Predicting House Price Using Machine Learning

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

Given data set:

index	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Address
0	79545.45857431678	5.682861321615587	7.009188142792237	4.09	23086.800502686456	1059033.5578701235	208 Michael Ferry Apt. 674 Laurabury, NE 37010-5101
1	79248.64245482568	6.0028998082752425	6.730821019094919	3.09	40173.07217364482	1505890.91484695	188 Johnson Views Suite 079 Lake Kathleen, CA 48958
2	61287.067178656784	5.865889840310001	8.512727430375099	5.13	36882.15939970458	1058987.9878760849	9127 Elizabeth Stravenue Danieltown, WI 06482-3489
3	63345.24004622798	7.1882360945186425	5.586728664827653	3.26	34310.24283090706	1260616.8066294468	USS Barnett FPO AP 44820
4	59982.197225708034	5.040554523106283	7.839387785120487	4.23	26354.109472103148	630943.4893385402	USNS Raymond FPO AE 09386
5	80175.7541594853	4.9884077575337145	6.104512439428879	4.04	26748.428424689715	1068138.0743935304	06039 Jennifer Islands Apt. 443 Tracyport, KS 16077
6	64698.46342788773	6.025335906887153	8.147759585023431	3.41	60828.24908540716	1502055.8173744078	4759 Daniel Shoals Suite 442 Nguyenburgh, CO 20247
7	78394.33927753085	6.9897797477182815	6.620477995185026	2.42	36516.35897249384	1573936.5644777217	972 Joyce Viaduct Lake William, TN 17778-6483
8	59927.66081334963	5.36212556960358	6.3931209805509015	2.3	29387.39600281585	798869.5328331633	USS Gilbert FPO AA 20957
9	81885.92718409566	4.423671789897876	8.167688003472351	6.1	40149.96574921337	1545154.8126419624	Unit 9446 Box 0958 DPO AE 97025
10	80527.47208292288	8.09351268063935	5.042746799645982	4.1	47224.35984022191	1707045.722158058	6368 John Motorway Suite 700 Janetbury, NM 26854
11	50593.69549704281	4.496512793097035	7.467627404008019	4.49	34343.991885578806	663732.3968963273	911 Castillo Park Apt. 717 Davisborough, PW 78603
12	39033.809236982364	7.671755372854428	7.250029317273495	3.1	39220.36146737246	1042814.0978200928	209 Natasha Stream Suite 961

Necessary step to follow:

Import Libraries:

Start by importing the necessary libraries:

Program:

```
import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler
```

Load the Dataset:

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

Program:

```
data = pd.read_csv("/content/USA_Housing.csv")
```

```
data
```

	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
...	...	...	...	...	...	...	...
4995	60567.944140	7.830362	6.137356	3.46	22837.361035	1.060194e+06	USNS Williams\nFPO AP 30153-7653

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:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   Avg. Area Income              5000 non-null   float64
 1   Avg. Area House Age           5000 non-null   float64
 2   Avg. Area Number of Rooms     5000 non-null   float64
 3   Avg. Area Number of Bedrooms  5000 non-null   float64
 4   Area Population               5000 non-null   float64
 5   Price                         5000 non-null   float64
 6   Address                       5000 non-null   object 
dtypes: float64(6), object(1)
memory usage: 273.6+ KB
```

Importance of loading and processing 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.

Challenges involved in loading and preprocessing a house price dataset:

There are a number of challenges involved in loading and preprocessing a house price dataset, including: Handling missing values:

House price datasets often contain missing values, which can be due to a variety of factors, such as human error or incomplete data collection. Common methods for handling missing values include dropping the rows with missing values, imputing the missing values with the mean or median of the feature, or using a more sophisticated method such as multiple imputation.

```
# looking for the null values
total_null_values = data.isnull().sum()

# calculating total values
total_values = data.count().sort_values(ascending=True)

# calculating the percentage of null values
null_values_percentage = total_null_values/total_values *100

# converting to dataframe of missing values
missing_values = pd.concat({'Total Values' : total_values, 'Null_values': total_null_values, 'Percentage of Missing Values': null_values_percentage}, axis=1)

# display missing values
print(missing_values)
```

```
                              Total Values  Null_values  \
Avg. Area Income                      5000            0
Avg. Area House Age                   5000            0
Avg. Area Number of Rooms             5000            0
Avg. Area Number of Bedrooms          5000            0
Area Population                       5000            0
Price                                 5000            0
Address                               5000            0

                              Percentage of Missing Values
Avg. Area Income                                       0.0
Avg. Area House Age                                    0.0
Avg. Area Number of Rooms                              0.0
Avg. Area Number of Bedrooms                           0.0
Area Population                                        0.0
Price                                                  0.0
Address                                                0.0
```

```
# looking for duplicated values
duplicated_values = data.duplicated()
```

```
# number of duplicated values in dataset
print("The number of duplicated records in dataset is {}".format(duplicated_values.sum()))
```

Output:

The number of duplicated records in dataset is 0

Split the Data:

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

```
from sklearn.model_selection import train_test_split

x = data.drop(["Price"], axis = 1)
y = data['Price']
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.2)
train_data = x_train.join(y_train)
train_data
```

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Address	Price
1401	45990.123742	6.788987	7.207151	3.36	41553.839562	237 Stephanie Corner\nWest Thomas, FL 46330	1.043968e+06
4001	76609.917237	3.770548	6.700456	3.21	55430.311566	624 Alyssa Plains Apt. 752\nNew Charlesstad, D...	1.214689e+06
3373	67395.329746	6.801526	7.810228	6.10	54687.821830	560 Renee Turnpike Suite 782\nBlanchardland, L...	1.670950e+06
2563	73480.214762	6.181028	6.677806	4.35	19481.385125	481 Marquez Plaza Suite 172\nNorth Jeffreyboro...	1.075550e+06
4104	61687.394423	5.507913	6.995603	3.34	45279.163966	908 Davis Mount\nHarrisonburgh, RI 63275	1.210827e+06
...	...	...	...	...	...	...	...
4128	63918.051581	5.099158	7.662942	4.26	48669.649744	637 Jodi Flat\nLake Jose, NV 35151-3084	1.095035e+06
806	57216.212818	6.855448	8.234531	6.32	35046.013145	773 Zachary Turnpike\nSouth Vanessamouth, NJ 8...	1.380715e+06
4735	65068.554529	4.710401	6.455830	2.31	29647.564306	854 Jessica Junction Suite 225\nNorth Elizabet...	6.919414e+05
1450	66252.144789	5.238126	5.709672	3.14	33656.230961	38431 Gomez Motorway\nPatriciahaven, AS	6.649789e+05

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 dataset may be stored in a local file, in a database, or in a cloud storage service.

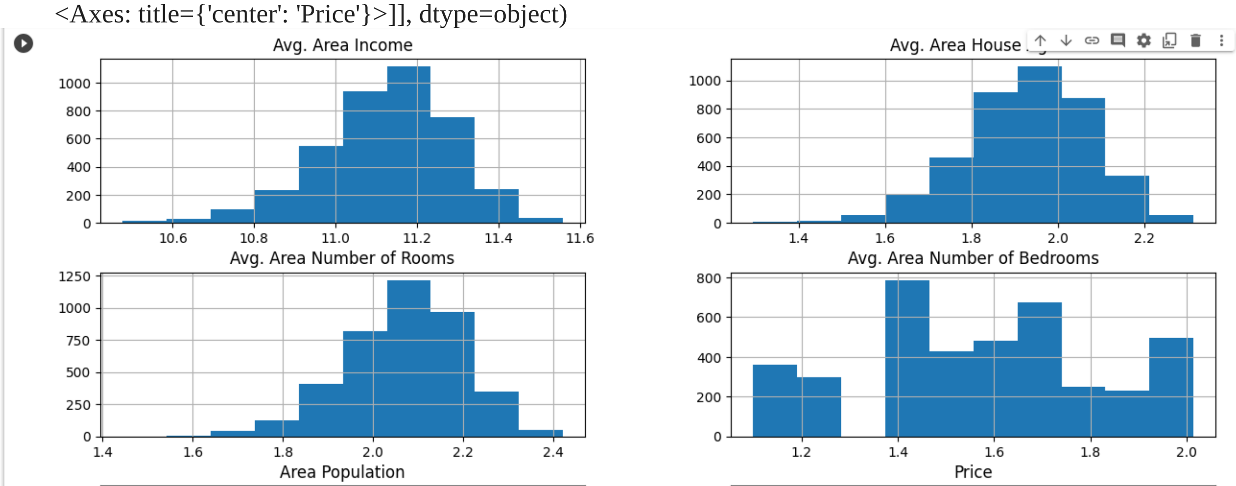
b. Load the dataset: Once you have identified the dataset, you need to load it into the machine learning environment. This may involve using a built-in function in the machine learning library, or it may involve writing your own code.

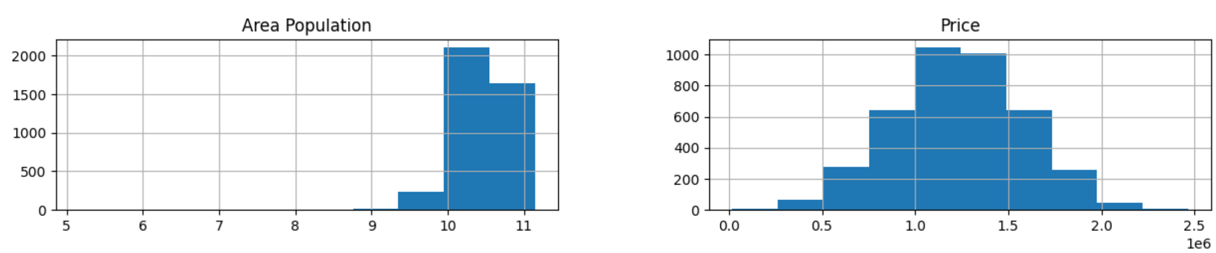
c. Preprocess the dataset: Once the dataset is loaded into the machine learning environment, you may need to preprocess it before you can start training and evaluating your model. This may involve cleaning the data, transforming P a g e| 9the data into a suitable format, and splitting the data into training and test sets.

Aggregating of Data:

```
train_data.hist(figsize=(15,8))
```

```
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)
```





Visualization of data:

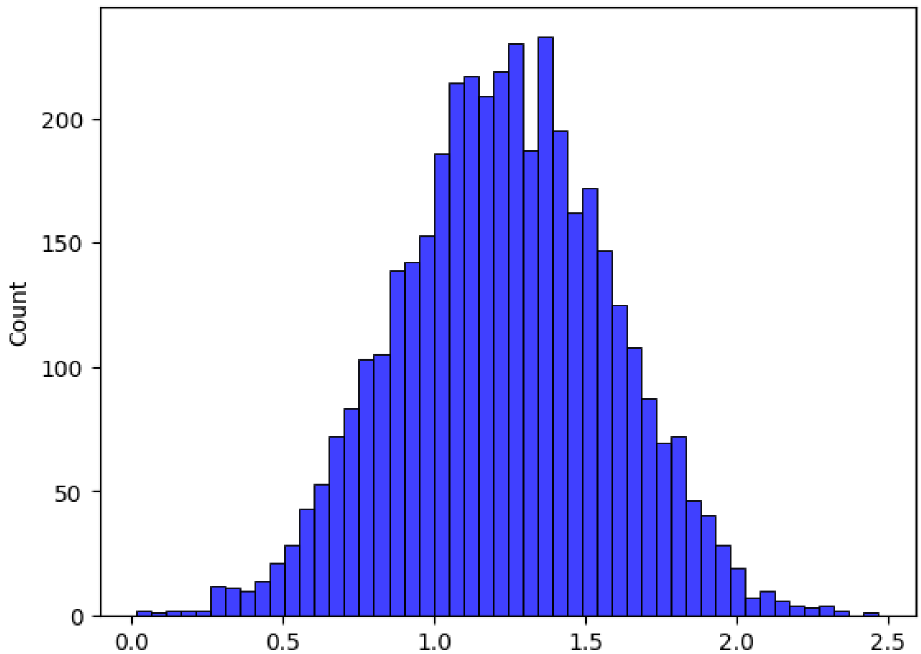
```
plt.figure(figsize=(15,8))
train_data["Avg. Area Income"] = np.log(train_data["Avg. Area Income"]+1)
train_data["Avg. Area House Age"] = np.log(train_data["Avg. Area House Age"]+1)
```

```python
train_data["Avg. Area Number of Rooms"] = np.log(train_data["Avg. Area Number of Rooms"] + 1)

train_data["Avg. Area Number of Bedrooms"] = np.log(train_data["Avg. Area Number of Bedrooms"] + 1)
train_data["Area Population"] = np.log(train_data["Area Population"] + 1)

train_data.hist(figsize=(15,8))
sns.histplot(train_data,x="Price",bins=50,color="blue")
```

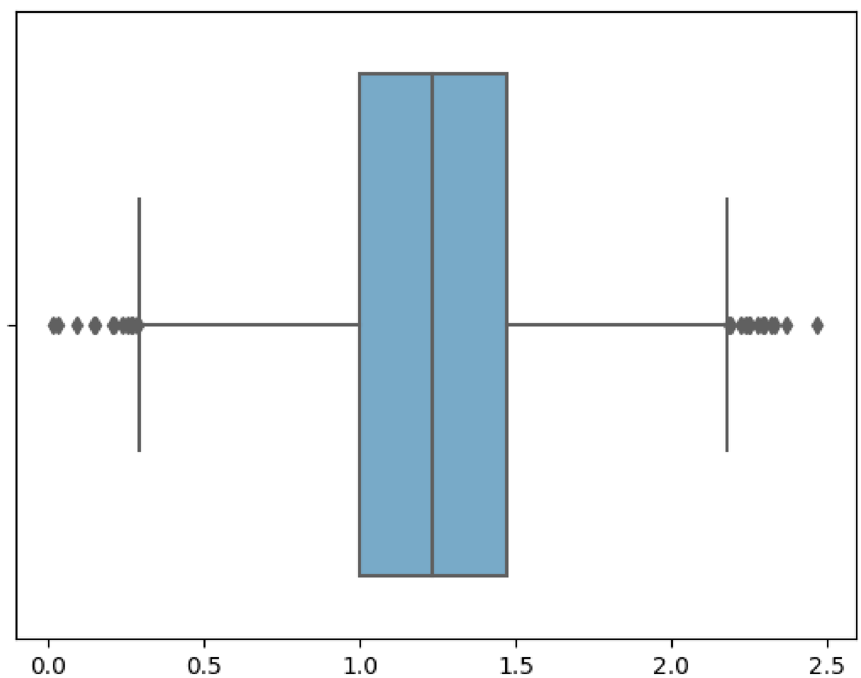
```
<Axes: xlabel='Price', ylabel='Count'>
```



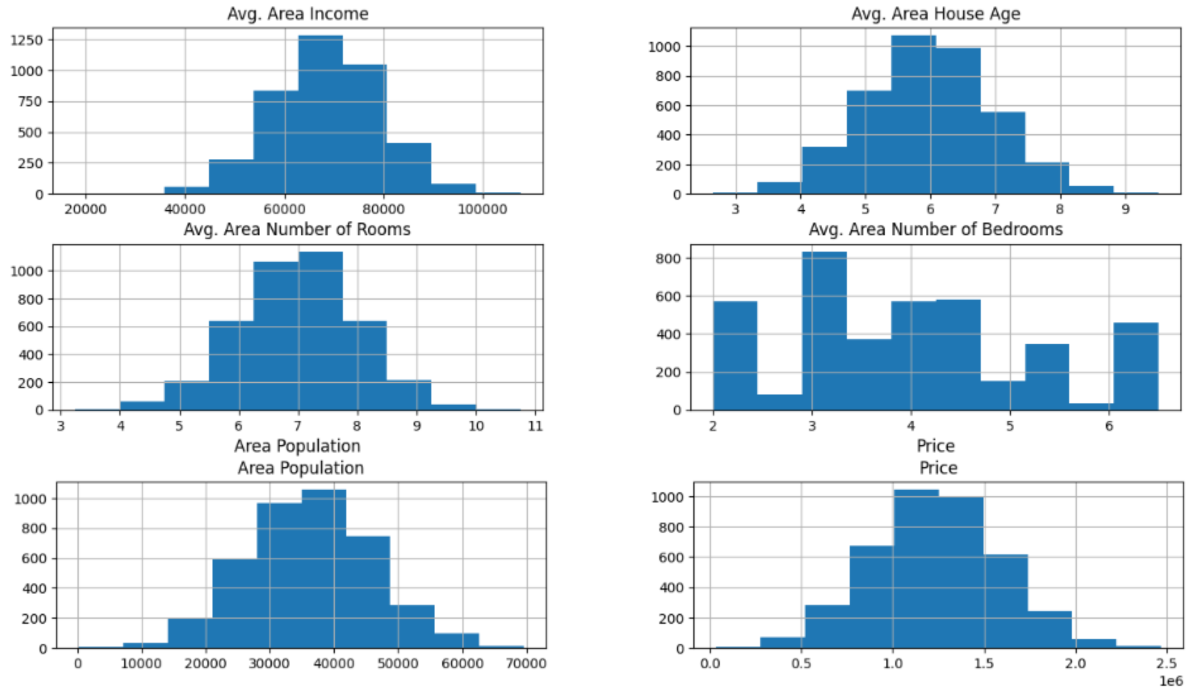

```python
plt.figure(figsize=(15,8))
sns.heatmap(train_data.corr(), annot=True, cmap="YlGnBu")
```


```
sns.boxplot(train_data,x="Price",palette="Blues")
```

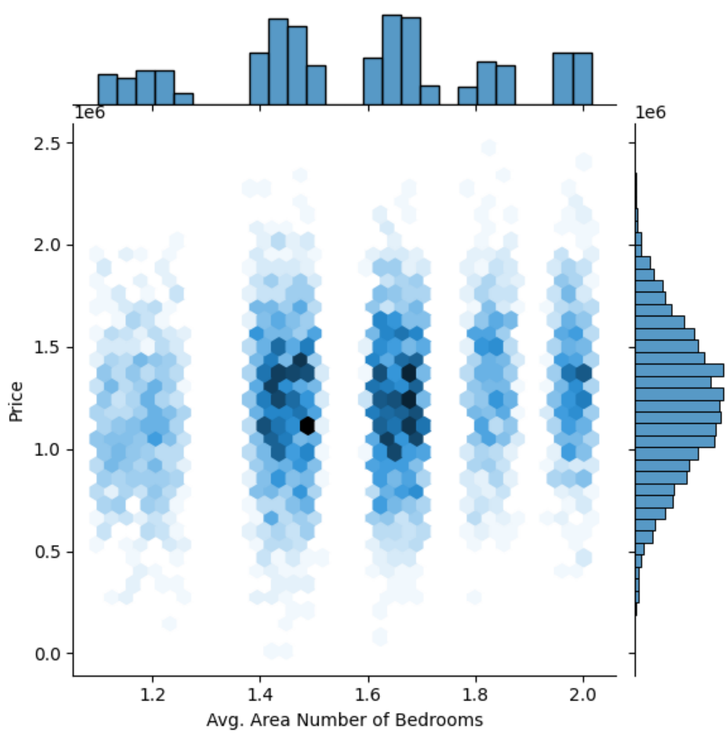
<Axes: xlabel='Price'>



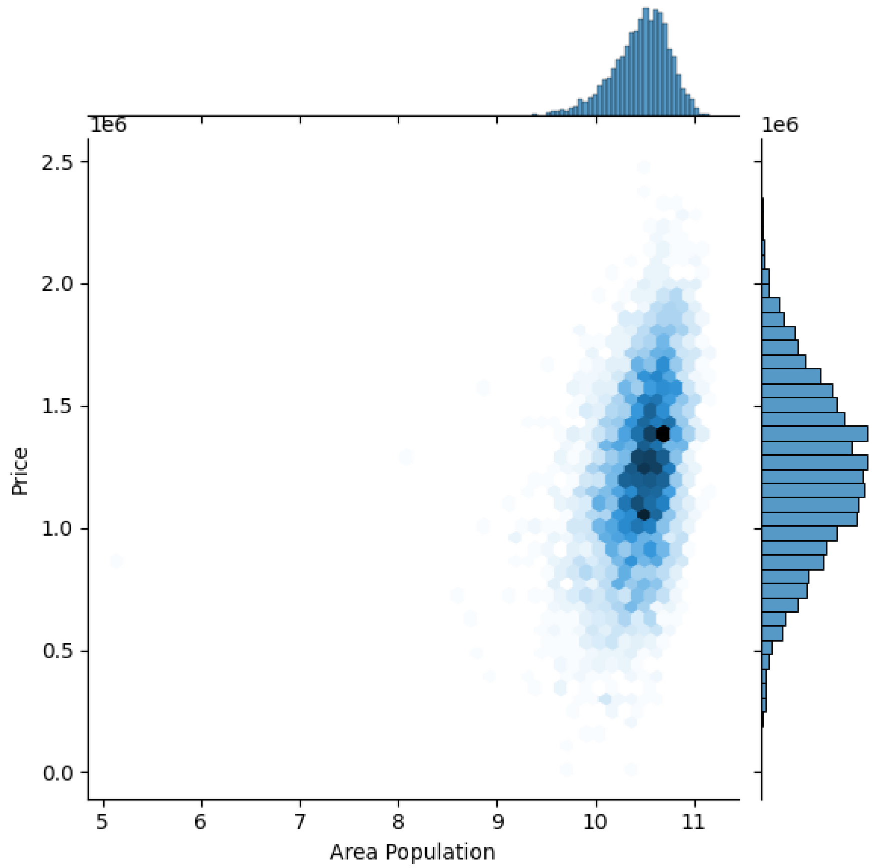

```
train_data.hist(figsize=(15,8))
```



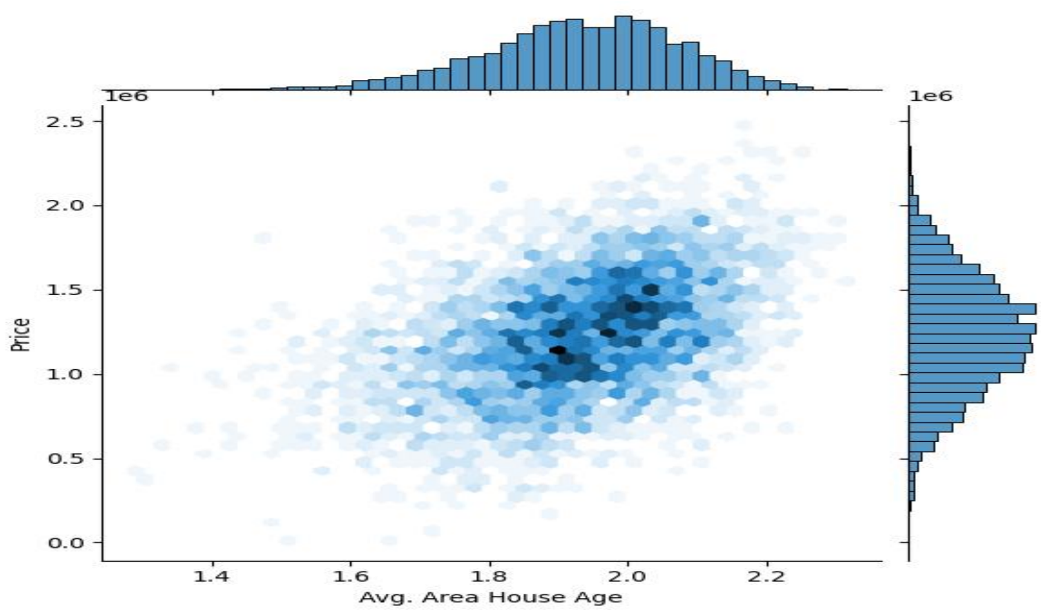

```
sns.jointplot(train_data,x="Avg. Area Number of Bedrooms",y="Price",kind="hex")
```



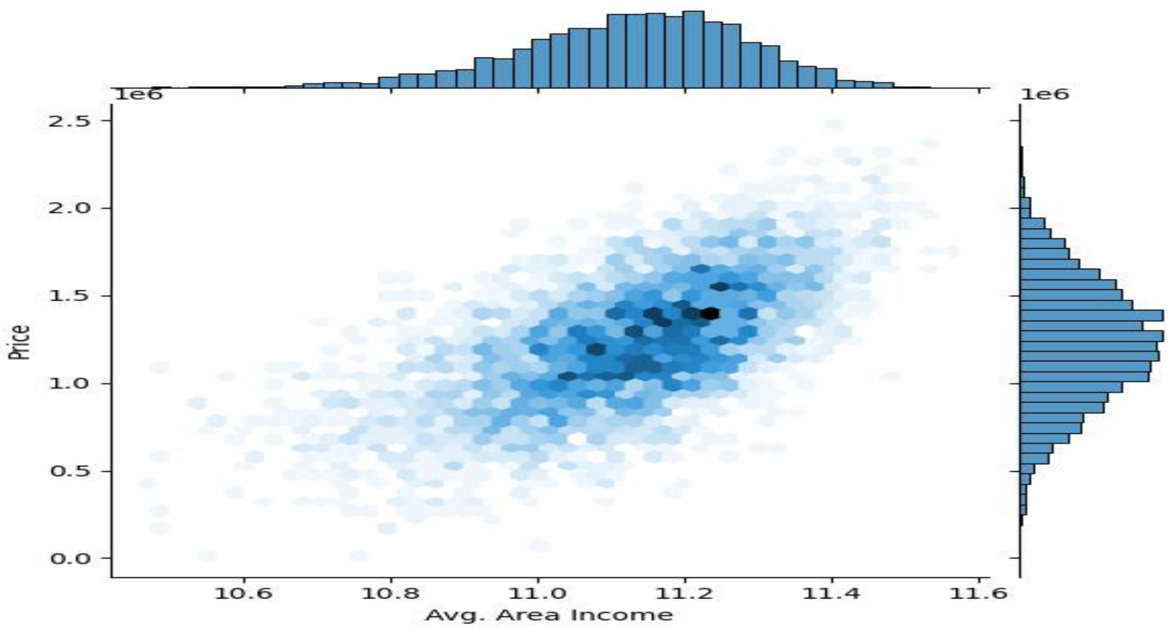
```
sns.jointplot(train_data,x="Area Population",y="Price",kind="hex"
```




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



```
from sklearn.linear_model import LinearRegression model = LinearRegression() model.fit(X_train, y_train)
sns.jointplot(train_data,x="Avg. Area Income",y="Price",kind="hex")
```



Visualizing Correlation:

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

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Bedrooms	Area Population	Price	1.4436652301430477	1.5994329669564435	1.5995840528111915	1.6150085109050225
Avg. Area Income	1.000000	-0.001322	0.010901	-0.011394	0.629439	0.009009	-0.001313	-0.011963	-0.023539
Avg. Area House Age	-0.001322	1.000000	0.009897	-0.021655	0.448345	0.016017	0.021656	-0.022727	0.006592
Avg. Area Number of Bedrooms	0.010901	0.009897	1.000000	-0.004996	0.171870	-0.005666	-0.024031	0.002921	-0.009976
Area Population	-0.011394	-0.021655	-0.004996	1.000000	0.398182	0.018637	-0.001690	-0.035974	-0.032664
Price	0.629439	0.448345	0.171870	0.398182	1.000000	0.006031	-0.004607	-0.051420	-0.043422
...	...	...	...	...	...	...	...	...	...
2.390735135426438	-0.007145	0.033484	0.014163	0.010021	0.029955	-0.000250	-0.000250	-0.000250	-0.000250
2.3911586972470578	0.005678	-0.014999	0.025847	-0.004252	0.007594	-0.000250	-0.000250	-0.000250	-0.000250
2.4109898810001003	0.000874	0.033592	-0.009822	-0.026947	0.026063	-0.000250	-0.000250	-0.000250	-0.000250
2.417689164002378	0.015218	0.024482	-0.005523	-0.012076	0.036884	-0.000250	-0.000250	-0.000250	-0.000250

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) is essential for informed decision-making.
- Data preprocessing emerged as a pivotal aspect of this process. It involves cleaning, transforming, and refining the dataset to ensure that 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