In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
df=pd.read_csv('Real_estates.csv')
```

In [3]:
```
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 [24]:
```
df.corr().style.background_gradient()
```

Out[24]:

	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 [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
```

```
There is no missing values or null values present in our dataset
```

```
Splitting the dataset into features and target
```

In [5]:
```
features=df.iloc[:,0:5]
features
```

Out[5]:

	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 [6]:
```
target=df.iloc[:,5:6]
target
```

Out[6]:

	Price
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

5000 rows × 1 columns

In [7]:
```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(features,target,test_size=0.2,random_state=1)
```

In [8]:
```
xtrain
```

Out[8]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population
1233	68562.024528	6.317286	7.305647	3.22	35476.168798
1056	68656.906773	7.354458	8.787908	6.36	43833.853437
1686	57869.268480	5.625299	7.601622	3.39	31818.932565
187	68844.764249	4.860453	6.916808	3.29	48392.497360
3840	62041.428293	6.692078	7.121939	3.25	46069.976912
...	...	...	...	...	...
2895	56734.350763	6.159101	8.280404	4.30	27982.271707
2763	50212.439535	6.645207	7.404114	5.44	20913.655444
905	80011.583519	6.448675	6.489268	2.49	26576.391994
3980	72899.658203	5.222040	6.861010	4.21	39311.147543
235	67056.840480	5.222169	7.163518	5.25	25134.681485

4000 rows × 5 columns

In [9]:
```
xtest
```

Out[9]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population
2764	75012.341660	6.742828	6.604335	4.10	42877.424147
4767	76187.273309	6.156222	7.166149	3.32	45084.394236
3814	67622.219611	5.813928	5.071112	4.16	35359.848465
3499	66933.165273	4.748787	5.879803	2.09	41834.042941
2735	65192.105635	6.275509	8.017889	4.47	26228.394577
...	...	...	...	...	...
448	66356.059961	7.480941	6.725864	3.19	38022.838199
921	68008.615434	4.357088	7.879545	5.41	28908.086785
4087	100741.298585	5.870726	6.644853	4.33	26041.487616
1242	62798.232983	5.890872	6.481651	3.05	34652.257887
2242	38868.250311	6.965104	8.966906	4.22	25432.076773

1000 rows × 5 columns

In [10]:
```
ytrain
```

Out[10]:

	Price
1233	1.336378e+06
1056	1.747911e+06
1686	8.868159e+05
187	1.211102e+06
3840	1.376898e+06
...	...
2895	1.063206e+06
2763	8.732420e+05
905	1.345963e+06
3980	1.270928e+06
235	1.039107e+06

4000 rows × 1 columns

In [11]:
```
ytest
```

Out[11]:

	Price
2764	1.413580e+06
4767	1.618721e+06
3814	8.413925e+05
3499	8.814439e+05
2735	1.174748e+06
...	...
448	1.309986e+06
921	1.059871e+06
4087	1.644923e+06
1242	1.106337e+06
2242	7.590447e+05

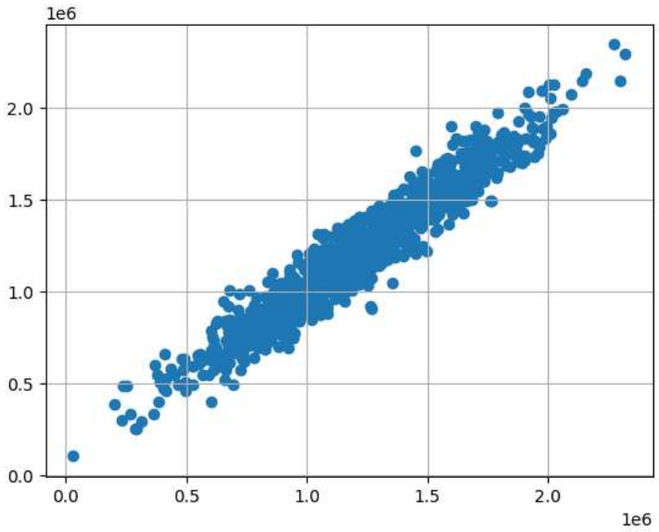
1000 rows × 1 columns

In [12]:
```
from sklearn.linear_model import LinearRegression
linreg=LinearRegression()
linreg.fit(xtrain,ytrain)
ypred=linreg.predict(xtest)
```

In [13]:
```
ypred
```

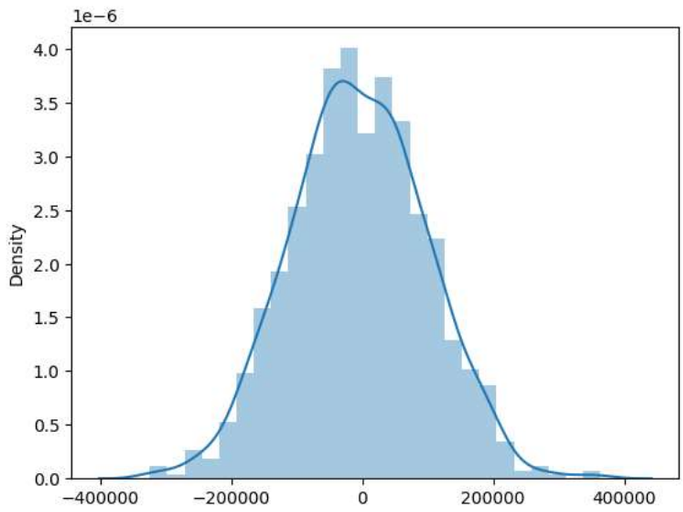
Out[13]:
```
array([[1554219.15758774],
       [1582983.38777546],
       [ 941849.34298559],
       [ 943752.69966419],
       [1183410.51047894],
       [ 330129.92089745],
       [1927632.21182627],
       [1069922.75415195],
       [1648778.05625173],
       [1073503.34526959],
       [ 626617.70408024],
       [1538963.80619366],
       [1631224.37750856],
       [1054158.78438377],
       [ 961171.82968555],
       [1234681.84695504],
       [1428653.88191322],
       [ 925913.80231812],
       [1128251.49779406],
```

In [28]:
```
plt.scatter(ytest,ypred)
plt.grid()
plt.show()
```



In [30]:
```
sns.distplot(ytest-ypred)
```

Out[30]: <AxesSubplot: ylabel='Density'>



In [14]:
```
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)
r2score=r2_score(ytest,ypred)
print(f'MAE={mae}\nMSE={mse}\nRMSE={rmse}\nR2Score={r2score}')
```

```
MAE=82494.73770125103
MSE=10543597313.62491
RMSE=102682.0204009685
R2Score=0.9215935236936348
```

In [15]:
```
features
```

Out[15]:

	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 [16]:
```
linreg.coef_
```

Out[16]:
```
array([[2.16667346e+01, 1.64990052e+05, 1.20784238e+05, 1.54252468e+03,
        1.51503697e+01]])
```

In [17]:
```
linreg.intercept_
```

Out[17]:
```
array([-2637185.64007627])
```

In [18]:
```
pd.DataFrame(linreg.coef_[0],index=features.columns,columns=['Coefficient'])
```

Out[18]:

	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

```
Above data shows that if x increases by 1 unit then average area income increases by $21.666735 and avg. area house age price
increases by $164990.0541829 and Avg. Area Number of Rooms price increases by $120784.238317 and Avg. Area Number of Bedrooms
price increases by $1542.524676 and Area Population income increases by $15.150370
```

In [19]:
```
df.corr()
```

Out[19]:

	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 [20]:
```
def getpredictions(aai,aaha,aanor,aanob,ap):
    newobs=[[aai,aaha,aanor,aanob,ap]]
    yp=linreg.predict(newobs)[0][0]
    print(f'The price of your dream house is = ${yp:.2f}')
```

In [21]:
```
getpredictions(70000,6,8,3,5000)
```

```
The price of your dream house is = $916079.42
```

In [22]:
```
getpredictions(70000,7,8,3,5000)
```

```
The price of your dream house is = $1081069.47
```

In [ ]:
```

```