Problem Statement

In [1]:

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

In [8]:

```
d=pd.read_csv(r"C:\Users\user\Downloads\Housing.csv")
d
```

Out[8]:

	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 (Kathl
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Stravenue\nD W
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 Willia AP 30
4996	78491.275435	6.999135	6.576763	4.02	25616.115489	1.482618e+06	PSC 8489\nAPO A
4997	63390.686886	7.250591	4.805081	2.13	33266.145490	1.030730e+06	4215 Trac 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\nE

5000 rows × 7 columns

In [9]:

```
d.head(10)
```

Out[9]:

	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 3
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\nFF
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymond\ AE
5	80175.754159	4.988408	6.104512	4.04	26748.428425	1.068138e+06	06039 Jennifer I Apt. 443\nTrac
6	64698.463428	6.025336	8.147760	3.41	60828.249085	1.502056e+06	4759 Daniel S 442\nNguyenburg
7	78394.339278	6.989780	6.620478	2.42	36516.358972	1.573937e+06	972 Viaduct\nLake W TN 17778
8	59927.660813	5.362126	6.393121	2.30	29387.396003	7.988695e+05	USS Gilbert\nFF
9	81885.927184	4.423672	8.167688	6.10	40149.965749	1.545155e+06	Unit 944 0958\nDPO AE

In [11]:

```
d.describe()
```

Out[11]:

	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 [12]:

```
d.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 [14]:

```
d.columns
```

Out[14]:

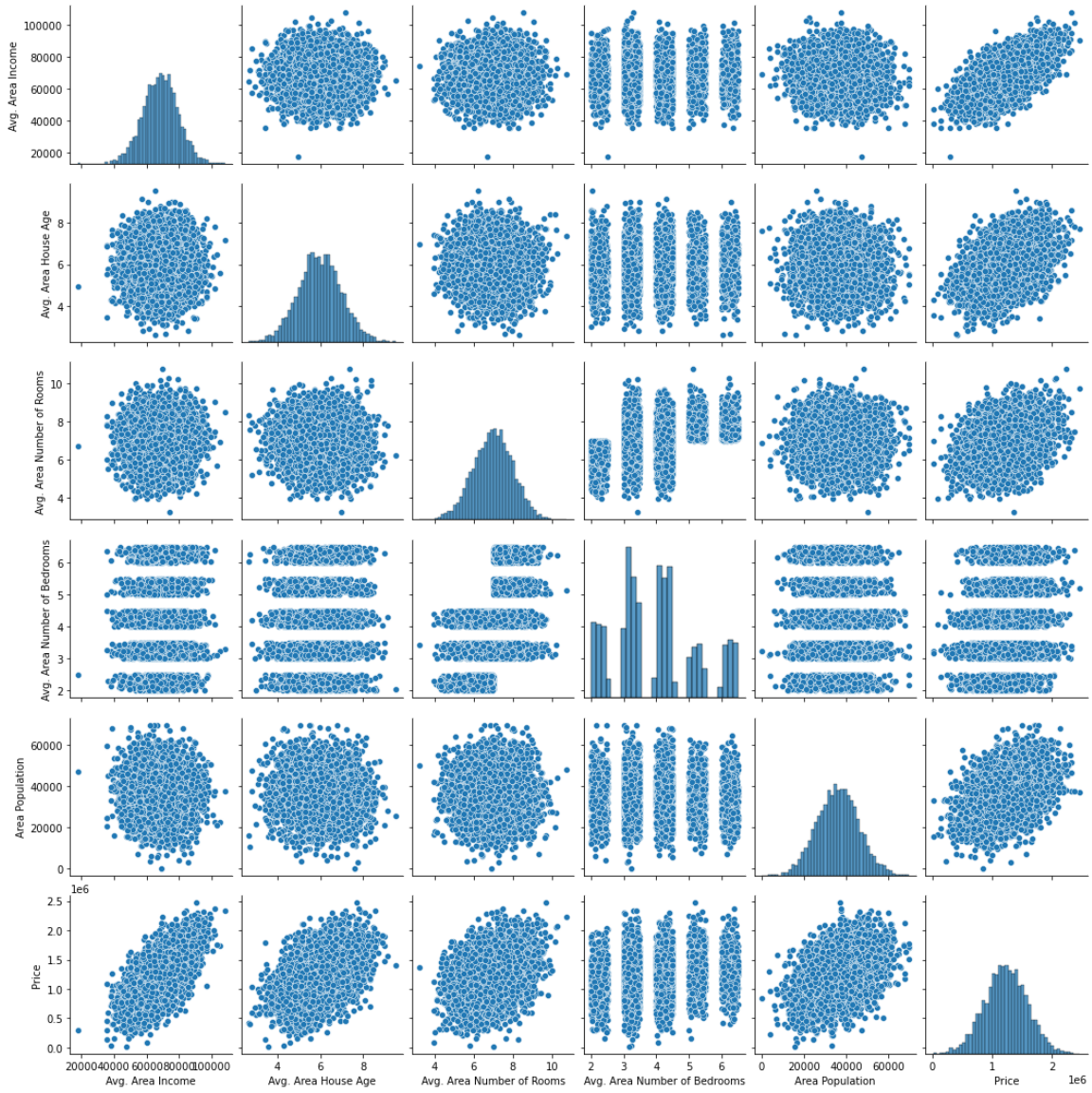
```
Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Roo
ms',
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price', 'Addres
s'],
      dtype='object')
```

In [15]:

```
sns.pairplot(d)
```

Out[15]:

```
<seaborn.axisgrid.PairGrid at 0x1718d8dcbb0>
```



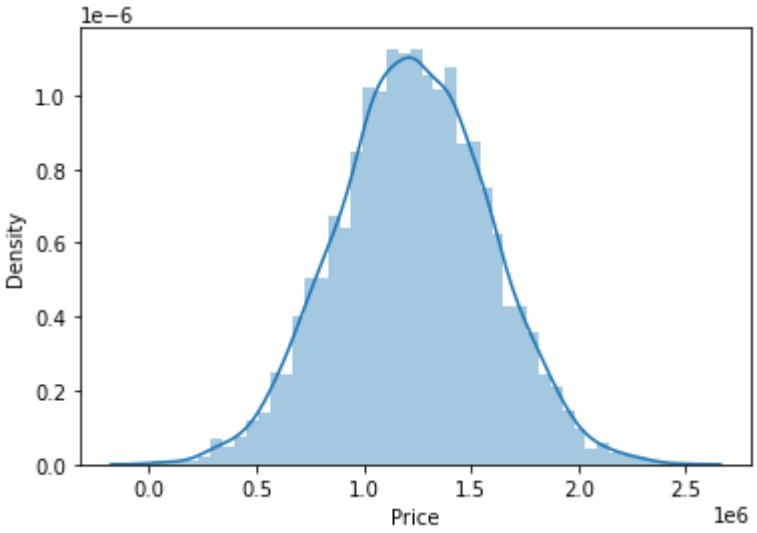
In [18]:

```
sns.distplot(d['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 fun
ction for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[18]:

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



In [19]:

```
da=d[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price', 'Address']]
```

In [20]:

```
# relation
sns.heatmap(da.corr())
```

Out[20]:

<AxesSubplot:>

to train the model

we are going to train linear regresssion model;we need to split out data into two values varible x and y where x is independent(input) and y is depdendent on x (output)we could ignore adrees column as it not requird foer model

In [31]:

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

In [32]:

```
# 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 [33]:

```
from sklearn.linear_model import LinearRegression

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

Out[33]:

```
LinearRegression()
```

In [34]:

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

```
-2632827.7811369104
```

In [35]:

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

Out[35]:

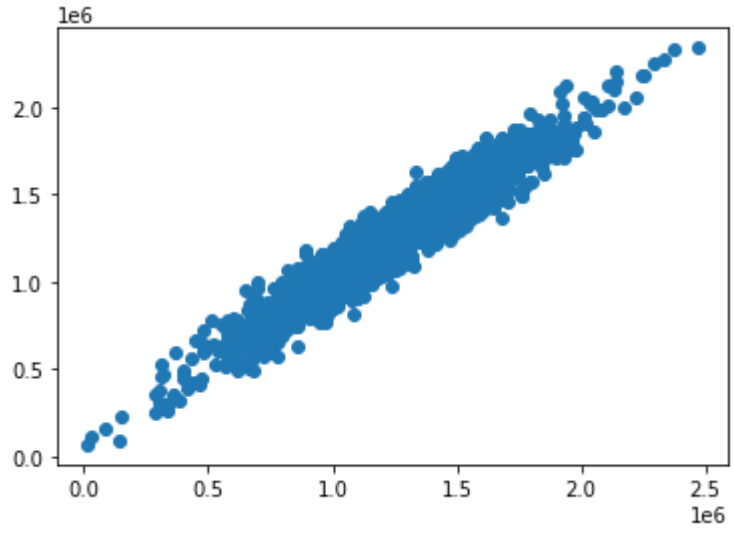
	Co-effecient
Avg. Area Income	21.554500
Avg. Area House Age	163874.290682
Avg. Area Number of Rooms	121897.427287
Avg. Area Number of Bedrooms	1275.355058
Area Population	15.252067

In [39]:

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

Out[39]:

```
<matplotlib.collections.PathCollection at 0x171922e0f10>
```



In [38]:

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

```
0.9194180849973701
```

In [ ]:

```

```