In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:

```
%matplotlib inline
```

In [4]:

```
df=pd.read_csv("USA_Housing.csv")
```

In [5]:

```
df.head()
```

Out[5]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Addr
0	79545.458574	5.682861	7.009188	4.09	23086.800503	1.059034e+06	208 Michael Ferry 674\nLaurabury, 37(
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnson Vi Suite 079\nL Kathleen, C
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Elizal Stravenue\nDanielto WI 064
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett\nFPO 44
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymond\nF AE 09

In [6]:

```
df.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
```

In [7]:

```
df.describe()
```

Out[7]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5.000000e+03
mean	68583.108984	5.977222	6.987792	3.981330	36163.516039	1.232073e+06
std	10657.991214	0.991456	1.005833	1.234137	9925.650114	3.531176e+05
min	17796.631190	2.644304	3.236194	2.000000	172.610686	1.593866e+04
25%	61480.562388	5.322283	6.299250	3.140000	29403.928702	9.975771e+05
50%	68804.286404	5.970429	7.002902	4.050000	36199.406689	1.232669e+06
75%	75783.338666	6.650808	7.665871	4.490000	42861.290769	1.471210e+06
max	107701.748378	9.519088	10.759588	6.500000	69621.713378	2.469066e+06

In [8]:

```
df.columns
```

Out[8]:

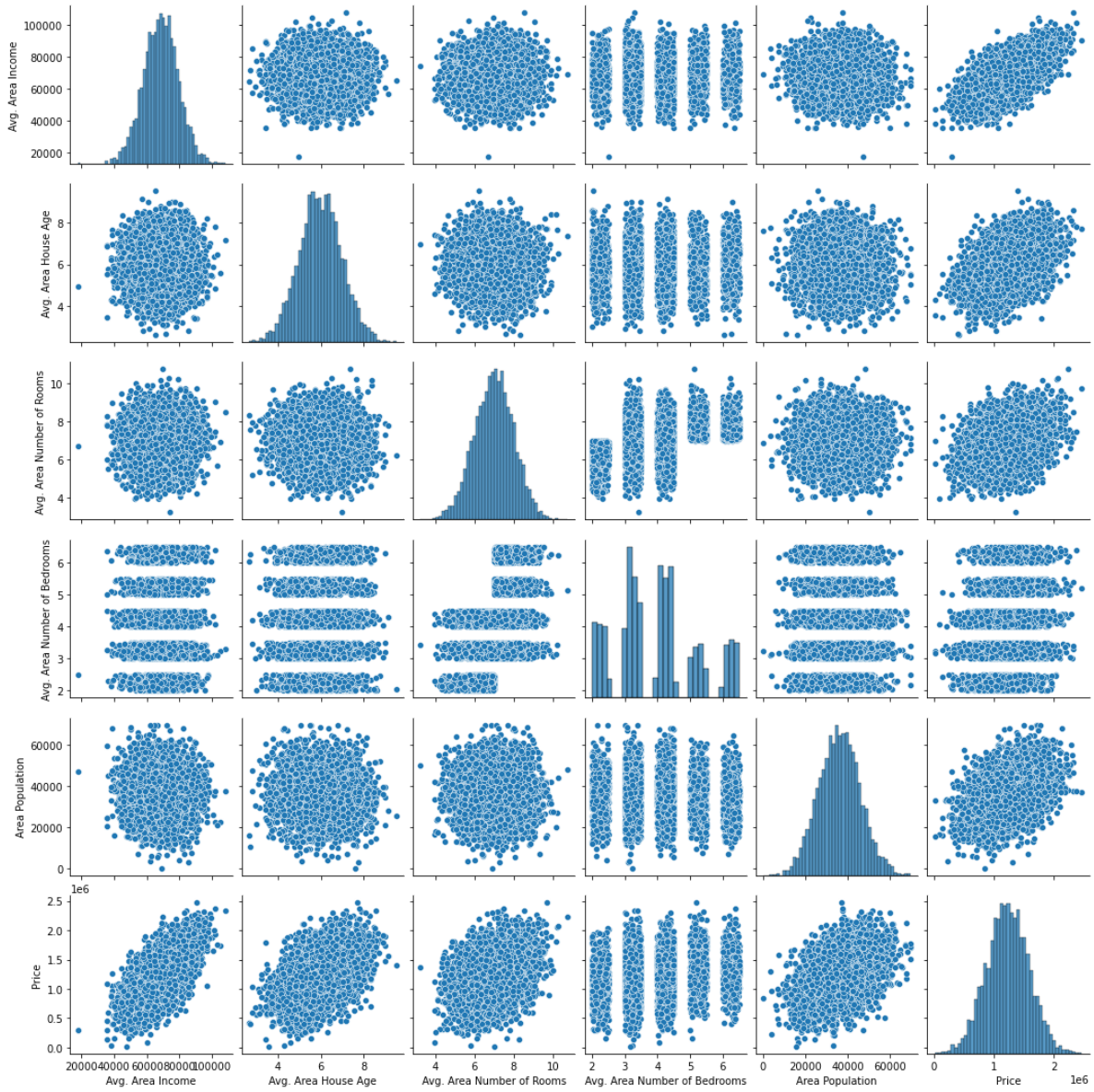
```
Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Room
s',
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price', 'Addres
s'],
      dtype='object')
```

In [9]:

```
sns.pairplot(df)
```

Out[9]:

```
<seaborn.axisgrid.PairGrid at 0x1d52f87a520>
```



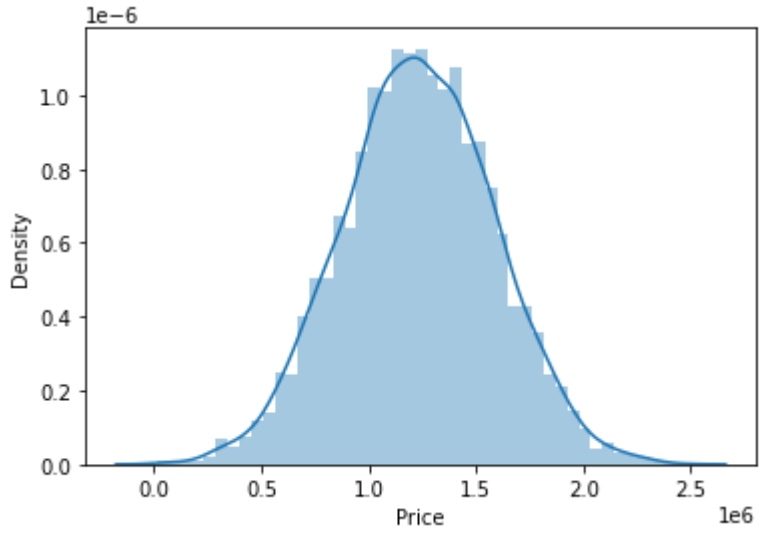
In [10]:

```
sns.distplot(df["Price"])
```

```
C:\Users\Krishna Reddy\anaconda3\lib\site-packages\seaborn\distributions.py:
2551: FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a figur
e-level function with similar flexibility) or `histplot` (an axes-level func
tion for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[10]:

```
<AxesSubplot:xlabel='Price', ylabel='Density'>
```



In [11]:

```
df.corr()
```

Out[11]:

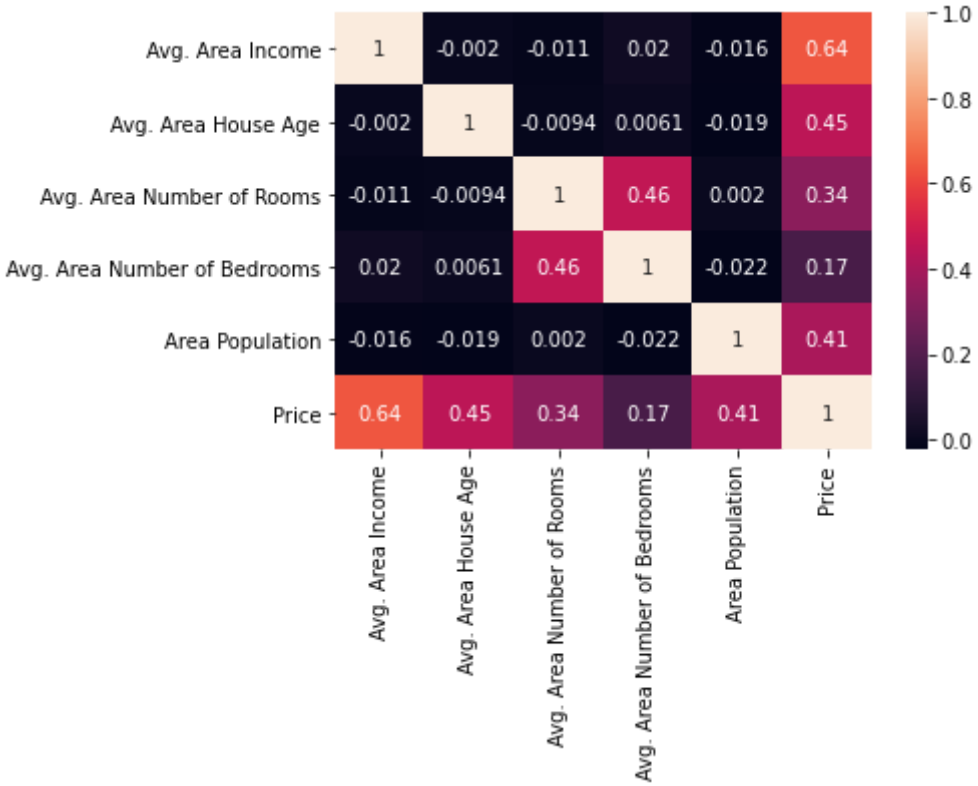
	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
Avg. Area Income	1.000000	-0.002007	-0.011032	0.019788	-0.016234	0.639734
Avg. Area House Age	-0.002007	1.000000	-0.009428	0.006149	-0.018743	0.452543
Avg. Area Number of Rooms	-0.011032	-0.009428	1.000000	0.462695	0.002040	0.335664
Avg. Area Number of Bedrooms	0.019788	0.006149	0.462695	1.000000	-0.022168	0.171071
Area Population	-0.016234	-0.018743	0.002040	-0.022168	1.000000	0.408556
Price	0.639734	0.452543	0.335664	0.171071	0.408556	1.000000

In [12]:

```
sns.heatmap(df.corr(), annot=True)
```

Out[12]:

```
<AxesSubplot:>
```



In [13]:

```
df.columns
```

Out[13]:

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

In [14]:

```
from sklearn.model_selection import train_test_split
```

In [15]:

```
x=df[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
       'Avg. Area Number of Bedrooms', 'Area Population']]
```

In [16]:

```
y=df[['Price']]
```

In [17]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y ,test_size=0.4, random_state=101)
```

In [18]:

```
from sklearn.linear_model import LinearRegression
```

In [19]:

```
lm=LinearRegression()
```

In [20]:

```
lm.fit(x_train, y_train)
```

Out[20]:

```
LinearRegression()
```

In [21]:

```
print(lm.intercept_)
```

```
[-2640159.79685191]
```

In [22]:

```
x.columns
```

Out[22]:

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

In [23]:

```
print(lm.coef_)
```

```
[[2.15282755e+01 1.64883282e+05 1.22368678e+05 2.23380186e+03
  1.51504200e+01]]
```

In [24]:

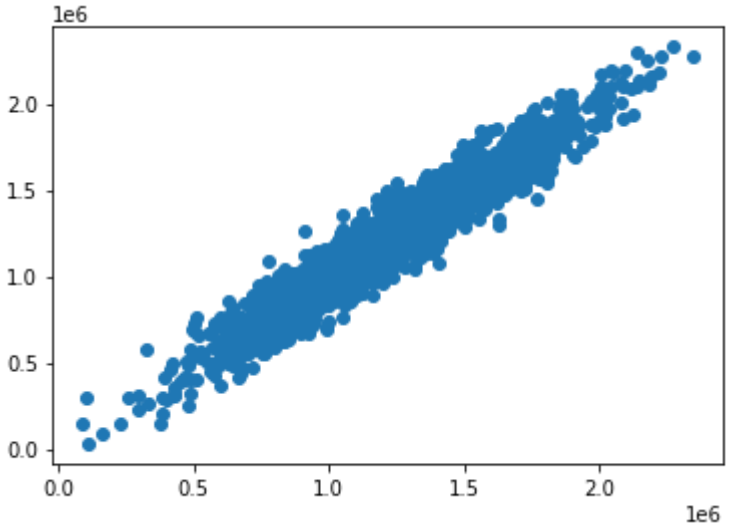
```
prediction=lm.predict(x_test)
```

In [25]:

```
plt.scatter(prediction,y_test)
```

Out[25]:

```
<matplotlib.collections.PathCollection at 0x1d537aabd30>
```



In [ ]:

```

```

In [26]:

```
from sklearn import metrics as mt
```

In [ ]:

```

```

mt.mean_absolute_error(y_test,predictions)

In [39]:

```
mt.mean_absolute_error(y_test,prediction)
```

Out[39]:

```
82288.22251914957
```

In [40]:

```
a=mt.mean_squared_error(y_test,prediction)
a
```

Out[40]:

```
10460958907.209507
```

In [41]:

```
np.sqrt(a)
```

Out[41]:

```
102278.82922291156
```

In [42]:

```
mt.explained_variance_score(y_test,prediction)
```

Out[42]:

```
0.9178179926151797
```

In [ ]:

```

```

In [ ]:

```

```