DAY 9:

Housing Dataset

In [1]:

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

In [2]:

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

Out[2]:

	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 3C
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\nE

5000 rows × 7 columns

In [3]:

```
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 [4]:

```
df.columns
```

Out[4]:

```
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')
```

Linear Regression

In [5]:

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

In [6]:

```
# to split my dataset into test and train 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 [7]:

```
from sklearn.linear_model import LinearRegression

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

Out[7]:

```
LinearRegression()
```

In [8]:

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

```
0.9206740737751782
```

In [9]:

```
lr.score(x_train,y_train)
```

Out[9]:

```
0.916798124057497
```

Ridge Regression

In [10]:

```
from sklearn.linear_model import Ridge,Lasso
```

In [11]:

```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
rr.score(x_test,y_test)
```

Out[11]:

```
0.9206843744134923
```

Lasso Regression

In [12]:

```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[12]:

```
Lasso(alpha=10)
```

In [13]:

```
la.score(x_test,y_test)
```

Out[13]:

```
0.9206738854547956
```

In [ ]:

```

```