LINEAR REGRESSION MACHINE LEARNING PROJECT ON HOUSE PRICES

Dataset used:- USA_Housing.csv

In [3]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

%matplotlib inline
```

In [5]:
```
HouseDF= pd.read_csv('USA_Housing.csv')
```

In [7]:
```
HouseDF.head()
```

Out[7]:

	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 [9]:
```
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 [11]:
```
HouseDF.describe()
```

Out[11]:

	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	1.232069e+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

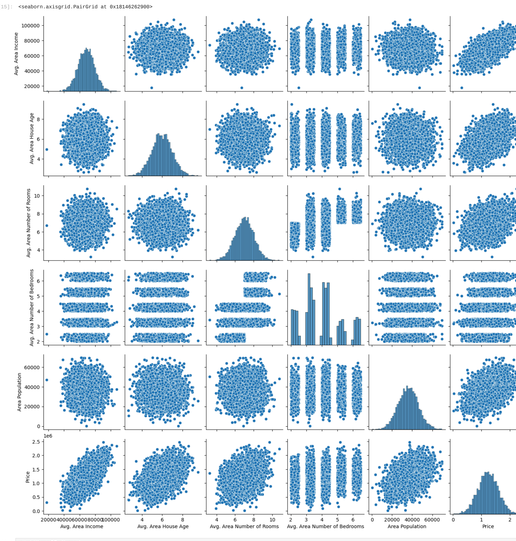
In [13]:
```
HouseDF.columns
```

Out[13]:
```
Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price', 'Address'],
      dtype='object')
```

Exploratory Data Analysis

In [15]:
```
sns.pairplot(HouseDF)
```

Out[15]:
```
<seaborn.axisgrid.PairGrid at 0x181402629d0>
```

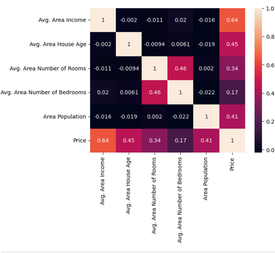


In [21]:
```
print(HouseDF.dtypes)
```

```
Avg. Area Income              float64
Avg. Area House Age           float64
Avg. Area Number of Rooms     float64
Avg. Area Number of Bedrooms  float64
Area Population               float64
Price                         float64
Address                        object
dtype: object
```

In [23]:
```
numeric_df = HouseDF.select_dtypes(include='number')
corr_matrix = numeric_df.corr()
sns.heatmap(corr_matrix, annot=True)
```

Out[23]:
```
<Axes: >
```



In [31]:
```
X=HouseDF[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
       'Avg. Area Number of Bedrooms', 'Area Population']]
y=HouseDF['Price']
```

Training the Model

In [33]:
```
from sklearn.model_selection import train_test_split
```

Train-Test Split

In [35]:
```
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.40, random_state=101)
```

In [37]:
```
X_train
```

Out[37]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population
1303	68091.179676	5.364208	7.502956	3.10	44557.379656
1051	75729.785546	5.580599	7.642973	4.21	29996.018448
4904	70885.420619	6.356747	7.250241	5.42	38627.301473
931	73288.407340	4.966300	7.915453	4.30	38413.490464
4976	75046.313791	5.351169	7.797825	5.23	34107.889619
...	...	...	...	...	...
4171	56610.642563	4.846832	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	75078.795516	7.644779	8.440726	4.33	56148.449322

3000 rows × 5 columns

In [39]:
```
from sklearn.linear_model import LinearRegression
```

In [41]:
```
lm=LinearRegression()
```

In [43]:
```
lm.fit(X_train,y_train)
```

Out[43]:
```
LinearRegression()
```

Model Evaluation

In [69]:
```
#print the intercept
print(lm.intercept_)
```

```
-2640159.79685267
```

In [71]:
```
print(lm.coef_)
```

```
[-2.15282755e+01 1.64883282e+05 1.22360678e+05 2.23380186e+03
 1.51504200e+01]
```

In [47]:
```
coeff_df=pd.DataFrame(lm.coef_,X.columns,columns=['Coefficient'])
```

In [49]:
```
coeff_df
```

Out[49]:

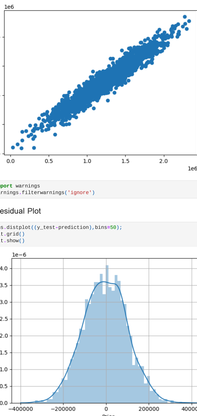
	Coefficient
Avg. Area Income	21.528276
Avg. Area House Age	164883.282027
Avg. Area Number of Rooms	122368.678027
Avg. Area Number of Bedrooms	2233.801864
Area Population	15.150420

Prediction

In [51]:
```
prediction=lm.predict(X_test)
```

In [53]:
```
plt.scatter(y_test, prediction)
```

Out[53]:
```
<matplotlib.collections.PathCollection at 0x1815043200>
```



In [61]:
```
import warnings
warnings.filterwarnings('ignore')
```

Residual Plot

In [63]:
```
sns.distplot((y_test-prediction),bins=50);
plt.grid()
plt.show()
```


In [ ]:
```

```