

Problem statement

Data collection

In [9]:

```
#to import libraries  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns
```

In [4]:

```
df=pd.read_csv(r"c:\Users\user\Downloads\10_USA_Housing.csv")
df
```

Out[4]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	
0	79545.458574	5.682861	7.009188	4.09	23086.800503	1.059034e+06	208 Michael 674\nLaur
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 John: Suite (K Kathl
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Stravenue\nD V
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raym
...	
4995	60567.944140	7.830362	6.137356	3.46	22837.361035	1.060194e+06	USNS Willie AP 30
4996	78491.275435	6.999135	6.576763	4.02	25616.115489	1.482618e+06	PSC 8489\nAPO /
4997	63390.686886	7.250591	4.805081	2.13	33266.145490	1.030730e+06	4215 Tra Suite 076\nJo
4998	68001.331235	5.534388	7.130144	5.44	42625.620156	1.198657e+06	USS Wallace
4999	65510.581804	5.992305	6.792336	4.07	46501.283803	1.298950e+06	37778 Geor Apt. 509\nf

5000 rows × 7 columns



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	Ad
0	79545.458574	5.682861	7.009188	4.09	23086.800503	1.059034e+06	208 Michael Fer 674\nLaurabu
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnson Suite 079\ Kathleen
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Eliz Stravenue\nDanie WI 06
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett\nFI
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymond\ AE

In [7]:

```
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 [8]:

```
#to display summary of statistics
df.describe()
```

Out[8]:

	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 [11]:

```
#to display cloumn heading
df.columns
```

Out[11]:

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

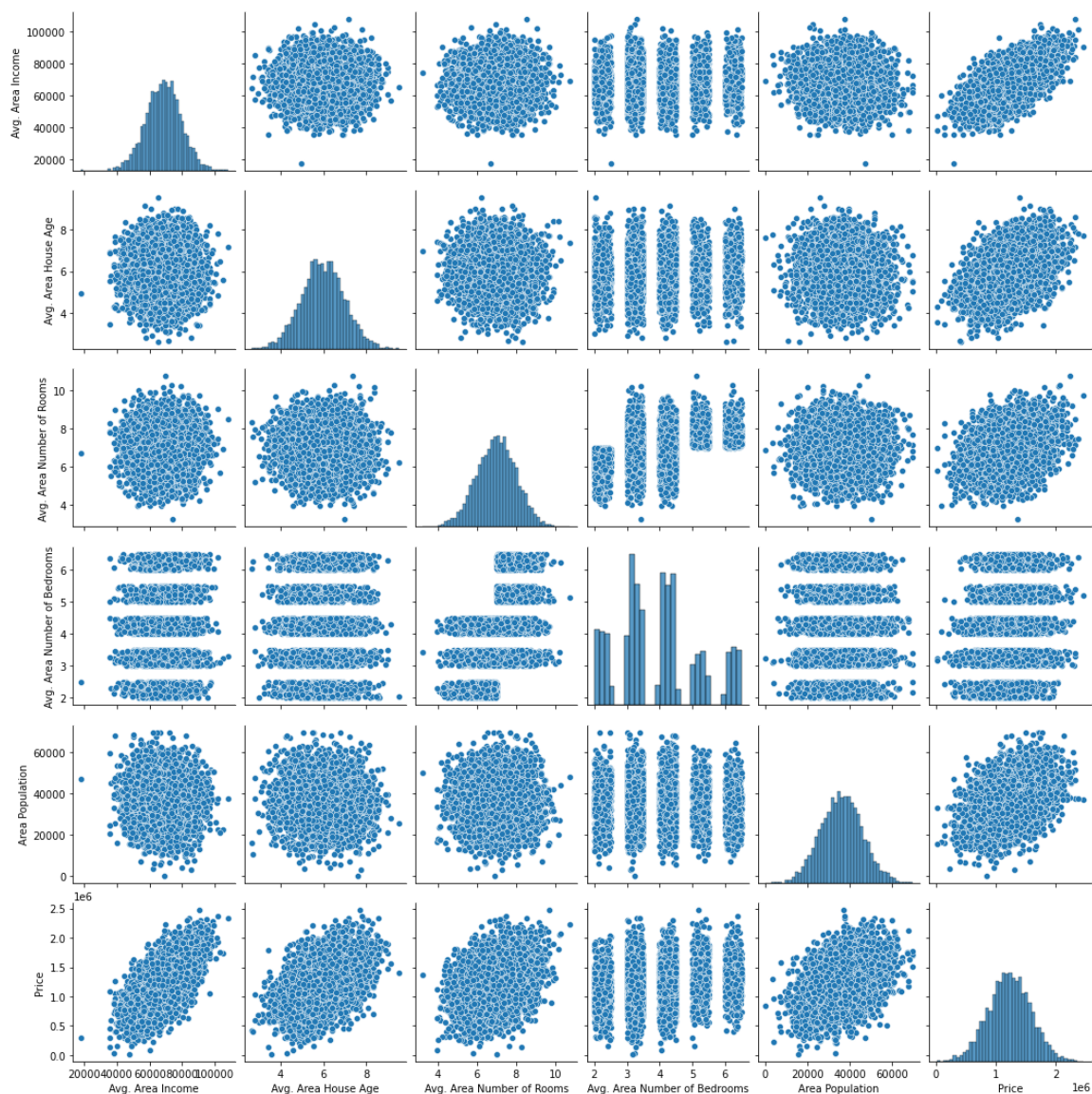
EDA and VISUALIZATION

In [12]:

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

Out[12]:

<seaborn.axisgrid.PairGrid at 0x22ddfa499a0>



In [13]:

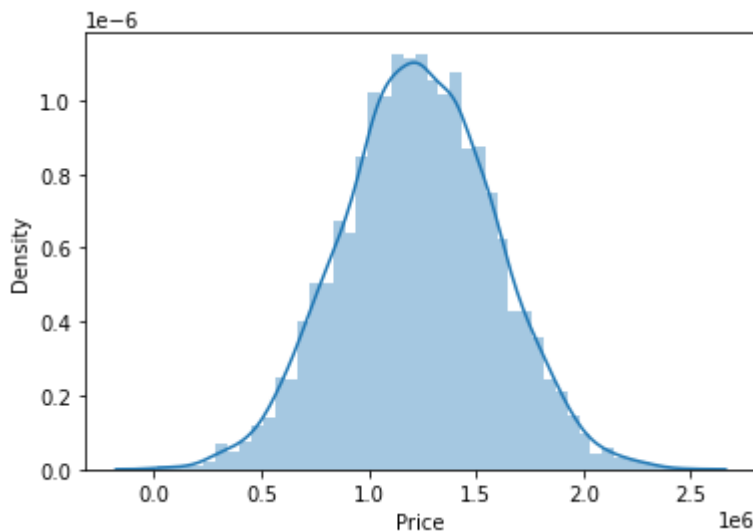
```
sns.distplot(df['Price'])
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: 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)
```

Out[13]:

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

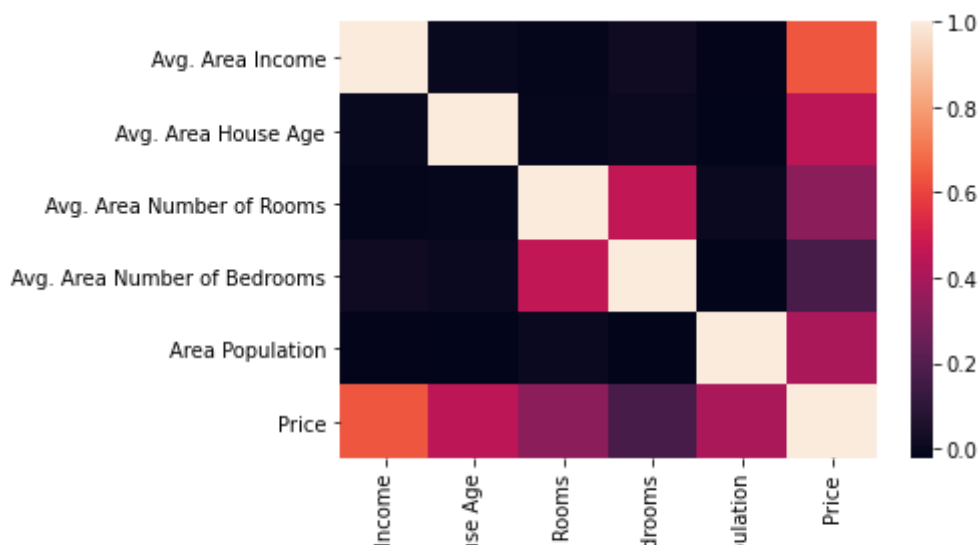


In [15]:

```
df1=df[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms', 'Avg. Area Number of Bedrooms', 'Area Population', 'Price']]
sns.heatmap(df1.corr())
```

Out[15]:

```
<AxesSubplot:>
```



to Train the model-Model buliding

we are going to split our data into two variable where x is a independent and y is dependent on x

In [16]:

```
x=df1[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
      'Avg. Area Number of Bedrooms', 'Area Population']]
y=df1['Price']
```

In [17]:

```
# to split my dataset into test and train data
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3)
```

In [18]:

```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[18]:

LinearRegression()

In [20]:

```
print(lr.intercept_)
```

-2643624.081213882

In [23]:

```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-effecient'])
coeff
```

Out[23]:

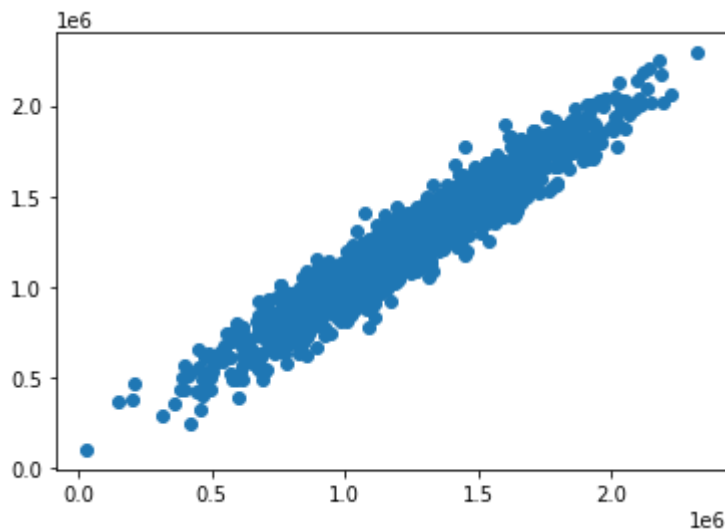
	Co-effecient
Avg. Area Income	21.513845
Avg. Area House Age	165869.031020
Avg. Area Number of Rooms	121582.366345
Avg. Area Number of Bedrooms	1659.652971
Area Population	15.294553

In [25]:

```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[25]:

<matplotlib.collections.PathCollection at 0x22de44d15b0>



In [24]:

```
print(lr.score(x_test,y_test))
```

0.92021134361623

In []: