In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
housedf=pd.read_csv("USA_Housing.csv")
```

In [3]:
```
housedf.head()
```

Out[3]:

	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 Ferry Apt. 674\nLaurabury, NE 3701...
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnson Views Suite 079\nLake Kathleen, CA-...
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Elizabeth Stravenue\nDanielton, WI 06482...
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett\nFPO AP 44820
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymond\nFPO AE 09386

In [4]:
```
housedf.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 [5]:
```
housedf.describe()
```

Out[5]:

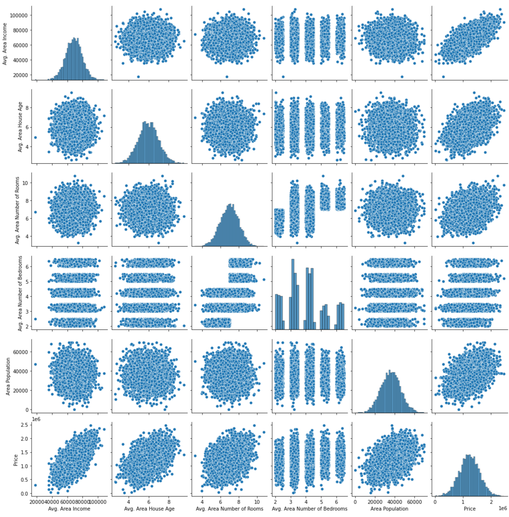
	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5.000000e+03
mean	68583.108984	5.977222	6.987792	3.981330	36163.516039	1.232073e+06
std	10657.991214	0.991456	1.005833	1.234137	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 [6]:
```
housedf.columns
```

Out[6]:
```
Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price', 'Address'],
      dtype='object')
```

In [7]:
```
sns.pairplot(housedf)
```

Out[7]:
```
<seaborn.axisgrid.PairGrid at 0x26f96e15b50>
```



In [8]:
```
sns.heatmap(housedf.corr(), annot=True)
```

Out[8]:
```
<AxesSubplot:>
```

In [9]:
```
X= housedf[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
       'Avg. Area Number of Bedrooms', 'Area Population']]
y=housedf['Price']
```

In [10]:
```
from sklearn.model_selection import train_test_split
```

In [11]:
```
X_train, X_test, y_train, y_test=train_test_split(X, y, train_size=0.40 , random_state=101)
```

In [12]:
```
X_train
```

Out[12]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population
1672	49775.405947	5.305108	6.178535	2.24	23557.361654
4557	75571.044023	6.114928	7.318214	4.44	33988.435859
1040	66250.316685	4.858520	7.758692	3.08	24201.753077
514	56978.222191	6.658346	7.889948	3.01	44071.604628
2148	73053.953218	5.712999	6.673142	4.30	41686.518927
...	...	...	...	...	...
4171	56610.642505	4.846632	7.558137	3.29	25494.740298
599	70596.850945	6.548274	6.539986	3.10	51614.830136
1361	55621.899104	3.735942	6.868291	2.30	63184.613147
1547	63044.460096	5.935261	5.913454	4.10	32725.279544
4959	75078.791516	7.644779	8.440726	4.33	56148.449322

2000 rows × 5 columns

In [13]:
```
y_train
```

Out[13]:
```
1672    4.540557e+05
4557    1.218264e+06
1040    1.094322e+06
514     1.308984e+06
2148    1.220724e+06
            ...     
4171    7.296417e+05
599     1.599479e+06
1361    1.102641e+06
1547    8.650995e+05
4959    2.108376e+06
Name: Price, Length: 2000, dtype: float64
```

In [14]:
```
from sklearn.linear_model import LinearRegression
```

In [11]:
```
lr=LinearRegression()
```

In [12]:
```
lr.fit(X_train, y_train)
```

Out[12]:
```
LinearRegression()
```

In [13]:
```
coeff_df=pd.DataFrame(lr.coef_, X.columns,columns=['coefficient'])
```

In [14]:
```
coeff_df
```

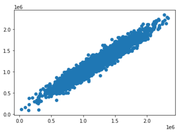
Out[14]:

	coefficient
Avg. Area Income	21.384444
Avg. Area House Age	162975.483408
Avg. Area Number of Rooms	121802.121132
Avg. Area Number of Bedrooms	1934.163056
Area Population	15.189607

In [15]:
```
Prediction= lr.predict(X_test)
```

In [31]:
```
plt.scatter(y_test, Prediction)
```

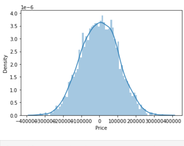
Out[31]:
```
<matplotlib.collections.PathCollection at 0x2635cb097f0>
```



In [38]:
```
sns.distplot((y_test-Prediction),bins=70)
```

```
C:\Users\91810\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)
```

Out[38]:
```
<AxesSubplot:xlabel='Price', ylabel='Density'>
```



In [34]:
```
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(y_test, Prediction)
r_square = r2_score(y_test, Prediction)
```

In [35]:
```
print('Mean_Squared_Error :' ,mse)
print('r_square_value :',r_square)
```

```
Mean_Squared_Error : 10354334181.172644
r_square_value : 0.9175452486480594
```

In [36]:
```
from math import sqrt
```

```
rms = sqrt(mse)
rms
```

Out[36]:
```
101756.24885564839
```

In [ ]: