Date - 23/10/2023

Team ID - 696

Project Title - House Pricing forecasting using ML

Importing Dependencies

In [2]:
```
1 import pandas as pd
2 import numpy as np
```

Loading Dataset

In [3]:
```
1 dataset = pd.read_csv("USA_Housing.csv")
```

Data importing

In [4]:
```
1 dataset
```

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 (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 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\nI

5000 rows × 7 columns

In [ ]:
```
1 import seaborn as sns
2 import matplotlib.pyplot as plt
3 from sklearn.model_selection import train_test_split
4 from sklearn.preprocessing import StandardScaler
5 from sklearn.metrics import r2_score, mean_absolute_error,mean_squared_
6 from sklearn.linear_model import LinearRegression
7 from sklearn.linear_model import Lasso
8 from sklearn.ensemble import RandomForestRegressor
9 from sklearn.svm import SVR
```

In [5]:
```
1 dataset.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 [6]:
```
1 dataset.describe()
```

Out[6]:

	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

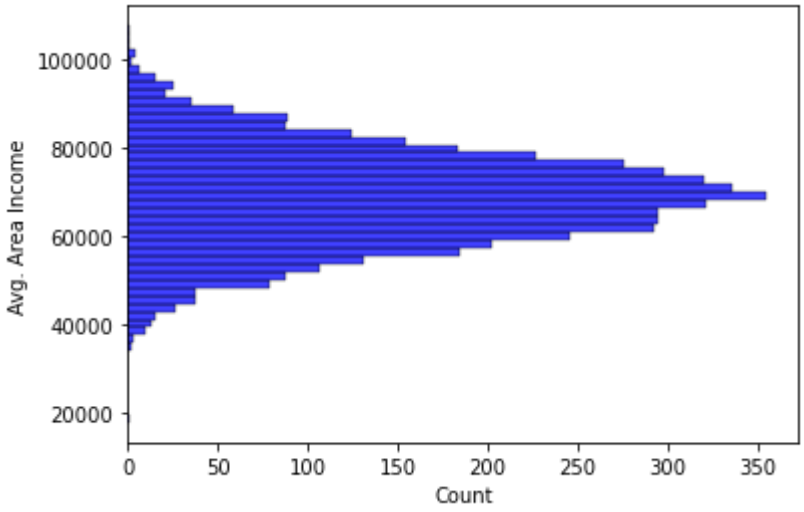
In [7]:
```
1 dataset.columns
```

Out[7]:
```
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')
```

Pre-Processing and Visualisation of Data

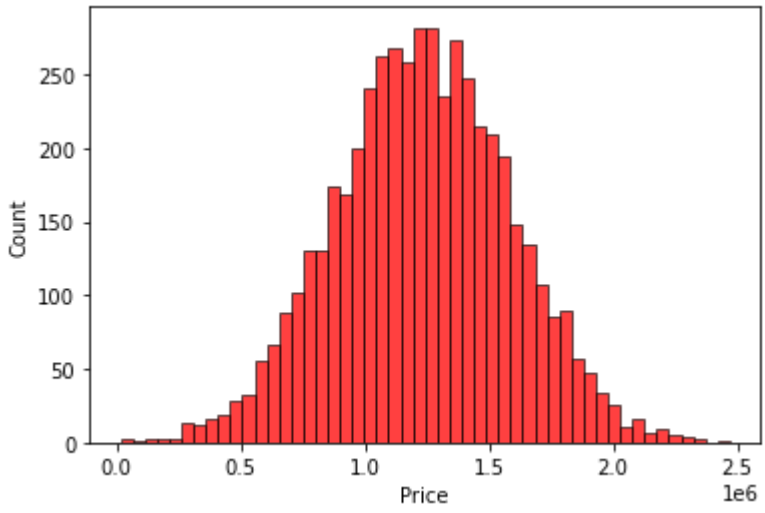
In [4]:
```
1 sns.histplot(dataset, y='Avg. Area Income', bins=50, color='b')
```

Out[4]: <AxesSubplot:xlabel='Count', ylabel='Avg. Area Income'>



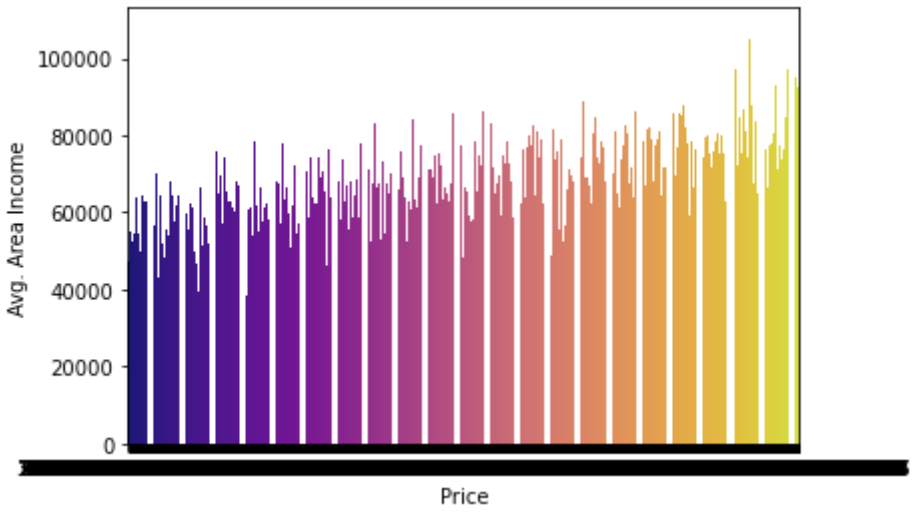
In [5]:
```
1 sns.histplot(dataset, x='Price', bins=50, color='r')
```

Out[5]: <AxesSubplot:xlabel='Price', ylabel='Count'>



In [6]:
```
sns.barplot(x ='Price', y ='Avg. Area Income', data = dataset,
            palette ='plasma')
```

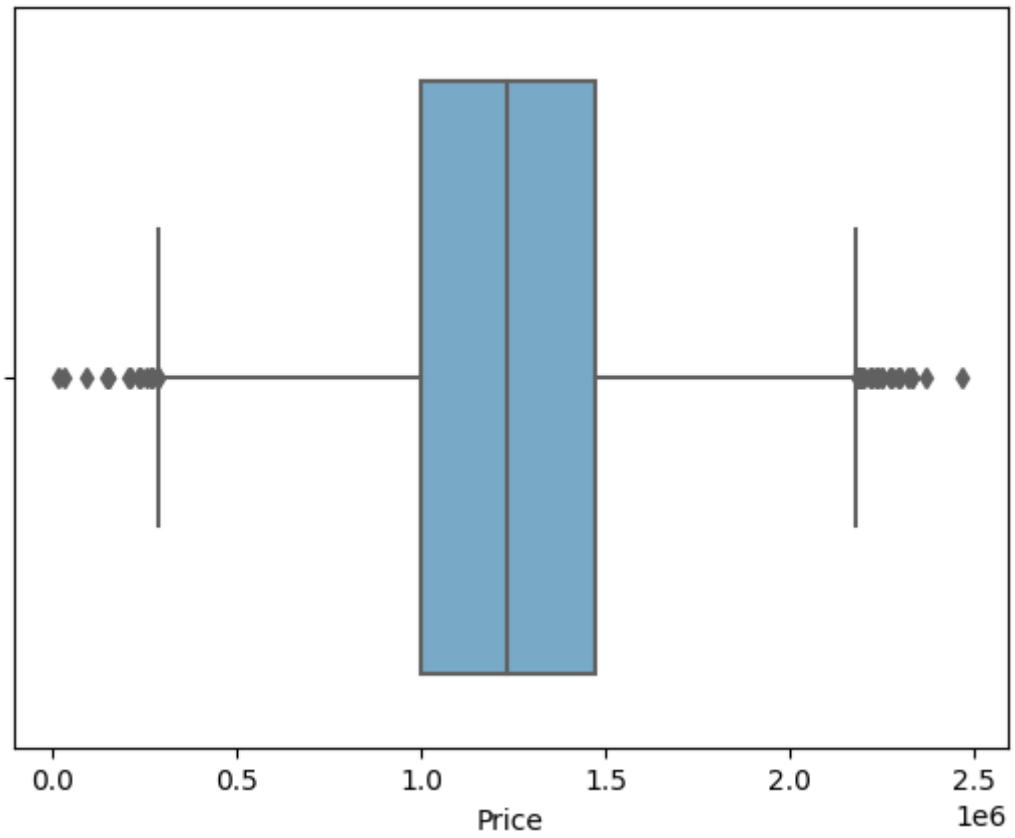
Out[6]: <AxesSubplot:xlabel='Price', ylabel='Avg. Area Income'>



In [7]:
```
plt.show()
```

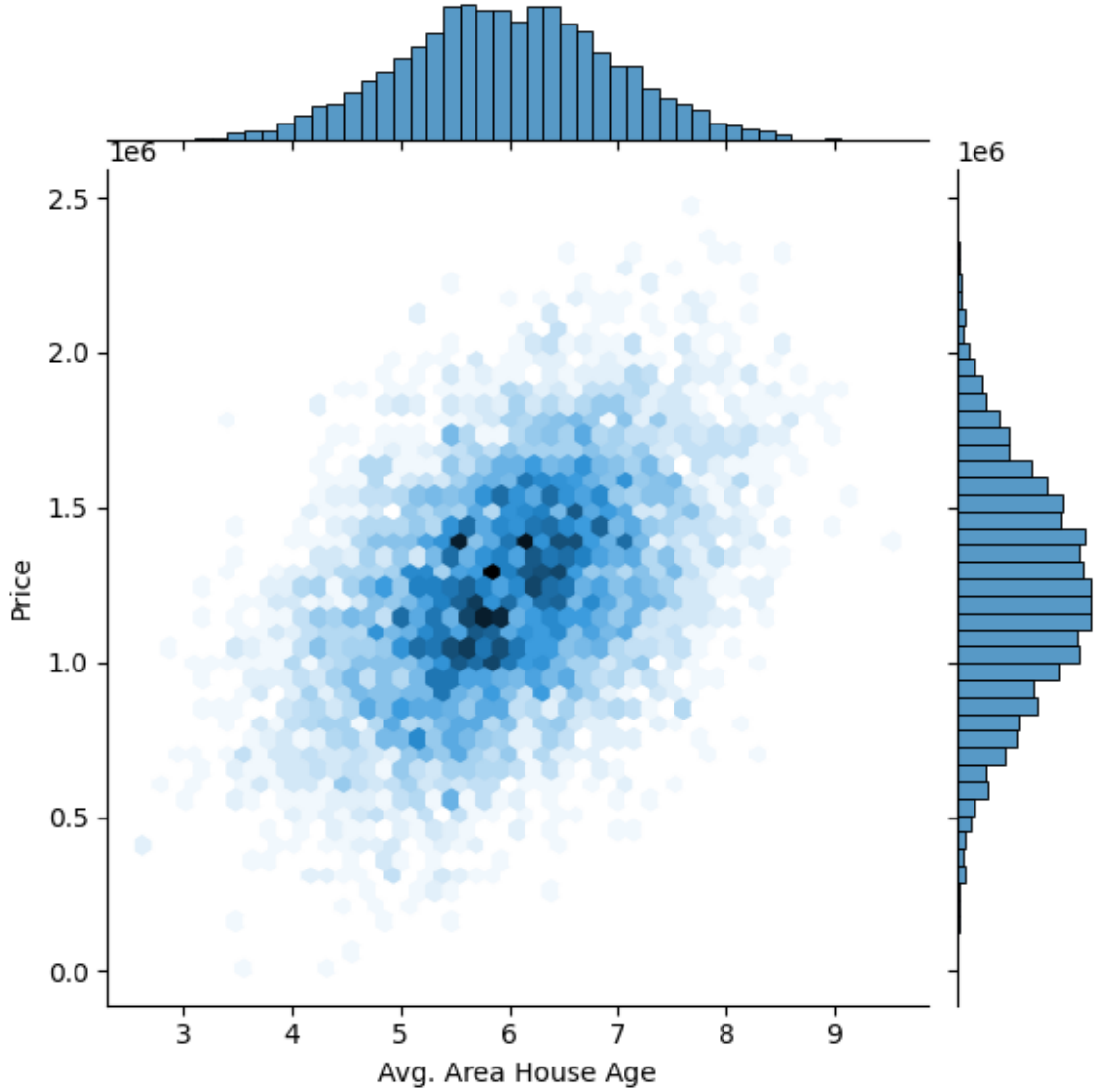
In [9]:
```
sns.boxplot(dataset, x='Price',  palette='Blues')
```

Out[9]: <AxesSubplot: xlabel='Price'>



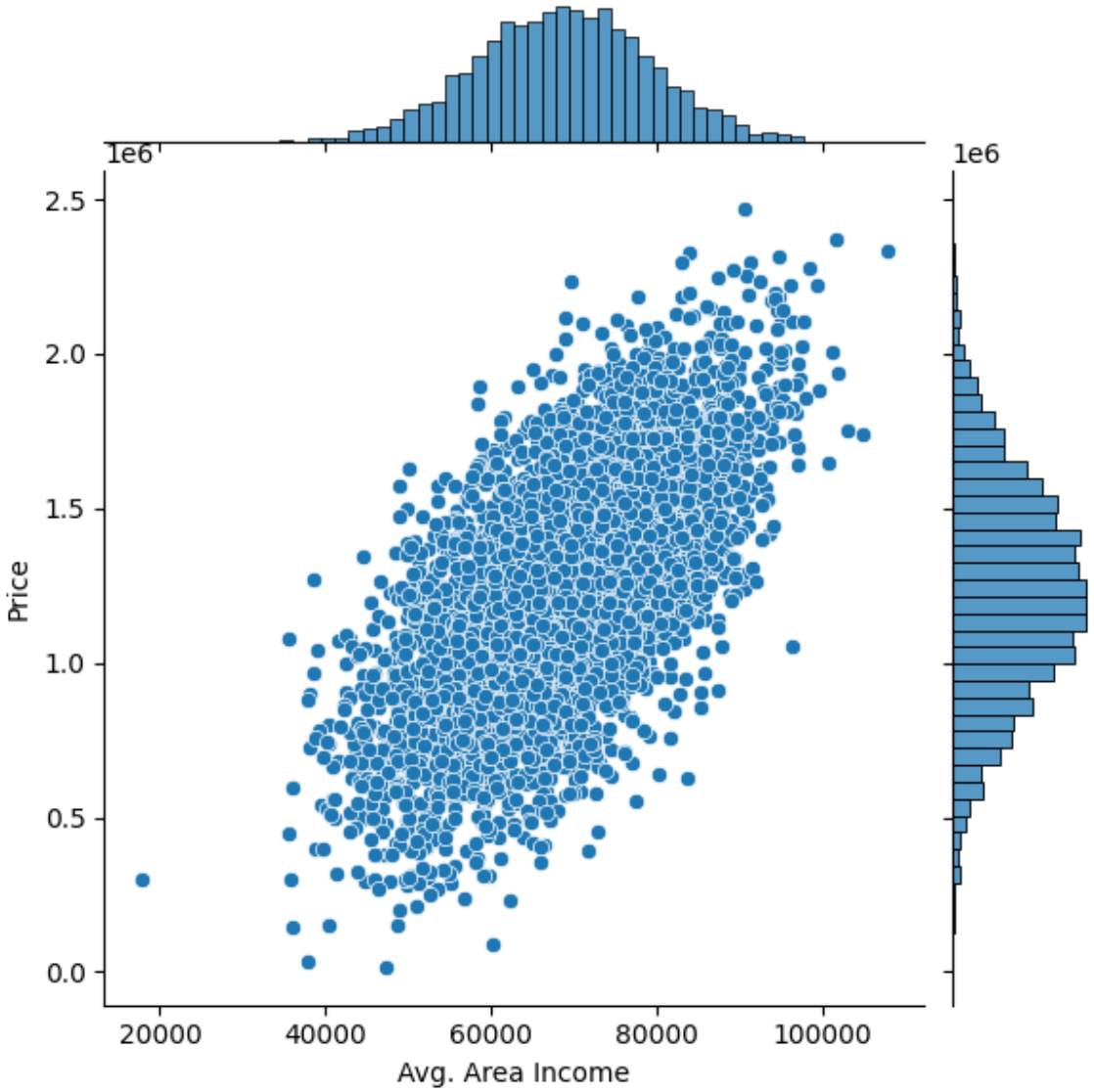
In [10]:
```
1 sns.jointplot(dataset, x='Avg. Area House Age', y='Price', kind='hex')
```

Out[10]:
```
<seaborn.axisgrid.JointGrid at 0x26a278d7910>
```



In [11]:
```
1 sns.jointplot(dataset, x='Avg. Area Income', y='Price')
```

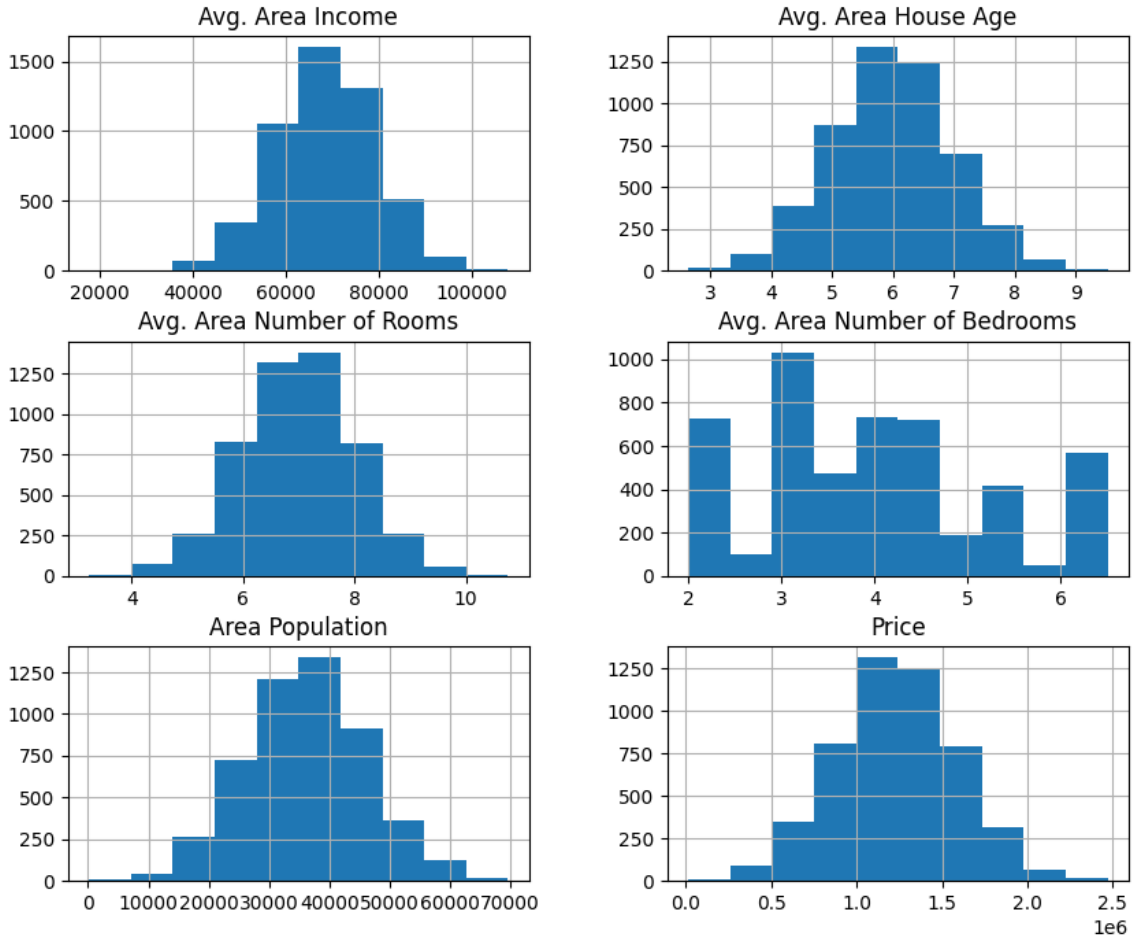
Out[11]:
```
<seaborn.axisgrid.JointGrid at 0x26a27c42890>
```



In [ ]:
```
1
```

In [13]:
```
1 dataset.hist(figsize=(10,8))
```

Out[13]:
```
array([[<AxesSubplot: title={'center': 'Avg. Area Income'}>,
        <AxesSubplot: title={'center': 'Avg. Area House Age'}>],
       [<AxesSubplot: title={'center': 'Avg. Area Number of Rooms'}>,
        <AxesSubplot: title={'center': 'Avg. Area Number of Bedrooms'}>],
       [<AxesSubplot: title={'center': 'Area Population'}>,
        <AxesSubplot: title={'center': 'Price'}>]], dtype=object)
```



Visualising Correlation

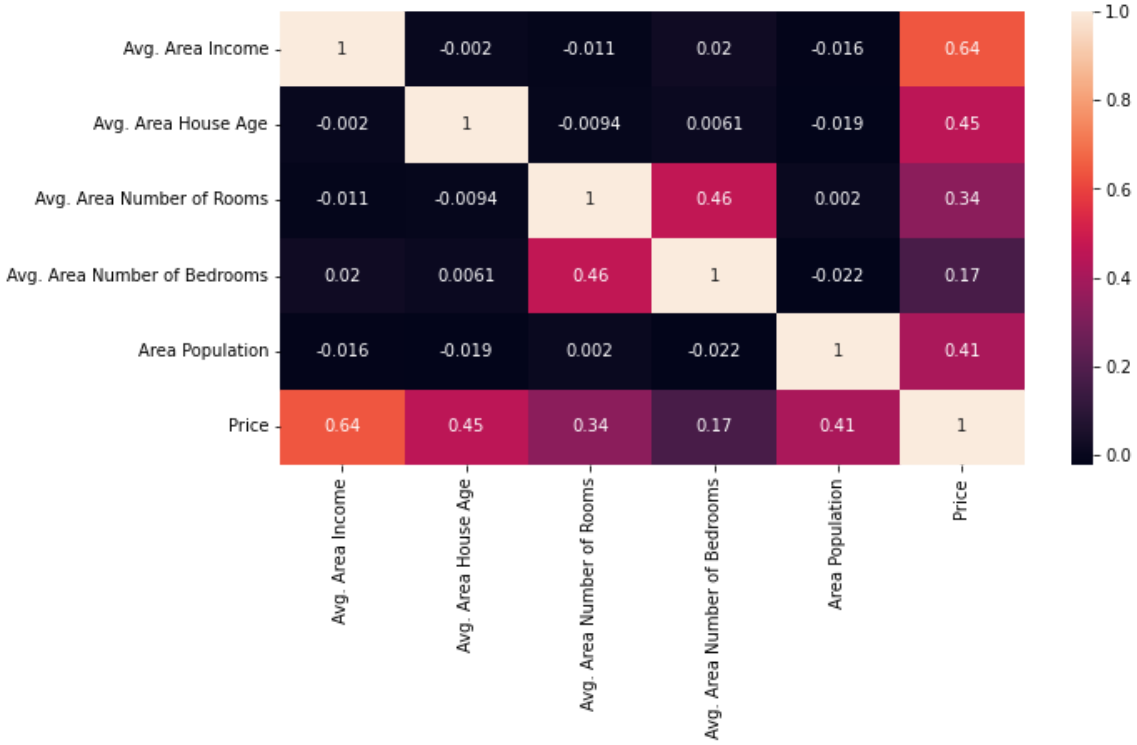
In [16]:
```
1 dataset.corr()
```

Out[16]:

	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 [8]:
```
1 plt.figure(figsize=(10,5))
2 sns.heatmap(dataset.corr(), annot=True)
```

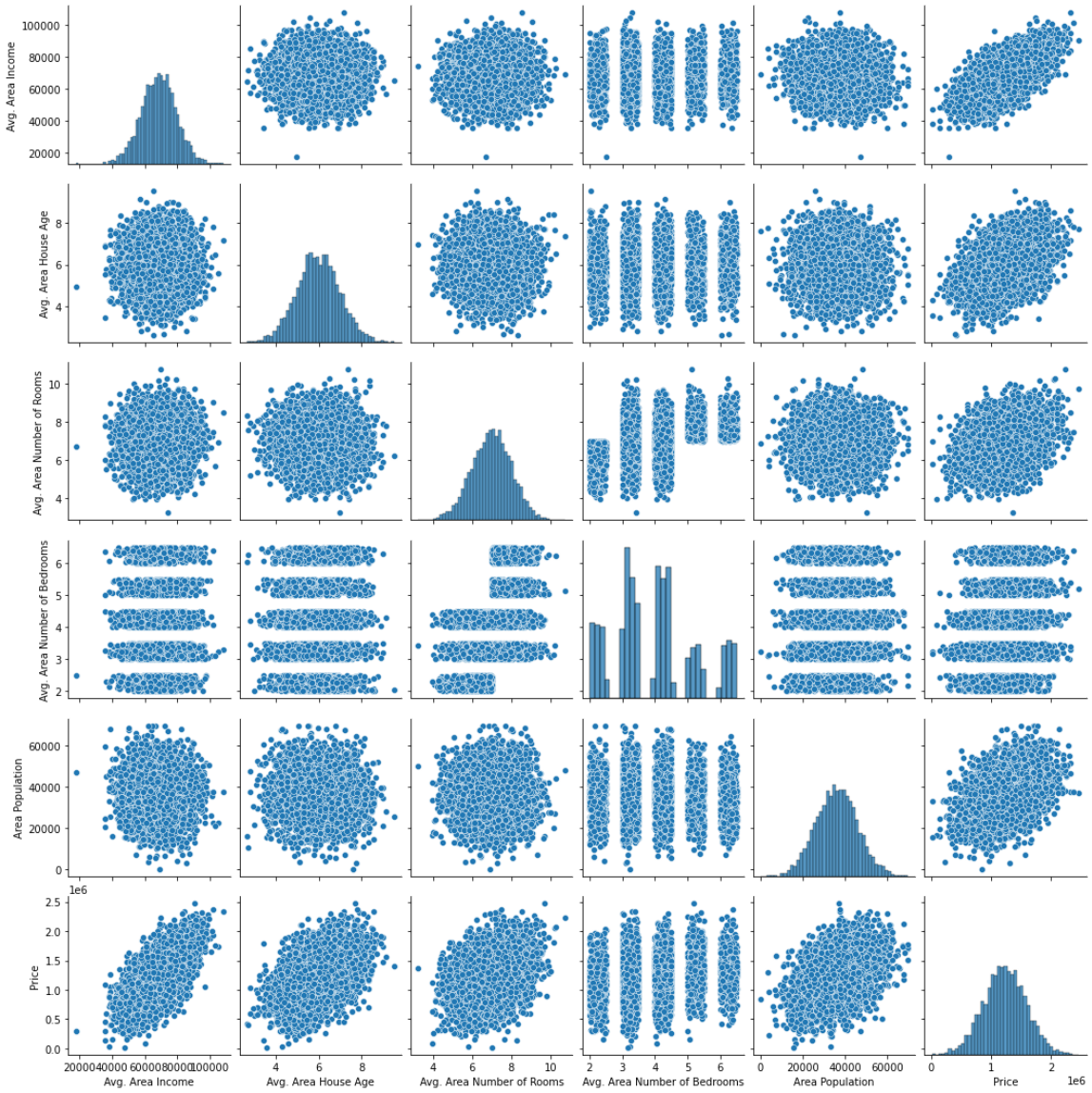
Out[8]: <AxesSubplot:>



In [9]:
```
1 plt.figure(figsize=(12,8))
2 sns.pairplot(dataset)
```

Out[9]:
```
<seaborn.axisgrid.PairGrid at 0x1f3be464fa0>

<Figure size 864x576 with 0 Axes>
```



In [ ]:
```
1
```