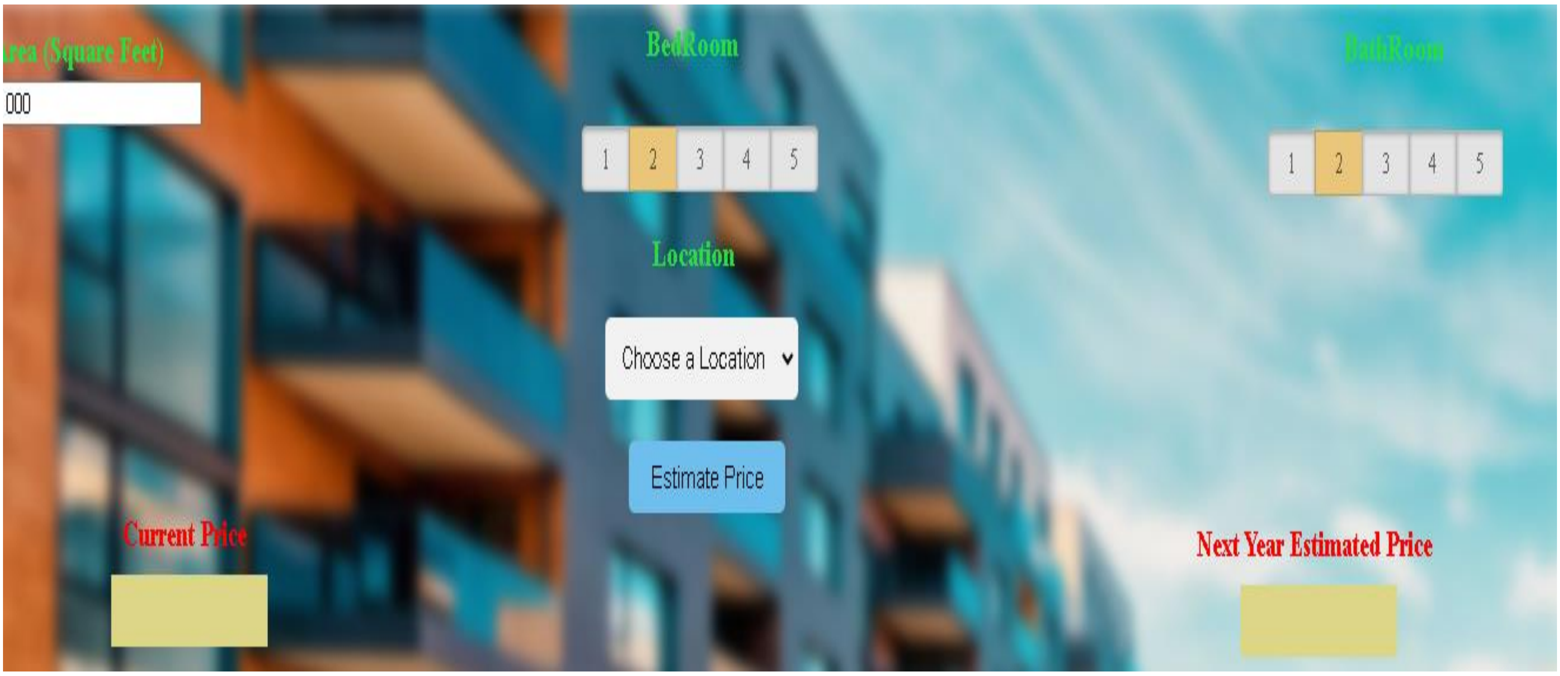
```
[ ] import json
    columns = {
        'data_columns' : [col.lower() for col in X.columns]
    }
    with open("columnss.json","w") as f:
        f.write(json.dumps(columns))
```

```
[ ] from google.colab import files
    files.download('bhpmini2.pickle')
```

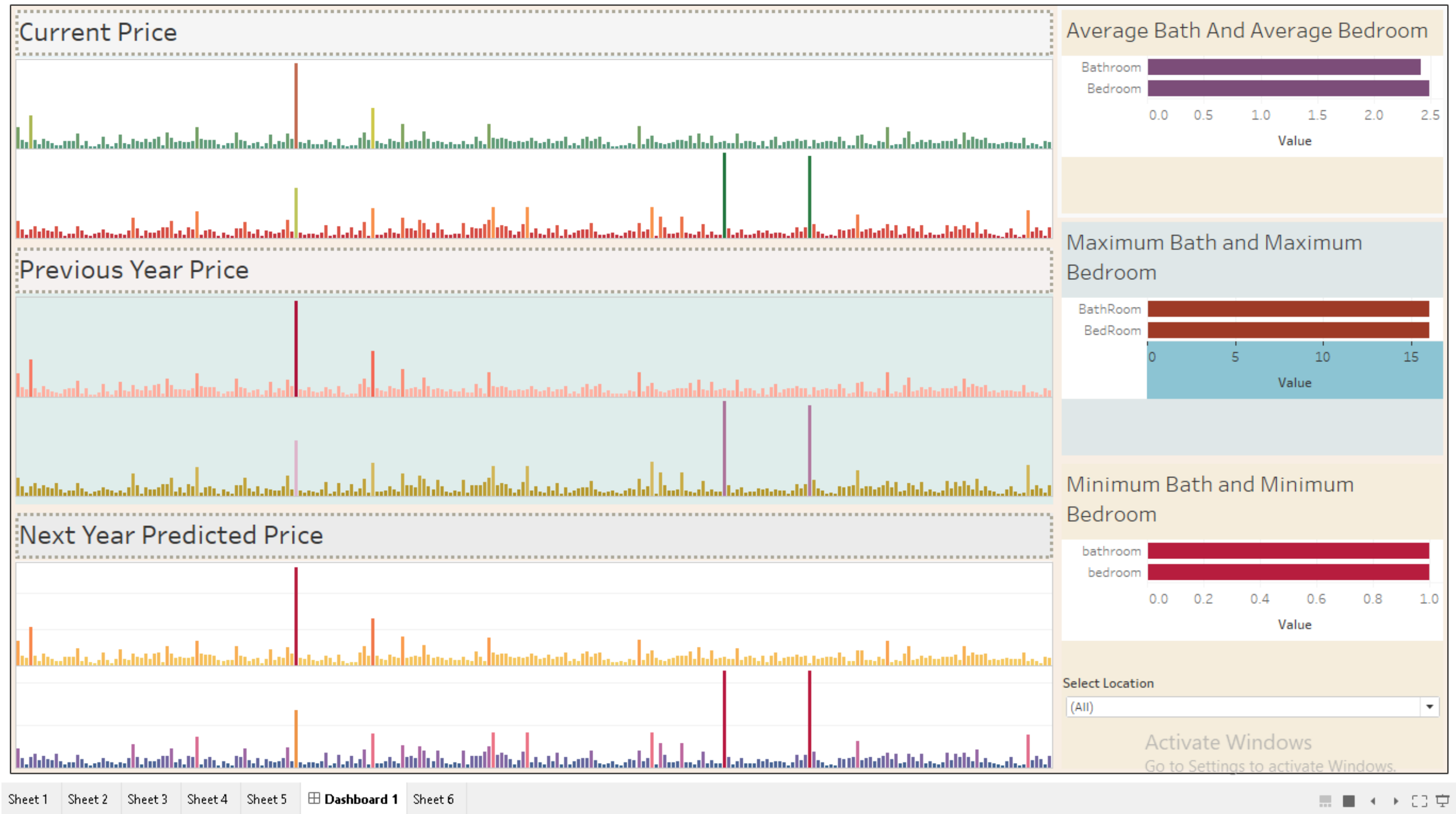
```
[ ] files.download('columnss.json')
```

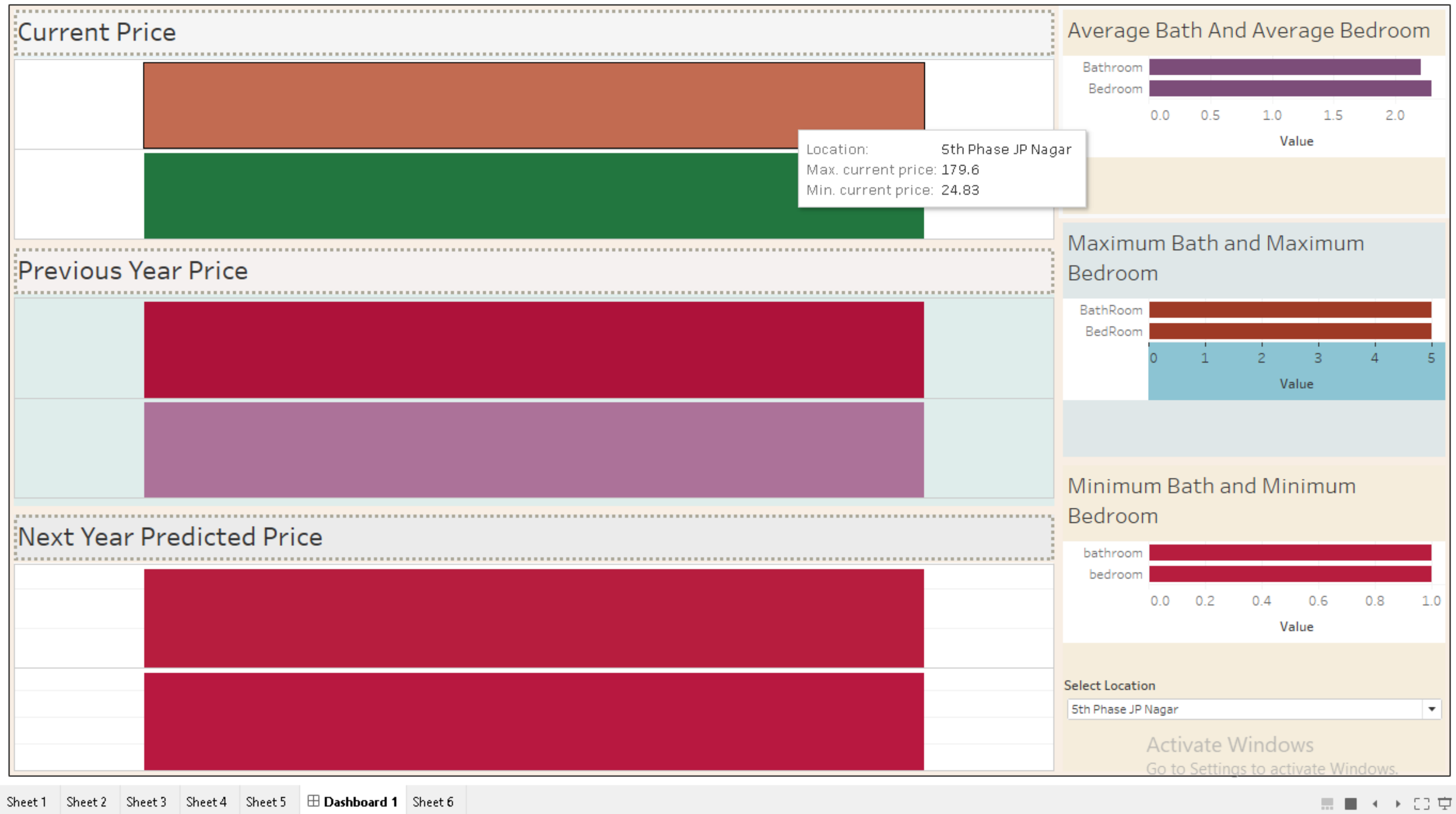
```
[ ] df12.to_csv('cleaned.csv')
```

```
[ ] df12.to_csv('df12.csv')
```



rea (Square Feet)
000
BedRoom
1 2 3 4 5
BathRoom
1 2 3 4 5
Location
Choose a Location
Estimate Price
Current Price
Next Year Estimated Price


Current Price
Previous Year Price
Next Year Predicted Price
Average Bath And Average Bedroom
Bathroom
Bedroom
Value
Maximum Bath and Maximum Bedroom
BathRoom
BedRoom
Value
Minimum Bath and Minimum Bedroom
bathroom
bedroom
Value
Select Location
(All)
Activate Windows
Go to Settings to activate Windows.
Sheet 1
Sheet 2
Sheet 3
Sheet 4
Sheet 5
Dashboard 1
Sheet 6


Current Price
Location: 5th Phase JP Nagar
Max. current price: 179.6
Min. current price: 24.83
Previous Year Price
Next Year Predicted Price
Average Bath And Average Bedroom
Bathroom
Bedroom
Value
Maximum Bath and Maximum Bedroom
BathRoom
BedRoom
Value
Minimum Bath and Minimum Bedroom
bathroom
bedroom
Value
Select Location
5th Phase JP Nagar
Activate Windows
Go to Settings to activate Windows.
Sheet 1
Sheet 2
Sheet 3
Sheet 4
Sheet 5
Dashboard 1
Sheet 6

Pre-requisite:

Here are some prerequisites for the successful completion of this project:

1. **Data:** You will need a large dataset of real estate listings, including details such as location, size, number of bedrooms/bathrooms, year built, and other relevant features.

2. **Python:** You will need to have a working knowledge of Python, as this is the most common language used in data science and machine learning.

3. **Data manipulation and visualization:** You should be familiar with libraries like pandas and matplotlib for data manipulation and visualization, as you will need to pre-process and analyse your data before building a model.

4. **Machine learning:** You should have a basic understanding of machine learning algorithms, such as linear regression, decision trees, and random forests, which are commonly used for real estate price prediction.

5. **Scikit-learn:** Scikit-learn is a popular machine learning library in Python that provides implementations of many common algorithms, as well as tools for data pre-processing, feature engineering, and model evaluation.

6. **Jupyter Notebook:** Jupyter Notebook is a popular tool for data analysis and machine learning in Python. You can use it to create and execute code in a web-based environment, as well as to create reports and share your work with others.

Main Objective:

The main objective of a real estate price prediction project is to develop a machine learning model that can accurately predict the selling price of a property based on various features such as location, size, number of bedrooms/bathrooms, year built, and other relevant factors. The model aims to assist real estate professionals, home buyers, and sellers in making informed decisions about property values.

Accurate real estate price prediction models have many potential applications, including helping buyers and sellers determine fair prices for properties, assisting real estate agents in pricing their listings, and informing investment decisions for developers and property investors. With accurate predictions, buyers and sellers can avoid overpaying or underpricing a property, while real estate agents can better market their listings to potential buyers. Additionally, property investors can use the model to identify undervalued properties and make informed investment decisions.

In summary, the main objective of a real estate price prediction project is to create a model that can accurately predict the price of a property based on relevant features, with the aim of assisting real estate professionals, buyers, and sellers in making informed decisions about property values.

What can we append in future?

There are several areas where we can append more features to improve the accuracy of real estate price prediction models in the future. Here are a few potential areas:

1. **Property history:** Including data about a property's past sale prices, previous owners, and any renovation or upgrade history could help predict its current value.

2. **Neighbourhood data:** Demographic data such as age, education level, and income of the neighbourhood where the property is located could provide insights into the value of the property.

3. **Accessibility:** Adding data about the availability of public transportation, proximity to commercial areas, and walkability scores could help predict the value of the property.

4**. Environmental factors:** Including data about environmental factors such as air quality, water quality, and natural disasters could provide additional information about the property's value.

5. **Property type:** Including data about the type of property, such as condo, townhouse, or single-family home, could provide insights into the value of the property.

6. **Economic indicators:** Including data about local and national economic indicators such as inflation rate, interest rates, and job growth could help predict the value of the property.

7. **Seasonality:** Incorporating data about seasonality, such as how different seasons impact real estate demand, could help improve the accuracy of the model.

By incorporating these additional features, we can develop more accurate real estate price prediction models, which can help real estate professionals, buyers, and sellers make more informed decisions about property values.

Bibliography

1. www.kaggle.com
2. www.colab.research.google.com

Thank you!