

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

import warnings
warnings.filterwarnings("ignore")
```

```
In [3]: df = pd.read_csv("Real_estates.csv")
df
```

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\nDanieltown, 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
...
4995	60567.944140	7.830362	6.137356	3.46	22837.361035	1.060194e+06	USNS Williams\nFPO AP 30153-7653
4996	78491.275435	6.999135	6.576763	4.02	25616.115489	1.482618e+06	PSC 9258, Box 8489\nAPO AA 42991-3352
4997	63390.686886	7.250591	4.805081	2.13	33266.145490	1.030730e+06	4215 Tracy Garden Suite 076\nJoshualand, VA 01...
4998	68001.331235	5.534388	7.130144	5.44	42625.620156	1.198657e+06	USS Wallace\nFPO AE 73316
4999	65510.581804	5.992305	6.792336	4.07	46501.283803	1.298950e+06	37778 George Ridges Apt. 509\nEast Holly, NV 2...

5000 rows × 7 columns

```
In [4]: 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
```

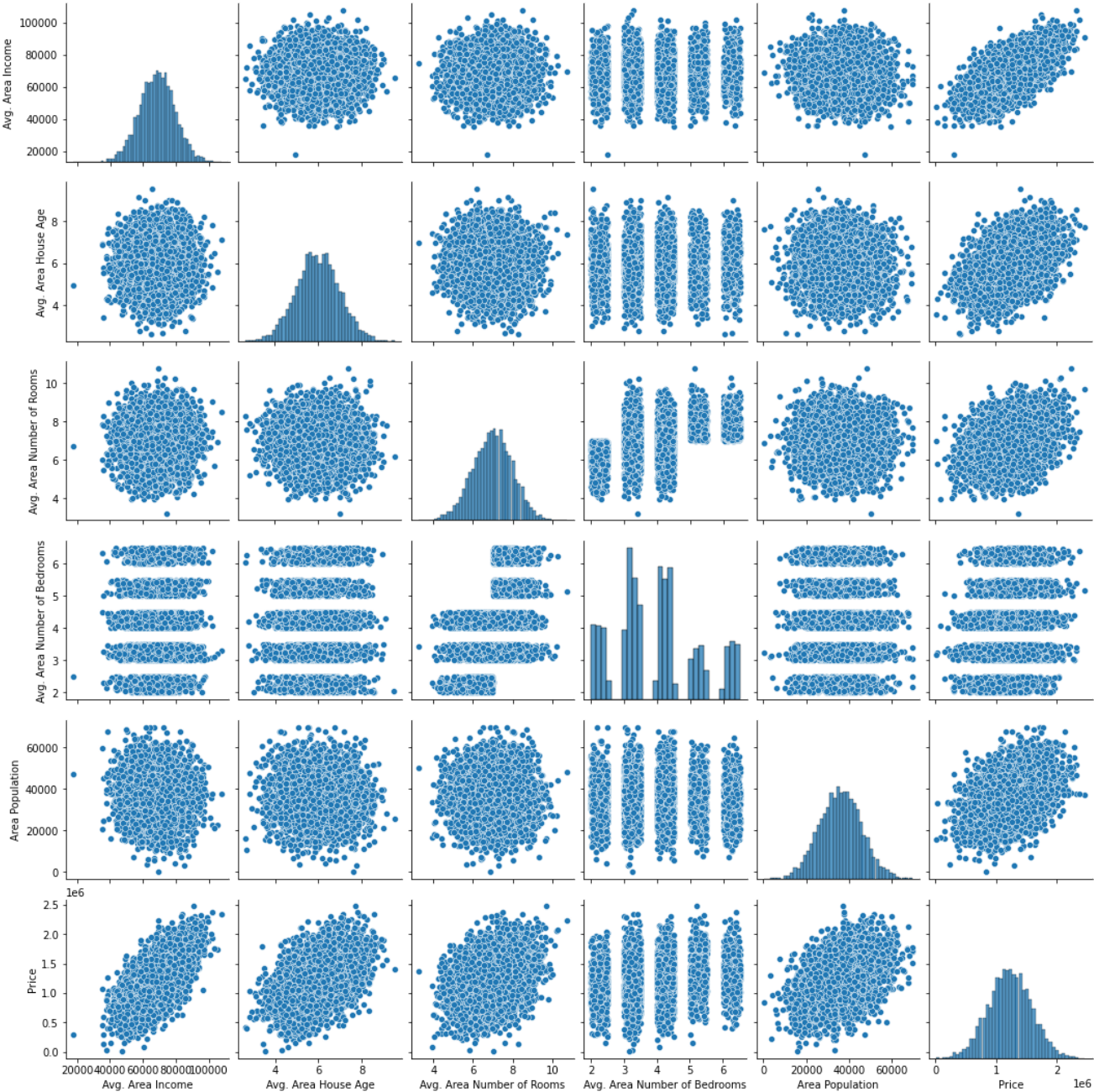
```
In [6]: df.describe()
```

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

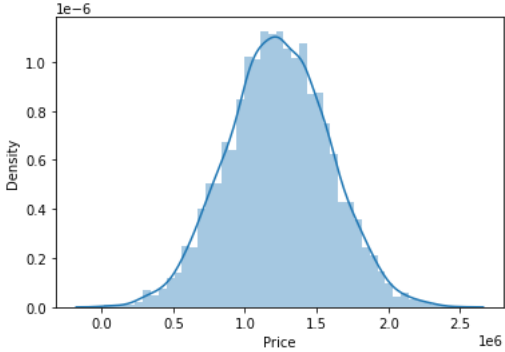
```
In [8]: sns.pairplot(df)
```

Out[8]: <seaborn.axisgrid.PairGrid at 0x1d83e68da80>



```
In [11]: sns.distplot(df["Price"])
```

Out[11]: <AxesSubplot: xlabel='Price', ylabel='Density'>



```
In [12]: df.corr()
```

Out[12]:

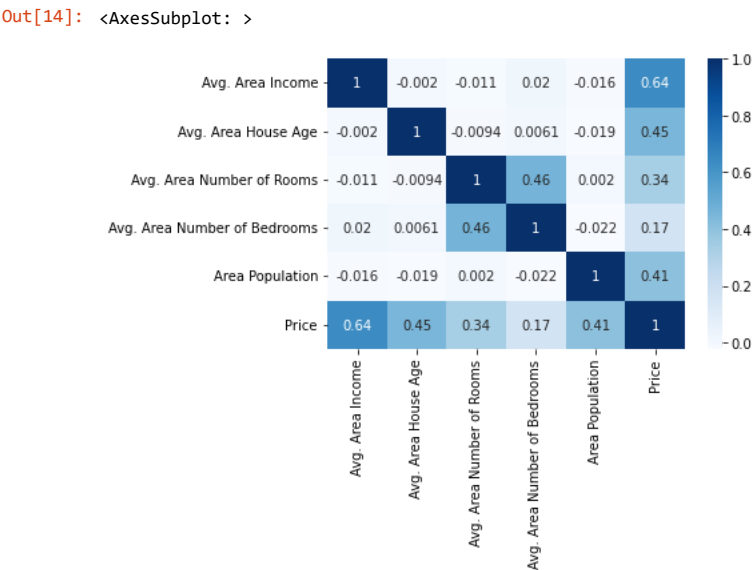
	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
Avg. Area Income	1.000000	-0.002007	-0.011032	0.019788	-0.016234	0.639734
Avg. Area House Age	-0.002007	1.000000	-0.009428	0.006149	-0.018743	0.452543
Avg. Area Number of Rooms	-0.011032	-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 [13]: df.corr().style.background_gradient()
```

Out[13]:

	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 [14]: sns.heatmap(df.corr(),annot=True,cmap="Blues")
```



Model Creation

```
In [15]: df.head()
```

Out[15]:

	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\nDanieltown, 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 [16]: x = df.iloc[:, :-2]
y = df.iloc[:, -2]
```

```
In [17]: x
```

Out[17]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population
0	79545.458574	5.682861	7.009188	4.09	23086.800503
1	79248.642455	6.002900	6.730821	3.09	40173.072174
2	61287.067179	5.865890	8.512727	5.13	36882.159400
3	63345.240046	7.188236	5.586729	3.26	34310.242831
4	59982.197226	5.040555	7.839388	4.23	26354.109472
...
4995	60567.944140	7.830362	6.137356	3.46	22837.361035
4996	78491.275435	6.999135	6.576763	4.02	25616.115489
4997	63390.686886	7.250591	4.805081	2.13	33266.145490
4998	68001.331235	5.534388	7.130144	5.44	42625.620156
4999	65510.581804	5.992305	6.792336	4.07	46501.283803

5000 rows × 5 columns

```
In [18]: y
```

Out[18]:

0	1.059034e+06
1	1.505891e+06
2	1.058988e+06
3	1.260617e+06
4	6.309435e+05
...	...
4995	1.060194e+06
4996	1.482618e+06
4997	1.030730e+06
4998	1.198657e+06
4999	1.298950e+06

Name: Price, Length: 5000, dtype: float64

```
In [19]: from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.2,random_state=1)
```

```
In [20]: from sklearn.linear_model import LinearRegression

linreg=LinearRegression()

linreg.fit(xtrain,ytrain)

ypred = linreg.predict(xtest)
```

```
In [21]: linreg.coef_
```

Out[21]: array([2.16667346e+01, 1.64990052e+05, 1.20784238e+05, 1.54252468e+03, 1.51503697e+01])

```
In [22]: linreg.intercept_
```

Out[22]: -2637185.64007627

```
In [23]: pd.DataFrame(linreg.coef_,index=x.columns, columns=["Coefficient"])
```

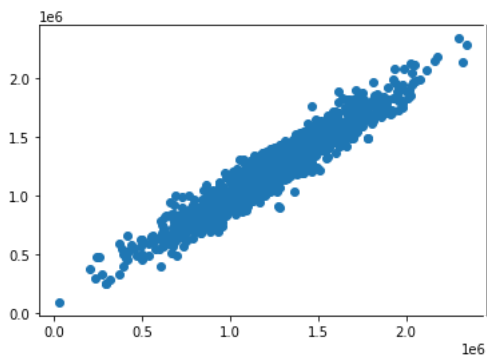
Out[23]:

	Coefficient
Avg. Area Income	21.666735
Avg. Area House Age	164990.051829
Avg. Area Number of Rooms	120784.238317
Avg. Area Number of Bedrooms	1542.524676
Area Population	15.150370

- for every growth in X, we estimate that Y will grow by M

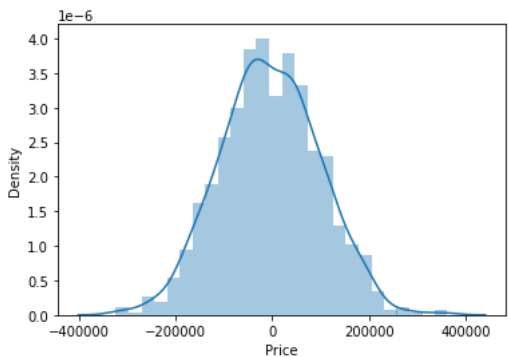
```
In [24]: plt.scatter(ytest,ypred)
```

Out[24]: <matplotlib.collections.PathCollection at 0x1d8494c6d10>



```
In [26]: sns.distplot((ytest-ypred))
```

Out[26]: <AxesSubplot: xlabel='Price', ylabel='Density'>



Regression Evaluation Metrics

```
In [27]: from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score

mae=mean_absolute_error(ytest,ypred)
mse=mean_squared_error(ytest,ypred)

rmse=np.sqrt(mse)

r2=r2_score(ytest,ypred)

print(f"MAE:- {mae}\n MSE:- {mse}\n RMSE:- {rmse}\n Accuracy:- {r2}")
```

MAE:- 82520.32996179513
MSE:- 10542495109.238657
RMSE:- 102676.65318483388
Accuracy:- 0.9216017201335713

```
In [28]: x.columns
```

Out[28]: Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms', 'Avg. Area Number of Bedrooms', 'Area Population'], dtype='object')

In [29]:

```
def predictprice(aai,aaha,aanor,aanob,ap):  
    newobs=[[aai,aaha,aanor,aanob,ap]]  
  
    yp=linreg.predict(newobs)[0]  
  
    print(f"The price of your Dream House is $ {yp:.2f}")
```

In [34]:

```
predictprice(70000,15,10,5,10000)
```

The price of your Dream House is \$ 2721395.26

In []: