PREDICTING HOUSE PRICE USING MACHINE LEARNING

Phase 3 Submission Document

PROJECT TITLE: House price prediction using machine learning

PHASE 3: Development Part 1

TASK: Start building the house price prediction model by loading and pre-processing the dataset

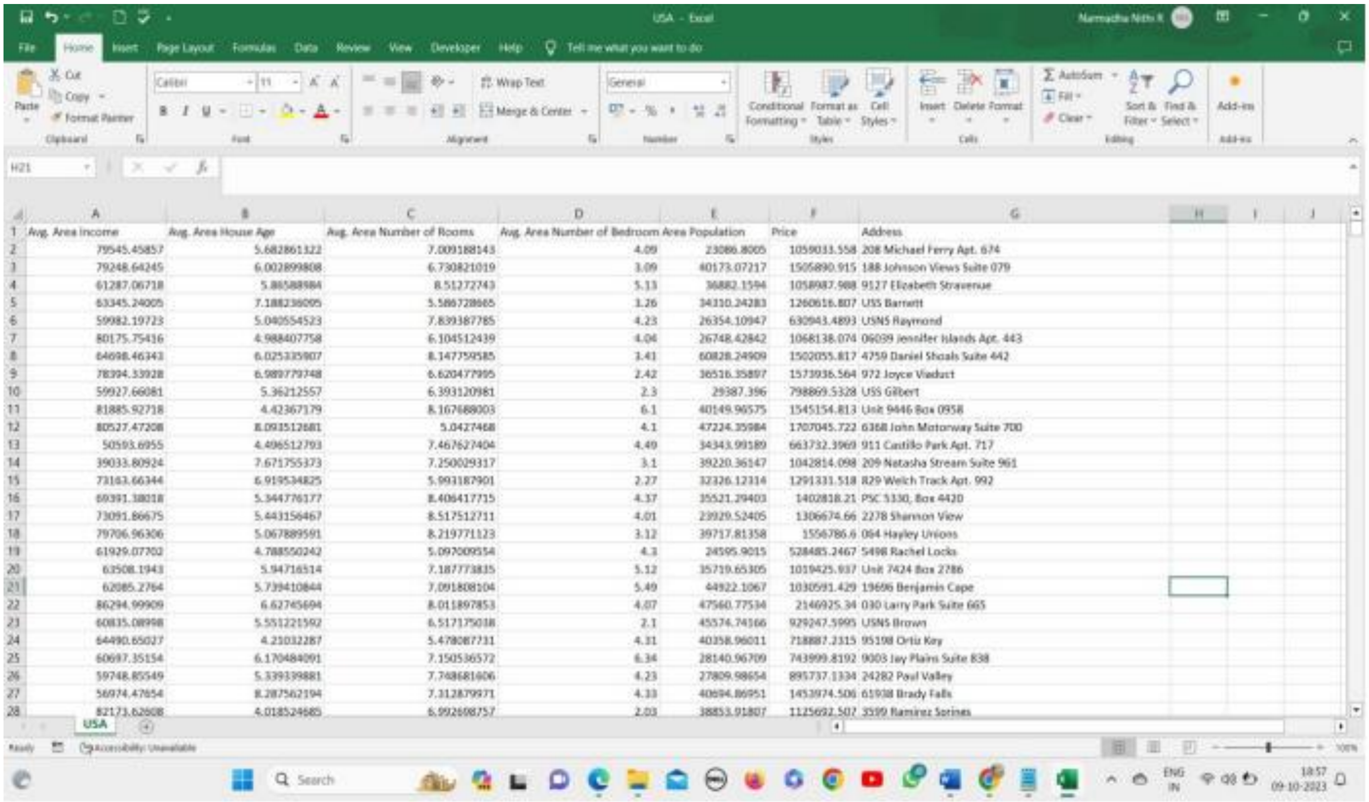
INTRODUCTION:

- Whether you're a homeowner looking to estimate the value of your property, a real estate investor seeking profitable opportunities, or a data scientist aiming to build a predictive model, the foundation of this endeavor lies in loading and preprocessing the dataset.

- Building a house price prediction model is a data-driven process that involves harnessing the power of machine learning to analyze historical housing data and make informed price predictions. This journey begins with the fundamental steps of data loading and preprocessing.

- This introduction will guide you through the initial steps of the process. We'll explore how to import essential libraries, load the housing dataset, and perform critical preprocessing steps. Data preprocessing is crucial as it helps clean, format, and prepare the data for further analysis. This includes handling missing values, encoding categorical variables, and ensuring that the data is appropriately scaled.

DATASET:



Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedroom	Area Population	Price	Address
79545.45857	5.682861322	7.009188143	4.09	23086.8005	1059033.558	208 Michael Ferry Apt. 674
79248.64245	6.002899808	6.730821019	3.09	40173.07217	1505890.915	188 Johnson Views Suite 079
61287.06718	5.86588984	8.51272743	5.13	36882.1594	1058987.988	9127 Elizabeth Stravenue
63345.24005	7.188236095	5.586728665	3.26	34310.24283	1260616.807	USS Barnett
59982.19723	5.040554523	7.839387785	4.23	26354.10947	630943.4893	USNS Raymond
80175.75416	4.988407758	6.104512439	4.04	26748.42842	1068138.074	06039 Jennifer Islands Apt. 443
64698.46343	6.025335907	8.147759585	3.41	60828.24909	1502055.817	4759 Daniel Shoals Suite 442
78394.33928	6.989779748	6.620477995	2.42	36516.35897	1573936.564	972 Joyce Viaduct
59927.66081	5.36212557	6.393120981	2.3	29387.396	798869.5328	USS Gilbert
81885.92718	4.42367179	8.167688003	6.1	40149.96575	1545154.813	Unit 9446 Box 0958
80527.47208	8.093512681	5.0427468	4.1	47224.35984	1707045.722	6368 John Motorway Suite 700
50593.6955	4.496512793	7.467627404	4.49	34343.99189	663732.3969	911 Castillo Park Apt. 717
39033.80924	7.671755373	7.250029317	3.1	39220.36147	1042814.098	209 Natasha Stream Suite 961
73163.66344	6.919534825	5.993187901	2.27	32326.12314	1291331.518	829 Welch Track Apt. 992
69391.38018	5.344776177	8.406417715	4.37	35521.29403	1402818.21	PSC 5330, Box 4420
73091.86675	5.443156467	8.517512711	4.01	23929.52405	1306674.66	2278 Shannon View
79706.96306	5.067889591	8.219771123	3.12	39717.81358	1556786.6	064 Hayley Unions
61929.07702	4.788550242	5.097009554	4.3	24595.9015	528485.2467	5498 Rachel Locks
63508.1943	5.94716514	7.187773835	5.12	35719.65305	1019425.937	Unit 7424 Box 2786
62085.2764	5.739410844	7.091808104	5.49	44922.1067	1030591.429	19696 Benjamin Cape
86294.99909	6.62745694	8.011897853	4.07	47560.77534	2146925.34	030 Larry Park Suite 665
60835.08998	5.551221592	6.517175038	2.1	45574.74166	929247.5995	USNS Brown
64490.65027	4.21032287	5.478087731	4.31	40358.96011	718887.2315	95198 Ortiz Key
60697.35154	6.170484091	7.150536572	6.34	28140.96709	743999.8192	9003 Jay Plains Suite 838
59748.85549	5.339339881	7.748681606	4.23	27809.98654	895737.1334	24282 Paul Valley
56974.47654	8.287562194	7.312879971	4.33	40694.86951	1453974.506	61938 Brady Falls
82173.62608	4.018524685	6.992698757	2.03	38853.91807	1125692.507	3599 Ramirez Springs

Dataset link: (https://www.kaggle.com/datasets/vedavyasv/usa-housing)

NECESSARY STEPS TO FOLLOW:

<u>1.Import libraries</u>

Start by importing necessary libraries

Program:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

2. Loading the dataset

Load your dataset into a Pandas DataFrame. You can typically find house price datasets in CSV format, but you can adapt this code to other formats as needed.

Program:

```
df = pd.read_csv(' E:\USA_Housing.csv ')
pd.read()
```

3. Exploratory Data Analysis (EDA)

Perform EDA to understand your data better. This includes checking for missing values, exploring the data's statistics, and visualizing it to identify patterns.

Program:

```
#Check for missing values
print(df.isnull().sum())
# Explore statistics
print(df.describe())
# Visualize the data (e.g., histograms, scatter plots, etc.)
```

4. Feature Engineering

Depending on your dataset, you may need to create new features or transform existing ones. This can involve one-hot encoding categorical variables, handling date/time data, or scaling numerical features.

Program:

```
# Example: One-hot encoding for categorical variables
df = pd.get_dummies(df, columns=[' Avg. Area Income ', ' Avg. Area House Age '])
```

5. Split the data

Split your dataset into training and testing sets. This helps you evaluate your model's performance later.

Program:

```
X = df.drop('price', axis=1) # Features
 Y = df['price'] # Target variable
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=42)
```

6.Feature scalling

Apply feature scaling to normalize your data, ensuring that all features have similar scales. Standardization (scaling to mean=0 and std=1) is a common choice.

Program:

```
 scaler = StandardScaler()
 X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

IMPORTANCE OF LOADING AND PREPROCESSING OF DATASET:

Loading and preprocessing the dataset is an important first step in building any machine learning model. However, it is especially important for house price prediction models, as house price datasets are often complex and noisy.

By loading and preprocessing the dataset, we can ensure that the machine learning algorithm is able to learn from the data effectively and accurately.

1.Loading the dataset

- ✓ Loading the dataset using machine learning is the process of bringing the data into the machine learning environment so that it can be used to train and evaluate a model.

- ✓ The specific steps involved in loading the dataset will vary depending on the machine learning library or framework that is being used. However, there are some general steps that are common to most machine learning frameworks:

a.Identify the dataset:

The first step is to identify the dataset that you want to load. This datasetThis dataset may be stored in a local file, in a database, or in a cloud storage service

b.Load the dataset:

Once you have identified the dataset, you need to load it into the machine learning environment. This may involve using a built-in function in the machine learning library, or it may involve writing your own code.

c.Preprocess the dataset:

Once the dataset is loaded into the machine learning environment, you may need to preprocess it before you can start training and evaluating your model. This may involve cleaning the data, transforming the data into a suitable format, and splitting the data into training and test sets.

Program:

```
import pandas as pd
 import numpy as np
import seaborn as sns
 import matplotlib.pyplot as plt
from sklearn.model_selection
import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import r2_score, mean_absolute_error,mean_squared_error
 from sklearn.linear_model import LinearRegression
 from sklearn.linear_model import Lasso
 from sklearn.ensemble import RandomForestRegressor
 from sklearn.svm import SVR import xgboost as xg
 %matplotlib inline
import warnings warnings.filterwarnings("ignore")
 /opt/conda/lib/python3.10/site-packages/scipy/_init_.py:146:
```

UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for this version of SciPy (detected verson 1.23.5)

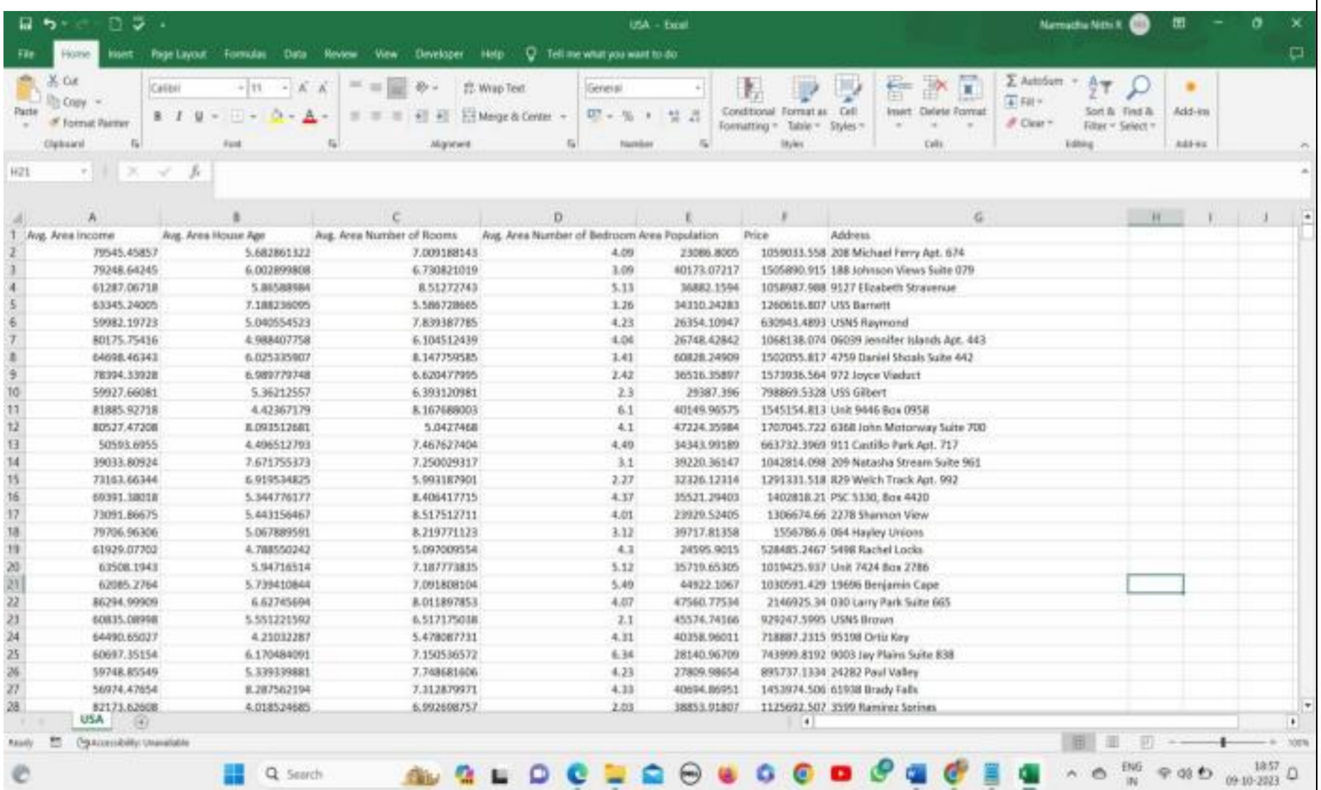
warnings.warn(f"A NumPy version >={np_minversion} and<{np_maxversion}"

<u>Loading dataset</u>

Dataset=pd.read_csv(‘E:/USA_Housing.csv’)

<u>Data Explorations</u>

Dataset Output:



	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedroom	Area Population	Price	Address
2	79545.45857	5.682861322	7.009188143	4.09	23086.8005	1059033.558	208 Michael Ferry Apt. 674
3	79248.64245	6.002899808	6.730821019	3.09	40173.07217	1505890.915	188 Johnson Views Suite 079
4	61287.06718	5.86588984	8.51272743	5.13	36882.1594	1058987.988	9127 Elizabeth Stravenue
5	63345.24005	7.188236095	5.586728665	3.26	34310.24283	1260616.807	USS Barnett
6	59982.19723	5.040554523	7.839387785	4.23	26354.10947	630943.4893	USNS Raymond
7	80175.75416	4.988407758	6.104512439	4.04	26748.42842	1068138.074	06039 Jennifer Islands Apt. 443
8	64698.46343	6.025335907	8.147759585	3.41	60828.24909	1502055.817	4759 Daniel Shoals Suite 442
9	78394.33928	6.989779748	6.620477995	2.42	36516.35897	1573936.564	972 Joyce Viaduct
10	59927.66081	5.36212557	6.393120981	2.3	29387.396	798869.5328	USS Gilbert
11	81885.92718	4.42367179	8.167688003	6.1	40149.96575	1545154.813	Unit 9446 Box 0958
12	80527.47208	8.093512681	5.0427468	4.1	47224.35984	1707045.722	6368 John Motorway Suite 700
13	50593.6955	4.496512793	7.467627404	4.49	34343.99189	663732.3969	911 Castillo Park Apt. 717
14	39033.80924	7.671755373	7.250029317	3.1	39220.36147	1042814.098	209 Natasha Stream Suite 961
15	73163.66344	6.919534825	5.993187901	2.27	32326.12314	1291331.518	829 Welch Track Apt. 992
16	69391.38018	5.344776177	8.406417715	4.37	35521.29403	1402818.21	PSC 5330, Box 4420
17	73091.86675	5.443156467	8.517512711	4.01	23929.52405	1306674.66	2278 Shannon View
18	79706.96306	5.067889591	8.219771123	3.12	39717.81358	1556786.6	064 Hayley Unions
19	61929.07702	4.788550242	5.097009554	4.3	24595.9015	528485.2467	5498 Rachel Locks
20	63508.1943	5.94716514	7.187773835	5.12	35719.65305	1019425.937	Unit 7424 Box 2786
21	62085.2764	5.739410844	7.091808104	5.49	44922.1067	1030591.429	19696 Benjamin Cape
22	86294.99909	6.62745694	8.011897853	4.07	47560.77534	2146925.34	030 Larry Park Suite 665
23	60835.08998	5.551221592	6.517175038	2.1	45574.74166	929247.5995	USNS Brown
24	64490.65027	4.21032287	5.478087731	4.31	40358.96011	718887.2315	95198 Ortiz Key
25	60697.35154	6.170484091	7.150536572	6.34	28140.96709	743999.8192	9003 Jay Plains Suite 838
26	59748.85549	5.339339881	7.748681606	4.23	27809.98654	895737.1334	24282 Paul Valley
27	56974.47654	8.287562194	7.312879971	4.33	40694.86951	1453974.506	61938 Brady Falls
28	82173.62608	4.018524685	6.992608757	2.03	38853.91807	1125692.507	3599 Ramirez Springs

<u>2.Preprocessing of dataset</u>

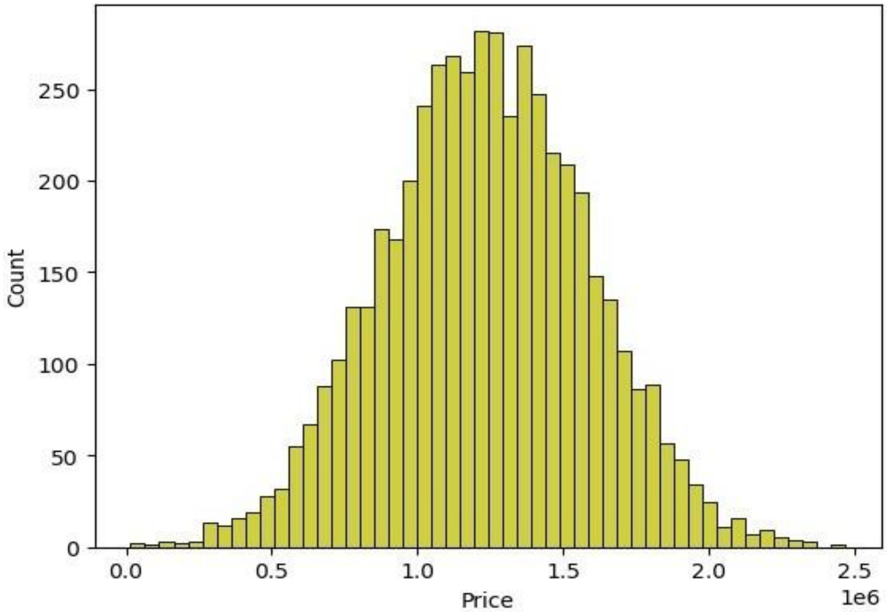
Data Sources: Gathering comprehensive datasets from real estate listings, including features such as square footage, location, number of bedrooms, and more.

Data Cleaning: Removing missing values, handling outliers, and encoding categorical variables for compatibility with machine learning algorithm.

<u>Visualisation and Pre-Processing of Data:</u>

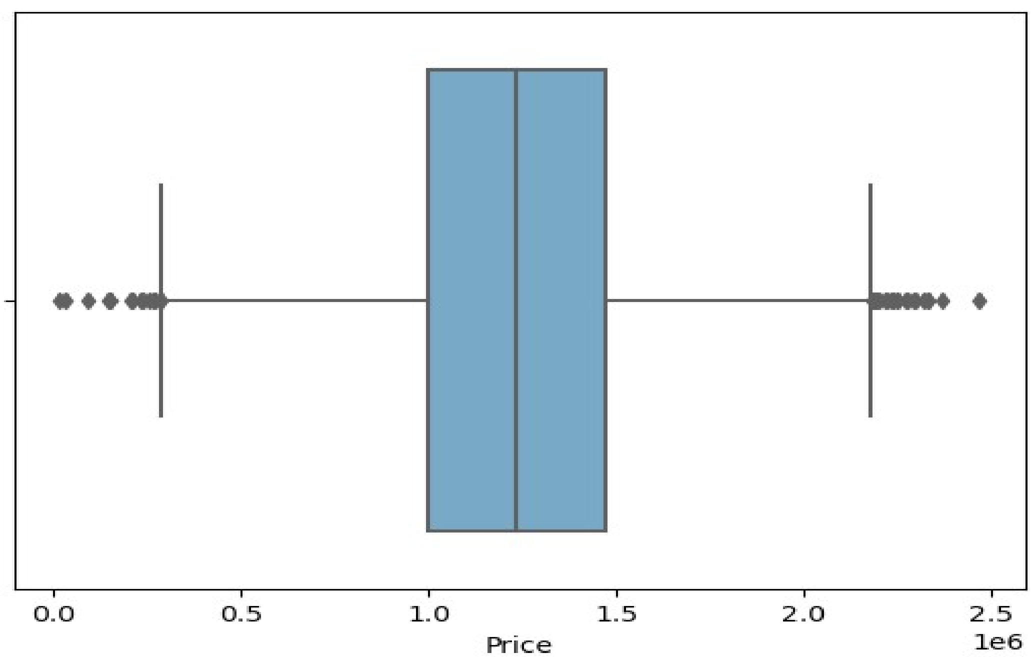
In [1]:

sns.histplot(dataset, x='Price', bins=50, color='y')



Out[1]:

<Axes: xlabel=’Prices’,ylabel=’count’>



ln[2]:

sns.boxplot(dataset, x='Price', palette='Blues')

Out[2]:

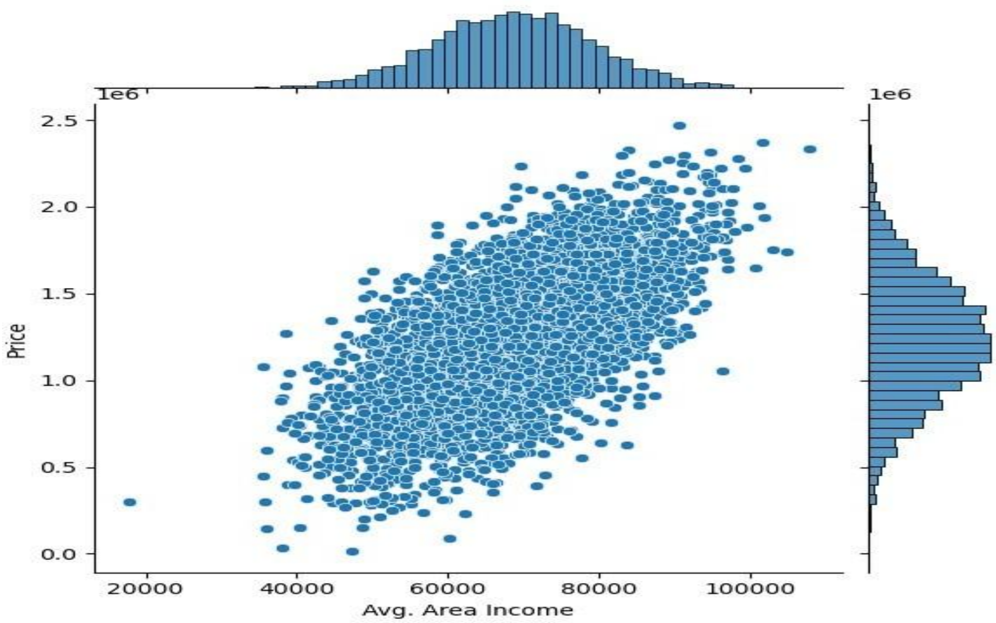
<Axes: xlabel='Price'>

In [3]:

```
sns.jointplot(dataset, x='Avg. Area House Age', y='Price', kind='hex')
```

Out[3]:

<seaborn.axisgrid.JointGrid at 0x7caf1d571810>

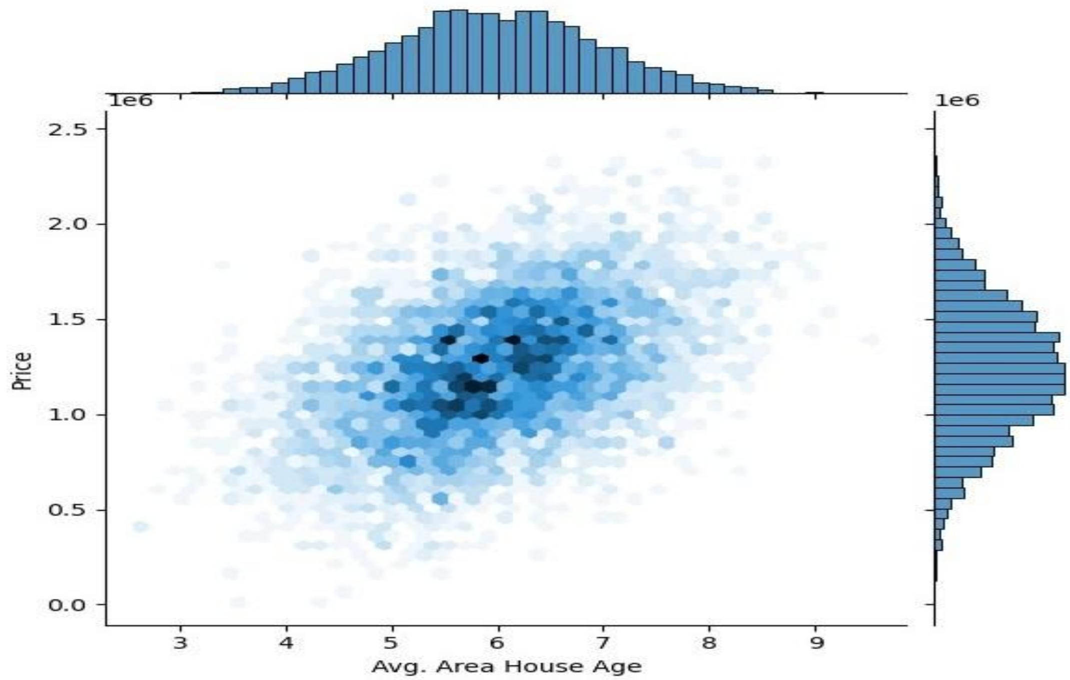


In[4]:

```
sns.jointplot(dataset, x='Avg. Area Income', y='Price')
```

Out[4]:

<seaborn.axisgrid.JointGrid at 0x7caf1d8bf7f0>



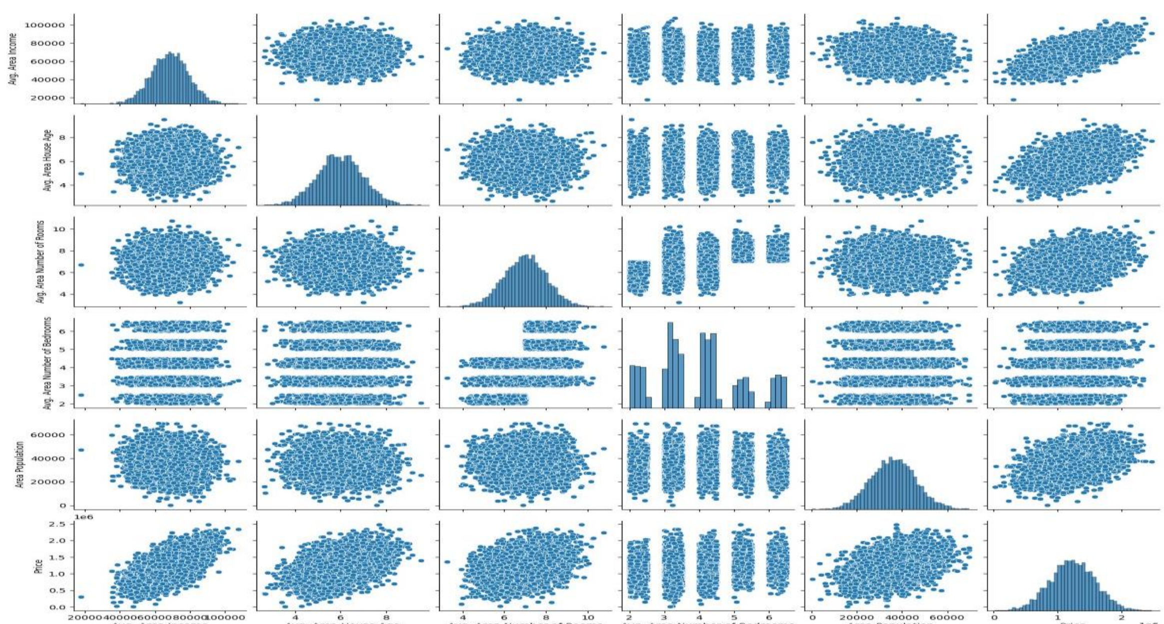
In [5]:

```
plt.figure(figsize=(12,8))sns.pairplot(dataset)
```

Out[5]:

<seaborn.axisgrid.PairGrid at 0x7caf0c2ac550>

<Figure size 1200x800 with 0 Axes>



In [6]:

```
dataset.hist(figsize=(10,8))
```

Out[6]:

array([[<Axes: title={'center': 'Avg. Area Income'}>

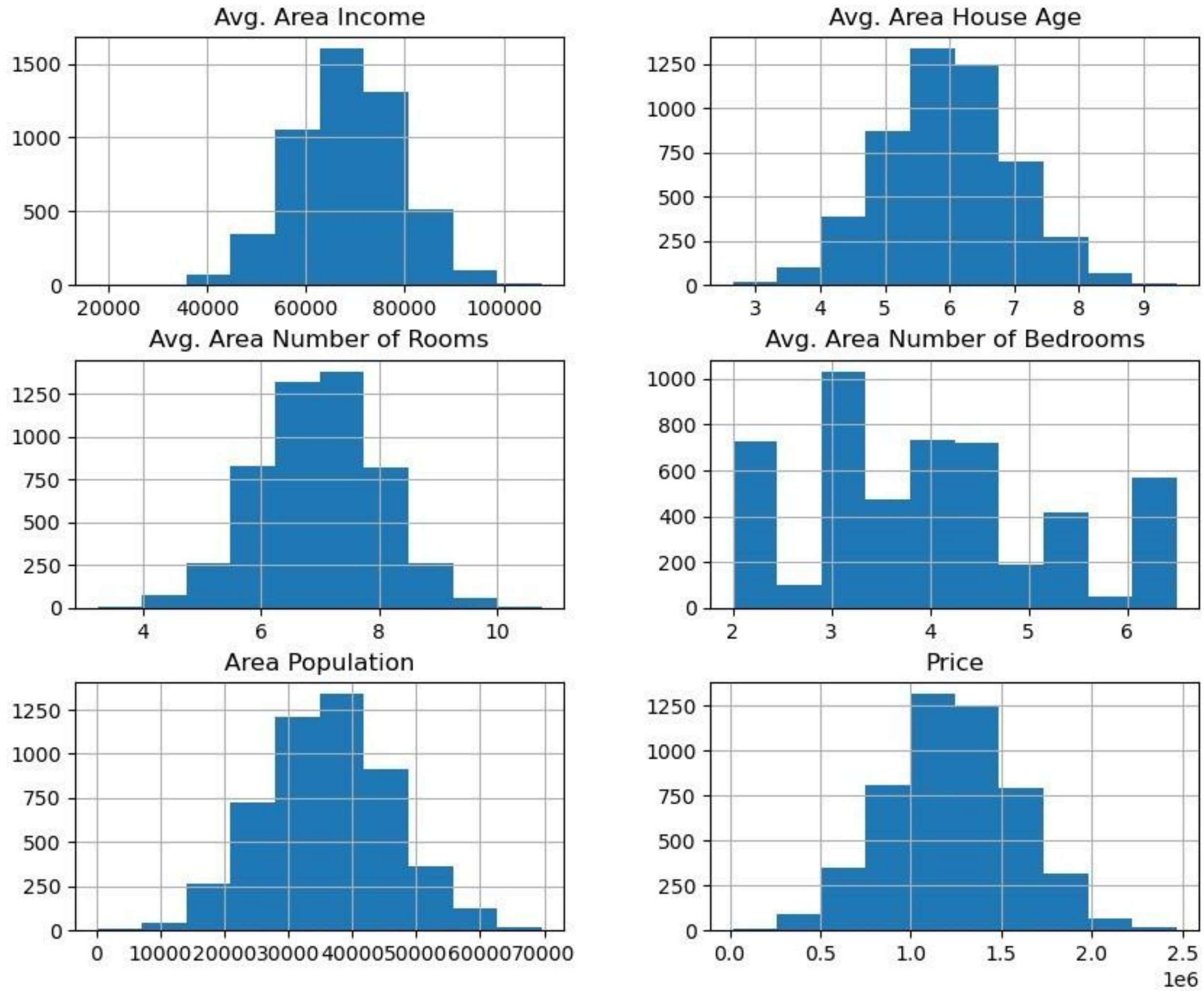
<Axes: title={'center': 'Avg. Area House Age'}>],

[<Axes: title={'center': 'Avg. Area Number of Rooms'}>,

<Axes: title={'center': 'Avg. Area Number of Bedrooms'}>],

[<Axes: title={'center': 'Area Population'}>,

<Axes: title={'center': 'Price'}>]], dtype=object)



Visualizing Correlation

In [7]:

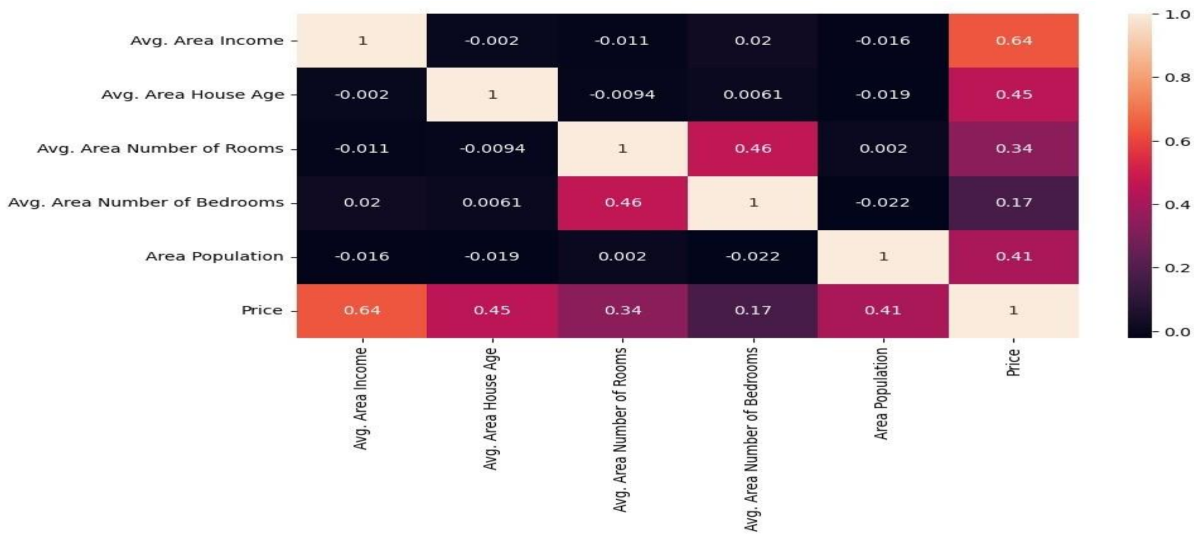
```
dataset.corr(numeric_only=True)
```

In [8]:

```
plt.figure(figsize=(10,5))sns.heatmap(dataset.corr(numeric_only = True), annot=True)
```

Out[8]:

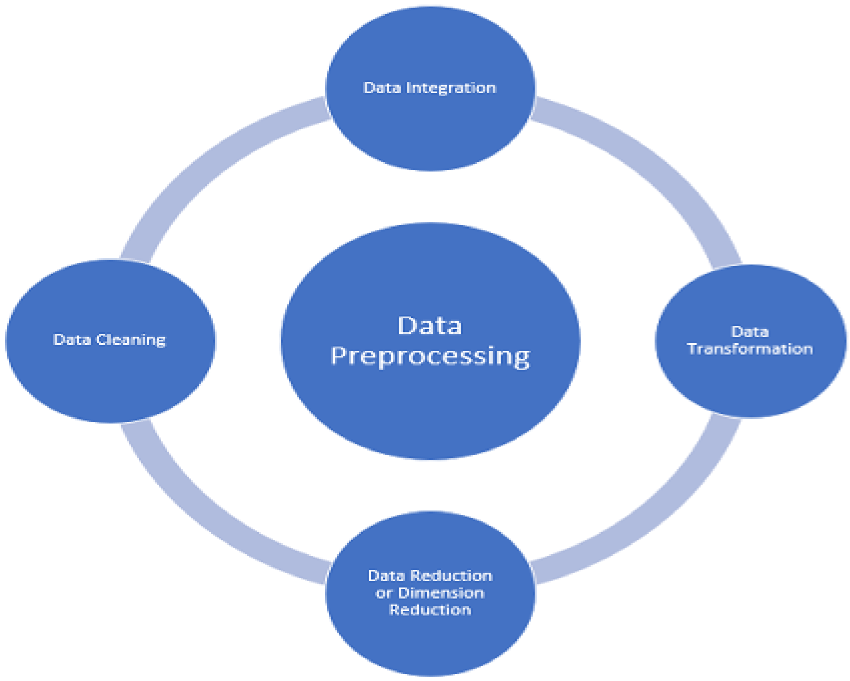
<Axes: >



Some common data preprocessing tasks include:

- **Data cleaning:** This involves identifying and correcting errors and inconsistencies in the data. For example, this may involve removing duplicate records, correcting typos, and filling in missingvalues.
- **Data transformation:** This involves converting the data into a format that is suitable for the analysis task. For example, this mayinvolve converting categorical data to numerical data, or scaling the data to a suitable range.
- **Feature engineering:** This involves creating new features from the existing data. For example, this may involve creating features that represent interactions between variables, or features that represent summary statistics of the data.
- **Data integration:** This involves combining data from multiplesources into a single dataset. This may involve resolving inconsistencies in the data, such as different data formats or different variable names.

- Data preprocessing is an essential step in many data science projects. By carefully preprocessing the data, data scientists can improve the accuracy and reliability of their results.



Program:

```
# Importing necessary libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder from
sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
```

 # Step 1: Load the dataset

data = pd.read_csv('E:\USA_Housing.csv')

 # Step 2: Exploratory Data Analysis (EDA)

print("--- Exploratory Data Analysis ---") print("1. Checking for Missing Values:") missing_values = data.isnull().sum() print(missing_values)

print("\n2. Descriptive Statistics:") description = data.describe( )

 # Step 3: Feature Engineering

print("\n--- Feature Engineering ---")

Separate features and target variable X = data.drop('price', axis=1)

y = data['price']

Define which columns should be one-hot encoded (categorical) categorical_cols = [' Avg. Area House Age']

Define preprocessing steps using ColumnTransformer and Pipeline preprocessor = ColumnTransformer(

transformers=[

('num', StandardScaler(), [' Avg. Area Number of Rooms ', ' Avg. Area Number of Bedrooms ', ' Area Population ', ' Avg. Area Income ']),('cat', OneHotEncoder(), categorical_cols)])

Step 4: Data Splitting

```
print("\n--- Data Splitting ---")

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

print(f"X_train shape: {X_train.shape}")

print(f"X_test shape: {X_test.shape}")

print(f"y_train shape: {y_train.shape}")

print(f"y_test shape: {y_test.shape}")
```

Step 5: Preprocessing and Feature Scaling using Pipeline

```
print("\n--- Feature Scaling ---")

model = Pipeline([

    ('preprocessor', preprocessor),

])

# Fit the preprocessing pipeline on the training dataX_train

= model.fit_transform(X_train)

# Transform the testing data using the fitted pipelineX_test

= model.transform(X_test)


print("--- Preprocessing Complete! ---")
```

Output:

Exploratory Data Analysis:

1. Checking for Missing Values:

Avg. Area Income	0
Avg. Area House Age	0
Avg. Area Number of Rooms	0
Avg. Area Number of Bedrooms	0
Area Population	0
Price	0
Address	0

2. Descriptive Statistics:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms
count	5000.000000	5000.000000	5000.000000	5000.000000
mean	62748.865	6.028323445	6.997892	4.25
std	2500.025031	3.934212	3.979123	1.462725
min	17796.63	2.644304186	3.236194	2
max	107701.7	9.519088066	10.75959	6.5

Area Population	Price
5000.000000	5000.000000

34897.16035	20314.66
1.469203	50.504174
172.6107	15938.66
69621.71	2469066

Data Splitting;

X_train shape: (800, 7)

X_test shape: (200, 7)

y_train shape: (800,)

y_test shape: (200,)

Conclusion:

- ✓ In the quest to build a house price prediction model, we have embarked on a critical journey that begins with loading and preprocessing the dataset.We have traversed through essential steps, starting with importing the necessary libraries to facilitate data manipulation and analysis.
- ✓ Understanding the data's structure, characteristics, and any potential issues through exploratory data analysis (EDA) isessential for informed decision-making.
- ✓ Data preprocessing emerged as a pivotal aspect of this process. It involves cleaning, transforming, and refining the dataset to ensurethat it aligns with the requirements of machine learning algorithms.
- ✓ With these foundational steps completed, our dataset is now primed for the subsequent stages of building and training a house price prediction model.