Write a python program to predict home prices using Linear Regression method.

Predicting Housing Prices for regions in the USA.

The data contains the following columns:

'Avg. Area Income': Avg. Income of residents of the city house is located in.

'Avg. Area House Age': Avg Age of Houses in same city

'Avg. Area Number of Rooms': Avg Number of Rooms for Houses in same city

'Avg. Area Number of Bedrooms': Avg Number of Bedrooms for Houses in same city

'Area Population': Population of city house is located in

'Price': Price that the house sold at

'Address': Address for the house

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
housing=pd.read_csv('Housing_USA.csv')
housing
```

```
      Avg. Area Income  Avg. Area House Age  Avg. Area Number of Rooms
\
0          79545.45857             5.682861                   7.009188

1          79248.64245             6.002900                   6.730821

2          61287.06718             5.865890                   8.512727

3          63345.24005             7.188236                   5.586729

4          59982.19723             5.040555                   7.839388

...                ...                  ...                        ...

4995       60567.94414             7.830362                   6.137356

4996       78491.27543             6.999135                   6.576763

4997       63390.68689             7.250591                   4.805081

4998       68001.33124             5.534388                   7.130144

4999       65510.58180             5.992305                   6.792336
```

```
      Avg. Area Number of Bedrooms  Area Population         Price  \
0                             4.09      23086.80050  1.059034e+06
1                             3.09      40173.07217  1.505891e+06
2                             5.13      36882.15940  1.058988e+06
3                             3.26      34310.24283  1.260617e+06
4                             4.23      26354.10947  6.309435e+05
...                            ...              ...           ...
4995                          3.46      22837.36103  1.060194e+06
4996                          4.02      25616.11549  1.482618e+06
4997                          2.13      33266.14549  1.030730e+06
4998                          5.44      42625.62016  1.198657e+06
4999                          4.07      46501.28380  1.298950e+06

                                                Address
0     208 Michael Ferry Apt. 674\nLaurabury, NE 3701...
1     188 Johnson Views Suite 079\nLake Kathleen, CA...
2     9127 Elizabeth Stravenue\nDanieltown, WI 06482...
3                           USS Barnett\nFPO AP 44820
4                          USNS Raymond\nFPO AE 09386
...                                               ...
4995                 USNS Williams\nFPO AP 30153-7653
4996              PSC 9258, Box 8489\nAPO AA 42991-3352
4997  4215 Tracy Garden Suite 076\nJoshualand, VA 01...
4998                          USS Wallace\nFPO AE 73316
4999  37778 George Ridges Apt. 509\nEast Holly, NV 2...

[5000 rows x 7 columns]
```

```
housing.head()
```

```
   Avg. Area Income  Avg. Area House Age  Avg. Area Number of Rooms  \
0       79545.45857             5.682861                   7.009188
1       79248.64245             6.002900                   6.730821
2       61287.06718             5.865890                   8.512727
3       63345.24005             7.188236                   5.586729
4       59982.19723             5.040555                   7.839388

   Avg. Area Number of Bedrooms  Area Population         Price  \
0                          4.09      23086.80050  1.059034e+06
1                          3.09      40173.07217  1.505891e+06
2                          5.13      36882.15940  1.058988e+06
3                          3.26      34310.24283  1.260617e+06
4                          4.23      26354.10947  6.309435e+05

                                             Address
0  208 Michael Ferry Apt. 674\nLaurabury, NE 3701...
1  188 Johnson Views Suite 079\nLake Kathleen, CA...
2  9127 Elizabeth Stravenue\nDanieltown, WI 06482...
```

```
3                          USS Barnett\nFPO AP 44820
4                        USNS Raymond\nFPO AE 09386

#checking columns and total records
housing.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
```

Generating descriptive statistics that summarize the central tendency, dispersion and shape of a dataset's distribution, excluding NaN value.

```
housing.describe()

       Avg. Area Income  Avg. Area House Age  Avg. Area Number of
Rooms  \
count       5000.000000          5000.000000
5000.000000
mean       68583.108984             5.977222
6.987792
std        10657.991214             0.991456
1.005833
min        17796.631190             2.644304
3.236194
25%        61480.562390             5.322283
6.299250
50%        68804.286405             5.970429
7.002902
75%        75783.338665             6.650808
7.665871
max       107701.748400             9.519088
10.759588

       Avg. Area Number of Bedrooms  Area Population         Price
count                   5000.000000      5000.000000  5.000000e+03
mean                       3.981330     36163.516039  1.232073e+06
std                        1.234137      9925.650114  3.531176e+05
min                        2.000000       172.610686  1.593866e+04
```

```
25%                          3.140000     29403.928700  9.975771e+05
50%                          4.050000     36199.406690  1.232669e+06
75%                          4.490000     42861.290770  1.471210e+06
max                          6.500000     69621.713380  2.469066e+06
```

```
housing.columns
```

```
Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of
Rooms',
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price',
'Address'],
      dtype='object')
```

Exploratory Data Analysis

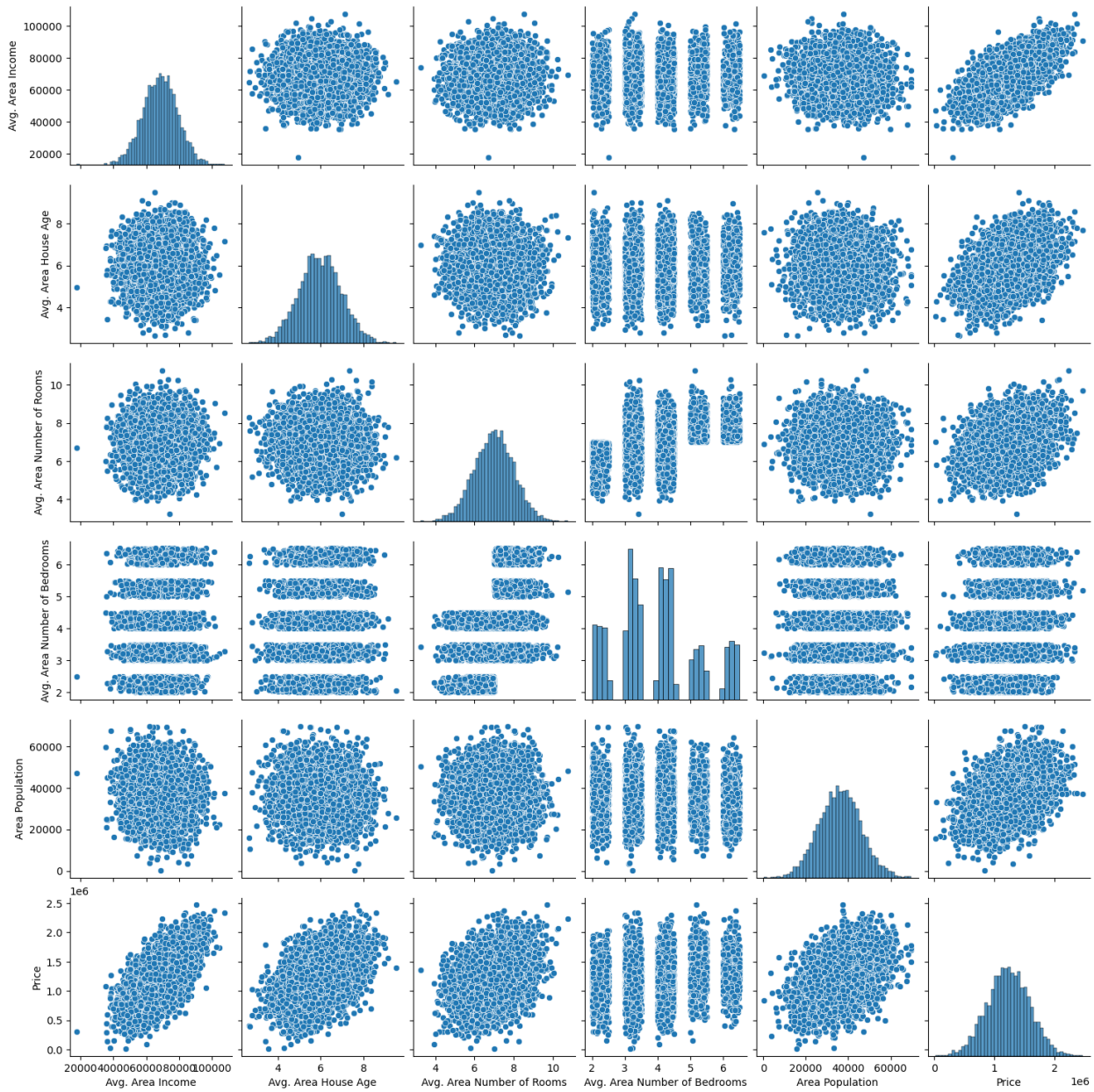
Visualizing the Entire Data using pairplot

Exploring types of relationship across the entire datase

Pairplot in Seaborn is a data visualization tool that creates a matrix of scatterplots, showing pairwise relationships between variables in a dataset, aiding in visualizing correlations and distributions.

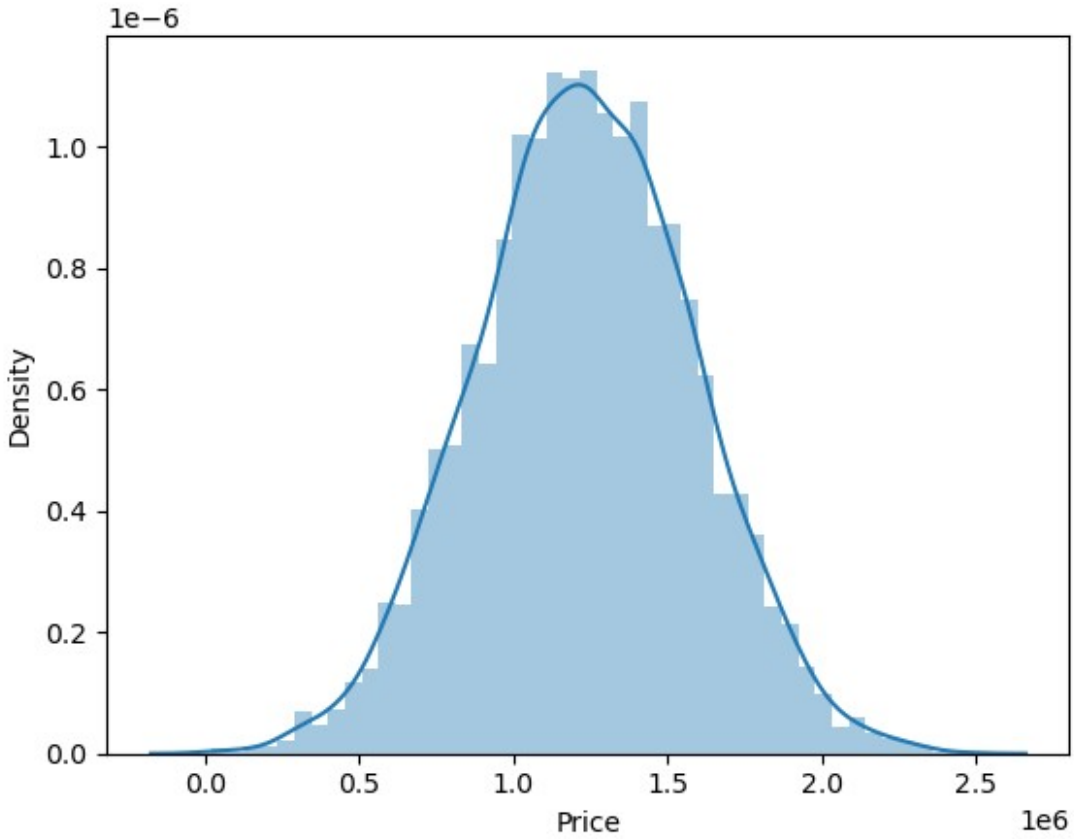
```
sns.pairplot(housing)
```

```
<seaborn.axisgrid.PairGrid at 0x26f601fdac0>
```



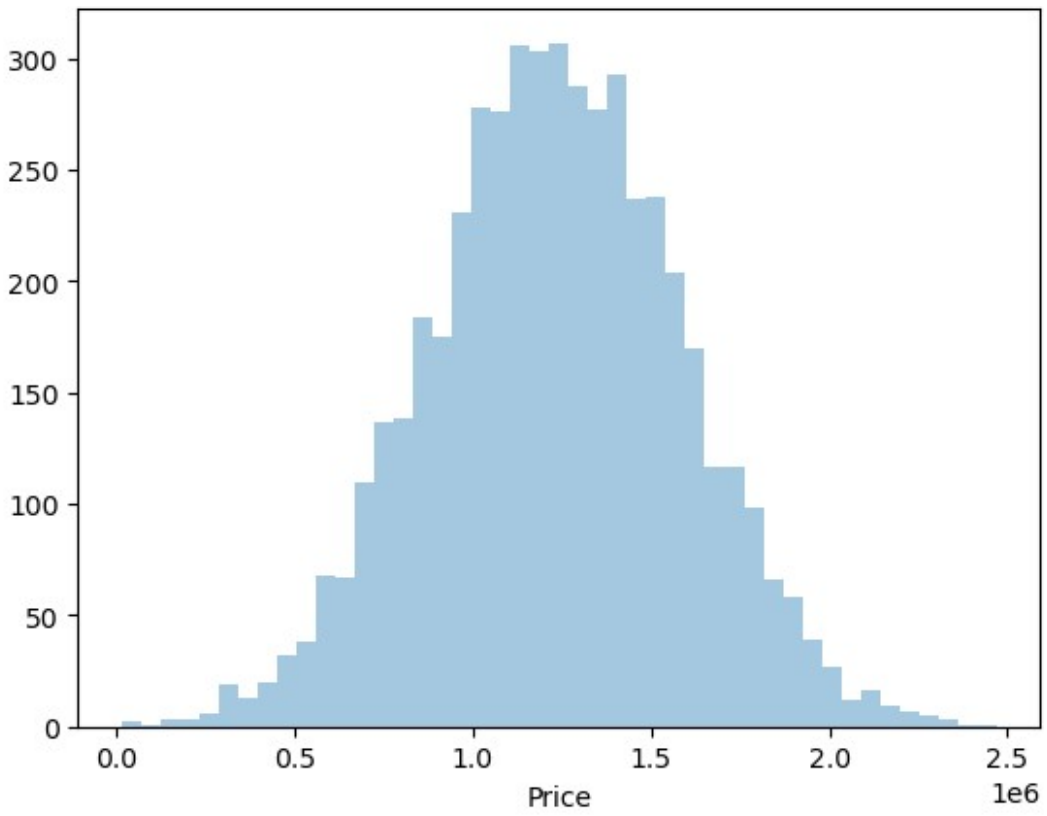

```
sns.distplot(housing.Price)
plt.show()
```

```
C:\Users\VISVINVIN\anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



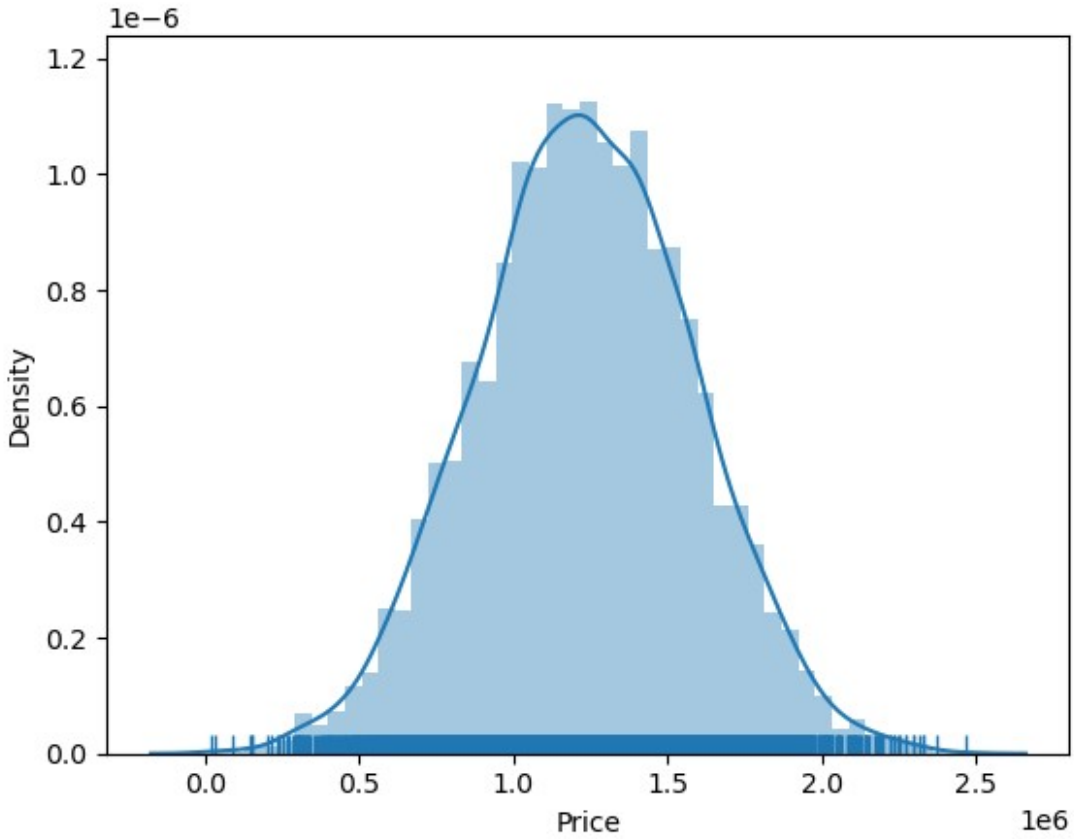

```
sns.distplot(housing.Price,kde=False)
plt.show()
```

```
C:\Users\VISVINVIN\anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



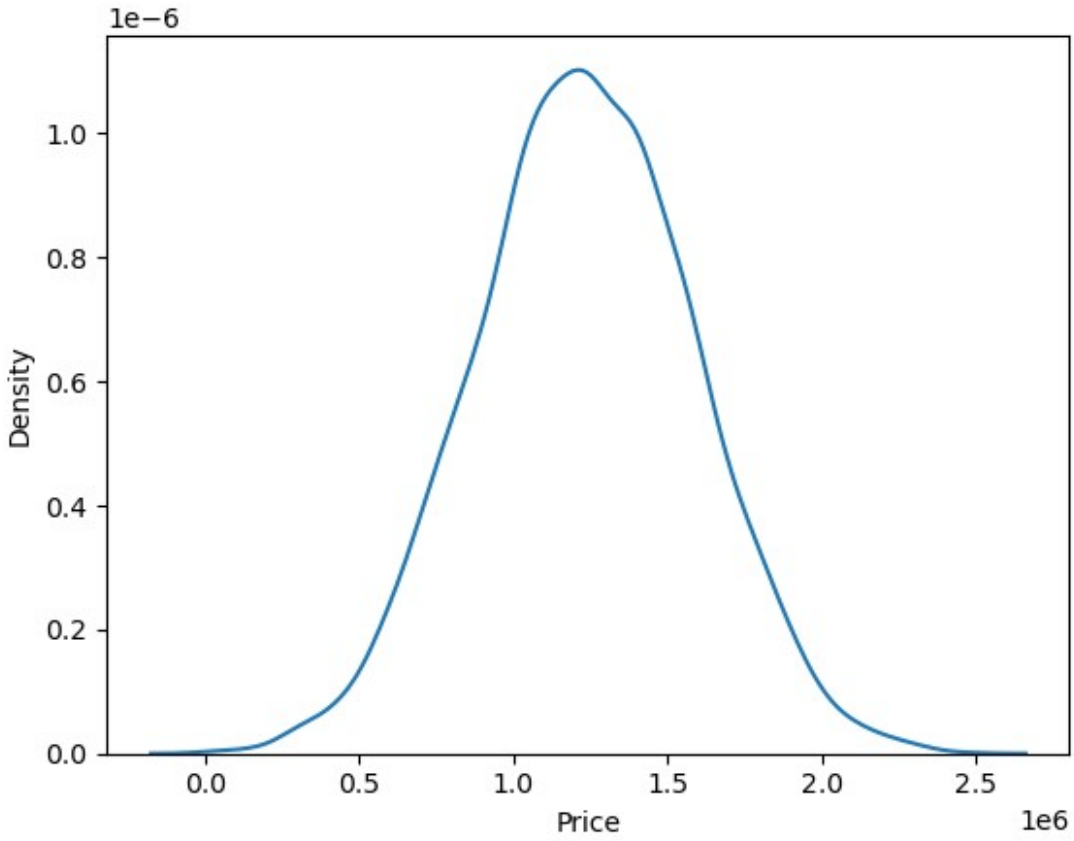

```
sns.distplot(housing.Price,rug=True)
plt.show()
```

```
C:\Users\VISVINVIN\anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\VISVINVIN\anaconda3\lib\site-packages\seaborn\
distributions.py:2103: FutureWarning: The `axis` variable is no longer
used and will be removed. Instead, assign variables directly to `x` or
`y`.
  warnings.warn(msg, FutureWarning)
```

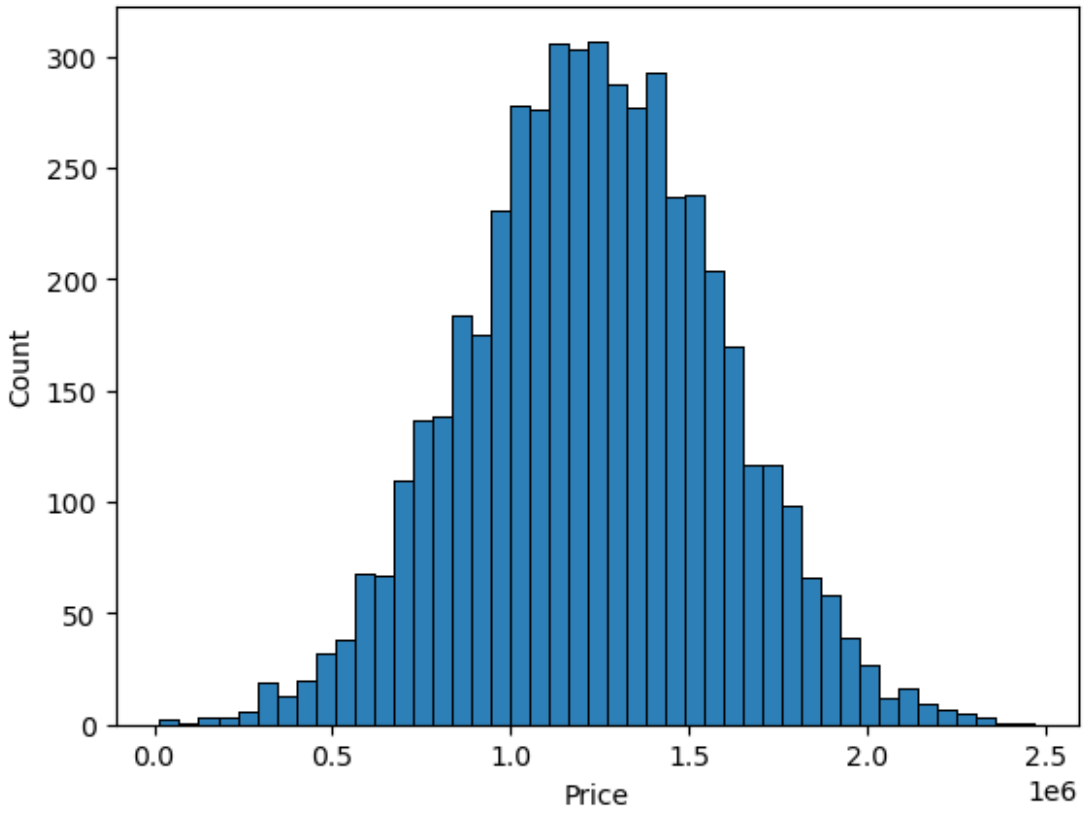



```
sns.distplot(housing.Price,hist=False)
plt.show()
```

```
C:\Users\VISVINVIN\anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `kdeplot` (an axes-level function for kernel density
plots).
  warnings.warn(msg, FutureWarning)
```




```
sns.histplot(housing.Price)
plt.show()
```



Displaying correlation among all the columns : Correlation matrix

```
housing.corr()
```

```
                             Avg. Area Income  Avg. Area House Age  \
Avg. Area Income                     1.000000            -0.002007
Avg. Area House Age                 -0.002007             1.000000
Avg. Area Number of Rooms           -0.011032            -0.009428
Avg. Area Number of Bedrooms         0.019788             0.006149
Area Population                     -0.016234            -0.018743
Price                                0.639734             0.452543

                             Avg. Area Number of Rooms  \
Avg. Area Income                             -0.011032
Avg. Area House Age                          -0.009428
Avg. Area Number of Rooms                     1.000000
Avg. Area Number of Bedrooms                  0.462695
Area Population                               0.002040
Price                                         0.335664
```

```
                              Avg. Area Number of Bedrooms  Area
Population  \
Avg. Area Income                                   0.019788        -
0.016234
Avg. Area House Age                                0.006149        -
0.018743
Avg. Area Number of Rooms                          0.462695
0.002040
Avg. Area Number of Bedrooms                       1.000000        -
0.022168
Area Population                                   -0.022168
1.000000
Price                                              0.171071
0.408556

                                 Price
Avg. Area Income              0.639734
Avg. Area House Age           0.452543
Avg. Area Number of Rooms     0.335664
Avg. Area Number of Bedrooms  0.171071
Area Population               0.408556
Price                         1.000000
```

Displaying correlation among all the columns using Heat Map

```
sns.heatmap(housing.corr(), annot = True)
plt.show()
```


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
Avg. Area Income
Avg. Area House Age
Avg. Area Number of Rooms
Avg. Area Number of Bedrooms
Area Population
Price
1.0
0.8
0.6
0.4
0.2
0.0
1.0
0.8
0.6
0.4
0.2
0.0

Linear regression is also a type of machine-learning algorithm more specifically a supervised machine-learning algorithm that learns from the labelled datasets and maps the data points to the most optimized linear functions, which can be used for prediction on new datasets.

Regression: It predicts the continuous output variables based on the independent input variable.

like the prediction of house prices based on different parameters like house age,

distance from the main road, location, area, etc.

Training a Linear Regression Model

We will need to first split up our data into an X array that contains the features to train on, and a y array with the target variable, in this case the Price column. We will remove out the Address column because it only has text info that the linear regression model can't use.

```
# Columns as Features
X = housing[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area
Number of Rooms',
       'Avg. Area Number of Bedrooms', 'Area Population']]

# Price is Target Variable, what we trying to predict
y = housing['Price']
```

Training the Model: split the data into training and testing sets.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.4, random_state=101)

#importing the Linear Regression Algorithm
from sklearn.linear_model import LinearRegression

#creating LinearRegression Object
lm = LinearRegression()

#Training the Data Model
lm.fit(X_train, y_train)
```

```
LinearRegression()
```

Model Evaluation

Evaluate the model by checking out it's coefficients

```
#Displaying the Intercept
print(lm.intercept_)
```

```
-2640159.7968132393
```

```
coeff_df = pd.DataFrame(lm.coef_, X.columns, columns=['Coefficient'])
coeff_df
```

```
                                Coefficient
Avg. Area Income                  21.528276
Avg. Area House Age           164883.282027
Avg. Area Number of Rooms     122368.678023
Avg. Area Number of Bedrooms    2233.801864
Area Population                   15.150420
```

Interpreting the coefficients:

Holding all other features fixed, a 1 unit increase in Avg. Area Income is associated with an increase of dollar 21.528

Holding all other features fixed,a 1 unit increase in Avg. Area House Age is associated with an increase of dollar 164883.28

Holding all other features fixed, a 1 unit increase in Avg. Area Number of Rooms is associated with an increase of dollar 122368.67

Holding all other features fixed, a 1 unit increase in Avg. Area Number of Bedrooms is associated with an increase of Dollar 2233.80 .

Holding all other features fixed, a 1 unit increase in Area Population is associated with an increase of dollar 15.15 .

Predictions from the Model

Perform predictions off our test set and analyse how well it did!

```
predictions = lm.predict(X_test)
```

```
predictions
```

```
array([1260960.70581766,  827588.7554465 , 1742421.24257428, ...,
        372191.4061303 , 1365217.15136993, 1914519.54191725])
```

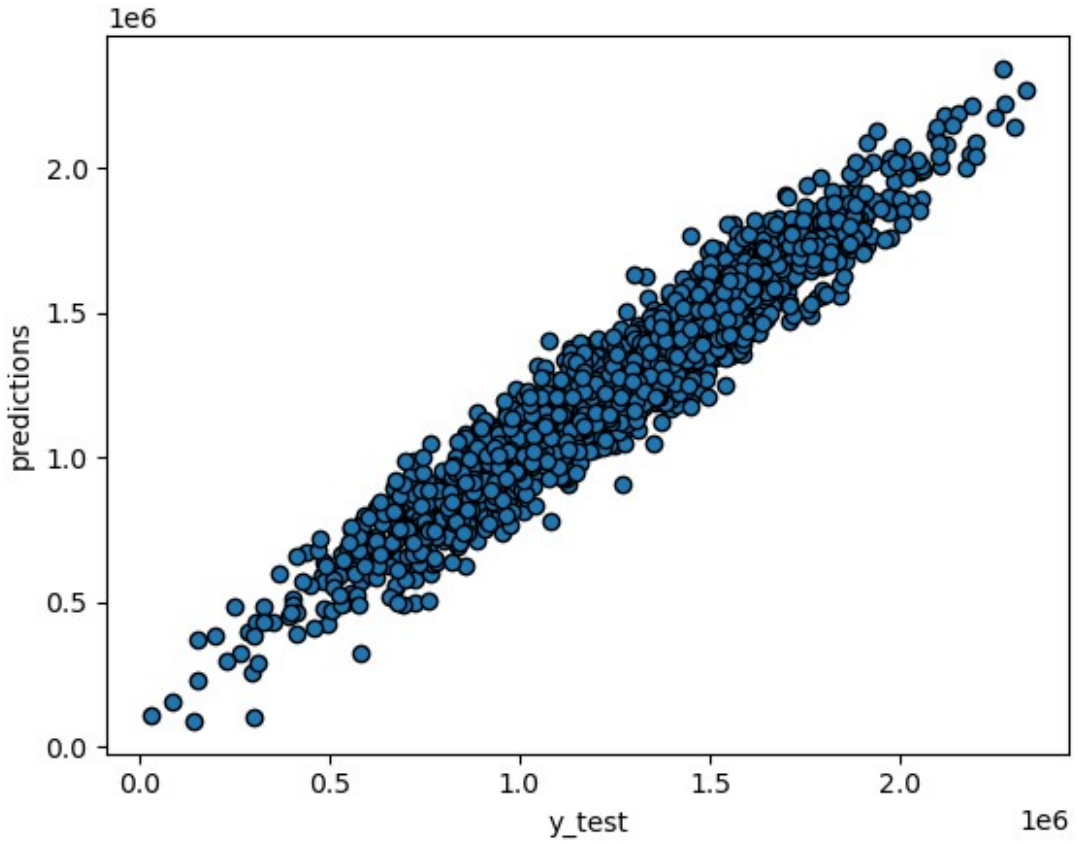
```
y_test
```

```
1718    1.251689e+06
2511    8.730483e+05
345     1.696978e+06
2521    1.063964e+06
54      9.487883e+05
           ...
1776    1.489520e+06
4269    7.777336e+05
1661    1.515271e+05
2410    1.343824e+06
2302    1.906025e+06
Name: Price, Length: 2000, dtype: float64
```

```
error=y_test-predictions
print(error)
```

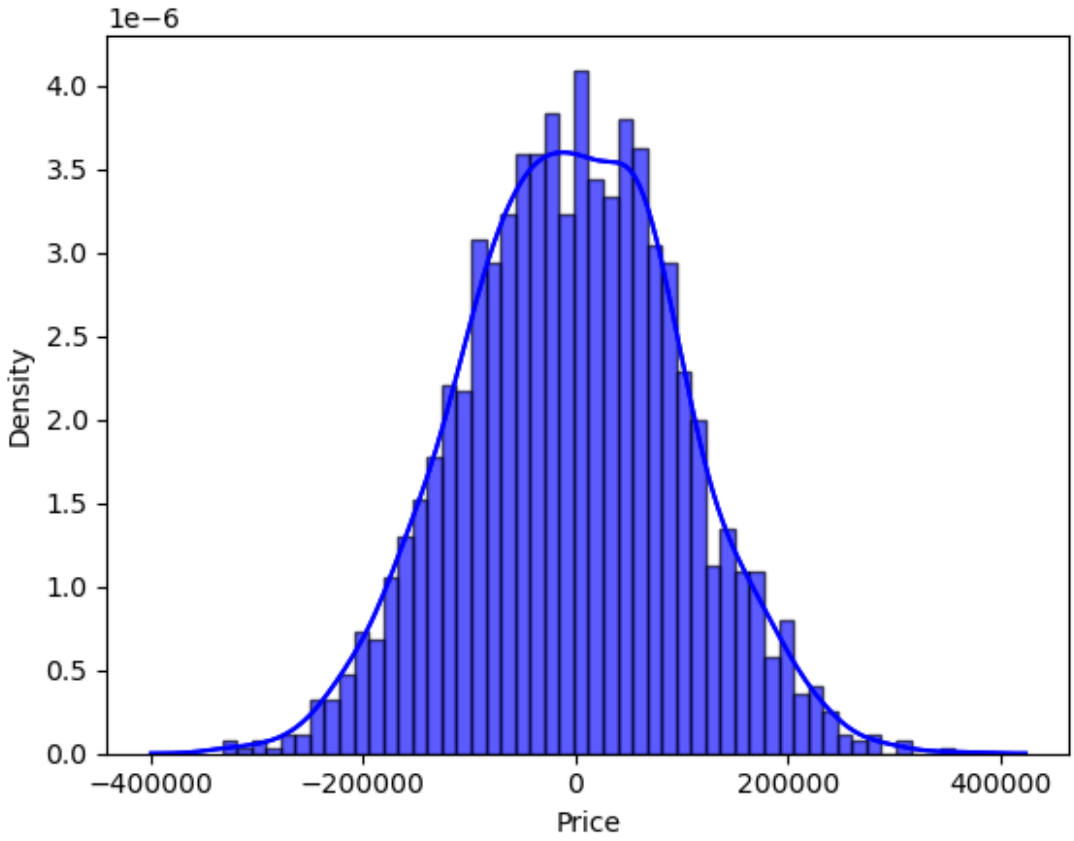
```
1718     -9272.089818
2511     45459.564154
345     -45443.579574
2521     89338.900507
54      -49929.566321
           ...
1776    -25522.673078
4269     31721.824514
1661   -220664.323530
2410    -21392.936370
2302     -8494.905917
Name: Price, Length: 2000, dtype: float64
```

```
plt.scatter(y_test, predictions, edgecolor='black')
plt.xlabel("y_test")
plt.ylabel("predictions")
plt.show()
```




```
#Residual Histogram
#Ploting a histogram of the residuals and making sure it looks
normally distributed.
sns.distplot((y_test - predictions), bins = 50,
hist_kws=dict(edgecolor="black", linewidth=1),color='Blue')
plt.show()
```

```
C:\Users\VISVINVIN\anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



Calculating the Mean Absolute Error, Mean Squared Error, and the Root Mean Squared Error

MAE is the easiest to understand, because it's the average error.

MSE is more popular than MAE, because MSE "punishes" larger errors, which tends to be useful in the real world.

RMSE is even more popular than MSE, because RMSE is interpretable in the "y" units.

All of these are loss functions, which should be minimized

```
from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test,
predictions)))
```

```
MAE: 82288.22250721784
MSE: 10460958905.77505
RMSE: 102278.82921589907
```

Result

A python program to predict homeprices using linaer regression model was developed and executed successfully