

PROBLEM STATEMENT

A real estate agent want help to predict the house price for regions in USA. He gave us the dataset to work on to use Linear Regression model. Create a model that helps him to estimate of what the house would sell for

DATA COLLECTION

In [1]:

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

In [3]:

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

Out[3]:

	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(5)
```

Out[5]:

	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 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\nFI
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymond\ AE

DATA CLEANING AND PRE-PROCESSING

In [7]:

```
#to find null values  
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 [9]:

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

Out[9]:

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

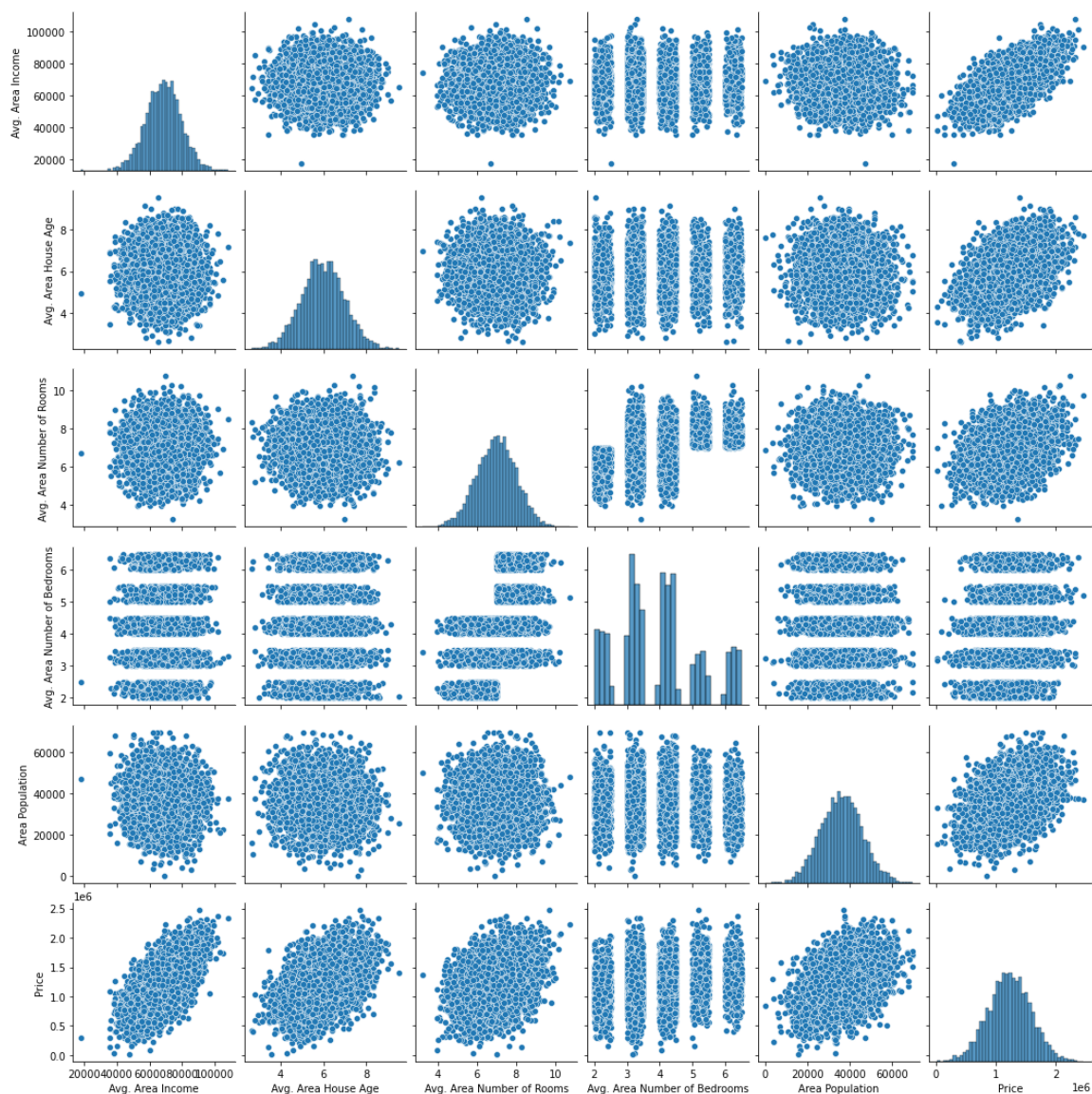
EDA and VISUALIZATION

In [10]:

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

Out[10]:

<seaborn.axisgrid.PairGrid at 0x1edb8531730>



In [31]:

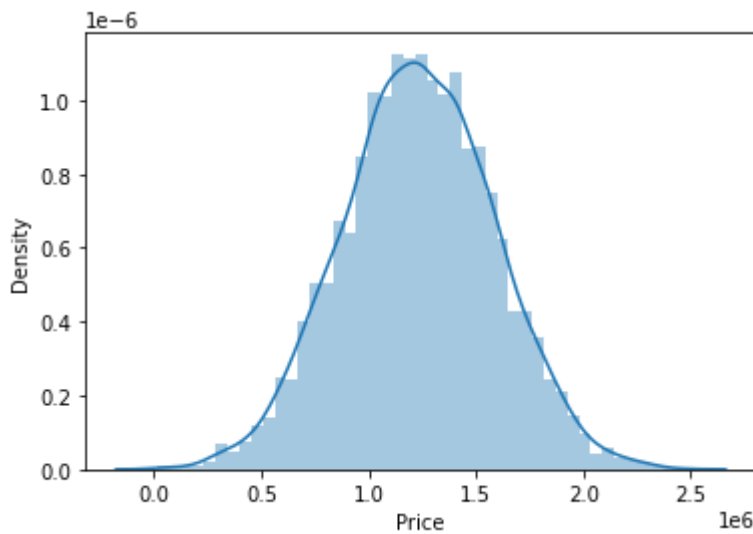
```
#normal distribution  
sns.distplot(df['Price'])
```

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 fun
ction for histograms).

```
warnings.warn(msg, FutureWarning)
```

Out[31]:

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



In [13]:

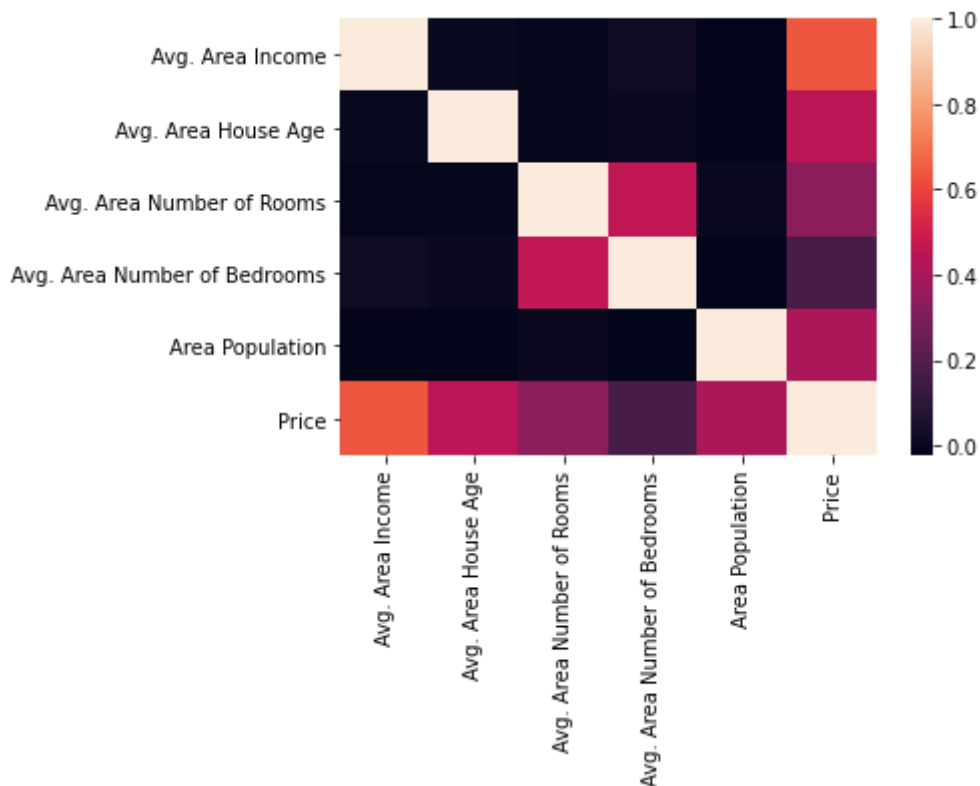
```
df1=df[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',  
        'Avg. Area Number of Bedrooms', 'Area Population', 'Price']]
```

In [15]:

```
#heat map
sns.heatmap(df1.corr())
```

Out[15]:

<AxesSubplot:>



To train the model-model building

We are going to train Linear Regression model; We need to split out data into two variables x and y where x is independent variable (input) and y is dependent on x (output) we could ignore address column as it is not required for our model

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 [20]:

```
# to split my dataset into training and test 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 [22]:

```
from sklearn.linear_model import LinearRegression

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

Out[22]:

LinearRegression()

In [23]:

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

-2627905.3551811427

In [24]:

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

Out[24]:

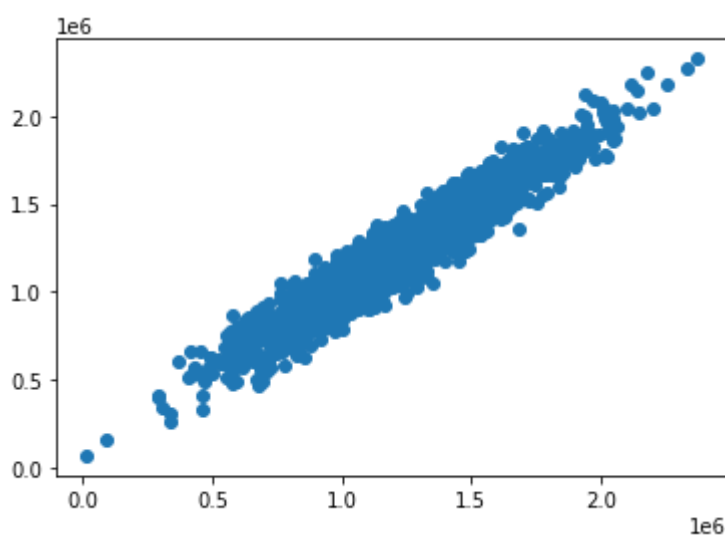
	Co-efficient
Avg. Area Income	21.541046
Avg. Area House Age	164758.849448
Avg. Area Number of Rooms	120924.705303
Avg. Area Number of Bedrooms	967.250894
Area Population	15.175661

In [27]:

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

Out[27]:

<matplotlib.collections.PathCollection at 0x1edbc1491f0>



In [29]:

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

0.9180544706943

In []: