


```
In [37]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

```
In [38]: df=pd.read_csv("Housing.csv")
```

```
In [39]: df.head()
```

```
Out[39]:
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheat
0	13300000	7420	4	2	3	yes	no	no	
1	12250000	8960	4	4	4	yes	no	no	
2	12250000	9960	3	2	2	yes	no	yes	
3	12215000	7500	4	2	2	yes	no	yes	
4	11410000	7420	4	1	2	yes	yes	yes	



```
In [40]: df.isnull().sum()
```

```
Out[40]: price      0
area      0
bedrooms   0
bathrooms  0
stories    0
mainroad   0
guestroom  0
basement   0
hotwaterheating  0
airconditioning  0
parking    0
prefarea   0
furnishingstatus  0
dtype: int64
```

```
In [41]: df.describe()
```

```
Out[41]:
```

	price	area	bedrooms	bathrooms	stories	parking
count	5.450000e+02	545.000000	545.000000	545.000000	545.000000	545.000000
mean	4.766729e+06	5150.541284	2.965138	1.286239	1.805505	0.693578
std	1.870440e+06	2170.141023	0.738064	0.502470	0.867492	0.861586
min	1.750000e+06	1650.000000	1.000000	1.000000	1.000000	0.000000
25%	3.430000e+06	3600.000000	2.000000	1.000000	1.000000	0.000000
50%	4.340000e+06	4600.000000	3.000000	1.000000	2.000000	0.000000
75%	5.740000e+06	6360.000000	3.000000	2.000000	2.000000	1.000000
max	1.330000e+07	16200.000000	6.000000	4.000000	4.000000	3.000000

```
In [42]: df.columns
```

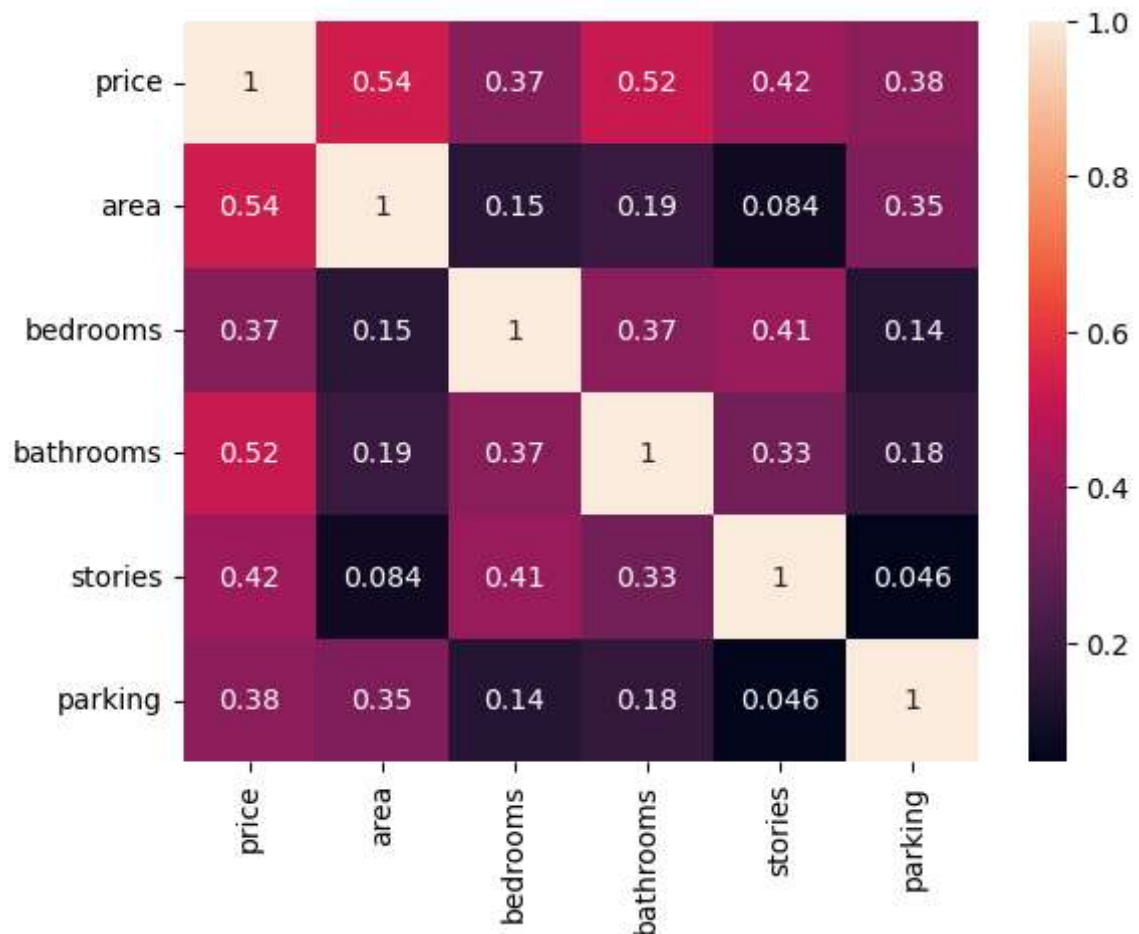
```
Out[42]: Index(['price', 'area', 'bedrooms', 'bathrooms', 'stories', 'mainroad',  
                'guestroom', 'basement', 'hotwaterheating', 'airconditioning',  
                'parking', 'prefarea', 'furnishingstatus'],  
               dtype='object')
```

```
In [43]: sns.heatmap(df.corr(),annot=True)
```

C:\Users\computer\AppData\Local\Temp\ipykernel_15404\4277794465.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.

```
sns.heatmap(df.corr(),annot=True)
```

Out[43]: <Axes: >



```
In [56]: x=df[['area', 'bedrooms', 'bathrooms']]
y=df['price']
```

```
In [57]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
```

```
In [58]: model = LinearRegression()
```

```
In [59]: model.fit(x_train,y_train)
```

Out[59]:

▼ LinearRegression

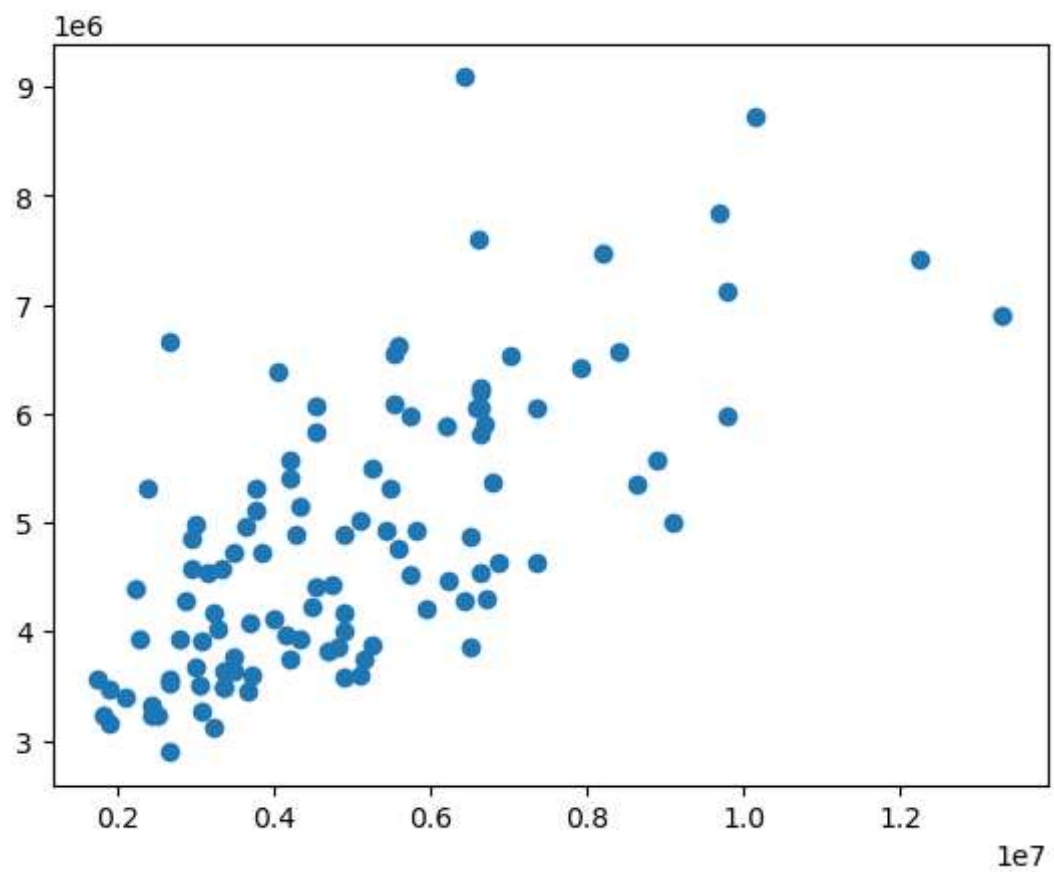
LinearRegression()

```
In [60]: print("[INFO] evaluating...")
         predictions = model.predict(x_test)
         print(predictions)
```

```
[INFO] evaluating...
[6383168.05981192 6230250.35140428 3597885.28090091 4289730.8386382
 3930445.60575177 4883933.33918115 5505773.16533075 6068793.48103629
 3321512.02483442 3495157.72744705 8731338.12527607 3561265.8244721
 3487335.97847431 3633344.35548029 3933900.2714526 6661080.95290716
 2893133.47793136 4635197.40872131 4583377.42320885 4274999.75826381
 4296640.17003986 4920207.32903988 3462807.85199841 3875170.95453847
 5412497.19140831 7425564.68389184 3269692.03932195 5021117.35611339
 7122556.71331971 3238600.04801447 5366583.9261965 3640253.68688195
 6057517.06636272 4847659.34932243 4572198.51082701 5573863.86824637
 4167435.67282878 4021179.33154444 3764621.65211187 5307730.62714319
 5319221.21748587 3456243.98716683 6202613.02579763 4013357.58257171
 4534099.68582614 4235271.10270425 6057517.06636272 4525560.52386579
 4983206.22693418 3238600.04801447 6567080.25723531 3238600.04801447
 4886475.58731091 4116997.55359665 4180093.95378268 3567705.7071645
 5971150.42384195 3391517.75642211 5150855.01571622 3514973.30408096
 4723103.68827084 4722288.7729915 4427005.04910037 3915714.52537737
 4414098.80386812 3861352.29173514 5904599.35795509 3598797.69847198
 5824514.82737829 4531557.43769638 4969297.37060092 4759565.37395123
 5104119.52646327 7612116.63173672 3129432.61186821 5816602.88487562
 3816441.63762434 3865621.87271532 4635197.40872131 4393370.80966313
 6542082.68205015 3971901.59416174 5884783.78132117 4866660.010677
 3149691.15736393 7466018.06120081 3529704.38445536 3754257.65500938
 6908277.24633825 7840034.37446164 3940907.10514598 5343403.87739169
 4084993.14471809 3740438.99220605 9091535.77573357 4200821.94798766
 4935974.80912452 5988423.7523461 4462464.12367976 6624994.6588701
 3681709.67529193 5573863.86824637 3588433.70136949 6547076.98477974
 4995395.05917882 5316678.96935611 6417714.71682023 6057517.06636272
 6092063.72337103]
```

```
In [62]: plt.scatter(y_test,predictions)
```

```
Out[62]: <matplotlib.collections.PathCollection at 0x239c57a5750>
```



```
In [63]: sns.distplot((y_test-predictions),bins=50)
```

C:\Users\computer\AppData\Local\Temp\ipykernel_15404\1061164399.py:1: UserWarning:

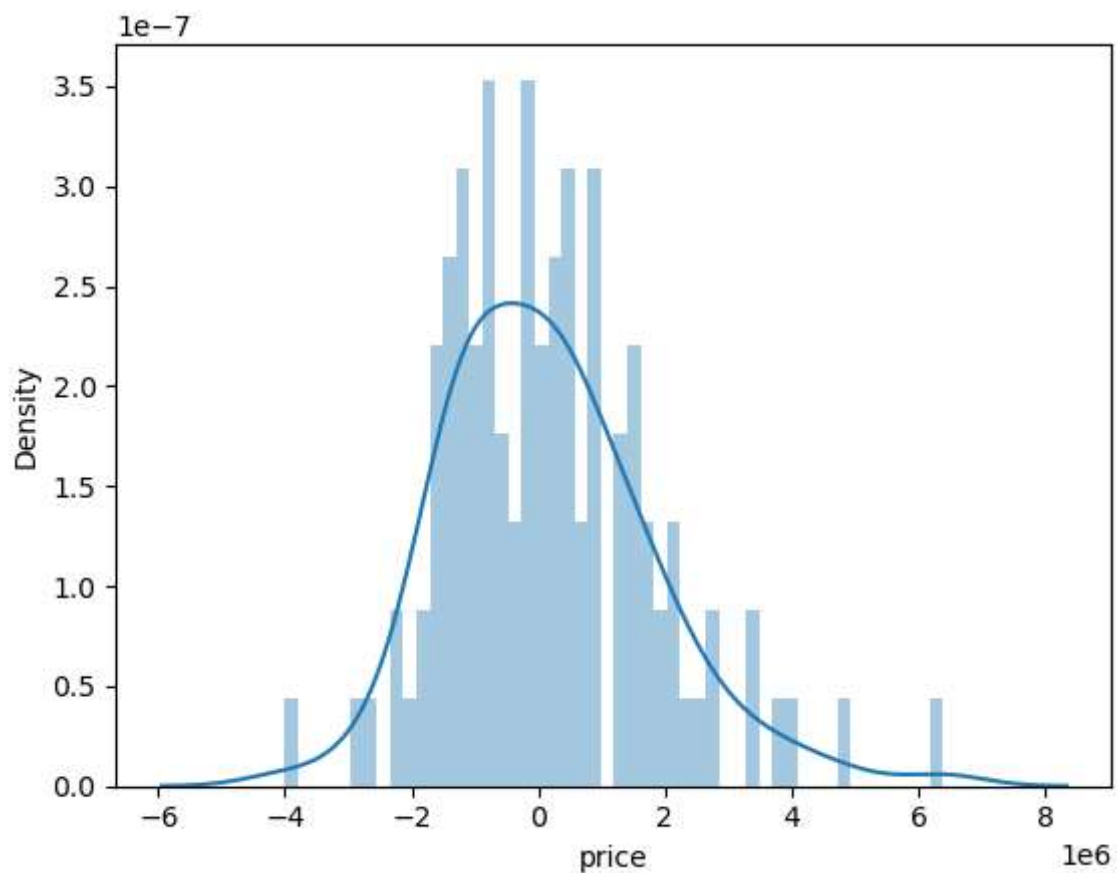
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751> (<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>)

```
sns.distplot((y_test-predictions),bins=50)
```

```
Out[63]: <Axes: xlabel='price', ylabel='Density'>
```



```
In [ ]:
```