🏠 House Price Prediction - ML Project

📌 Overview

This project predicts house prices using a **synthetic dataset** generated for properties in Indian cities like Vadodara, Ankleshwar, and Surat. It includes data generation, EDA, pre-processing, model training, evaluation, and visualization.

📚 Objective

- Generate synthetic housing price data.
- Conduct Exploratory Data Analysis (EDA).
- Preprocess data using scaling and encoding.
- Train two models:
  - Linear Regression
  - Random Forest Regressor
- Evaluate models using **MSE** and **R² Score**.
- Visualize predictions against actual prices.

🗃️ Dataset Description

Feature	Description
Size	Size in square feet
HouseType	Flat, Bungalow, Duplex, Triplex, Tenament
City	Vadodara, Ankleshwar, Surat
Area	Localities within each city
Rooms	Number of rooms (1-7)
Age	Age of property in years (0-30)
Price	Final price (Target variable)

- **Total Samples: 50**
- **Pricing Logic:**
  - Based on city and house type multipliers.
  - Adjustments for rooms, property age, and random noise.

📊 Exploratory Data Analysis (EDA):

🔎 Data Info

- No missing data.
- Mixed types: numerical & categorical.

📉 Data Description

- Basic statistical summary via `.describe()`.

📊 Visualizations

- **Price Distribution:** Slightly skewed, multimodal.
- **Correlation Heatmap:**
  - High correlation between Size and Price.
  - Moderate influence from Rooms and Age.

⚙️ **Data Pre-processing:**

- Numerical Features: Size, Rooms, Age
- Categorical Features: HouseType, City, Area

Pre-processing Techniques:
- Standard Scaler: Scales numerical values.
- One Hot Encoder: Encodes categorical variables with drop-first to prevent dummy variable trap.

Combined via Column Transformer for efficient transformation.

🤖 Model Training

📌 Train-Test Split

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

📌 Models

1. **Linear Regression**
2. **Random Forest Regressor (n_estimators=100)**

Both models are integrated with pre-processing using a **Pipeline**.

📈 Evaluation Metrics

- **Mean Squared Error (MSE)**
- **R² Score (Coefficient of Determination)**

Results:

Model	MSE (↓ Better)	R² Score (↑ Better)
Linear Regression	Low (near zero)	Close to 1
Random Forest	Very Low	Close to 1

⚠️ **Note:** Due to small dataset (50 samples), metrics can show overfitting-like perfect scores.

📊 Visualization

- **Scatter Plot:** Actual vs Predicted Prices.
- **Regression Line:** Helps visualize prediction accuracy.
- Both models plotted for comparison.

💻 Tech Stack

- **Python 3.13**

- Libraries:
  - pandas
  - numpy
  - matplotlib
  - seaborn
  - scikit-learn

💡 Conclusion

- Preprocessing pipelines streamline model input preparation.
- Both models perform well on synthetic data.
- The methodology is robust and can be adapted for real-world datasets.

🚀 Future Enhancements

- Incorporate real datasets.
- Add more features like property amenities, locality ratings.
- Tune Random Forest hyperparameters.
- Explore advanced models like:
  - Gradient Boosting
  - XGBoost
  - LightGBM

📁 Repository Structure

```
/House-Price-Prediction
│
├── synthetic_house_prices.csv  # Generated Dataset
├── house_price_prediction.ipynb  # Jupyter Notebook with full code
├── house_price_prediction.py     # Python script version
├── README.md                     # Project Documentation
```

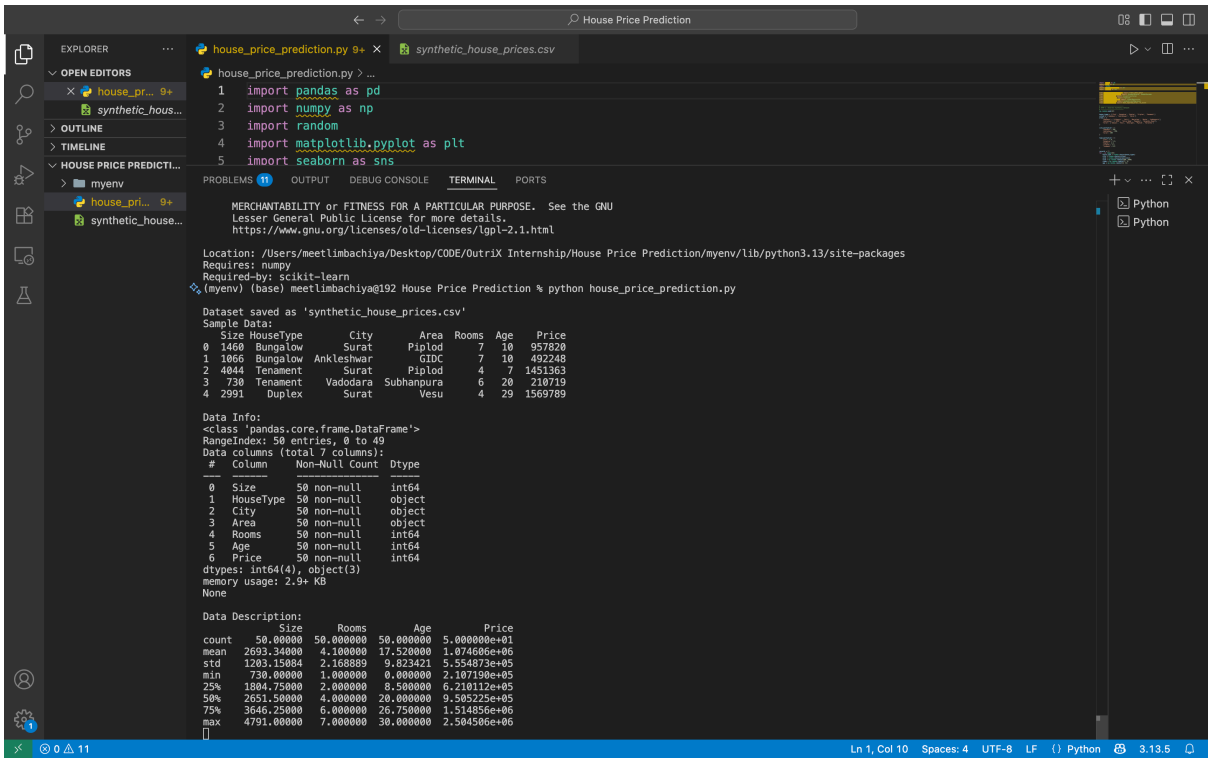
👨‍💻 Author

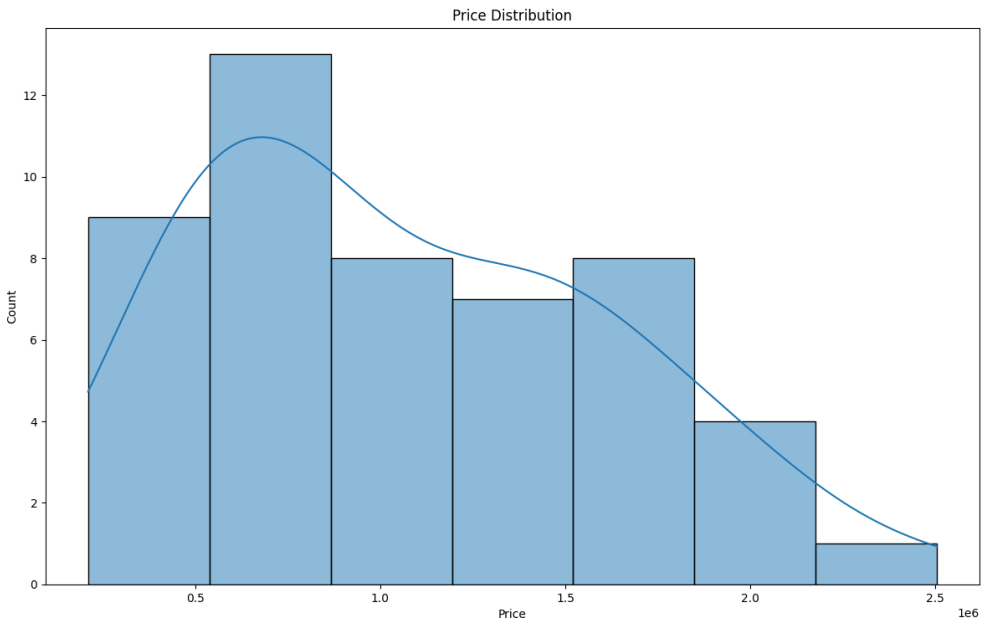
Developed as part of an internship task to demonstrate EDA, ML modeling, and data visualization in Python.

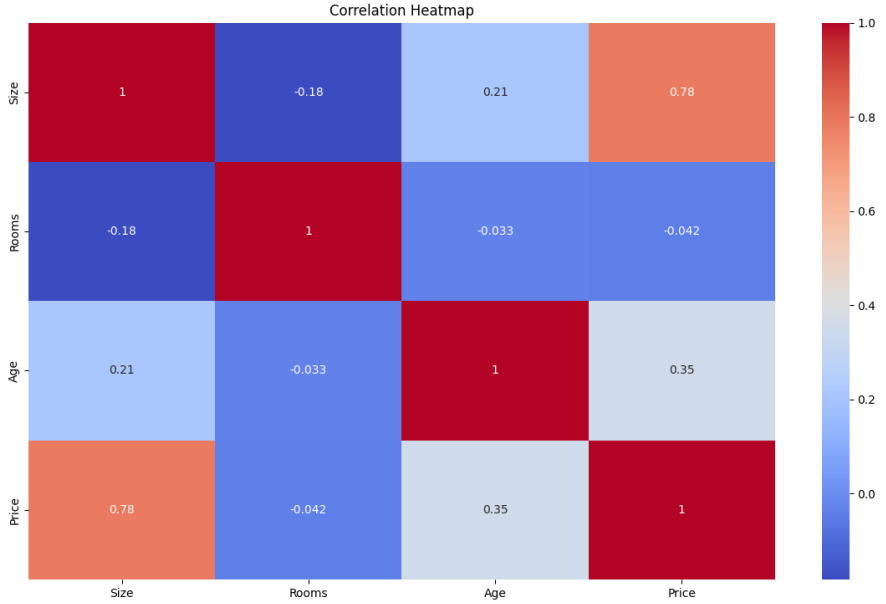
End of Report

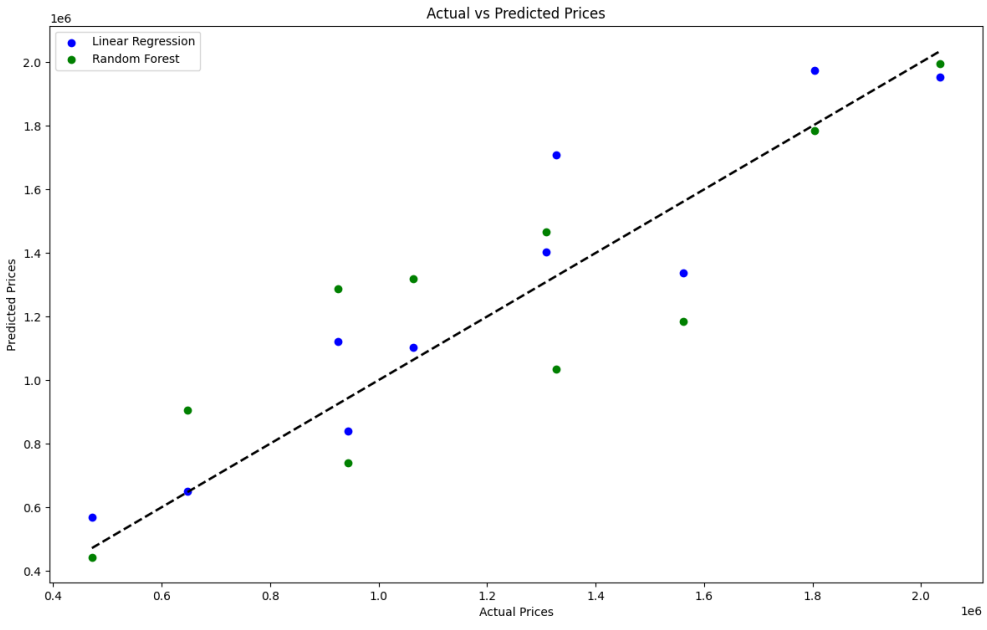
MEET LIMBACHIYA

OUTPUTS:


House Price Prediction
EXPLORER
OPEN EDITORS
house_price_prediction.py 9+
synthetic_house_prices.csv
OUTLINE
TIMELINE
HOUSE PRICE PREDICTI...
myenv
synthetic_house...
1 import pandas as pd
2 import numpy as np
3 import random
4 import matplotlib.pyplot as plt
5 import seaborn as sns
PROBLEMS 11
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.
https://www.gnu.org/licenses/old-licenses/lgpl-2.1.html
Location: /Users/meetlimbachiya/Desktop/CODE/OutriX Internship/House Price Prediction/myenv/lib/python3.13/site-packages
Requires: numpy
Required-by: scikit-learn
(myenv) (base) meetlimbachiya@192 House Price Prediction % python house_price_prediction.py
Dataset saved as 'synthetic_house_prices.csv'
Sample Data:
Size HouseType City Area Rooms Age Price
0 1460 Bungalow Surat Piplod 7 10 957820
1 1066 Bungalow Ankleshwar GIDC 7 10 492248
2 4044 Tenament Surat Piplod 4 7 1451363
3 730 Tenament Vadodara Subhanpura 6 20 210719
4 2991 Duplex Surat Vesu 4 29 1569789
Data Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 7 columns):
Column Non-Null Count Dtype
0 Size 50 non-null int64
1 HouseType 50 non-null object
2 City 50 non-null object
3 Area 50 non-null object
4 Rooms 50 non-null int64
5 Age 50 non-null int64
6 Price 50 non-null int64
dtypes: int64(4), object(3)
memory usage: 2.9+ KB
None
Data Description:
Size Rooms Age Price
count 50.00000 50.000000 50.000000 5.000000e+01
mean 2693.34000 4.100000 17.520000 1.074606e+06
std 1203.15084 2.168889 9.823421 5.554873e+05
min 730.00000 1.000000 0.000000 2.107190e+05
25% 1804.75000 2.000000 8.500000 6.210112e+05
50% 2651.50000 4.000000 20.000000 9.505225e+05
75% 3646.25000 6.000000 26.750000 1.514856e+06
max 4791.00000 7.000000 30.000000 2.504506e+06
Python
Python
Ln 1, Col 10
Spaces: 4
UTF-8
LF
Python
3.13.5


Price Distribution
Count
Price
1e6
0
2
4
6
8
10
12
0.5
1.0
1.5
2.0
2.5


Correlation Heatmap
Size
Rooms
Age
Price
1
-0.18
0.21
0.78
-0.18
1
-0.033
-0.042
0.21
-0.033
1
0.35
0.78
-0.042
0.35
1
Size
Rooms
Age
Price
1.0
0.8
0.6
0.4
0.2
0.0


Actual vs Predicted Prices
1e6
Linear Regression
Random Forest
Predicted Prices
Actual Prices
1e6
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0