HOUSE PRICE PREDICTION OF INDIAN METROPOLITAN USING MACHINE LEARNING & PYTHON

AN INTERNSHIP REPORT

Submitted by

AKSHAT SHARMA

[EC2331201010120]

Under the Guidance of

Dr C. Muthamizhchelvan, (Assistant Professor, Directorate of Online Education)

in partial fulfillment for the award of the degree of

BACHELOR OF COMPUTER APPLICATIONS

DIRECTORATE OF ONLINE EDUCATION

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, KATTANKULATHUR

JUNE 2024

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

DIRECTORATE OF ONLINE EDUCATION

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, KATTANKULATHUR

BONAFIDE CERTIFICATE

This internship report titled **"HOUSE PRICE PREDICTION OF INDIAN METROPOLITAN USING MACHINE LEARNING & PYTHON"** is the Bonafede work of **"AKSHAT SHARMA [**EC2331201010120**]"**, who carried outthe internship work under my supervision along with the company mentor. Certified further, that to the best of my knowledge the work reported herein does not form any other internship report or dissertation based on which a degree or award was conferred on an earlier occasion on this or any other candidate

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be **University** *u/s 3 of UGC Act, 1956*

INTERNSHIP OFFER LETTER

CORIZO

Empowering Tomorrow's Leaders

Internship Offer Letter

26th February 2024

Dear Akshat Sharma ,

With reference of your application, we are pleased to offer you an internship with **Corizo Edutech.**

We take this opportunity in wishing you the very best in your new employment as well as advising you that our offer letter is on the following terms and conditions:

1. Period of Service: Two(2) Months of your employment will be probationary.
You shall, for the purpose of your employment with us, sign this offer letter for submission and approval of the management.

2.Designation: You shall be employed as a **Data Science Intern.**

3. Remuneration: You will be not eligible for remuneration.

Internship Start Date : 05/03/2024

Internship End Date: 05/05/2024

Your responsibilities will include those for which you are engaged, as well as any other duties given to you by your mentor from time to time. By accepting this offer you agree to perform all responsibilities assigned to you with due care and diligence and in compliance with the management norms and clauses.

By accepting this offer letter of Employment, you acknowledge that you will keep all this information strictly confidential and refrain from using it for your own purposes, that is, disclosing it to anyone outside of the company.

By accepting this offer letter, you agree that throughout your internship, you will observe all policies and practices governing the conduct of our business and employees. This letter sets forth the complete offer we are extending to you and supersedes and replaces any prior inconsistent statements or discussions. **Official communication either within the company or outside the company should be through the official Email of the HR or support only.**

Corizo, D-247, B-12, Sector-63, Noida supportlms@corizo.in corizo.in

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

ACKNOWLEDGEMENTS

We express our humble gratitude to **Dr C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support. We extend our sincere thanks to Director DOE, SRM Institute of Science and Technology, Prof. **Dr Manoranjan Pon Ram,** for his invaluable support. SRM Institute of Science and Technology, for her support throughout the project work. We want to convey our thanks to Programme Coordinator **Dr. G. Babu**, Directorate of online Education, SRM Institute of Science and Technology, for their inputs during the project reviews and support. Our inexpressible respect and thanks to my guide, **Dr G.Babu.**, Assistant Professor & Programme Coordinator Directorate of online Education,, SRM Institute of Science and Technology, for providing me with an opportunity to pursue my project under her mentorship. She provided me with the freedom and support to explore the research topics of my interest. Her passion for solving problems and making a difference in the world has always been inspiring. We sincerely thank the Directorate of online Education, staff and students, SRM Institute of Science and Technology, for their help during our project. Finally, we would like to thank parents, family members, and friends for their unconditional love, constant support, and encouragement.

- AKSHAT SHARMA

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

TABLE OF CONTENTS

1. ABSTRACT

In the dynamic landscape of real estate, accurately predicting house prices plays a pivotal role for various stakeholders including buyers, sellers, and investors. Leveraging the power of Python programming and machine learning (ML) algorithms, this project aims to develop a robust system for **house price prediction** in **Mumbai, Delhi , Chennai, Hyderabad Bengluru.**

1. Introduction: The real estate market is characterized by its complexity, influenced by factors such as location, size, amenities, and market trends. Accurate price prediction is crucial for informed decision-making and maximizing returns on investment. However, traditional methods often fall short in capturing the nuanced relationships within the data. This project seeks to address this challenge by harnessing the capabilities of **Python and ML techniques**.

2. Methodology: The project follows a systematic development process, starting with data collection from Kaggle. The collected data undergoes rigorous preprocessing, including cleaning, encoding, and feature engineering. Various **ML algorithms** including linear regression, decision trees, random forests, and gradient boosting are compared and then best one is applied to develop predictive models. Model performance is evaluated using metrics such as mean absolute error and **R-squared.**

3. Results: Through experimentation and evaluation, the project demonstrates the effectiveness of the proposed approach in accurately predicting house prices. The trained models offer valuable insights into market trends and provide actionable information for **stakeholders**.

4. Conclusion: The project contributes to the growing body of literature on ML applications in real estate and underscores the significance of data-driven decision-making in property transactions. By leveraging Python and ML techniques, it offers a powerful tool for navigating the complexities of the real estate market and making **informed decisions**.

2. System Analysis

2.1 Existing System

In the existing system, we address challenges and present an approach to the efficient, incremental consolidation of requirements. Following common practice, our method iterates over information requirements to create the final design. We show how to efficiently accommodate a new information requirement into an existing design and how to update a design in response to evolving information requirements. The final design satisfying all requirements is created.

2.2 Proposed System

Following the defined requirements, we propose to create a **web application** to predict house prices in Indian cities using Python and machine learning. This application will enable users to input property details and obtain accurate price predictions. For this app, I have taken the initiative to propose a blueprint of the output, demonstrating how users can input property details and **receive predictions**.

3. SOFTWARE REQUIREMENTS SPECIFICATIONS

System configurations

The software requirement specification can produce at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by established a complete information description, a detailed functional description, a representation of system behavior, and indication of performance and design constrain, appropriate validate criteria, and other information pertinent to requirements.

Software Requriements

- **Operating system** : Windows- Windows 7, 8, 10,11,12 or later , Mac- macOS 10.9 or later, Linux- Ubuntu, Debian, Red Hat, Fedora, or SUSE.
- **Coding Language** : Python.
- **IDE(s)** : Microsoft Visual Studio Code, Spyder.

4. TECHNOLOGY

4.1 PYTHON

Python is a high-level, interpreted programming language known for its simplicity and readability. Its syntax is clear and easy to understand, making it an ideal choice for both beginners and experienced developers. Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

- Interpreted Language: Python code is executed line-by-line, making debugging easier.

- **Dynamic Typing:** Variables do not require an explicit declaration to reserve memory space.

- **High-Level Language:** Python abstracts many low-level details, allowing developers to focus on solving problems.

- **Extensive Libraries:** Python boasts a vast standard library and numerous third-party modules that extend its capabilities.

Python is widely used in web development, data analysis, artificial intelligence, scientific computing, and more. Its simplicity and powerful features make it a popular choice among developers.

4.2 XGBOOST

XGBoost (Extreme Gradient Boosting) is a powerful machine learning algorithm based on the gradient boosting framework. It is designed to be highly efficient, flexible, and portable, and it provides superior performance for both regression and classification tasks.

- **Gradient Boosting:** Combines the predictions of multiple weak models to produce a strong learner.

- **Efficiency:** Optimized for speed and performance, handling large datasets effectively.

- **Regularization:** Incorporates L1 and L2 regularization to prevent overfitting.

- **Parallel Processing:** Supports parallel and distributed computing to accelerate training.

XGBoost is widely used in data science and real-world applications due to accuracy and speed.

4.3 PANDAS

Pandas is a powerful data manipulation and analysis library for Python. It provides data structures and functions needed to manipulate structured data seamlessly.

- **DataFrame:** Two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes.
- **Series:** One-dimensional labeled array capable of holding any data type.
- **Data Cleaning:** Functions to handle missing data and duplicates.
- **Data Transformation:** Tools for merging, reshaping, selecting, and aggregating data.

Pandas is essential for data wrangling and preprocessing, making it a go-to library for data scientists.

4.4 NUMPY

NumPy (Numerical Python) is the fundamental package for scientific computing with Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.

- **N-dimensional Array**: Efficiently stores large datasets.
- **Mathematical Functions**: Extensive library of functions for performing mathematical operations.
- **Broadcasting:** Allows arithmetic operations on arrays of different shapes.`
- **Integration:** Works seamlessly with other scientific computing libraries like SciPy and Matplotlib.

NumPy forms the backbone of numerical computations in Python and is widely used in data analysis and machine learning.

4.5 PICKLE

Pickle is a Python module used for serializing and deserializing Python object structures, also called marshalling or flattening.

- **Serialization:** Converts a Python object into a byte stream.
- **Deserialization:** Converts a byte stream back into a Python object.
- **Storage:** Allows objects to be stored in files or databases and later retrieved.
- **Simple Usage:** Easy to use with a few lines of code.

Pickle is useful for saving machine learning models and other Python objects for later use, ensuring the persistence of data across sessions.

House Price Prediction Model of Indian Metropolitan Cities

The house price prediction system aims to accurately forecast real estate prices based on various factors such as location, property size, number of rooms, and other relevant attributes. This system will be useful for potential buyers, sellers, and real estate agents who need to make informed decisions.

Overview

- **Objective:** Develop a model to predict house prices with high accuracy.
- **Technologies:** Python, XGBoost, Pandas, NumPy, and Pickle.
- **Functionality:** Users input property details through a user-friendly interface, and the system displays the predicted price based on the input details. Visualization of prediction results and historical data is provided for better understanding.

This house price prediction system leverages the power of machine learning and Python's robust ecosystem to deliver reliable and accurate property price forecasts.

5. CODE

Bangalore_Model

"Predicting House Prices in Indian Metropolitan Cities: A Comprehensive Analysis and Price Forecasting Model"

"House Price Prediction Model for Bangalore, India"

Components of the House Price Prediction Model:

1. Importing Dependencies
2. Data Importing
3. Preprocessing
4. Visualization
5. Standardizing the Data
6. Label Encoding
7. Model Training
8. Model Deployment

1. Importing dependencies.

We imported NumPy and Pandas for data handling, Matplotlib and Seaborn for visualization, and essential machine learning tools such as RandomForestRegressor and XGBRegressor for predictive modeling. Additionally, we included preprocessing tools like StandardScaler and LabelEncoder for data preprocessing and evaluation metrics like r2_score for assessing model performance and pickle for deployment purpose.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.preprocessing import StandardScaler,LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
import pickle
```

[1] Python

2. Importing Bangalore House Price Dataset.

```
data = pd.read_csv('C:/Users/aksha/OneDrive/Documents/SRM_Internship_project/Indian_cities/Bangalore.csv')
```

[2] Python

3. Preprocessing

3.1 We quickly review a subset of the dataset using to gain insights into its contents.

```
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
data.head()
```

Python

	Price	Area	Location	No. of Bedrooms	Resale	MaintenanceStaff	Gymnasium	SwimmingPool	LandscapedGardens	JoggingTrack	RainWaterHarvesting	In
0	30000000	3340	JP Nagar Phase 1	4	0	1	1	1	1	1	1	
1	7888000	1045	Dasarahalli on Tumkur Road	2	0	0	1	1	1	1	1	
2	4866000	1179	Kannur on Thanisandra Main Road	2	0	0	1	1	1	1	1	
3	8358000	1675	Doddanekundi	3	0	0	0	0	0	0	0	
4	6845000	1670	Kengeri	3	0	1	1	1	1	1	1	

Overview of Each Column:

1. **Price**: Target variable representing the price of the property.
2. **Area**: Non-categorical column indicating the area (size) of the property.
3. **Location**: Non-categorical column indicating the location of the property.
4. **No. of Bedrooms**: Number of bedrooms in the property.
5. **Resale**: Binary indicator (0 or 1) representing whether the property is a resale.
6. **MaintenanceStaff**: Binary indicator representing the availability of maintenance staff.
7. **Gymnasium**: Binary indicator representing the availability of a gymnasium.
8. **SwimmingPool**: Binary indicator representing the availability of a swimming pool.
9. **LandscapedGardens**: Binary indicator representing the availability of landscaped gardens.
10. **JoggingTrack**: Binary indicator representing the availability of a jogging track.
11. **RainWaterHarvesting**: Binary indicator representing the availability of rainwater harvesting.
12. **IndoorGames**: Binary indicator representing the availability of indoor games facilities.
13. **ShoppingMall**: Binary indicator representing the proximity to a shopping mall.
14. **Intercom**: Binary indicator representing the availability of an intercom system.
15. **SportsFacility**: Binary indicator representing the availability of sports facilities.
16. **ATM**: Binary indicator representing the proximity to an ATM.
17. **ClubHouse**: Binary indicator representing the availability of a clubhouse.
18. **School**: Binary indicator representing the proximity to a school.
19. **24X7Security**: Binary indicator representing the availability of 24x7 security.
20. **PowerBackup**: Binary indicator representing the availability of power backup.
21. **CarParking**: Binary indicator representing the availability of car parking.
22. **StaffQuarter**: Binary indicator representing the availability of staff quarters.
23. **Cafeteria**: Binary indicator representing the availability of a cafeteria.
24. **MultipurposeRoom**: Binary indicator representing the availability of a multipurpose room.
25. **Hospital**: Binary indicator representing the proximity to a hospital.
26. **WashingMachine**: Binary indicator representing the availability of a washing machine.
27. **Gasconnection**: Binary indicator representing the availability of a gas connection.
28. **AC**: Binary indicator representing the availability of air conditioning.
29. **Wifi**: Binary indicator representing the availability of Wi-Fi.
30. **Children'splayarea**: Binary indicator representing the availability of a children's play area.
31. **LiftAvailable**: Binary indicator representing the availability of a lift.
32. **BED**: Binary indicator representing the availability of a bed.
33. **VaastuCompliant**: Binary indicator representing whether the property is Vaastu compliant.
34. **Microwave**: Binary indicator representing the availability of a microwave.
35. **GolfCourse**: Binary indicator representing the proximity to a golf course.
36. **TV**: Binary indicator representing the availability of a television.
37. **DiningTable**: Binary indicator representing the availability of a dining table.
38. **Sofa**: Binary indicator representing the availability of a sofa.
39. **Wardrobe**: Binary indicator representing the availability of a wardrobe.
40. **Refrigerator**: Binary indicator representing the availability of a refrigerator.

In this dataset, 'Price' is the target variable we aim to predict, while 'Area' and 'Location' are the non-categorical columns. All other columns represent categorical variables indicating various features or amenities of the properties.

3.2 We're counting how many rows and columns there are in the dataset.

```
data.shape
```

Python

```
(1951, 40)
```

- No of rows in dataset = 1951
- No of columns in dataset = 40

3.3 Stroing name of columns in columns variable. This code snippet iterates through each column in the dataset, excluding 'Price', 'Area', and 'Location'. For each column, it prints the count of unique values and a separator.This code checks if any columns contain only zeros as categorical values and drops them if found.

+ Code + Markdown

```
columns = data.columns
```

Python

```
for column in columns:
    if column == 'Price':
        continue
    elif column == 'Area':
        continue
    elif column == 'Location':
        continue
    else:
        print(column)
        print(data[column].value_counts(),'\n======================\n')
```

Python

```
No. of Bedrooms
No. of Bedrooms
2    888
3    879
4    101
1     73
5     10
Name: count, dtype: int64
======================

Resale
Resale
0    1771
1     180
Name: count, dtype: int64
======================
```

3.4 Now we will take a look on statistical data

```
data.describe()
```

Python

	Price	Area	No. of Bedrooms	Resale	MaintenanceStaff	Gymnasium	SwimmingPool	LandscapedGardens	JoggingTrack	RainWaterHarvesting	IndoorGames	ShoppingMall	Intercom	SportsFacility	ATM	Cl
count	1.951000e+03	1951.000000	1951.000000	1951.000000	1951.000000	1951.000000	1951.000000	1951.000000	1951.000000	1951.000000	1951.000000	1951.000000	1951.000000	1951.000000	1951.000000	195
mean	9.953003e+06	1515.602768	2.532035	0.092260	0.119938	0.924141	0.843157	0.672476	0.717581	0.645310	0.564839	0.114300	0.791902	0.512558	0.124552	
std	1.293016e+07	764.069332	0.677002	0.289467	0.324973	0.264840	0.363746	0.469431	0.450292	0.478542	0.495905	0.318257	0.406052	0.499970	0.330294	
min	2.096000e+06	525.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	4.973500e+06	1147.000000	2.000000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	
50%	6.950000e+06	1330.000000	3.000000	0.000000	0.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	1.000000	1.000000	0.000000	
75%	1.000000e+07	1610.000000	3.000000	0.000000	0.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	1.000000	1.000000	0.000000	
max	2.027000e+08	9900.000000	5.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	

```
data.info()
```

Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1951 entries, 0 to 1950
Data columns (total 38 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Price                1951 non-null   int64
 1   Area                 1951 non-null   int64
 2   Location             1951 non-null   object
 3   No. of Bedrooms      1951 non-null   int64
 4   Resale               1951 non-null   int64
 5   MaintenanceStaff     1951 non-null   int64
 6   Gymnasium            1951 non-null   int64
 7   SwimmingPool         1951 non-null   int64
 8   LandscapedGardens    1951 non-null   int64
 9   JoggingTrack         1951 non-null   int64
 10  RainWaterHarvesting  1951 non-null   int64
 11  IndoorGames          1951 non-null   int64
```

key observations:

1. The 'Location' column is the only one with a data type of 'object', while all other columns are of type 'int'.
2. The lowest price recorded in the dataset is 2 million INR.
3. The highest price recorded in the dataset is 202 million INR.
4. The smallest area of a house in the dataset is 525 square units.
5. The largest area of a house in the dataset is 99,000 square units.

3.5 Checking Is there any null values.

```
data.isnull().sum()
```

```
Price                 0
Area                  0
Location              0
No. of Bedrooms       0
Resale                0
MaintenanceStaff      0
Gymnasium             0
SwimmingPool          0
LandscapedGardens     0
JoggingTrack          0
RainWaterHarvesting   0
IndoorGames           0
ShoppingMall          0
Intercom              0
SportsFacility        0
ATM                   0
ClubHouse             0
School                0
24X7Security          0
PowerBackup           0
CarParking            0
StaffQuarter          0
Cafeteria             0
MultipurposeRoom      0
Hospital              0
...
TV                    0
DiningTable           0
```

3.6 'Counting Houses by Location'

```
pd.set_option('display.max_rows', None)
data.Location.value_counts()
```

Python

```
Location
RR Nagar                                143
Begur                                    96
Electronic City Phase 1                  74
Bommasandra                              69
Electronic City Phase 2                  69
Whitefield Hope Farm Junction            65
Varthur                                  59
Kanakapura Road Beyond Nice Ring Road    49
Horamavu                                 46
Narayanapura on Hennur Main Road         45
Krishnarajapura                          45
JP Nagar Phase 3                         43
Anagalapura Near Hennur Main Road        40
Kannur on Thanisandra Main Road          37
ITPL                                     35
Harlur                                   35
Hebbal                                   34
Attibele                                 33
JP Nagar Phase 8                         31
Muneshwara Nagar                         28
Jalahalli                                28
Devanahalli                              28
Uttarahalli                              27
Anjanapura                               27
...
Kannamangala                              1
Padmanabhanagar                           1
Amruthahalli                              1
RMV                                       1
Name: count, dtype: int64
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

3.7 Identifying Outliers: Locations with Few Houses Many locations have only a few houses, which may act as outliers in the data and potentially impact our model's performance. To address this, we will group locations with fewer than 11 houses together and categorize them under a separate variable.

```
location_count = data['Location'].value_counts()
location_count_less_10 = location_count[location_count<=10]
```

[12] Python

```
data['Location'] = data['Location'].apply(lambda x: 'other' if x in location_count_less_10 else x)
data['Location'].value_counts()
```

[13] Python

```
Location
other                                 213
RR Nagar                              143
Begur                                  96
Electronic City Phase 1                74
Bommasandra                            69
Electronic City Phase 2                69
Whitefield Hope Farm Junction          65
Varthur                                59
Kanakapura Road Beyond Nice Ring Road  49
Horamavu                               46
Narayanapura on Hennur Main Road       45
Krishnarajapura                        45
JP Nagar Phase 3                       43
Anagalapura Near Hennur Main Road      40
Kannur on Thanisandra Main Road        37
Harlur                                 35
ITPL                                   35
Hebbal                                 34
Attibele                               33
JP Nagar Phase 8                       31
Muneshwara Nagar                       28
Devanahalli                            28
Jalahalli                              28
Uttarahalli                            27
...
Banashankari                           12
JP Nagar Phase 7                       11
Nayandahalli                           11
```

4. Visualization

4.1 This code generates a heatmap to visualize the correlation between different features in the dataset, excluding the 'Location' column. Each cell in the heatmap represents the correlation coefficient between two features, with annotations provided to indicate the strength of correlation.

```
corr_data = data.drop(columns='Location')
```

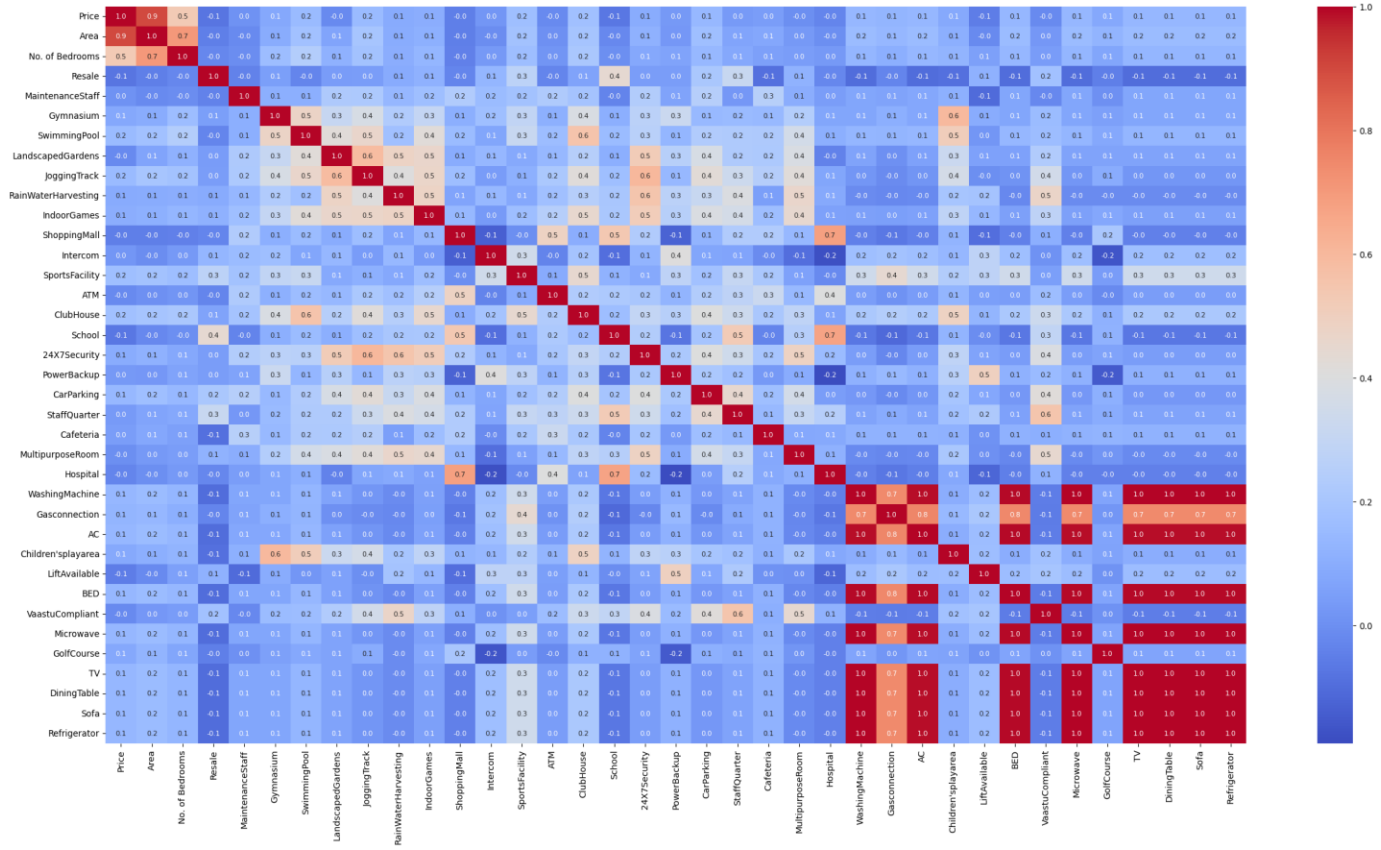
[14] Python

```
corr = corr_data.corr()
```

[15] Python

```
plt.figure(figsize=(30,15))
sns.heatmap(corr,fmt='.1f',annot=True,annot_kws={'size':8} ,cmap='coolwarm')
```

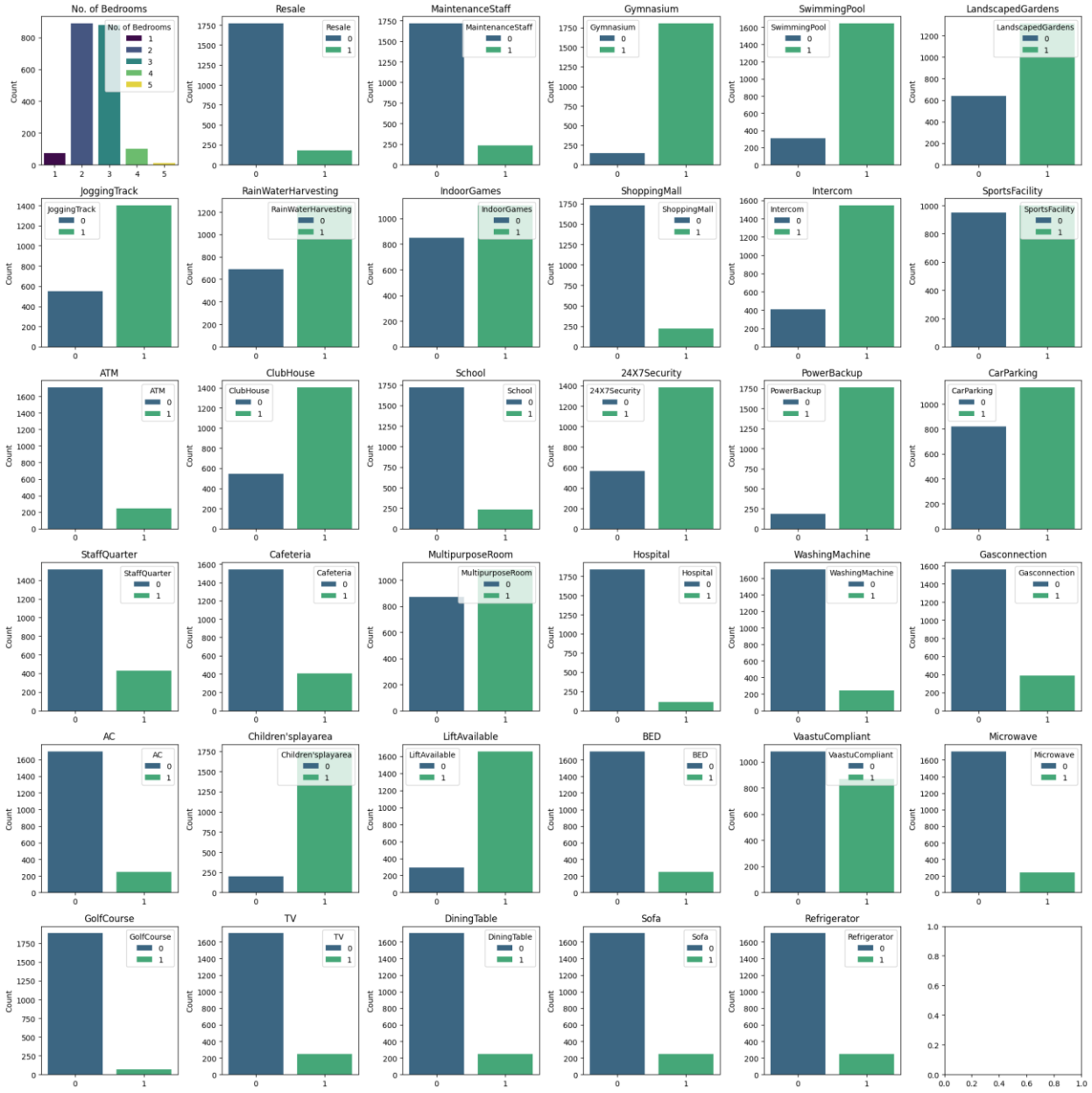
[16] Python



Insights from the Heat Map:

1. Dark brown color indicates strong positive correlation, while lighter brown or blue colors signify weaker positive correlation. Dark blue represents strong negative correlation.
2. For example:
   - There is a strong positive correlation between 'Sofa' and 'TV', which is expected as these items are often found together in a household.
   - Like this We can see positive or negative correlation between various features.

```
import seaborn as sns
import matplotlib.pyplot as plt

# Selecting only the categorical columns
categorical_columns = ['No. of Bedrooms', 'Resale', 'MaintenanceStaff', 'Gymnasium', 'SwimmingPool', 'LandscapedGardens',
                       'JoggingTrack', 'RainWaterHarvesting', 'IndoorGames', 'ShoppingMall', 'Intercom', 'SportsFacility',
                       'ATM', 'ClubHouse', 'School', '24X7Security', 'PowerBackup', 'CarParking', 'StaffQuarter',
                       'Cafeteria', 'MultipurposeRoom', 'Hospital', 'WashingMachine', 'Gasconnection', 'AC',
                       "Children'splayarea", 'LiftAvailable', 'BED', 'VaastuCompliant', 'Microwave', 'GolfCourse',
                       'TV', 'DiningTable', 'Sofa', 'Refrigerator']

# Plotting count plots for each categorical variable
fig, axes = plt.subplots(nrows=6, ncols=6, figsize=(20, 20))

for i, column in enumerate(categorical_columns):
    sns.countplot(x=column, data=data, ax=axes[i//6, i%6],palette='viridis',hue=column)
    axes[i//6, i%6].set_title(column)
    axes[i//6, i%6].set_xlabel('')
    axes[i//6, i%6].set_ylabel('Count')

plt.tight_layout()
plt.show()
```

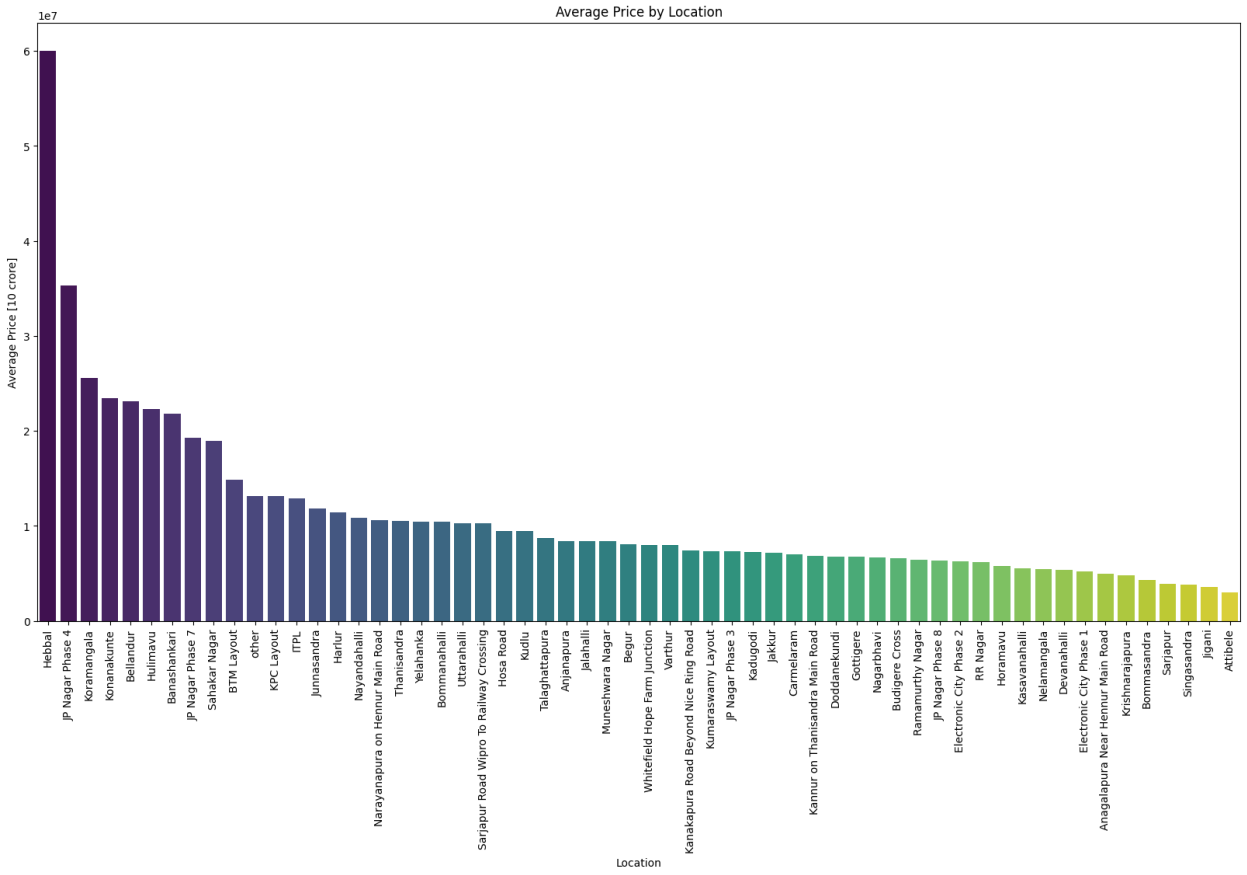


Insights from Count Graphs:

1. The dataset includes houses with a varying number of rooms, ranging from 1 to 8.
2. Observing the count plots for features such as 'Hospital','Golfcourse','Gymnasium', we notice significant uneven distribution or skewness in the data. means all this facility are not provided in most of cases while purchasing of a housing.

4.3 "Plotting a Bar Graph: Mean House Prices by Location"

```
import seaborn as sns
import matplotlib.pyplot as plt

# Aggregate data by location (e.g., by taking the mean price for each location)
average_prices = data.groupby('Location')['Price'].mean().reset_index()

# Sort the data by average price
average_prices = average_prices.sort_values(by='Price', ascending=False)

# Plotting the bar plot
plt.figure(figsize=(20, 10))
sns.barplot(x='Location', y='Price', data=average_prices,palette='viridis')
plt.xticks(rotation=90)  # Rotate x-axis labels for better readability
plt.xlabel('Location')
plt.ylabel('Average Price [10 crore]')
plt.title('Average Price by Location')
plt.show()
```



Insights:

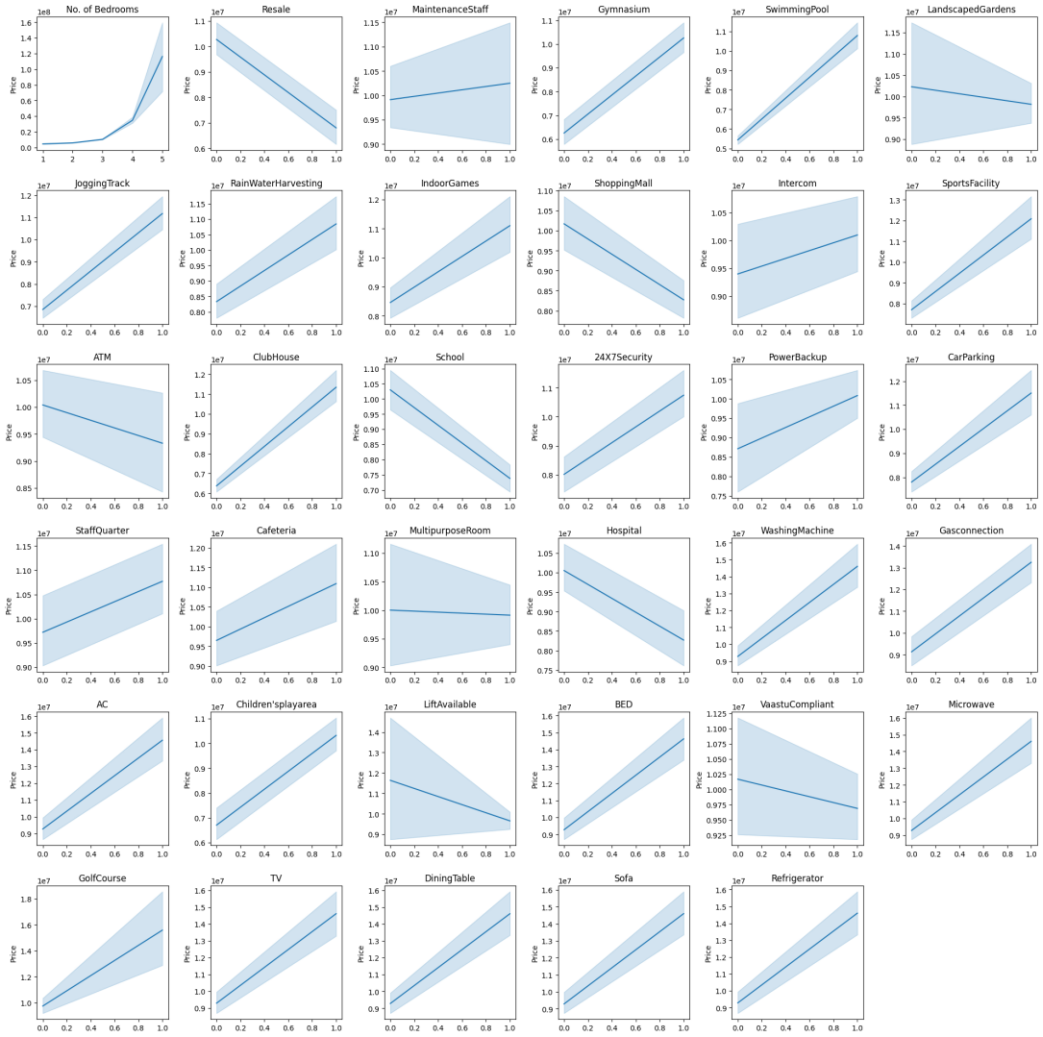
1. The most affluent areas, such as Banjara Hills, Jubilee Hills, and Madhapur, are likely the luxurious neighborhoods of Bangalore, while areas with the least expensive houses include Krishna Reddy and Adibatla.
2. The majority of house prices fall within the range of 5 million to 15 million INR.

+ Code + Markdown

4.4 Line plot showing relation between each feature and price

```python
import seaborn as sns
import matplotlib.pyplot as plt

# Setting up the figure with appropriate size
plt.figure(figsize=(20, 20))

# Plotting line plots for each categorical variable against Price
for i, column in enumerate(categorical_columns):
    plt.subplot(6, 6, i+1)
    sns.lineplot(x=data[column], y=data['Price'], estimator='mean')  # You can change 'mean' to 'median' if you prefer
    plt.title(column)
    plt.xlabel('')
    plt.ylabel('Price')

plt.tight_layout()
plt.show()
```

Python



Insights:

1. The line plots display shadows, with wider shadows behind the line indicating a higher concentration of data points in that range, and vice versa.
2. The presence or absence of features such as 'MultipurposeRoom','MaintenanceStaff','LandscapedGardens' near the house does not have a significant impact on price. Hence, in the next step, we will drop these columns to simplify the model and reduce complexity.
3. Most features show a positive correlation with price, implying that the presence of any feature tends to increase the price of the house.
4. People of Bangalore are much aware of vaastu, if there any complaint of vastu the price of hoouse decreases.
5. The combined availability of a lift, hospital, school, shopping mall, and ATM could decrease property prices due to increased noise, traffic congestion, security concerns, diminished privacy, potential for higher crime rates, and reduced parking availability.

```python
# multipurpose , maintanance staff, landscape garder
data = data.drop(columns=['MultipurposeRoom','MaintenanceStaff','LandscapedGardens'])
```

Python

5 Standardizing the Data

5.1 Since we only have the 'Area' column as the feature with numerical or continuous values, we will standardize this column only.

```
[22] std = StandardScaler()
```
Python

```
[23] data['Area'] = std.fit_transform(data[['Area']])
```
Python

6 LabelEncoding

```
[24] len(data['Location'].value_counts())
```
Python

```
... 58
```

6.1 Assigning Numerical Values to Locations

- We have 58 unique locations repeated across all 2434 data points. Label encoding will assign a unique numerical value to each location, facilitating numerical processing by machine learning models. This transformation is crucial as models require numerical input rather than strings, ensuring efficient data interpretation and model performance.

```
[25] encoder = LabelEncoder()
```
Python

```
[26] data['Location'] = encoder.fit_transform(data['Location'])
```
Python

Splitting Data

```
[27] x = data.drop(columns='Price',axis=1)
     y = data['Price']
```

```
[28] xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.2,random_state=2)
```

7. Model Selection And Training

7.1 Defining Model Evaluation Function:

- We are utilizing two models, RandomForestRegressor and XGBRegressor, to predict housing prices. A function is defined to take features and corresponding labels, assessing their accuracy using the R2 score metric.

```
[29] models = [RandomForestRegressor(),XGBRegressor()]

     def accuracy(columns,target):
         for model in models:
             print(model)
             model.fit(columns,target)
             pred = model.predict(columns)
             accuracy = r2_score(target,pred)
             print('R2 score is ',accuracy,'\n----------------------------------------------------------------------------------')
```

7.2 Evaluating Models with Train Data:

- Initially, we calculate the R2 score using the training data itself. This step is crucial to ensure that the models do not overfit and can generalize well to unseen data.

+ Code + Markdown

```
print('====================================')
print('CHECHKING ACCURACY SCORES OF MODEL')
print('====================================\n')
accuracy(xtrain,ytrain)
```

```
====================================
CHECHKING ACCURACY SCORES OF MODEL
====================================

RandomForestRegressor()
R2 score is  0.99029539683593
----------------------------------------------------------------------------------
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=None, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             multi_strategy=None, n_estimators=None, n_jobs=None,
             num_parallel_tree=None, random_state=None, ...)
R2 score is  0.9989331862285571
----------------------------------------------------------------------------------
```

7.3 Evaluating Models with Train and Test Data:

- Initially, we calculate the R2 score using the training data itself. Similarly, we evaluate the models using the test data to confirm their performance on unseen data.

```
print('=============================================================================================================')
print('CHECHKING ACCURACY SCORES OF MODEL')
print('=============================================================================================================\n')
accuracy(xtest,ytest)
```

```
=============================================================================================================
CHECHKING ACCURACY SCORES OF MODEL
=============================================================================================================

RandomForestRegressor()
R2 score is  0.9847882079253627
-------------------------------------------------------------------------------
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=None, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             multi_strategy=None, n_estimators=None, n_jobs=None,
             num_parallel_tree=None, random_state=None, ...)
R2 score is  0.9999714531030359
-------------------------------------------------------------------------------
```

Insights-

In both training and testing data , the accuracy of XGBregressor is high, thus we will take this model for deployment.

7.4 Final model - XGBRegressor

```
model = XGBRegressor()
```

Python

```
model.fit(xtrain,ytrain)
```

Python

XGBRegressor

```
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=None, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             multi_strategy=None, n_estimators=None, n_jobs=None,
             num_parallel_tree=None, random_state=None, ...)
```

7.5 Testing Model Prediction on Random Data:

- To assess the model's predictive capability, we randomly select a data point using data.sample() and provide its features to our model. We then evaluate whether the model accurately predicts the corresponding label for this data point.

```
# pd.set_option('display.max_rows', None)
# pd.set_option('display.max_columns', None)
# data.sample()
```

Python

8 Ready for Deployment

- After thorough testing, our model demonstrates the ability to make accurate predictions. With satisfactory performance, the model is now deemed ready for deployment .

8.1 Saving the Model.

```
filename = 'Bangalore_model.sav'
pickle.dump(model,open(filename,'wb'))
```

Python

8.2 Saving the Standard scaler function.

```
standard = 'StandardScaler_Bangalore.sav'
pickle.dump(std,open(standard,'wb'))
```

Python

8.3 Saving the Label Encoder Function.

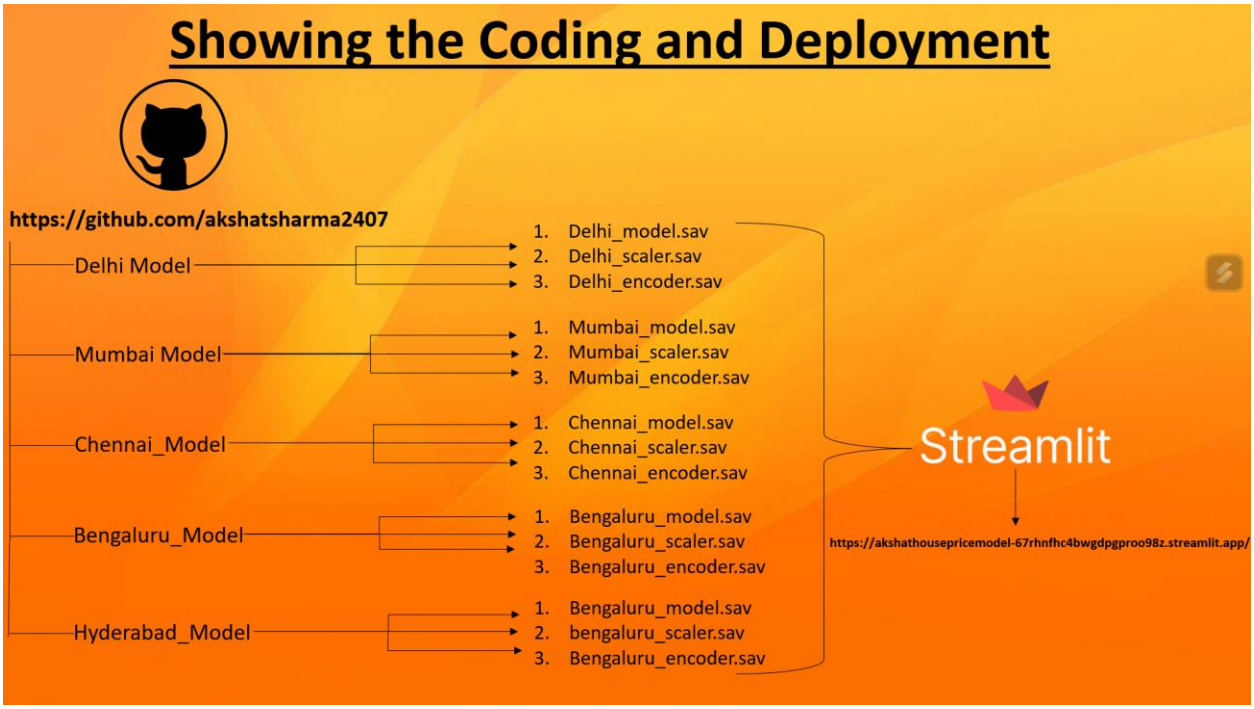
```
encoding = 'encoder_bangalore.sav'
pickle.dump(encoder,open(encoding,'wb'))
```

Python

The same Data cleaning,Visualization and model training and evaluation is done for Delhi,Mumbai,Hyderabad,Chennai which can be seen in by github profile-

GITHUB LINK - akshatsharma2407 · GitHub

STREAMLIT (WEB APP) - **https://akshathousepricemodel67rhnfhc4bwgdpgproo98z.streamlit.app/**



WEB APP FOR FINAL DEPLOYMENT – USING STREAMLIT

Code for Streamlit

```
# -*- coding: utf-8 -*-

import pickle
import streamlit as st
from streamlit_option_menu import option_menu
import numpy as np

#loading the models

delhi_model = pickle.load(open('Delhi_model.sav','rb'))
delhi_scaler = pickle.load(open("StandardScaler_Delhi.sav",'rb'))
delhi_encoder = pickle.load(open("encoder_Delhi.sav",'rb'))

mumbai_model = pickle.load(open("mumbai_model.sav",'rb'))
mumbai_scaler = pickle.load(open("StandardScaler_mumbai.sav",'rb'))
mumbai_encoder = pickle.load(open("encoder_mumbai.sav",'rb'))

chennai_model = pickle.load(open("Chennai_model.sav",'rb'))
chennai_scaler = pickle.load(open("StandardScaler_Chennai.sav",'rb'))
chennai_encoder = pickle.load(open("encoder_Chennai.sav",'rb'))

bangalore_model = pickle.load(open("Bangalore_model.sav",'rb'))
bangalore_scaler = pickle.load(open("StandardScaler_Bangalore.sav",'rb'))
bangalore_encoder = pickle.load(open("encoder_bangalore.sav",'rb'))

Hyderabad_model = pickle.load(open("Hyderabad_model.sav",'rb'))
Hyderabad_scaler = pickle.load(open("StandardScaler_Hyderabad.sav",'rb'))
Hyderabad_encoder = pickle.load(open('encoder_Hyderabad.sav','rb'))



with st.sidebar:
    selected = option_menu('House Price Prediction of Indian Metropolitan Cities',
                           [ 'Home Page',
```

```
                'Chennai Model',
                'Delhi Model',
                'Mumbai Model',
                'Bengaluru Model',
                'Hyderabad Model'],
                icons=['house-fill','building-fill','buildings','buildings-fill','houses-fill','buildings'],
                default_index=0)


if (selected == 'Home Page'):
    st.title("MetropolitanHouse: Predicting Real Estate Prices in Indian Urban Centers")
    st.image('House_steamlit.jpg', use_column_width=True)
    st.write("This app predicts house prices of Indian metropolitan cities using machine learning and Python code.")
    st.write('Welcome to our House Price Prediction App! Whether you are a prospective buyer eager to find your dream home or a seller curious about the market value of your property, our innovative tool is designed just for you. Powered by cutting-edge machine learning algorithms and crafted with precision in Python, our model considers a myriad of factors, from the area and location to the number of bedrooms and a plethora of amenities including gyms, pools, and landscaped gardens. With our intuitive interface, making informed decisions about real estate has never been easier. Join us on this journey as we revolutionize the way you navigate the housing market, empowering you to buy and sell with confidence.')
    st.write("Our prediction model takes into account several features such as area, location, number of bedrooms, resale value, and various amenities.")
    st.write("These amenities include maintenance staff, gymnasium, swimming pool, landscaped gardens, jogging track, rainwater harvesting, indoor games, shopping mall, intercom, sports facility, ATM, club house, 24x7 security, power backup, car parking, staff quarter, cafeteria, multipurpose room, hospital facilities, washing machine, gas connection, air conditioning, children's play area, availability of lift, vaastu compliance, microwave, TV, dining table, sofa, refrigerator, and more.")


if (selected == 'Chennai Model'):
    st.title('Chennai House Price Prediction Model Using Machine Learning')
    st.image("chennai_temple.jpeg", use_column_width=True)
    # Area, Location, and Number of Bedrooms
    st.header("Basic Information")
    area = st.text_input('Area')
    location = st.selectbox('Location',['Perungalathur', 'Madhavaram', 'Karapakkam', 'Thiruvidandhai',
      'Iyappanthangal', 'Mevalurkuppam', 'Kolapakkam', 'Kundrathur',
      'Pammal', 'Puzhal', 'Selaiyur', 'Thoraipakkam OMR', 'Anna Nagar',
      'Mogappair', 'Sholinganallur', 'Medavakkam', 'Avadi',
      'Tiruvottiyur', 'Manapakkam', 'Madipakkam', 'Thiruvanmiyur',
      'Ramapuram', 'Saidapet', 'Poonamallee', 'Pallavaram',
      'Maraimalai Nagar', 'Madambakkam', 'Perungudi', 'Villivakkam',
      'Adyar', 'Navallur', 'Moolacheri', 'Chromepet', 'Nandambakkam',
      'Kelambakkam', 'Vadapalani', 'Kumananchavadi', 'Porur',
```

```
    'Periyapanicheri', 'Manikandan Nagar', 'Kodambakkam', 'Velachery',
    'East Tambaram', 'Gopalapuram', 'Sunnambu Kolathur S Kolathur',
    'Perumbakkam', 'Cholambedu', 'Urapakkam', 'Raja Annamalai Puram',
    'Besant Nagar', 'Peerakankaranai', 'Nanmangalam', 'Jamalia',
    'Guduvancheri', 'Sembakkam', 'Adambakkam', 'Nungambakkam',
    'T Nagar', 'K K Nagar', 'Ambattur', 'Valasaravakkam',
    'Kanathur Reddikuppam', 'Mugalivakkam', 'Purasaiwakkam',
    'Maduravoyal', 'Gowrivakkam', 'Mudichur', 'West Tambaram',
    'Alwarpet', 'Annanagar West', 'Thiruverkadu', 'tambaram west',
    'Guindy', 'Korattur', 'Tambaram Sanatoruim', 'Irumbuliyur',
    'Kolathur', 'Thirumullaivoyal', 'Singaperumal Koil', 'Ayapakkam',
    'Perambur', 'Chetpet', 'Kilpauk', 'Egmore', 'Alandur', 'Kovur',
    'Vandalur', 'Pozhichalur', 'Vanagaram', 'Thoraipakkam',
    'Ullagaram', 'Kovilambakkam', 'Kattupakkam', 'Thirumazhisai',
    'Kattankulathur', 'Ayanambakkam', 'Sithalapakkam', 'Vengaivasal',
    'Kilkattalai', 'Annanagar', 'Chengalpattu', 'Pallikaranai',
    'Rajakilpakkam', 'Chitlapakkam', 'Palavakkam', 'Kotturpuram',
    'Nenmeli', 'Ramavaram', 'Padi', 'NehruNagar', 'Pazavanthangal',
    'Thatchoor', 'Padur', 'Iyyappanthangal', 'Mambakkam', 'Egatoor',
    'Semmancheri', 'Virugambakkam', 'Moolakadai', 'Siruseri',
    'Velappanchavadi', 'Ekkatuthangal', 'Royapettah', 'Nandanam',
    'Vellakkal', 'Annamalai Colony', 'Thalambur', 'Nanganallur',
    'Chembarambakkam', 'Teynampet', 'Mannivakkam', 'Thaiyur',
    'Injambakkam', 'Aminjikarai', 'CIT Nagar', 'Koyambedu',
    'Kil Ayanambakkam', 'Choolaimedu'])
num_bedrooms = st.selectbox('Number of Bedrooms', [1, 2, 3, 4, 5])

# Amenities in two columns
st.header("Amenities")
col1, col2 = st.columns(2)

# Column 1
with col1:
    maintenance_staff = st.checkbox('Maintenance Staff')
    gymnasium = st.checkbox('Gymnasium')
    swimming_pool = st.checkbox('Swimming Pool')
    landscaped_gardens = st.checkbox('Landscaped Gardens')
```

```
    jogging_track = st.checkbox('Jogging Track')
    rainwater_harvesting = st.checkbox('Rainwater Harvesting')
    indoor_games = st.checkbox('Indoor Games')
    shopping_mall = st.checkbox('Shopping Mall')
    intercom = st.checkbox('Intercom')
    sports_facility = st.checkbox('Sports Facility')
    atm = st.checkbox('ATM')
    club_house = st.checkbox('Club House')
    dining_table = st.checkbox('Dining Table')
    sofa = st.checkbox('Sofa')
    refrigerator = st.checkbox('Refrigerator')
    tv = st.checkbox('TV')

    # Column 2
with col2:
    security_24x7 = st.checkbox('24X7 Security')
    power_backup = st.checkbox('Power Backup')
    car_parking = st.checkbox('Car Parking')
    staff_quarter = st.checkbox('Staff Quarter')
    cafeteria = st.checkbox('Cafeteria')
    multipurpose_room = st.checkbox('Multipurpose Room')
    hospital = st.checkbox('Hospital')
    washing_machine = st.checkbox('Washing Machine')
    gas_connection = st.checkbox('Gas Connection')
    ac = st.checkbox('AC')
    children_play_area = st.checkbox("Children's Play Area")
    lift_available = st.checkbox('Lift Available')
    bed = st.checkbox('BED')
    vaastu_compliant = st.checkbox('Vaastu Compliant')
    microwave = st.checkbox('Microwave')
    resale = st.checkbox('Resale')

price_predict = ''

if st.button('Predict price'):

    # Transform area using scaler
```

```
        area = chennai_scaler.transform(np.array([[float(area)]]))  # Convert to float and 2D array

            # Transform location using encoder
        try:
            location = chennai_encoder.transform([location])[0]  # Ensure to access the first element of the resulting array
        except:
            location = 'other'
            location = chennai_encoder.transform([location])[0]  # Ensure to access the first element of the resulting array

        # Convert input data to array
        input_data = np.array([
            area[0][0], location, num_bedrooms, resale, maintenance_staff, gymnasium, swimming_pool, landscaped_gardens,
            jogging_track, rainwater_harvesting, indoor_games, shopping_mall, intercom, sports_facility, atm,
            club_house, security_24x7, power_backup, car_parking, staff_quarter, cafeteria, multipurpose_room,
            hospital, washing_machine, gas_connection, ac, children_play_area, lift_available, bed, vaastu_compliant,
            microwave, tv, dining_table, sofa, refrigerator
        ]).reshape(1, -1)

        # Predict price using the model
        price_predict = chennai_model.predict(input_data)
    st.success(price_predict)




if (selected == 'Delhi Model'):
    st.title('Delhi House Price Prediction Model Using Machine Learning')
    st.image("Inida_gate.jpg", use_column_width=True)
    # Area, Location, and Number of Bedrooms
    st.header("Basic Information")
    area = st.text_input('Area')
    location = st.selectbox('Location',['Sector 10 Dwarka', 'Uttam Nagar', 'Sarita Vihar', 'Dwarka Mor',
        'Sector 7 Dwarka', 'Sector 6 Dwarka', 'Sector 5 Dwarka',
        'Sector 23 Rohini', 'Mayur Vihar II', 'Sector 24 Rohini',
        'Sector 11 Dwarka', 'Sector 23 Dwarka', 'Sector 12 Dwarka',
        'West End', 'Sector 9 Rohini', 'Mundka', 'Sector 13 Rohini',
        'Jamia Nagar', 'Sector 19 Dwarka', 'Sector 17 Dwarka', 'Bindapur',
        'Sector-18 Dwarka', 'Vasant Kunj', 'Shastri Nagar',
```

```
    'Sector-8 Rohini', 'Sector 9 Dwarka', 'Shanti Park Dwarka',
    'Govindpuri', 'Sector 22 Dwarka', 'Matiala', 'Saket',
    'Mahavir Enclave', 'Burari', 'Shahdara', 'Babarpur', 'Khanpur',
    'Sector 13 Dwarka', 'Mansa Ram Park', 'Green Park', 'Kalkaji',
    'Sector 4 Dwarka', 'DLF Phase 5', 'Sector 3 Dwarka',
    'Chittaranjan Park', 'Chattarpur', 'Greater Kailash',
    'Sector-14 Rohini', 'Paschim Vihar', 'Pitampura',
    'Sector 18B Dwarka', 'Sector 2 Dwarka', 'Jasola',
    'Pochanpur Colony', 'Palam', 'Saidabad', 'Budh Vihar',
    'Sector 25 Rohini', 'Sector 18A Dwarka', 'Sewak Park',
    'Sector 23B Dwarka', 'Rohini sector 24', 'Sector 28 Rohini',
    'Rohini Sector 9', 'Rohini Extension', 'nawada', 'Alaknanda',
    'Sector 22 Rohini', 'Lajpat Nagar', 'South Extension 2',
    'Sector 16B Dwarka', 'Sheikh Sarai', 'Sidhartha Nagar',
    'Sector-D Vasant Kunj', 'Hauz Khas', 'Kalkaji Extension',
    'Greater kailash 1', 'Lajpat Nagar III', 'Safdarjung Enclave',
    'Greater Kailash II', 'Sainik Farms', 'Sector 20 Rohini',
    'greater kailash Enclave 1', 'DLF Farms', 'Mehrauli', 'Mahipalpur',
    'mayur vihar phase 1', 'Sarvodaya Enclave', 'Karol Bagh',
    'West Sagarpur', 'Ashok Vihar', 'Sector 21 Dwarka',
    'East of Kailash', 'Khirki Extension', 'Dashrath Puri',
    'SULTANPUR', 'Patparganj', 'Kaushambi', 'Shakurbasti',
    'Hari Nagar', 'Siri Fort', 'Katwaria Sarai', 'Mayur Vihar',
    'Nasirpur'])
num_bedrooms = st.selectbox('Number of Bedrooms', [1, 2, 3, 4, 5])

# Amenities in two columns
st.header("Amenities")
col1, col2 = st.columns(2)

# Column 1
with col1:
    maintenance_staff = st.checkbox('Maintenance Staff')
    gymnasium = st.checkbox('Gymnasium')
    swimming_pool = st.checkbox('Swimming Pool')
    landscaped_gardens = st.checkbox('Landscaped Gardens')
    jogging_track = st.checkbox('Jogging Track')
```

```
    rainwater_harvesting = st.checkbox('Rainwater Harvesting')
    indoor_games = st.checkbox('Indoor Games')
    shopping_mall = st.checkbox('Shopping Mall')
    intercom = st.checkbox('Intercom')
    sports_facility = st.checkbox('Sports Facility')
    atm = st.checkbox('ATM')
    club_house = st.checkbox('Club House')
    school = st.checkbox('School')
    dining_table = st.checkbox('Dining Table')
    sofa = st.checkbox('Sofa')
    refrigerator = st.checkbox('Refrigerator')
    tv = st.checkbox('TV')

    # Column 2
with col2:
    security_24x7 = st.checkbox('24X7 Security')
    power_backup = st.checkbox('Power Backup')
    car_parking = st.checkbox('Car Parking')
    staff_quarter = st.checkbox('Staff Quarter')
    cafeteria = st.checkbox('Cafeteria')
    multipurpose_room = st.checkbox('Multipurpose Room')
    washing_machine = st.checkbox('Washing Machine')
    gas_connection = st.checkbox('Gas Connection')
    ac = st.checkbox('AC')
    children_play_area = st.checkbox("Children's Play Area")
    lift_available = st.checkbox('Lift Available')
    bed = st.checkbox('BED')
    vaastu_compliant = st.checkbox('Vaastu Compliant')
    microwave = st.checkbox('Microwave')
    resale = st.checkbox('Resale')

price_predict = ''

if st.button('Predict price'):

    # Transform area using scaler
    area = delhi_scaler.transform(np.array([[float(area)]]))  # Convert to float and 2D array
```

```
            # Transform location using encoder
        try:
            location = delhi_encoder.transform([location])[0]  # Ensure to access the first element of the resulting array
        except:
            location = 'other'
            location = delhi_encoder.transform([location])[0]  # Ensure to access the first element of the resulting array

        # Convert input data to array
        input_data = np.array([
            area[0][0], location, num_bedrooms, resale, maintenance_staff, gymnasium, swimming_pool, landscaped_gardens,
            jogging_track, rainwater_harvesting, indoor_games, shopping_mall, intercom, sports_facility, atm,
            club_house,school,security_24x7, power_backup, car_parking, staff_quarter, cafeteria, multipurpose_room,
            washing_machine, gas_connection, ac, children_play_area, lift_available, bed, vaastu_compliant,
            microwave, tv, dining_table, sofa, refrigerator
        ]).reshape(1, -1)

        # Predict price using the model
        price_predict = delhi_model.predict(input_data)
    st.success(price_predict)

if (selected == 'Mumbai Model'):
    st.title('Mumbai House Price Prediction Model Using Machine Learning')
    st.image("tajhotel.webp", use_column_width=True)
    st.header("Basic Information")
    area = st.text_input('Area')
    location = st.selectbox('Location',['Kharghar', 'Sector-13 Kharghar', 'Sector 18 Kharghar',
      'Sector 20 Kharghar', 'Sector 15 Kharghar', 'Dombivali',
      'Churchgate', 'Prabhadevi', 'Jogeshwari West', 'Kalyan East',
      'Malad East', 'Virar East', 'Virar', 'Malad West', 'Borivali East',
      'Mira Road East', 'Goregaon West', 'Kandivali West',
      'Borivali West', 'Kandivali East', 'Andheri East', 'Goregaon East',
      'Wadala', 'Ulwe', 'Dahisar', 'kandivali', 'Goregaon',
      'Bhandup West', 'thakur village kandivali east', 'Santacruz West',
      'Kanjurmarg', 'I C Colony', 'Dahisar W', 'Marol', 'Parel',
      'Lower Parel', 'Worli', 'Jogeshwari East', 'Chembur Shell Colony',
      'Central Avenue', 'Chembur East', 'Diamond Market Road', 'Mulund',
```

'Nalasopara West', 'raheja vihar', 'Powai Lake', 'MHADA Colony 20',
'Tolaram Colony', 'Taloja', 'Thane West', 'Vangani',
'Sector 5 Ulwe', 'Sector12 New Panvel', 'Sector 17 Ulwe',
'Sector9 Kamothe', 'Sector 19 Kharghar', 'Navi Basti',
'Sector12 Kamothe', 'Sector 21 Kamothe', 'Rutu Enclave',
'taloja panchanand', 'Virar West', 'Chembur', 'Sector 20 Kamothe',
'Sector 22 Kamothe', 'Sector 18 Kamothe', 'Sector-5 Kamothe',
'Sector-6A Kamothe', 'Sector 11 Kamothe', 'Sector-18 Ulwe',
'Sector-12 Kamothe', 'azad nagar', 'Sindhi Society Chembur',
'Kurla', 'Sahkar Nagar', 'Deonar', 'Thane', 'Jankalyan Nagar',
'Badlapur', 'Ambarnath', 'Ambernath West', 'Vakola', 'Kamothe',
'Kamothe Sector 16', 'Almeida Park', 'Khar', 'Bandra West',
'Pali Hill', '15th Road', 'Palghar', 'Sector13 Kharghar',
'Sector 21 Kharghar', 'Sector 12 Kharghar', 'Vivek Vidyalaya Marg',
'Vasai east', 'Nahur', 'Badlapur West', 'Panvel', 'Kalyan',
'Badlapur East', 'Mira Bhayandar', 'Juhu', 'Naigaon East',
'Sector 21 Ulwe', 'Bandra East', 'Dronagiri', 'Nerul', 'Karanjade',
'Sanpada', 'Sector-8 Ulwe', 'Sector-3 Ulwe', 'Sector 23 Ulwe',
'ULWE SECTOR 19', 'Ghodbunder Road', 'Bhiwandi', 'Vasai',
'Nala Sopara', 'Dadar East', 'Ghatkopar', 'Breach Candy',
'Worli South Mumbai', 'Asangaon', 'Koparkhairane Station Road',
'Kopar Khairane Sector 19A', 'Koper Khairane',
'Eastern Express Highway Vikhroli', 'Magathane', 'Rawal Pada',
'Ambernath East', 'Dokali Pada', 'Dattapada', 'Rajendra Nagar',
'Kulupwadi', 'Samata Nagar Thakur Village', 'Mira Road and Beyond',
'West Amardeep Colony', 'Pant Nagar', 'mumbai', 'Four Bungalows',
'no 9', 'kolshet', 'Hiranandani Meadows', 'Kalpataru', 'Petali',
'Kharghar Sector 34C', 'Ghatkopar East',
'Mumbai Agra National Highway', 'vasant vihar thane west',
'Kalyan West', 'Shirgaon', 'Pokhran 2', 'juhu tara', 'Peddar Road',
'Palm Beach', 'Sector 10', 'Sector 19 Kamothe', 'Tilak Nagar',
'Ghatkopar West', 'Tardeo', 'Napeansea Road', 'Mahalaxmi',
'Dahisar West', 'Mulund West', 'Natakwala Lane', 'Link Road',
'Devidas Cross Lane', 'Soniwadi Road', 'Haridas Nagar', 'Shimpoli',
'TPS Road', 'Off Shimpoli road', 'Rustomjee Global City',
'Sunil Nagar', 'Sector 30 Kharghar', 'Sector 12 A', 'Sector 18',
'Sector13 Khanda Colony', 'Sector16 Airoli', 'Ranjanpada',

```
    'Sector 15', 'Sector 35G', 'Sector 5', 'Sector 35I Kharghar',
    'Sector35D Kharghar', 'Sector34 A Kharghar', 'Sector 30',
    'Sector 36 Kharghar', 'Sector 11 Belapur', 'Sector-34B Kharghar',
    'Dombivali East', 'Roadpali', 'Sector-50 Seawoods',
    'Mumbai Highway', 'Sector 7 Kharghar', 'Lokhandwala Township',
    'Andheri', 'Andheri West', 'Shastri Nagar', 'Wadala East Wadala',
    'Kalwa', 'PARSIK NAGAR', 'Maharashtra Nagar', 'Patlipada',
    'Belapur', 'Seawoods', 'Majiwada', '4 Bunglows', 'Airoli',
    'Kolshet Road', 'Sector 10 Khanda Colony', 'Pokharan Road',
    'Kharegaon', 'Panch Pakhadi', 'Sector 36 Kamothe',
    'Dombivli (West)', 'DN Nagar Road', 'Godrej Hill', 'Ganesh Nagar',
    'Haware City', 'Mahatma Gandhi Road', 'Akurli Nagar',
    'Kasar vadavali', 'Vasai West', 'Mumbai Nashik Expressway',
    'Katrap', 'Mira Road', 'Kasheli',
    'Western Express Highway Kandivali East', 'Vasind', 'KASHELI',
    'Thakurli', 'Shakti Nagar', 'Bhayandar East', 'Dahisar East',
    'ulhasnagar 4', 'Sector-26 Taloja', 'Koproli'])
num_bedrooms = st.selectbox('Number of Bedrooms', [1, 2, 3, 4, 5])

# Amenities in two columns
st.header("Amenities")
col1, col2 = st.columns(2)

# Column 1
with col1:
    maintenance_staff = st.checkbox('Maintenance Staff')
    gymnasium = st.checkbox('Gymnasium')
    swimming_pool = st.checkbox('Swimming Pool')
    landscaped_gardens = st.checkbox('Landscaped Gardens')
    jogging_track = st.checkbox('Jogging Track')
    rainwater_harvesting = st.checkbox('Rainwater Harvesting')
    indoor_games = st.checkbox('Indoor Games')
    shopping_mall = st.checkbox('Shopping Mall')
    intercom = st.checkbox('Intercom')
    sports_facility = st.checkbox('Sports Facility')
    atm = st.checkbox('ATM')
    club_house = st.checkbox('Club House')
```

```
    school = st.checkbox('School')
    dining_table = st.checkbox('Dining Table')
    sofa = st.checkbox('Sofa')
    refrigerator = st.checkbox('Refrigerator')
    tv = st.checkbox('TV')

    # Column 2
with col2:
    power_backup = st.checkbox('Power Backup')
    staff_quarter = st.checkbox('Staff Quarter')
    cafeteria = st.checkbox('Cafeteria')
    multipurpose_room = st.checkbox('Multipurpose Room')
    washing_machine = st.checkbox('Washing Machine')
    gas_connection = st.checkbox('Gas Connection')
    ac = st.checkbox('AC')
    wifi = st.checkbox('Wifi')
    children_play_area = st.checkbox("Children's Play Area")
    lift_available = st.checkbox('Lift Available')
    bed = st.checkbox('BED')
    vaastu_compliant = st.checkbox('Vaastu Compliant')
    microwave = st.checkbox('Microwave')
    resale = st.checkbox('Resale')
    golfcourse = st.checkbox('Golfcourse')
    wardrobe = st.checkbox('Wardrobe')
    hospital = st.checkbox('Hospital')

price_predict = ''

if st.button('Predict price'):

    # Transform area using scaler
    area = mumbai_scaler.transform(np.array([[float(area)]]))  # Convert to float and 2D array

        # Transform location using encoder
    try:
        location = mumbai_encoder.transform([location])[0]  # Ensure to access the first element of the resulting array
    except:
```

```
            location = 'other'
            location = mumbai_encoder.transform([location])[0]  # Ensure to access the first element of the resulting array

        # Convert input data to array
        input_data = np.array([
            area[0][0], location, num_bedrooms, resale, maintenance_staff, gymnasium, swimming_pool, landscaped_gardens,
            jogging_track, rainwater_harvesting, indoor_games, shopping_mall, intercom, sports_facility, atm,
            club_house,school, power_backup, staff_quarter, cafeteria, multipurpose_room,hospital,
            washing_machine, gas_connection, ac,wifi,children_play_area, lift_available, bed, vaastu_compliant,
            microwave,golfcourse,tv, dining_table, sofa,wardrobe,refrigerator
        ]).reshape(1, -1)

        # Predict price using the model
        price_predict = mumbai_model.predict(input_data)
    st.success(price_predict)

if (selected == 'Bengaluru Model'):
    st.title('Bengaluru House Price Prediction Model Using Machine Learning')
    st.image("bengaluru.jpg", use_column_width=True)

    st.header("Basic Information")
    area = st.text_input('Area')
    location = st.selectbox('Location',['JP Nagar Phase 1', 'Dasarahalli on Tumkur Road',
      'Kannur on Thanisandra Main Road', 'Doddanekundi', 'Kengeri',
      'Horamavu', 'Thanisandra', 'Ramamurthy Nagar',
      'Whitefield Hope Farm Junction', 'Electronic City Phase 1',
      'Yelahanka', 'Anjanapura', 'Jalahalli', 'Kasavanahalli',
      'Bommasandra', 'Bellandur', 'RR Nagar', 'Begur', 'Hosa Road',
      'Sahakar Nagar', 'Kadugodi', 'Jakkur', 'Jigani', 'Krishnarajapura',
      'Brookefield', 'Banashankari', 'Nelamangala', 'Attibele',
      'Banaswadi', 'Kodigehalli', 'ITPL', 'Uttarahalli Hobli',
      'Chikkagubbi on Hennur Main Road', 'Varthur', 'Vidyaranyapura',
      'Electronic City Phase 2', 'J. P. Nagar', 'K. Chudahalli',
      'Narayanaghatta', 'Anekal City', 'Sarjapur', 'Koramangala',
      'Hebbal', 'Budigere Cross', 'Bommanahalli', 'Electronics City',
      'Chikkalasandra', 'Kogilu', 'Nayandahalli', 'Bilekahalli',
      'Muneshwara Nagar', 'Junnasandra',
```

```
    'Narayanapura on Hennur Main Road', 'Kothanur',
    'Kadugodi Industrial Area',
    'Sarjapur Road Wipro To Railway Crossing', 'RMV Extension Stage 2',
    'Kudlu', 'Talaghattapura', 'Kumbalgodu', 'Carmelaram',
    'Uttarahalli', 'Anagalapura Near Hennur Main Road',
    'Avalahalli Off Sarjapur Road', 'R T  Nagar', 'JP Nagar Phase 7',
    'Subramanyapura', 'JP Nagar Phase 4', 'JP Nagar Phase 8',
    'Amruthahalli', 'Nagarbhavi', 'Chandapura', 'Marsur',
    'JP Nagar Phase 3', 'JP Nagar Phase 9', 'Gottigere',
    'Kanakapura Road Beyond Nice Ring Road', 'Harlur', 'Konanakunte',
    'Richmond Town', 'Jayanagar', 'Domlur', 'Devanahalli', 'Hulimavu',
    'Kumaraswamy Layout', 'Bikasipura', 'Singasandra',
    'JP Nagar Phase 6', 'Sanjaynagar', 'CV Raman Nagar',
    'Padmanabhanagar', 'Hennur', 'KPC Layout', 'R.K. Hegde Nagar',
    'Kannamangala', 'Yerthiganahalli', 'Badamanavarthekaval',
    'Kanakapura', 'Bannerughatta', 'BTM Layout',
    'Kuvempu Layout on Hennur Main Road', 'Marathahalli',
    'Rajajinagar', 'Whitefield', 'RMV'])
num_bedrooms = st.selectbox('Number of Bedrooms', [1, 2, 3, 4, 5])

# Amenities in two columns
st.header("Amenities")
col1, col2 = st.columns(2)

# Column 1
with col1:
    gymnasium = st.checkbox('Gymnasium')
    swimming_pool = st.checkbox('Swimming Pool')
    jogging_track = st.checkbox('Jogging Track')
    rainwater_harvesting = st.checkbox('Rainwater Harvesting')
    indoor_games = st.checkbox('Indoor Games')
    shopping_mall = st.checkbox('Shopping Mall')
    intercom = st.checkbox('Intercom')
    sports_facility = st.checkbox('Sports Facility')
    atm = st.checkbox('ATM')
    club_house = st.checkbox('Club House')
    school = st.checkbox('School')
```

```
    dining_table = st.checkbox('Dining Table')
    sofa = st.checkbox('Sofa')
    refrigerator = st.checkbox('Refrigerator')
    tv = st.checkbox('TV')
    hospital = st.checkbox('Hospital')

    # Column 2
with col2:
    security = st.checkbox('24X7 Security')
    power_backup = st.checkbox('Power Backup')
    carparking = st.checkbox('Carparking')
    staff_quarter = st.checkbox('Staff Quarter')
    cafeteria = st.checkbox('Cafeteria')
    washing_machine = st.checkbox('Washing Machine')
    gas_connection = st.checkbox('Gas Connection')
    ac = st.checkbox('AC')
    children_play_area = st.checkbox("Children's Play Area")
    lift_available = st.checkbox('Lift Available')
    bed = st.checkbox('BED')
    vaastu_compliant = st.checkbox('Vaastu Compliant')
    microwave = st.checkbox('Microwave')
    resale = st.checkbox('Resale')
    golfcourse = st.checkbox('Golfcourse')

price_predict = ''

if st.button('Predict price'):

    # Transform area using scaler
    area = bangalore_scaler.transform(np.array([[float(area)]]))  # Convert to float and 2D array

        # Transform location using encoder
    try:
        location = bangalore_encoder.transform([location])[0]  # Ensure to access the first element of the resulting array
    except:
        location = 'other'
        location = bangalore_encoder.transform([location])[0]  # Ensure to access the first element of the resulting array
```

```
        # Convert input data to array
        input_data = np.array([
            area[0][0], location, num_bedrooms, resale,gymnasium, swimming_pool,
            jogging_track, rainwater_harvesting, indoor_games, shopping_mall, intercom, sports_facility, atm,
            club_house,school,security, power_backup,carparking, staff_quarter, cafeteria,hospital,
            washing_machine, gas_connection, ac,children_play_area, lift_available, bed, vaastu_compliant,
            microwave,golfcourse,tv, dining_table, sofa,refrigerator
        ]).reshape(1, -1)

        # Predict price using the model
        price_predict = bangalore_model.predict(input_data)
    st.success(price_predict)




if (selected == 'Hyderabad Model'):
    st.title('Hyderabad House Price Prediction Model Using Machine Learning')
    st.image("Hyderabad.webp", use_column_width=True)
    st.header("Basic Information")
    area = st.text_input('Area')
    location = st.selectbox('Location',['Nizampet', 'Hitech City', 'Manikonda', 'Alwal', 'Kukatpally',
       'Gachibowli', 'Tellapur', 'Kokapet', 'Hyder Nagar', 'Mehdipatnam',
       'Narsingi', 'Khajaguda Nanakramguda Road', 'Madhapur',
       'Puppalaguda', 'Begumpet', 'Banjara Hills', 'AS Rao Nagar',
       'Pragathi Nagar Kukatpally', 'Miyapur', 'Mallampet',
       'Nanakramguda', 'Attapur', 'West Marredpally', 'Kompally',
       'Sri Nagar Colony', 'Hakimpet', 'Pocharam', 'Nagole', 'LB Nagar',
       'Meerpet', 'Kachiguda', 'Masab Tank', 'Kondapur', 'Saroornagar',
       'Uppal Kalan', 'Mallapur', 'Rajendra Nagar', 'Beeramguda',
       'Moosapet', 'Bachupally', 'Toli Chowki', 'Lakdikapul', 'Tarnaka',
       'Kistareddypet', 'Hafeezpet', 'Shaikpet', 'Amberpet', 'Kapra',
       'Trimalgherry', 'Habsiguda', 'Sanath Nagar', 'Darga Khaliz Khan',
       'Kothaguda', 'Balanagar', 'Jubilee Hills', 'raidurgam',
       'Murad Nagar', 'Chandanagar', 'East Marredpally', 'Aminpur',
       'Gajularamaram', 'Serilingampally', 'Malkajgiri', 'Mettuguda',
```

'Venkat Nagar Colony', 'Kondakal', 'Gopanpally', 'Somajiguda',
'Nallagandla Gachibowli', 'Krishna Reddy Pet', 'Bolarum',
'Zamistanpur', 'Madhura Nagar', 'Ghansi Bazaar', 'Chintalakunta',
'Chinthal Basthi', 'Nallakunta', 'Bowenpally', 'Bandlaguda Jagir',
'Boduppal', 'Neknampur', 'Appa Junction Peerancheru',
'Ambedkar Nagar', 'Vanasthalipuram', 'Moula Ali', 'Gandipet',
'Nacharam', 'Appa Junction', 'Qutub Shahi Tombs', 'Abids',
'Dilsukh Nagar', 'Quthbullapur', 'Sainikpuri', 'KTR Colony',
'Bollaram', 'Karmanghat', 'Gajulramaram Kukatpally', 'Uppal',
'Cherlapalli', 'Himayat Nagar', 'Rhoda Mistri Nagar', 'Chintalmet',
'Hitex Road', 'ECIL', 'Boiguda', 'ECIL Main Road',
'ECIL Cross Road', 'Rajbhavan Road Somajiguda',
'Ramachandra Puram', 'TellapurOsman Nagar Road', 'Mansoorabad',
'KRCR Colony Road', 'Pragati Nagar', 'Padmarao Nagar',
'Paramount Colony Toli Chowki', 'BK Guda Internal Road',
'muthangi', 'Pragathi Nagar', 'Yapral', 'Narayanguda', 'Kollur',
'Bachupally Road', 'Old Bowenpally', 'Alapathi Nagar',
'Arvind Nagar Colony', 'Matrusri Nagar', 'Pragathi Nagar Road',
'Padma Colony', 'Happy Homes Colony', 'Old Nallakunta',
'Sangeet Nagar', 'NRSA Colony', 'Adibatla', 'Methodist Colony',
'Ameerpet', 'ALIND Employees Colony', 'Khizra Enclave', 'Medchal',
'Dammaiguda', 'Suchitra', 'Whitefields', 'Mayuri Nagar',
'Adda Gutta', 'Miyapur HMT Swarnapuri Colony',
'Central Excise Colony Hyderabad', 'Basheer Bagh', 'Gopal Nagar',
'Bachupaly Road Miyapur', 'Kushaiguda', 'Ashok Nagar',
'Barkatpura', 'Madinaguda', 'Bagh Amberpet', 'new nallakunta',
'BHEL', 'Sun City', 'Hydershakote', 'BK Guda Road',
'Nallagandla Road', 'IDPL Colony', 'Ramnagar Gundu',
'Alkapur township', 'Banjara Hills Road Number 12',
'Panchavati Colony Manikonda', 'New Maruthi Nagar',
'Madhavaram Nagar Colony', 'Miyapur Bachupally Road',
'nizampet road', 'Kokapeta Village', 'HMT Hills', 'Tilak Nagar',
'Chititra Medchal', 'Isnapur', 'D D Colony', 'DD Colony',
'Patancheru Shankarpalli Road', 'Patancheru', 'Jhangir Pet',
'Almasguda', 'Allwyn Colony', 'financial District',
'Beeramguda Road', 'Pati', 'Karimnagar', 'Kollur Road',
'Sun City Padmasri Estates', 'Chaitanyapuri', 'Nandagiri Hills',

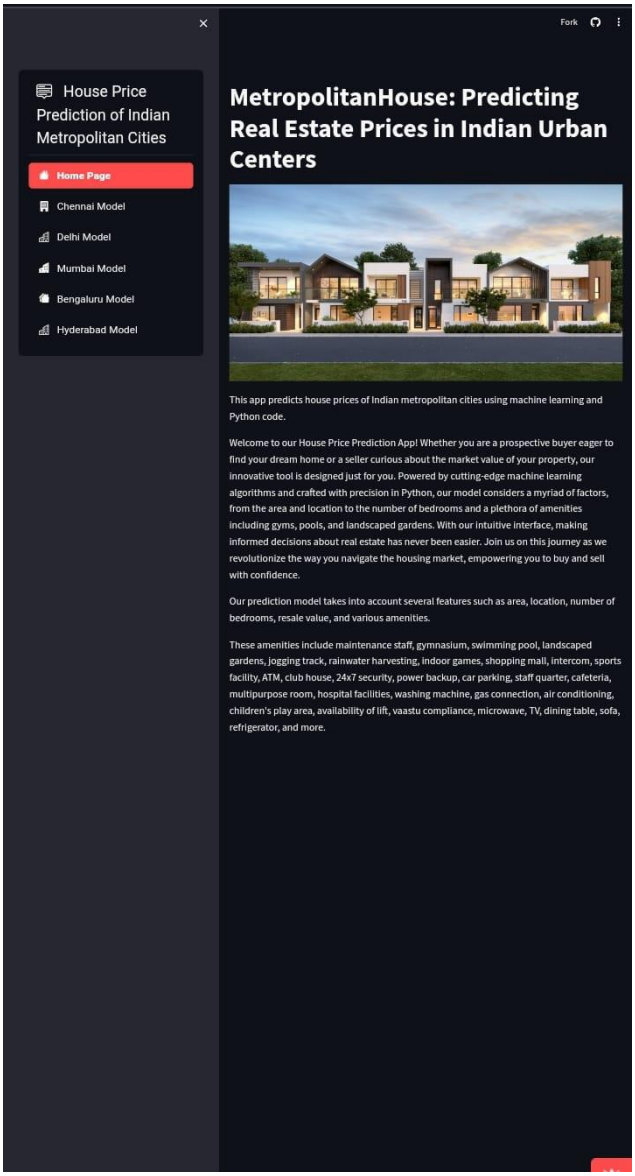
```
    'Whitefield', 'Film Nagar', 'Kismatpur', 'Dr A S Rao Nagar Rd',
    'Dullapally', 'KPHB', 'Vivekananda Nagar Colony', 'Ameenpur',
    'Chintradripet', 'Ring Road', 'Saket', 'Kavuri Hills', 'manneguda',
    'Moti Nagar', 'Usman Nagar', 'Shadnagar', 'Bongloor',
    'Mailardevpally', 'Uppalguda', 'Tirumalgiri', 'Chikkadapally',
    'JNTU', 'hyderabad', 'Shamshabad', 'Srisailam Highway',
    'Domalguda', 'Lingampalli', 'Residential Flat Machavaram',
    'Whisper Valley', 'Tukkuguda Airport View Point Road',
    'Santoshnagar', 'Tolichowki', 'Domalguda Road', 'Shankarpalli',
    'Kothapet', 'Baghlingampally', 'Picket', 'Safilguda',
    'Sikh Village', 'Neredmet', 'Macha Bolarum', 'Kowkur',
    'Rakshapuram', 'west venkatapuram', 'Vidyanagar Adikmet',
    'Aushapur', 'Old Alwal', 'Secunderabad Railway Station Road',
    'Balapur', 'Hastinapur', 'chandrayangutta'])
num_bedrooms = st.selectbox('Number of Bedrooms', [1, 2, 3, 4, 5])

# Amenities in two columns
st.header("Amenities")
col1, col2 = st.columns(2)

# Column 1
with col1:
    maintenancestaff = st.checkbox('Maintenancestaff')
    gymnasium = st.checkbox('Gymnasium')
    swimming_pool = st.checkbox('Swimming Pool')
    landscaped_gardens = st.checkbox('landscaped Gardens')
    jogging_track = st.checkbox('Jogging Track')
    rainwater_harvesting = st.checkbox('Rainwater Harvesting')
    indoor_games = st.checkbox('Indoor Games')
    shopping_mall = st.checkbox('Shopping Mall')
    intercom = st.checkbox('Intercom')
    sports_facility = st.checkbox('Sports Facility')
    atm = st.checkbox('ATM')
    club_house = st.checkbox('Club House')
    dining_table = st.checkbox('Dining Table')
    sofa = st.checkbox('Sofa')
    wardrobe = st.checkbox('Wardrobe')
```

```
    refrigerator = st.checkbox('Refrigerator')
    tv = st.checkbox('TV')

    # Column 2
with col2:
    security = st.checkbox('24X7 Security')
    power_backup = st.checkbox('Power Backup')
    carparking = st.checkbox('Carparking')
    staff_quarter = st.checkbox('Staff Quarter')
    cafeteria = st.checkbox('Cafeteria')
    multipurpose_room = st.checkbox('Multipurpose Room')
    washing_machine = st.checkbox('Washing Machine')
    gas_connection = st.checkbox('Gas Connection')
    ac = st.checkbox('AC')
    wifi = st.checkbox('Wifi')
    children_play_area = st.checkbox("Children's Play Area")
    lift_available = st.checkbox('Lift Available')
    bed = st.checkbox('BED')
    vaastu_compliant = st.checkbox('Vaastu Compliant')
    resale = st.checkbox('Resale')
    golfcourse = st.checkbox('Golfcourse')

price_predict = ''

if st.button('Predict price'):

    # Transform area using scaler
    area = Hyderabad_scaler.transform(np.array([[float(area)]]))  # Convert to float and 2D array

        # Transform location using encoder
    try:
        location = Hyderabad_encoder.transform([location])[0]  # Ensure to access the first element of the resulting array
    except:
        location = 'other'
        location = Hyderabad_encoder.transform([location])[0]  # Ensure to access the first element of the resulting array

    # Convert input data to array
```

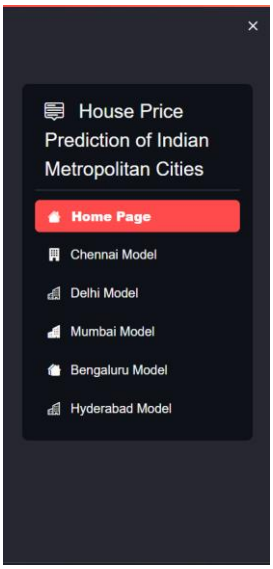
```
    input_data = np.array([
        area[0][0], location, num_bedrooms, resale,maintenancestaff,gymnasium, swimming_pool,landscaped_gardens,
        jogging_track, rainwater_harvesting, indoor_games, shopping_mall, intercom, sports_facility, atm,
        club_house,security, power_backup,carparking, staff_quarter, cafeteria,multipurpose_room,
        washing_machine, gas_connection, ac,wifi,children_play_area, lift_available, bed, vaastu_compliant,
        golfcourse,tv, dining_table, sofa,wardrobe,refrigerator
    ]).reshape(1, -1)

    # Predict price using the model
    price_predict = Hyderabad_model.predict(input_data)
st.success(price_predict)
```

6. RESULT

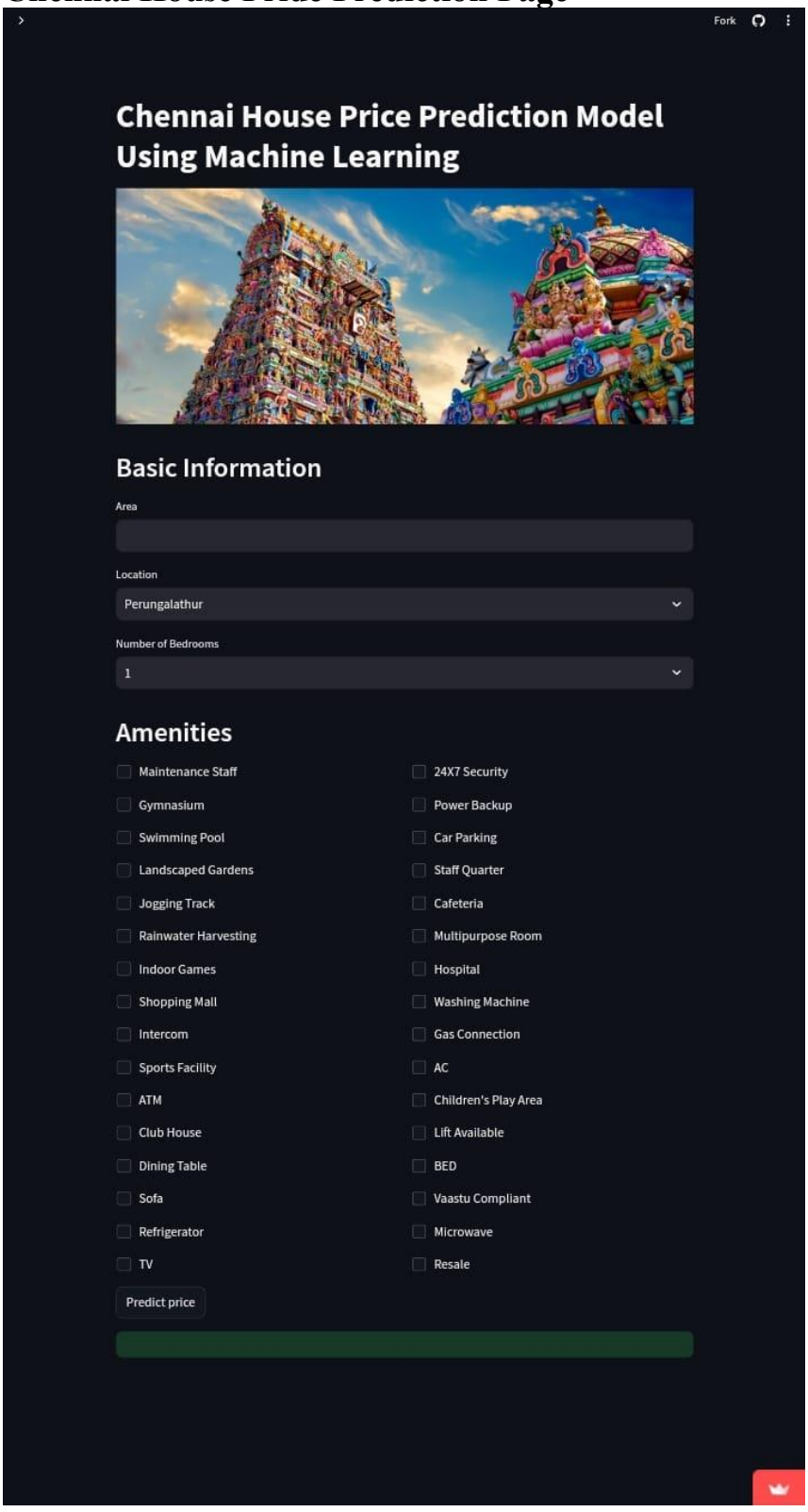
1. HOME PAGE



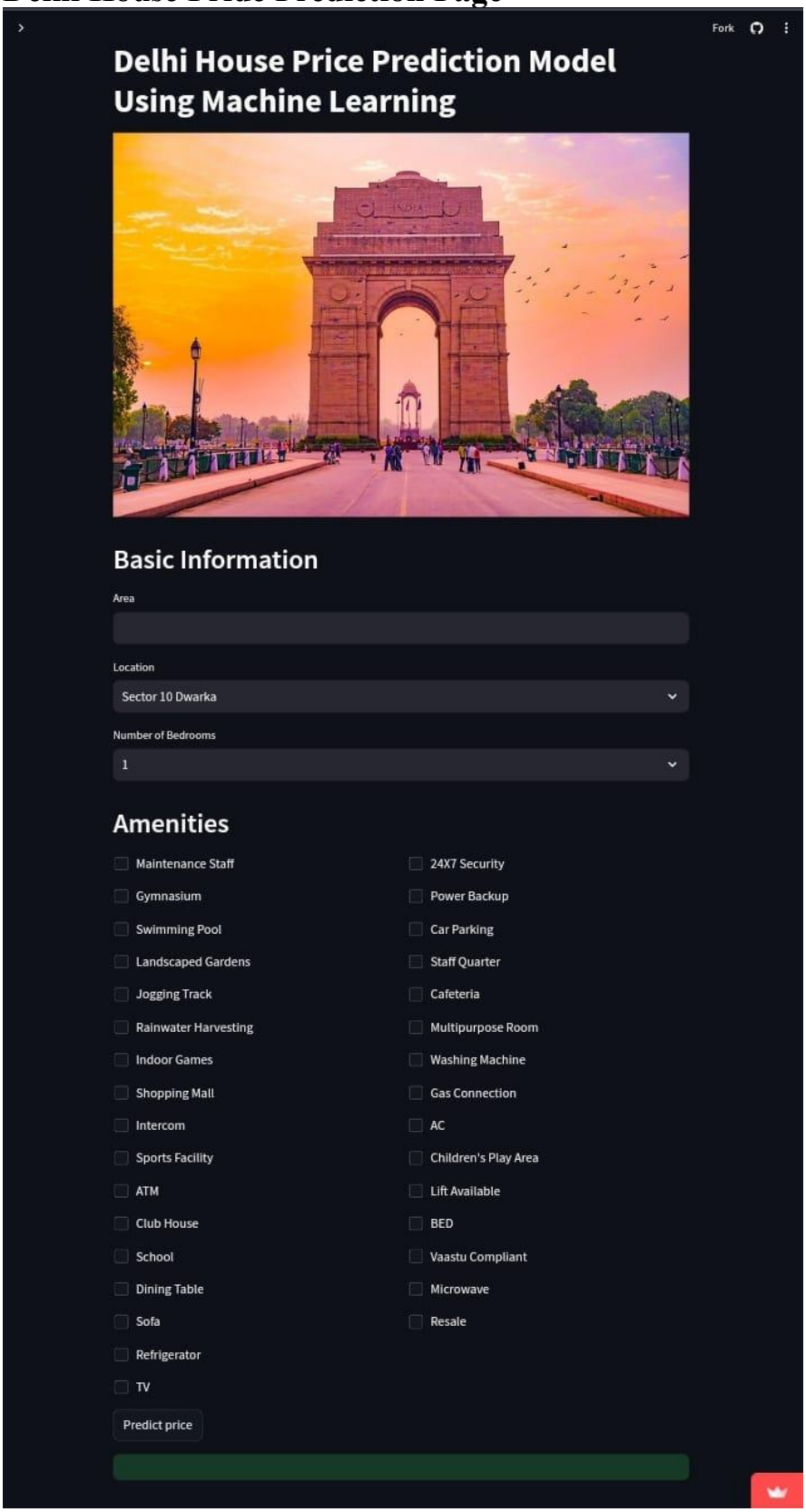
2. MENU BAR


House Price Prediction of Indian Metropolitan Cities
Home Page
Chennai Model
Delhi Model
Mumbai Model
Bengaluru Model
Hyderabad Model

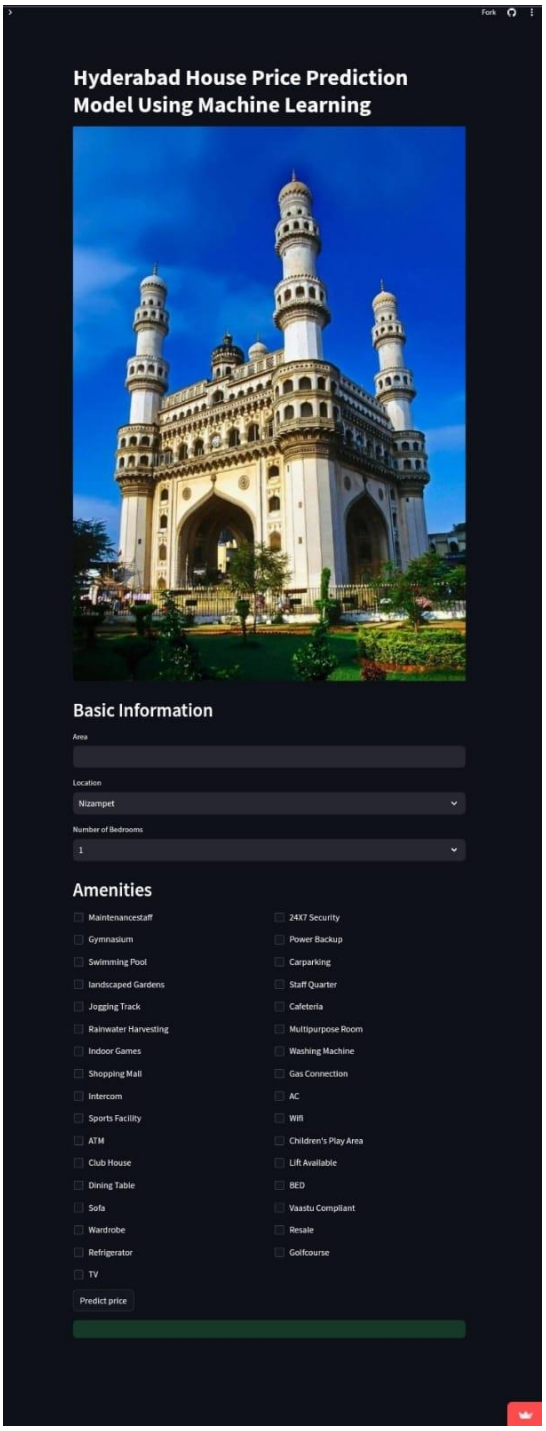
3. Chennai House Pride Prediction Page



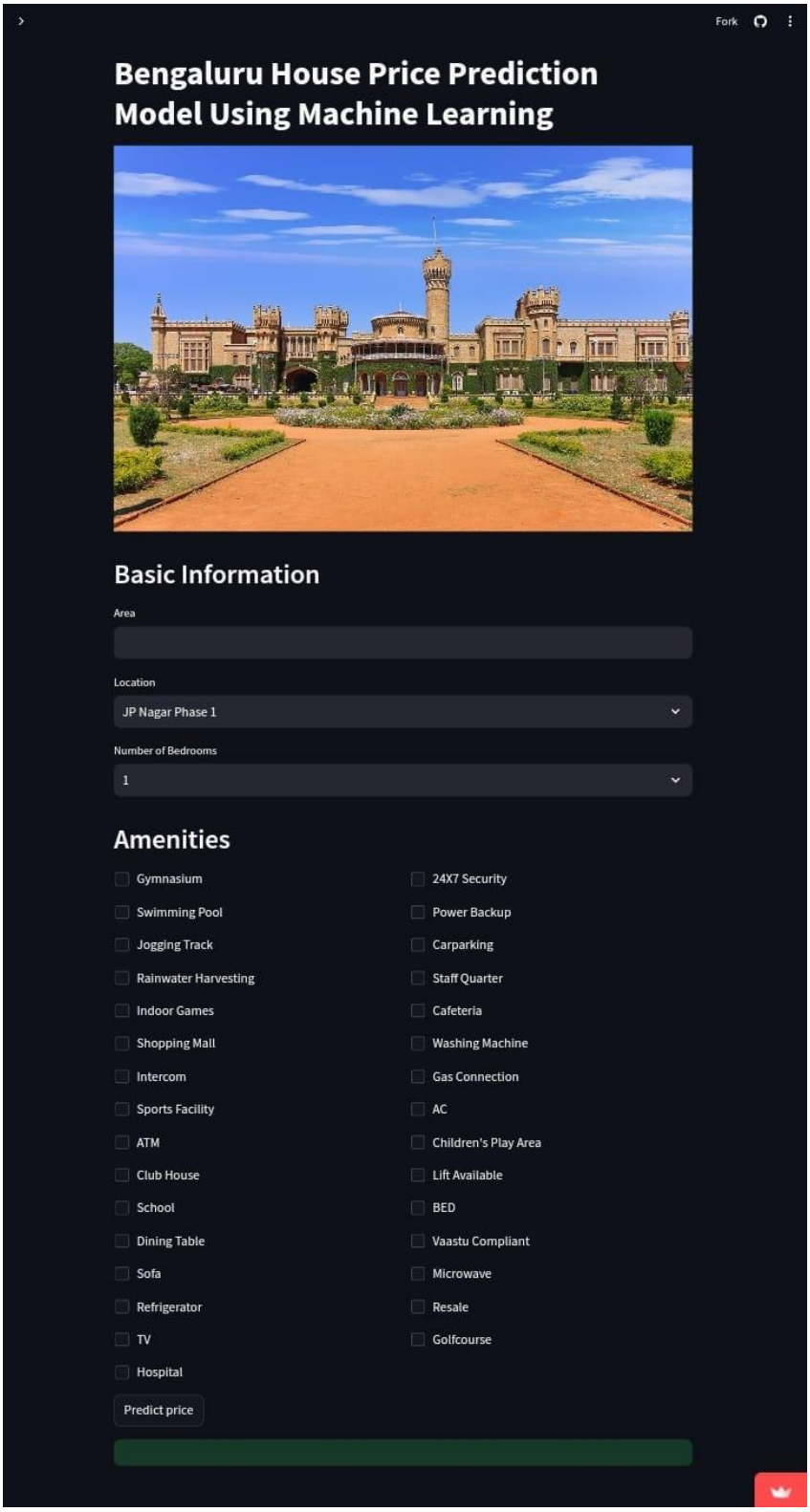
4. Delhi House Pride Prediction Page



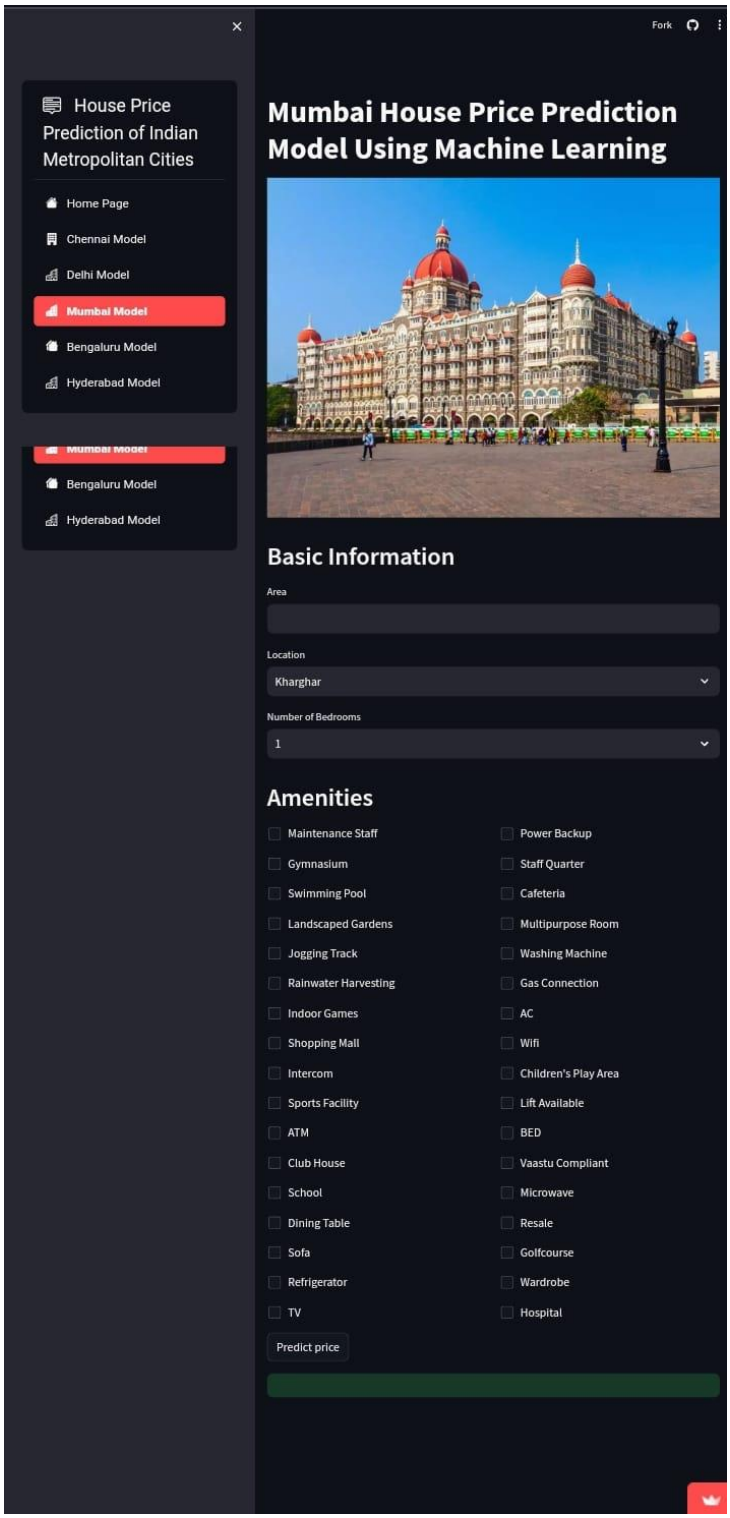
5. Hyderabad House Pride Prediction Page



6. Bengaluru House Pride Prediction Page



7. Mumbai House Price Prediction Page



7. CONCLUSION

The house price prediction system for **Indian metropolitan areas** addresses the challenge of accurately forecasting real estate prices by leveraging advanced machine learning algorithms and a user-friendly web interface. This system allows users, including **buyers, sellers, and real estate agents**, to input various property attributes and receive **precise price predictions**, aiding in informed **decision-making**. By providing a reliable tool for price forecasting, this system enhances the transparency and efficiency of the real estate market, helping Indians to better understand property values, make smarter investments, and navigate the housing market with greater confidence.

7. REFERENCES

Best reference sites **for Python,Data Science and Machine Learning** :

W3schools: Python , Pandas, Numpy,.

CampusX : Python, Machine Learning, DSMP

Sidharthan : Machine Learning Playlist

AI tools : Chatgpt, Blackbox.ai , Gemini AI

Other tools : Python visualizer, Jenny CS IT