In [27]:
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import metrics
```

In [40]:
```
data=pd.read_csv("Housing.csv")
```

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

Out[39]:

| | price | area | bedrooms | bathrooms | stories | mainroad | guestroom | basement | hotwaterhea |
|---|---|---|---|---|---|---|---|---|---|
| **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 [42]:
```
data.describe()
```

Out[42]:

| | 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 [43]:
```
data.shape
```

Out[43]:
```
(545, 13)
```

In [45]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 545 entries, 0 to 544
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   price             545 non-null    int64 
 1   area              545 non-null    int64 
 2   bedrooms          545 non-null    int64 
 3   bathrooms         545 non-null    int64 
 4   stories           545 non-null    int64 
 5   mainroad          545 non-null    object
 6   guestroom         545 non-null    object
 7   basement          545 non-null    object
 8   hotwaterheating   545 non-null    object
 9   airconditioning   545 non-null    object
 10  parking           545 non-null    int64 
 11  prefarea          545 non-null    object
 12  furnishingstatus  545 non-null    object
dtypes: int64(6), object(7)
memory usage: 55.5+ KB
```

In [46]:
```
data.isnull().sum()
```

Out[46]:
```
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 [48]:
```
data['bathrooms']=data['bathrooms'].fillna(data['bathrooms'].median())
```

In [49]:
```
data.describe()
```

Out[49]:

| | 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 [ ]:

In [ ]:

In [29]:
```
#dataframe
df = pd.DataFrame(data)
```

In [31]:
```
# Split the data into features (X) and target variable (y)
X = df[['area', 'bedrooms', 'bathrooms']]
y = df['price']
```

In [32]:
```
# Split the data into a training set and a testing set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, rando
```

In [33]:
```
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[33]:
```
LinearRegression()
```

In [51]:
```
coeff_data= pd.DataFrame(model.coef_,X.columns,columns=['Coefficient'])
```

In [52]:
```
coeff_data
```

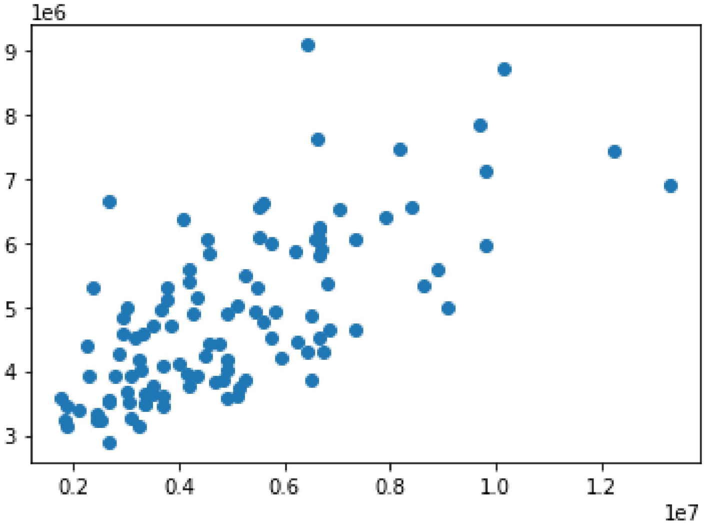
Out[52]:

| | Coefficient |
|---|---|
| **area** | 3.454666e+02 |
| **bedrooms** | 3.601977e+05 |
| **bathrooms** | 1.422320e+06 |

In [55]:
```
prediction = model.predict(X_test)
```

In [56]:
```
plt.scatter(y_test, prediction)
```

Out[56]:
```
<matplotlib.collections.PathCollection at 0x1df7e42beb0>
```



In [35]:
```

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse:.2f}")
print(f"R-squared: {r2:.2f}")
```

```
Mean Squared Error: 2750040479309.05
R-squared: 0.46
```

In [38]:
```
#prediction for a new house
new_house = np.array([[7420, 4, 2]])  # Replace with the features of the new h
predicted_price = model.predict(new_house)
print(f"Predicted Price for the New House: Rupees {predicted_price[0]:,.2f}")
```

```
Predicted Price for the New House: Rupees 6,908,277.25

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning:
X does not have valid feature names, but LinearRegression was fitted with fea
ture names
  warnings.warn(
```