In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sb
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

In [2]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [3]:
```
data = pd.read_csv("Housing.csv")
print(data.head())
```

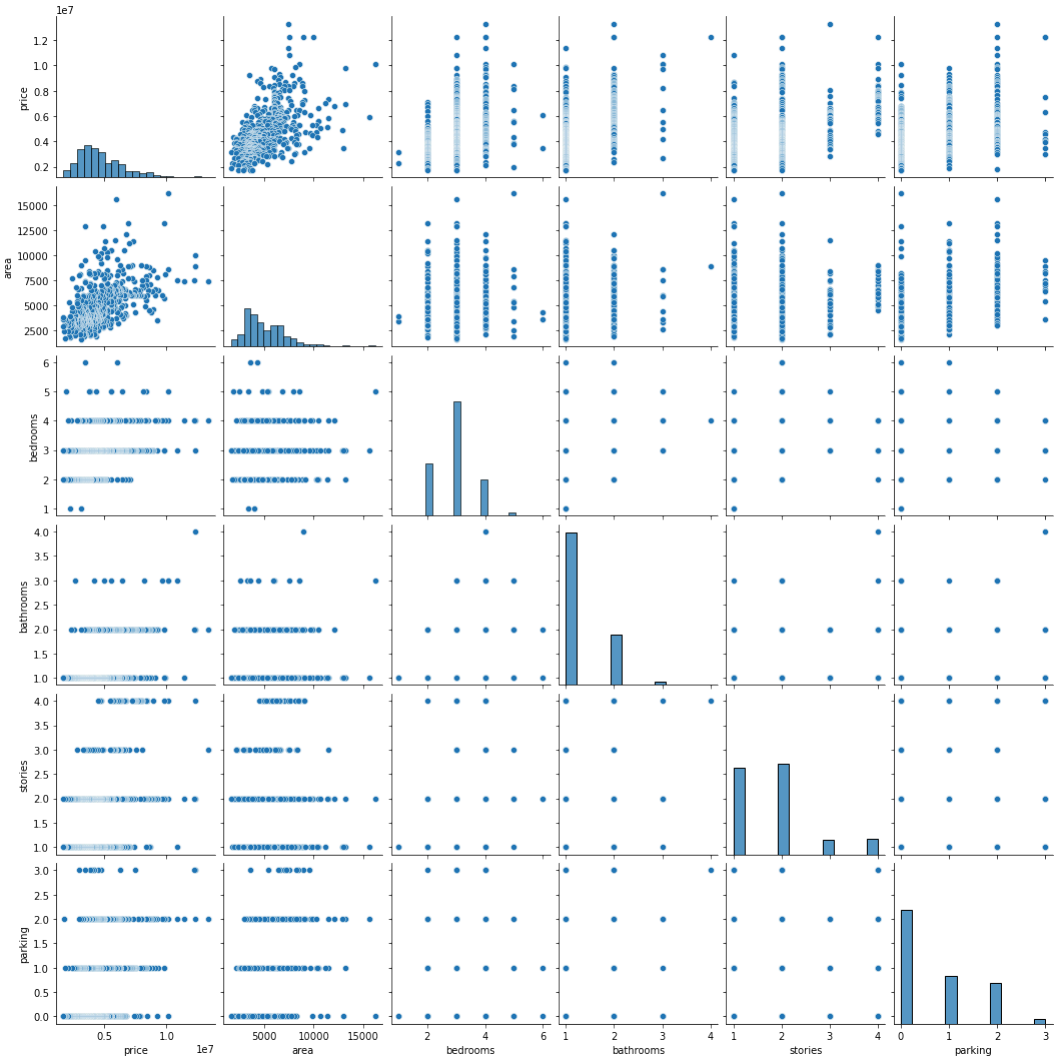
```
      price  area  bedrooms  bathrooms  stories mainroad guestroom basement  \
0  13300000  7420         4          2        3      yes        no       no
1  12250000  8960         4          4        4      yes        no       no
2  12250000  9960         3          2        2      yes        no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

  hotwaterheating airconditioning  parking prefarea furnishingstatus
0              no             yes        2      yes        furnished
1              no             yes        3       no        furnished
2              no              no        2      yes   semi-furnished
3              no             yes        3      yes        furnished
4              no             yes        2       no        furnished
```

In [4]:
```
print(data.isnull().sum())
```

```
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 [5]:
```
sb.pairplot(data)
plt.show()
```



In [6]:
```
#Encoding Categorical Variables
data = pd.get_dummies(data, drop_first=True)
```

In [7]:
```
X = data.drop('price', axis=1)
y = data['price']
```

In [8]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print(f"Training set size: {X_train.shape}")
print(f"Testing set size: {X_test.shape}")
```

```
Training set size: (436, 13)
Testing set size: (109, 13)
```

In [9]:
```
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[9]:
```
▾ LinearRegression
LinearRegression()
```

In [10]:
```
print(f"Model coefficients: {model.coef_}")
print(f"Model intercept: {model.intercept_}")
```

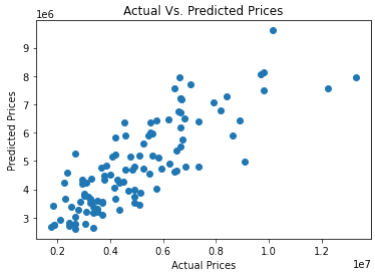
```
Model coefficients: [ 2.35968805e+02  7.67787016e+04  1.09444479e+06  4.07476595e+05
  2.24841913e+05  3.67919948e+05  2.31610037e+05  3.90251176e+05
  6.84649885e+05  7.91426736e+05  6.29890565e+05 -1.26881818e+05
 -4.13645062e+05]
Model intercept: 260032.35760741495
```

In [11]:
```
y_pred = model.predict(X_test)
```

In [12]:
```
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"Mean Squared Error: {mse}")
print(f"R-squared value: {r2}")
```

```
Mean Squared Error: 1754318687330.6643
R-squared value: 0.6529242642153184
```

In [13]:
```
plt.scatter(y_test, y_pred)
plt.xlabel("Actual Prices")
plt.ylabel("Predicted Prices")
plt.title("Actual Vs. Predicted Prices")
plt.show()
```



In [ ]: