PREDICTING HOUSE PRICE USING MACHINE LEARNING

TEAM MEMBER

810621104007: BHAVANI.S

Phase 5 submission document

Project title: House price prediction

Phase 5: Project Document & submission

Topic: In this section we will document the complete project and prepare it for submission.

House Price Prediction

Introduction:

- Development of civilization is the foundation of the increase in demand for houses day by day. Accurate prediction of house prices has been always a fascination for buyers, sellers, and bankers also. Many researchers have already worked to unravel the mysteries of the prediction of house prices. Many theories have been given birth as a consequence of the research work contributed by various researchers all over the world. Some of these theories believe that the geographical location and culture of a particular area determine how the home prices will increase or decrease whereas other schools of thought emphasize the socio-economic conditions that largely play behind these house price rises.

- We all know that a house price is a number from some defined assortment, so obviously prediction of prices of houses is a regression task. To forecast house prices one person usually tries to locate similar properties in his or

her neighborhood and based on collected data that person will try to predict the house price.

- All these indicate that house price prediction is an emerging research area of regression that requires the knowledge of machine learning. This has motivated me to work in this domain.

- Real estate appraisal is an integral part of the property buying process. Traditionally, the appraisal is performed by professional appraisers specially trained for real estate valuation. For the buyers of real estate properties, an automated price estimation system can be useful to estimate the prices of properties currently on the market. Such a system can be particularly helpful for novice buyers who are buying a property for the first time, with little to no experience.

DATASET: https://www.kaggle.com/datasets/vedavyasv/usa-housing

GIVEN DATA SET:

Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Address	
0	79545.458574	5.682861	7.009188	4.09	23086.800503	1.059034e+06	208 Michael Ferry Apt. 674\nLaurabury, NE 3701...
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnson Views Suite 079\nLake Kathleen, CA...
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Elizabeth Stravenue\nDanieltown, WI 06482...
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett\nFPO AP 44820
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymond\nFPO AE 09386

Here's a list of tools and software commonly used in theprocess:

1.Programming Language:

Python is the most popular language for machine learning due to its extensive libraries and frameworks. You can use libraries like Numpy , Pandas, scikit-learn and more.

2. Integrated Development Environment(IDE):

Choose an IDE for coding and running machine learning experiments. Some popular options include jupyter Notebook, Google colab, or traditional IDEs like PyCharm.

3. Machine Learning Libraries:

You'll need various learning libraries ,including.

Scikit-learn for building and evaluating machine learning models.

4.Data Virtualization Tools:

Tools like matplotlib, seaborn, or Plotly are essential for data exploration and virtualizations.

5.Data Preprocessing Tools:

Libraries like pandas help with data cleaning, manipulation, and preprocessing.

1.DESIGN THINKING AND PRESENT IN FORM OF DOCUMENT

1. Empathize:

- Understand the needs and challenges of all stakeholders involved in the house price prediction process ,including homebuyers.

2. Define:

- Clearly articulate the problem statement, such as 'How might we predict house prices more accuracy and transparency using machine.

3. Ideate:

- Brainstorm creative solution and data sources that can enhance the accuracy and transparency of house price prediction.

4. Prototype:

- Create prototype machine learning model based on the ideas generated during the ideation phase.

- Test and iterate on these prototypes to determine which approaches are most promising in terms of accuracy and usability.

5. Test:

- Gather feedback from user and stakeholders by testing the machine learning models with real-world data and scenarios.

6. Implement:

- Develop a production ready machine learning solution for predicting house prices, integrating the best-performing algorithms and data sources.

- Implement transparency measures such as model interpretability tools to ensures users understand how prediction are generated.

7. Educate and Train:

- Provide training and educational resources to help users understand .

- Foster a culture of data literacy among stakeholders to enhance trust in the technology.

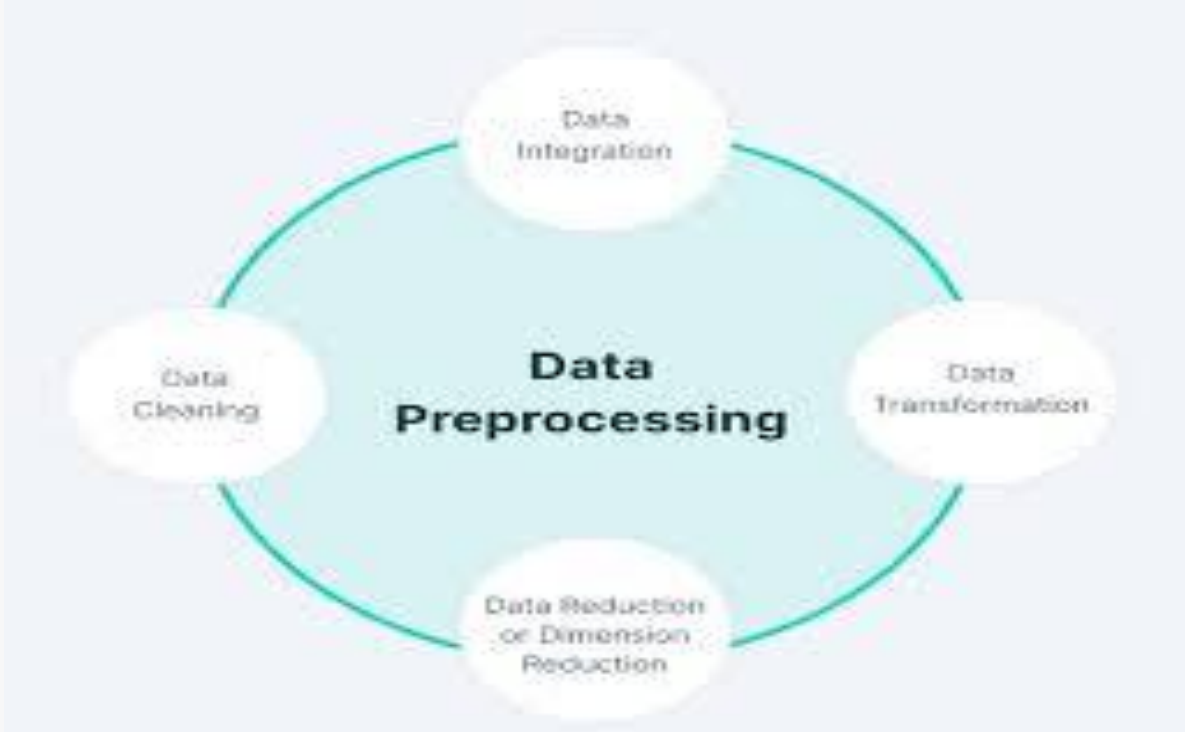
2. DESIGN INTO INNOVATION

1. Data collection:

Collect a diverse dataset that includes property features, location data, historical sales prices, and other relevant information.

2. Data Preprocessing:

Preprocess the data by handling missing values, removing duplicates, and encoding categorical variables.



PYTHON PROGRAM

```
import requests
from bs4 import BeautifulSoup

# Define the URL of the real estate listings page you want to scrape
url = "https://example-real-estate-website.com/listings"

# Send an HTTP GET request to the URL
response = requests.get(url)

# Check if the request was successful
if response.status_code == 200:
    # Parse the HTML content of the page using BeautifulSoup
    soup = BeautifulSoup(response.text, 'html.parser')

    # Find the HTML elements that contain the data you need
    property_listings = soup.find_all('div', class_='property-listing')
```

```
    # Initialize lists to store data
    property_data = []

    # Loop through each property listing
    for listing in property_listings:
        property_info = {}
        # Extract relevant information from the HTML elements
        property_info['property_name'] = listing.find('h2').text
        property_info['price'] = listing.find('span', class_='price').text
        property_info['bedrooms'] = listing.find('div', class_='bedrooms').text
        property_info['bathrooms'] = listing.find('div', class_='bathrooms').text
        property_info['sqft'] = listing.find('div', class_='sqft').text

        # Add the property information to the list
        property_data.append(property_info)
```

```
    # Print the collected data (you can save it to a file or a database)
    for property_info in property_data:
        print(property_info)
else:
    print("Failed to retrieve the webpage. Status code:", response.status_code)
```

1. **Feature Engineering:**
    - Create new features or transform existing ones to make them more informative for the prediction task. For example, calculate the price per square foot or create distance-based features.
2. **Data Splitting:**
    - Split the dataset into training, validation, and test sets. This allows you to train and evaluate your model on different subsets of the data.
3. **Model Selection:**
    - Choose an appropriate machine learning algorithm. Common choices include:
        - **Linear Regression:** A simple model that assumes a linear relationship between input features and house prices.
        - **Decision Trees and Random Forests:** These models can capture non-linear relationships and handle both numerical and categorical data.

- **Gradient Boosting Algorithms:** Such as XGBoost or LightGBM, which are powerful for regression tasks.
- **Neural Networks:** Deep learning models can capture complex patterns but may require a larger amount of data.

4. **Model Training:**
   - Train your selected model on the training dataset. During training, the model learns to make predictions by minimizing a loss function.
5. **Hyperparameter Tuning:**
   - Fine-tune the model's hyperparameters to optimize its performance. This can include adjusting learning rates, tree depths, regularization parameters, and more.
6. **Model Evaluation:**
   - Assess the model's performance using appropriate evaluation metrics, such as Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE).
   - Also, consider using visualizations like scatter plots of predicted vs. actual prices to understand the model's behavior.
7. **Model Interpretability:**
   - Use techniques like feature importance analysis or SHAP (SHapley Additive exPlanations) values to interpret the model's predictions and understand which features are most influential.
8. **Cross-Validation:**
   - Implement k-fold cross-validation to ensure the model's performance is robust and not overfitting to the training data.
9. **Ensemble Models:**
   - Combine multiple models, such as blending the predictions of different algorithms, to improve predictive accuracy.
10. **Regularization:**
    - Apply regularization techniques, like L1 (Lasso) or L2 (Ridge) regularization, to prevent overfitting and enhance model generalization.

11. **Data Scaling and Normalization:**
    - Scale or normalize features to ensure that they have similar ranges, which can help some algorithms converge faster.
12. **Time-Series Analysis (Optional):**
    - If predicting future prices, consider time-series analysis and forecasting techniques to account for seasonality and trends in historical data.
13. **Deployment:**
    - Once you have a satisfactory model, deploy it in a real-world environment, such as a web application or API, to make predictions.
14. **Monitoring and Maintenance:**
    - Continuously monitor the model's performance and retrain it as new data becomes available to account for changing market conditions and data drift.
15. **Ethical Considerations:**
    - Be mindful of potential biases in the data and model predictions, and ensure fairness in the model's outcomes.

3. BUILDING AND PREPROCESSING DATASET

```
# Import necessary libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer

# Load the raw data into a DataFrame (replace 'data.csv' with your dataset file)
data = pd.read_csv('data.csv')

# Explore the data to understand its structure and features
print(data.head())  # Display the first few rows of the dataset

# Data preprocessing
```

```
# 1. Handle missing values
imputer = SimpleImputer(strategy='median')
data['feature_with_missing_values'] = imputer.fit_transform(data['feature_with_missing_values'].values.reshape(-1, 1))

# 2. Encode categorical variables (if applicable)
# Example: Use one-hot encoding for categorical variables
data = pd.get_dummies(data, columns=['categorical_feature1', 'categorical_feature2'])

# 3. Split the data into features (X) and the target variable (y)
X = data.drop(columns=['target_variable'])  # Features
y = data['target_variable']  # Target variable

# 4. Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# 5. Standardize features (optional but often beneficial for some algorithms)

scaler = StandardScaler()

X_train = scaler.fit_transform(X_train)

X_test = scaler.transform(X_test)


# At this point, you have a preprocessed dataset ready for machine learning modeling.


# You can proceed to build and train your machine learning model, such as linear regression, decision trees, random forests, or other algorithms, using X_train and y_train.


# Then, evaluate the model's performance using X_test and y_test and make predictions on new data.


# Example:

# from sklearn.linear_model import LinearRegression

# model = LinearRegression()

# model.fit(X_train, y_train)
```

y_pred = model.predict(X_test)

Evaluate the model's performance and make predictions as needed for your house price prediction task.

3.Feature engineering:

Feature engineering is a critical step in house price prediction using machine learning. It involves creating new features, transforming existing ones, and selecting the most relevant attributes to improve the predictive power of your model. Below is a Python program for feature engineering with an example output. Please note that this is a simplified example, and you may need to adapt it to your specific dataset.

4.FEATRURES ENGINEERING , MODEL ENGINEERIN AND EVALUATION

1.FEATURE ENGINEERING:

As mentioned earlier , feature engineering is crucial. It involves creating new features or transforming existing ones to provide meaningful information for your model.

 # Combining bedrooms and bathrooms to create TotalRooms feature

data['TotalRooms'] = data['Bedrooms'] + data['Bathrooms']

 # Calculating price per square foot

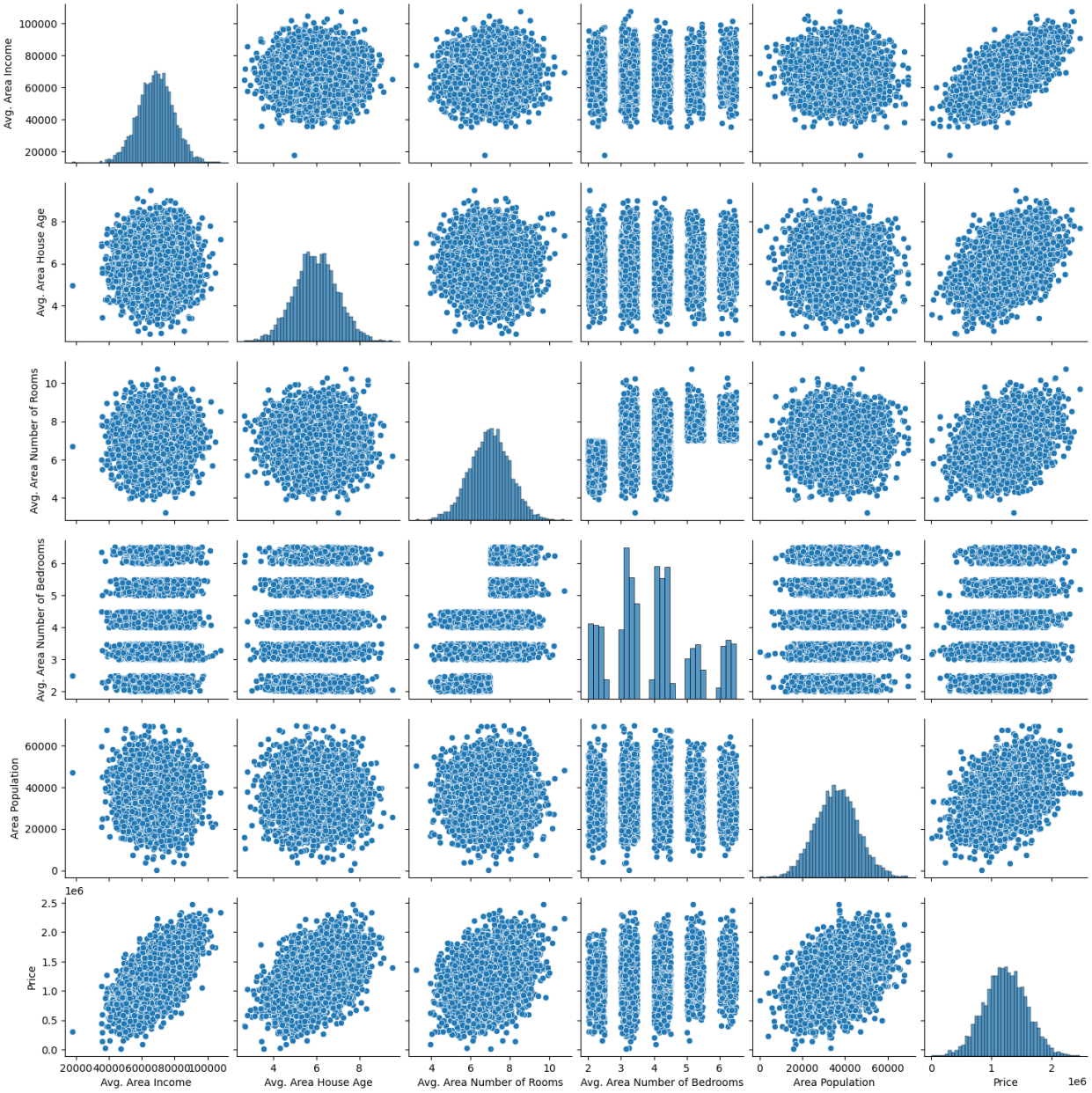
data['PricePerSqFt'] = data['Price'] / data['SquareFootage']

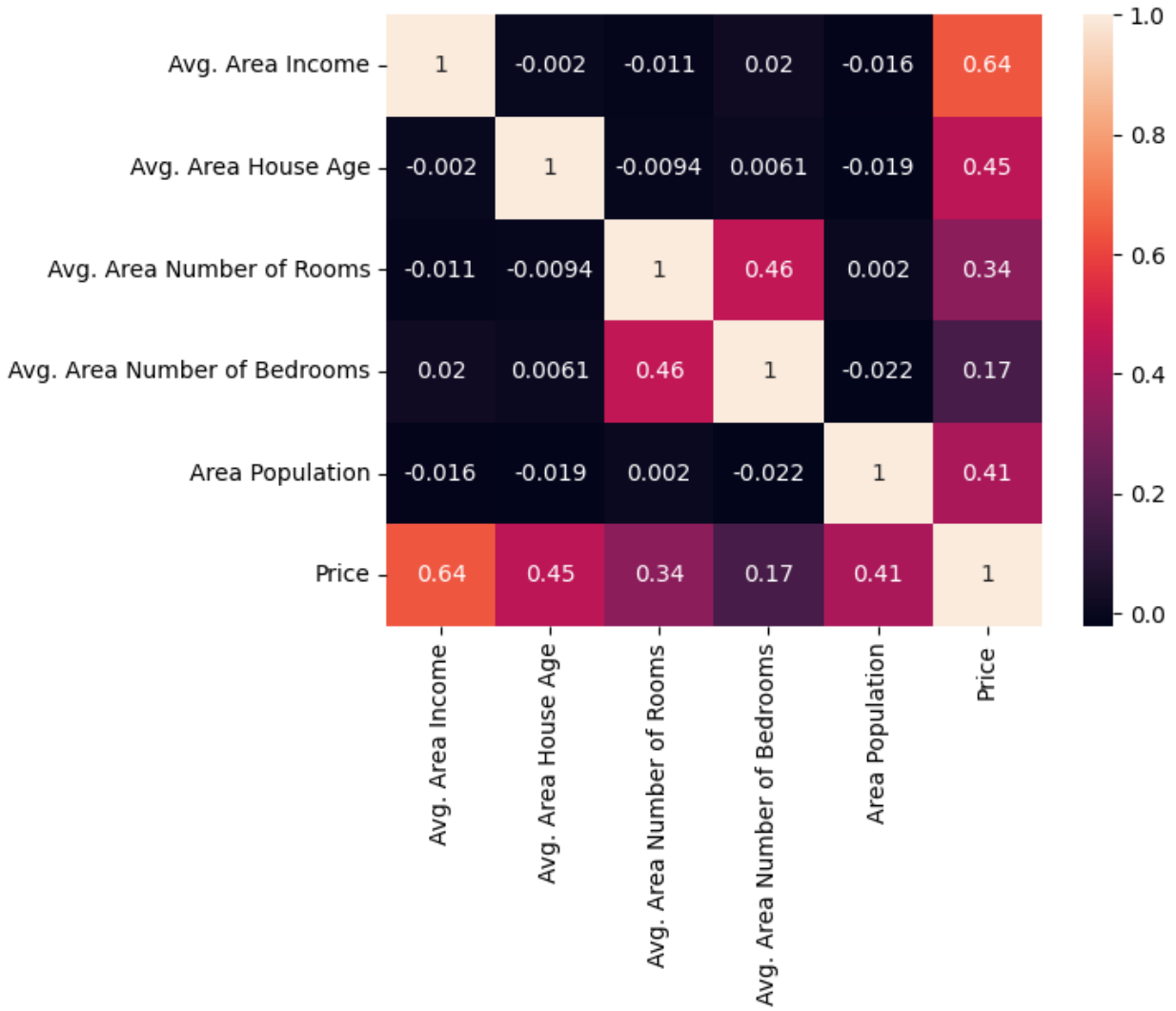
 # Encoding property type as a categorical variable

data = pd.get_dummies(data, columns=['PropertyType'])

 # Calculating distance to a key location (e.g., school)

data['DistanceToSchool'] = data.apply(lambda row: calculate_distance(row['Latitude'], row['Longitude'], school_latitude, school_longitude), axis=1)




Avg. Area Income
Avg. Area House Age
Avg. Area Number of Rooms
Avg. Area Number of Bedrooms
Area Population
Price
1
-0.002
-0.011
0.02
-0.016
0.64
-0.002
1
-0.0094
0.0061
-0.019
0.45
-0.011
-0.0094
1
0.46
0.002
0.34
0.02
0.0061
0.46
1
-0.022
0.17
-0.016
-0.019
0.002
-0.022
1
0.41
0.64
0.45
0.34
0.17
0.41
1
1.0
0.8
0.6
0.4
0.2
0.0

MODEL TRAINING:

1. **Data Collection**:

   Gather a dataset that includes information about various houses and their corresponding sale prices. This data can be collected from public sources, real estate websites, or other sources. Your dataset should include features like the number of bedrooms, bathrooms, square footage, location, etc., as well as the target variable, which is the house price.

2. **Data Preprocessing**:

   - **Data Cleaning**: Handle missing values, outliers, and any inconsistencies in the data.
   - **Feature Selection/Engineering**: Choose relevant features and create new ones if necessary. For example, you might calculate the price per square foot or the age of the house from the year it was built.
   - **Normalization/Scaling**: Scale numerical features to ensure they have similar scales, which can help improve model performance.
   - **Encoding Categorical Variables**: Convert categorical variables (like location or type of house) into numerical format using techniques like one-hot encoding or label encoding.

3. **Splitting the Data**:

Split your dataset into a training set and a testing set. The training set will be used to train the model, and the testing set will be used to evaluate its performance.

4. **Choosing a Machine Learning Algorithm**:

- Linear Regression: A simple and interpretable model.
- Decision Trees: Can capture non-linear relationships in the data.
- Random Forest, Gradient Boosting: Ensemble methods that often perform well.
- Neural Networks: Deep learning models that can capture complex patterns (optional).

5. **Training the Model**:

Train the selected model on the training data. The model will learn the relationship between the input features and the target variable (house prices).

6. **Model Evaluation**:

- Use metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error

(RMSE) to evaluate the model's performance on the testing set.

- Visualize the results using plots or graphs to understand how well your model is performing.

7. **Hyperparameter Tuning**:

8. Fine-tune the model's hyperparameters to optimize its performance. You can use techniques like grid search or random search to find the best hyperparameters.

9. **Cross-Validation** (optional): Perform k-fold cross-validation to ensure the model's generalization performance. This helps assess how well the model will perform on unseen data.

10.Deployment:

Once you're satisfied with the model's performance, you can deploy it as part of a web application, mobile app, or any other platform where users can input house features and get price predictions.

Monitoring and Maintenance:

Continuously monitor the model's performance in production and update it as needed. House price trends can change, so the model may need periodic retraining.