

In [64]:

```
import pandas as pd
```

In [65]:

```
import numpy as np
```

In [66]:

```
import seaborn as sns
```

In [67]:

```
import matplotlib.pyplot as plt
```

```
%matplotlib inline
```

In [68]:

```
HouseDF =pd.read_csv("house.csv")
```

In [69]:

```
HouseDF.head()
```

Out[69]:

| | price | area | latitude | longitude | Bedrooms | Bathrooms | Price_sqft |
|---|----------|------|-----------|-----------|----------|-----------|-------------|
| 0 | 5600000 | 1350 | 28.608850 | 77.460560 | 3 | 3 | 4148.148148 |
| 1 | 8800000 | 1490 | 28.374236 | 76.952416 | 3 | 3 | 5906.040268 |
| 2 | 16500000 | 2385 | 28.645769 | 77.385110 | 4 | 5 | 6918.238994 |
| 3 | 3810000 | 1050 | 28.566914 | 77.436434 | 2 | 2 | 3628.571429 |
| 4 | 6200000 | 1350 | 28.520732 | 77.356491 | 2 | 2 | 4592.592593 |

In [70]:

```
HouseDF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7738 entries, 0 to 7737
Data columns (total 7 columns):
#   Column          Non-Null Count  Dtype
---  -
0   price           7738 non-null   int64
1   area            7738 non-null   int64
2   latitude        7738 non-null   float64
3   longitude       7738 non-null   float64
4   Bedrooms        7738 non-null   int64
5   Bathrooms       7738 non-null   int64
6   Price_sqft      7738 non-null   float64
dtypes: float64(3), int64(4)
memory usage: 423.3 KB
```

In [71]:

```
HouseDF
```

Out[71]:

| | price | area | latitude | longitude | Bedrooms | Bathrooms | Price_sqft |
|------|----------|------|-----------|-----------|----------|-----------|-------------|
| 0 | 5600000 | 1350 | 28.608850 | 77.460560 | 3 | 3 | 4148.148148 |
| 1 | 8800000 | 1490 | 28.374236 | 76.952416 | 3 | 3 | 5906.040268 |
| 2 | 16500000 | 2385 | 28.645769 | 77.385110 | 4 | 5 | 6918.238994 |
| 3 | 3810000 | 1050 | 28.566914 | 77.436434 | 2 | 2 | 3628.571429 |
| 4 | 6200000 | 1350 | 28.520732 | 77.356491 | 2 | 2 | 4592.592593 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 7733 | 7900000 | 1095 | 28.635272 | 77.370395 | 2 | 2 | 7214.611872 |
| 7734 | 4510000 | 1060 | 28.581431 | 77.452819 | 2 | 2 | 4254.716981 |
| 7735 | 7000000 | 1898 | 28.625850 | 77.435336 | 4 | 3 | 3688.092729 |
| 7736 | 6500000 | 1400 | 28.701622 | 77.430153 | 3 | 3 | 4642.857143 |
| 7737 | 6500000 | 1750 | 28.693590 | 77.344376 | 3 | 2 | 3714.285714 |

7738 rows × 7 columns

In [72]:

```
HouseDF.describe()
```

Out[72]:

| | price | area | latitude | longitude | Bedrooms | Bathrooms | Pr |
|-------|--------------|-------------|-------------|-------------|-------------|-------------|-------|
| count | 7.738000e+03 | 7738.000000 | 7738.000000 | 7738.000000 | 7738.000000 | 7738.000000 | 7738 |
| mean | 8.320635e+06 | 1409.506591 | 28.552092 | 77.273476 | 2.708193 | 2.501163 | 5543 |
| std | 7.223197e+06 | 718.929581 | 0.107420 | 0.180606 | 0.877026 | 0.867050 | 2408 |
| min | 1.700000e+06 | 501.000000 | 28.240023 | 76.884101 | 2.000000 | 2.000000 | 2100 |
| 25% | 4.200000e+06 | 990.000000 | 28.455539 | 77.078590 | 2.000000 | 2.000000 | 3950 |
| 50% | 6.000000e+06 | 1250.000000 | 28.574637 | 77.345320 | 3.000000 | 2.000000 | 4972 |
| 75% | 9.500000e+06 | 1650.000000 | 28.642520 | 77.421054 | 3.000000 | 3.000000 | 6350 |
| max | 8.500000e+07 | 9500.000000 | 28.799748 | 77.688028 | 10.000000 | 10.000000 | 44378 |

In [73]:

```
HouseDF.columns
```

Out[73]:

```
Index(['price', 'area', 'latitude', 'longitude', 'Bedrooms', 'Bathrooms',  
      'Price_sqft'],  
      dtype='object')
```

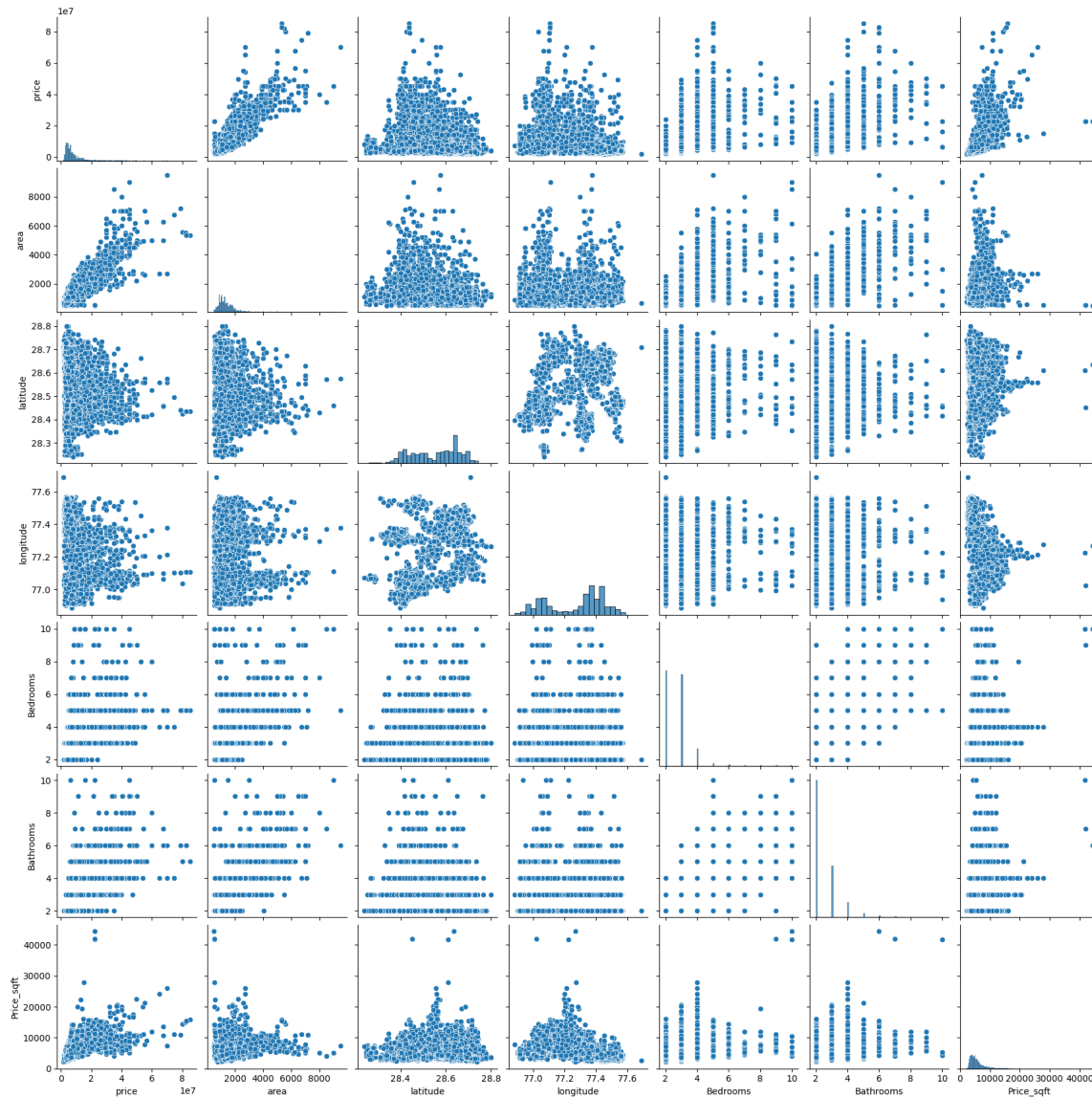
In [74]:

```
sns.pairplot(HouseDF)
```

C:\Users\DIVYA\AppData\Local\Programs\Python\Python39\lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to tight
self._figure.tight_layout(*args, **kwargs)

Out[74]:

<seaborn.axisgrid.PairGrid at 0x1c007ec6a60>

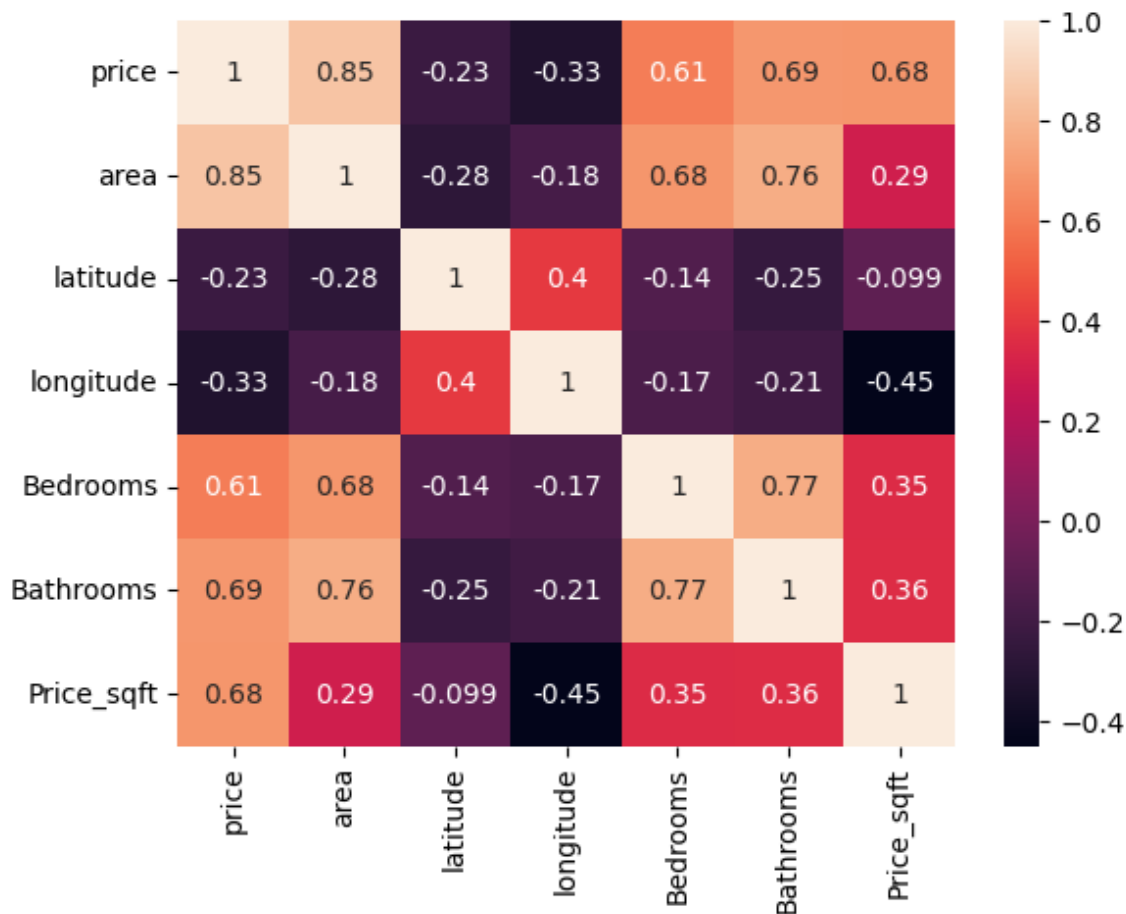


In [75]:

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

Out[75]:

<Axes: >



In [76]:

```
X=HouseDF[['price', 'area', 'latitude', 'longitude', 'Bedrooms', 'Bathrooms',  
           'Price_sqft']]
```

```
y=HouseDF['price']
```

In [56]:

```
from sklearn.model_selection import train_test_split
```

In [77]:

```
X_train, X_test, y_train, y_test = train_test_split(  
    ...    X, y, test_size=0.40, random_state=101)
```

In [78]:

```
from sklearn.linear_model import LinearRegression
```

In [79]:

```
lm = LinearRegression()
```

In [80]:

```
lm.fit (X_train, y_train)
```

Out[80]:

```
▼ LinearRegression  
LinearRegression()
```

In [81]:

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

In [82]:

```
coeff_df
```

Out[82]:

| | Coefficient |
|-------------------|---------------|
| price | 1.000000e+00 |
| area | -1.083464e-12 |
| latitude | 1.627659e-09 |
| longitude | 8.644755e-10 |
| Bedrooms | -1.515772e-10 |
| Bathrooms | 2.954013e-10 |
| Price_sqft | -2.356993e-13 |

In [84]:

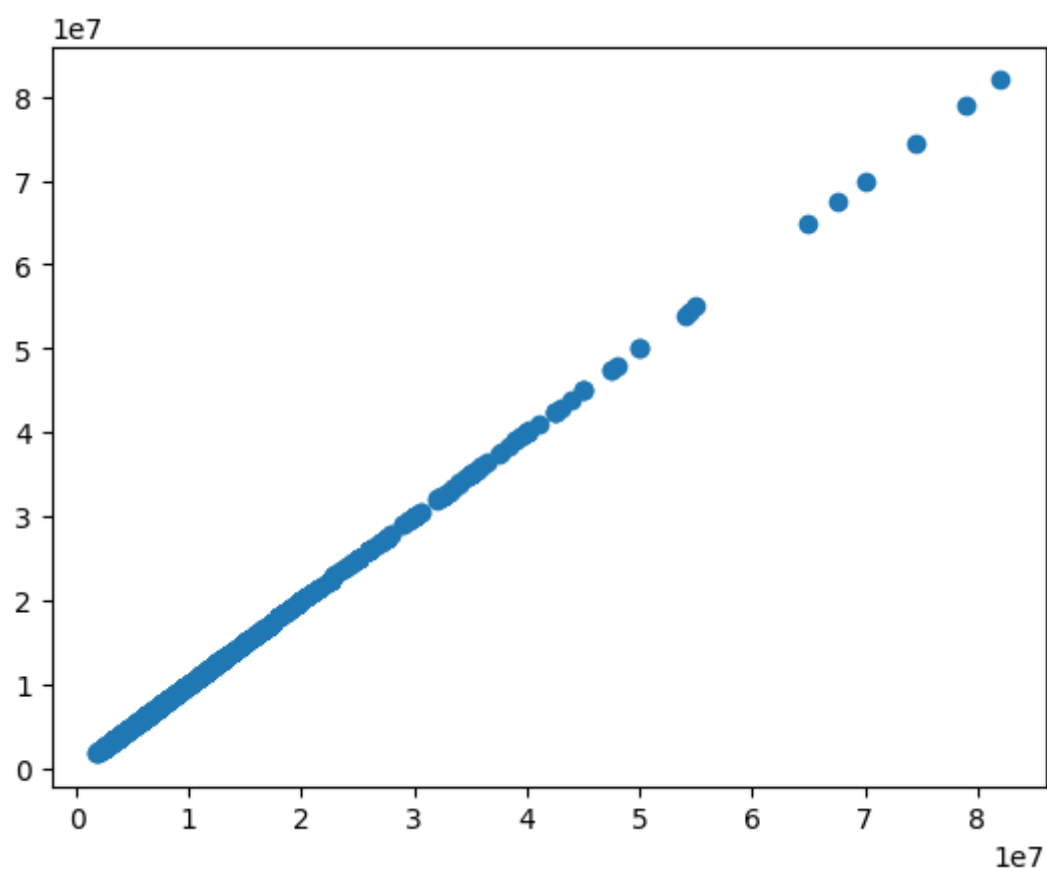
```
predictions = lm.predict (X_test)
```

In [85]:

```
plt.scatter (y_test, predictions)
```

Out[85]:

<matplotlib.collections.PathCollection at 0x1c00b4a6d90>



In [86]:

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

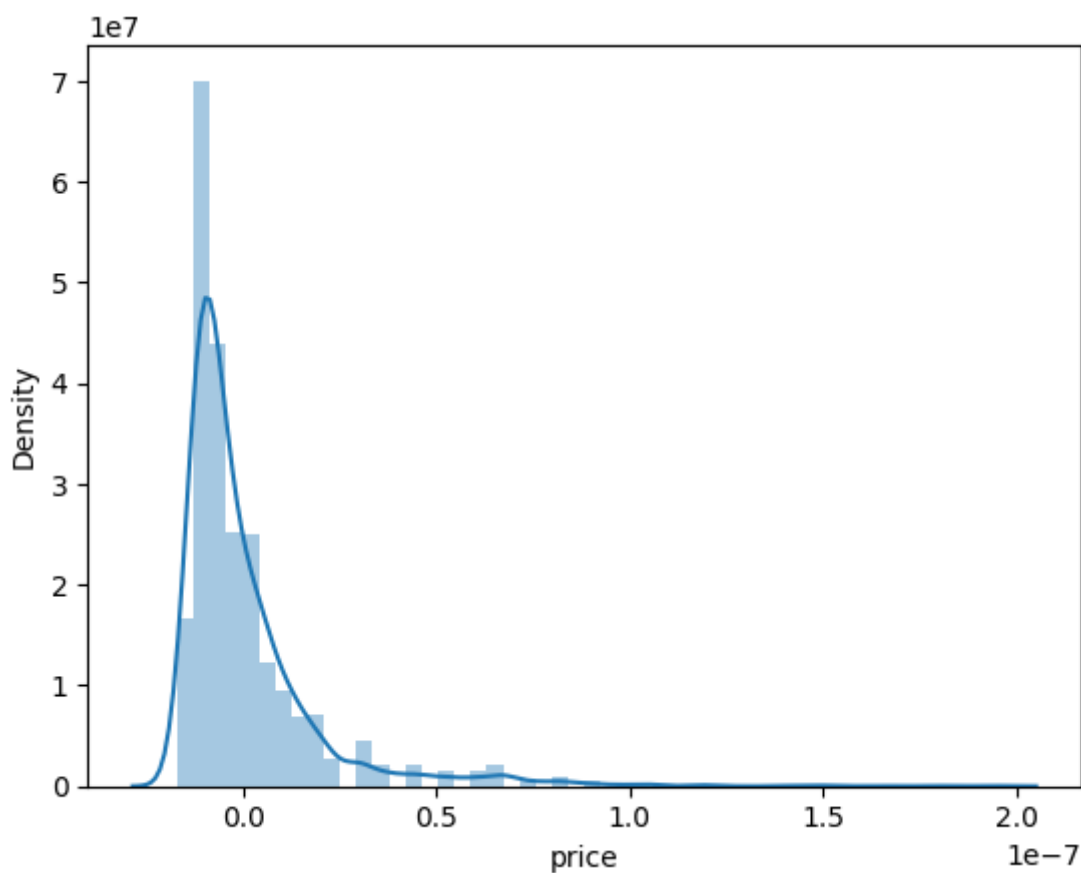
C:\Users\DIVYA\AppData\Local\Temp\ipykernel_6564\1326397652.py:1: UserWarning:

`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);
```



In []: