

Deena 20104016

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as pp
```

Problem Statement

LINEAR REGRESSION

```
In [2]: a = pd.read_csv("Housing.csv")
```

```
Out[2]:
```

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Addr
0	79545.45857	5.682861	7.009188	4.09	23086.80050	1.059034e+06	208 Michael Ferry , 674\nLaurabury, 370
1	79248.64245	6.002900	6.730821	3.09	40173.07217	1.505891e+06	188 Johnson Vi Suite 079\nL Kathleen, C
2	61287.06718	5.865890	8.512727	5.13	36882.15940	1.058988e+06	9127 Elizak Stravenue\nDanielto WI 0648
3	63345.24005	7.188236	5.586729	3.26	34310.24283	1.260617e+06	USS Barnett\nFPO 44
4	59982.19723	5.040555	7.839388	4.23	26354.10947	6.309435e+05	USNS Raymond\nF AE 09
...
4995	60567.94414	7.830362	6.137356	3.46	22837.36103	1.060194e+06	USNS Williams\nF AP 30153-7
4996	78491.27543	6.999135	6.576763	4.02	25616.11549	1.482618e+06	PSC 9258, 8489\nAPO 42991-3
4997	63390.68689	7.250591	4.805081	2.13	33266.14549	1.030730e+06	4215 Tracy Gar Suite 076\nJoshual VA (
4998	68001.33124	5.534388	7.130144	5.44	42625.62016	1.198657e+06	USS Wallace\nFPO 73
4999	65510.58180	5.992305	6.792336	4.07	46501.28380	1.298950e+06	37778 George Rid Apt. 509\nEast H NV

5000 rows × 7 columns

HEAD

In [3]:

Out[3]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Address
0	79545.45857	5.682861	7.009188	4.09	23086.80050	1.059034e+06	208 Michael Ferry Apt. 674\nLaurabury, NE 3701...
1	79248.64245	6.002900	6.730821	3.09	40173.07217	1.505891e+06	188 Johnson Views Suite 079\nLake Kathleen, CA...
2	61287.06718	5.865890	8.512727	5.13	36882.15940	1.058988e+06	9127 Elizabeth Stravenue\nDanielstown, WI 06482...
3	63345.24005	7.188236	5.586729	3.26	34310.24283	1.260617e+06	USS Barnett\nFPO AP 44820
4	59982.19723	5.040555	7.839388	4.23	26354.10947	6.309435e+05	USNS Raymond\nFPO AE 09386

Data Cleaning and Preprocessing

In [4]:

Out[4]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Address
0	79545.45857	5.682861	7.009188	4.09	23086.80050	1.059034e+06	208 Michael Ferry Apt. 674\nLaurabury, NE 3701...
1	79248.64245	6.002900	6.730821	3.09	40173.07217	1.505891e+06	188 Johnson Views Suite 079\nLake Kathleen, CA...
2	61287.06718	5.865890	8.512727	5.13	36882.15940	1.058988e+06	9127 Elizabeth Stravenue\nDanielstown, WI 06482...
3	63345.24005	7.188236	5.586729	3.26	34310.24283	1.260617e+06	USS Barnett\nFPO AP 44820
4	59982.19723	5.040555	7.839388	4.23	26354.10947	6.309435e+05	USNS Raymond\nFPO AE 09386

In [5]:

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%	61480.562390	5.322283	6.299250	3.140000	29403.928700	9.975771e+05
50%	68804.286405	5.970429	7.002902	4.050000	36199.406690	1.232669e+06
75%	75783.338665	6.650808	7.665871	4.490000	42861.290770	1.471210e+06
max	107701.748400	9.519088	10.759588	6.500000	69621.713380	2.469066e+06

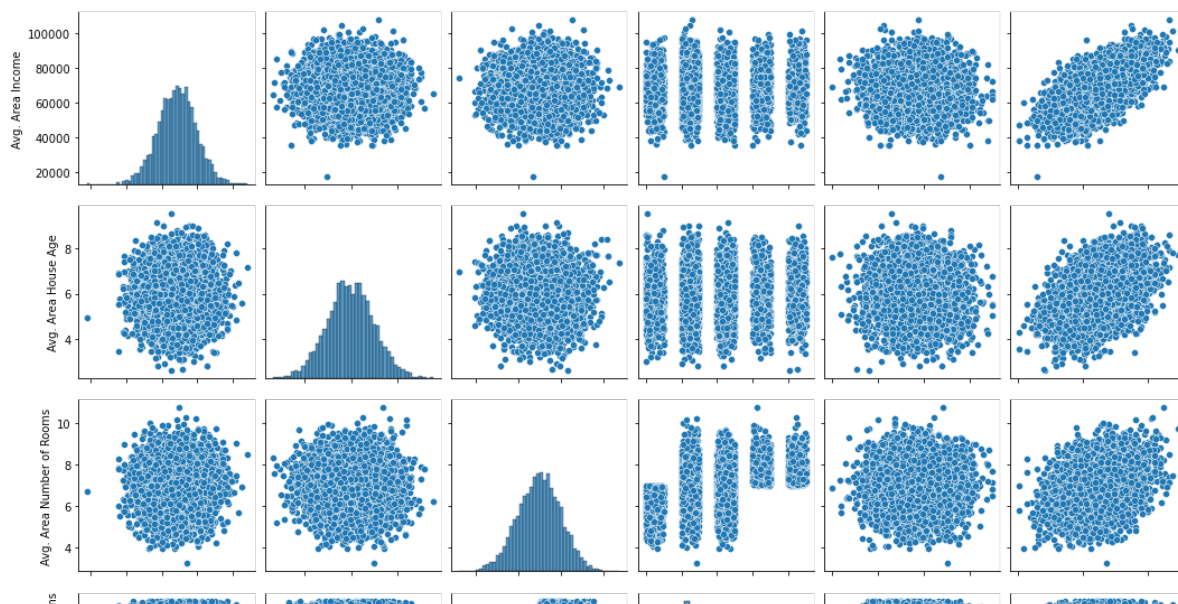
To display heading

In [6]:

```
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')
```

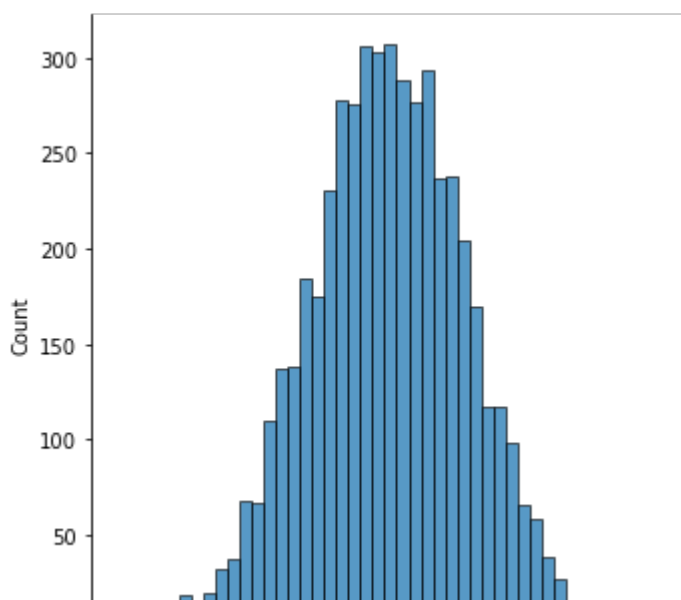
In [7]:

```
Out[7]: <seaborn.axisgrid.PairGrid at 0x265d3080a90>
```



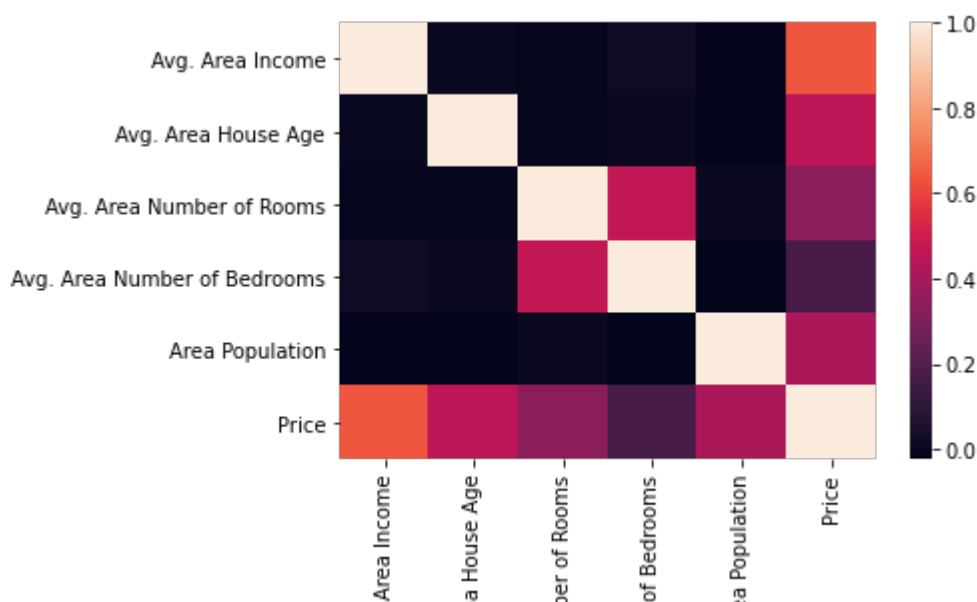
In [8]:

Out[8]: <seaborn.axisgrid.FacetGrid at 0x265d1f88880>



In [9]:

Out[9]: <AxesSubplot:>



TO TRAIN THE MODEL - MODEL BUILDING

```
In [10]: x = a[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
               'Avg. Area Number of Bedrooms', 'Area Population']]
```

```
In [11]: # 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 [12]: from sklearn.linear_model import LinearRegression
lr = LinearRegression()
```

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

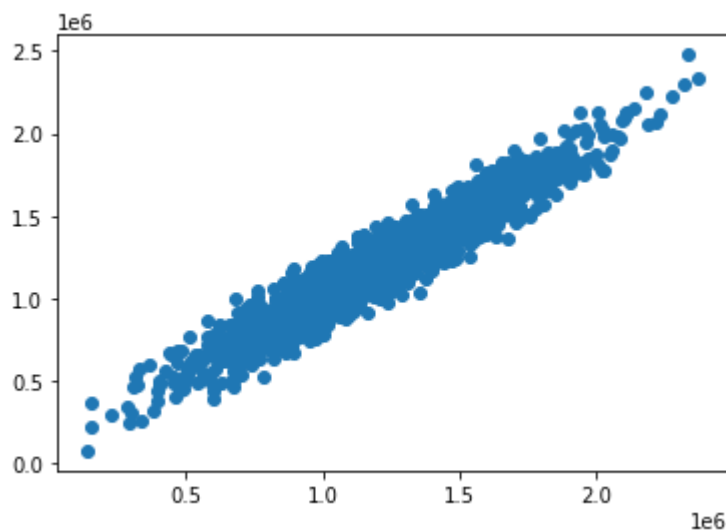
```
In [13]: coeff = pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
```

```
Out[13]:
```

	Co-efficient
Avg. Area Income	21.688828
Avg. Area House Age	165869.952778
Avg. Area Number of Rooms	119584.191352
Avg. Area Number of Bedrooms	2425.669921
Area Population	15.151147

```
In [14]: prediction= lr.predict(x_test)
```

```
Out[14]: <matplotlib.collections.PathCollection at 0x265d6982340>
```



```
In [15]:
```

```
Out[15]: 0.9135870549702687
```

```
In [16]:
```

```
Out[16]: 0.9199015853244594
```

RIDGE & LASSO

```
In [17]: from sklearn.linear_model import Ridge,Lasso
```

```
In [18]: rr=Ridge(alpha=10)
         rr.fit(x_train,y_train)
```

Out[18]: Ridge(alpha=10)

```
In [19]:
```

Out[19]: 0.9135713242765168

```
In [20]: la=Lasso(alpha=10)
         la.fit(x_train,y_train)
```

Out[20]: Lasso(alpha=10)

```
In [21]:
```

Out[21]: 0.9135870177805252

```
In [22]: from sklearn.linear_model import ElasticNet
         a=ElasticNet()
```

Out[22]: ElasticNet()

```
In [24]: print(a.coef_)
         print(a.intercept_)
         print(a.score(x_test,y_test))
```

```
[2.15524772e+01 1.10257389e+05 7.51686440e+04 1.43943316e+04
 1.50144379e+01]
-2030571.9563129405
0.8774184817895994
[1437559.4115755 1395848.16851562 1221937.05450338 ... 982201.88014986
1302551.79137851 1304294.89302719]
```

```
In [26]: from sklearn import metrics
         print(" Mean Absolute Error :",metrics.mean_absolute_error(y_test,prediction))
         print(" Mean Squared Error :",metrics.mean_squared_error(y_test,prediction))
```

```
Mean Absolute Error : 84143.71264095792
Mean Squared Error : 10945504154.969105
Root Mean Absolute Error : 290.0753568315618
```