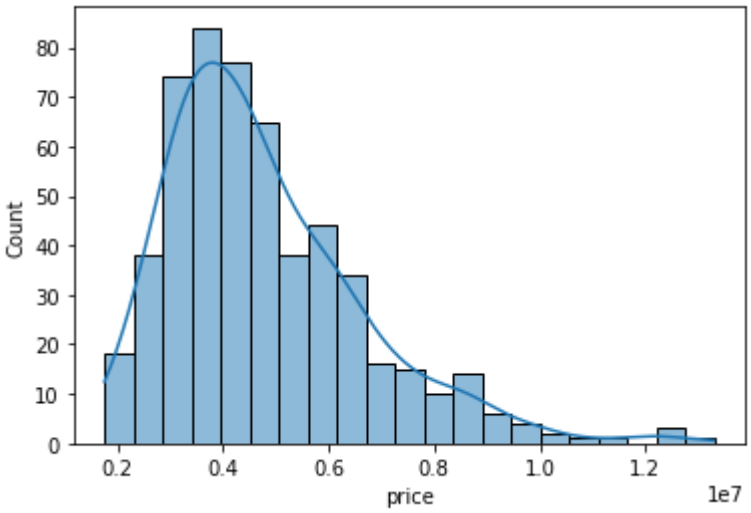
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Step 2: Load the dataset
df = pd.read_csv("C:/Users/91950/Downloads/Housing.csv")
```

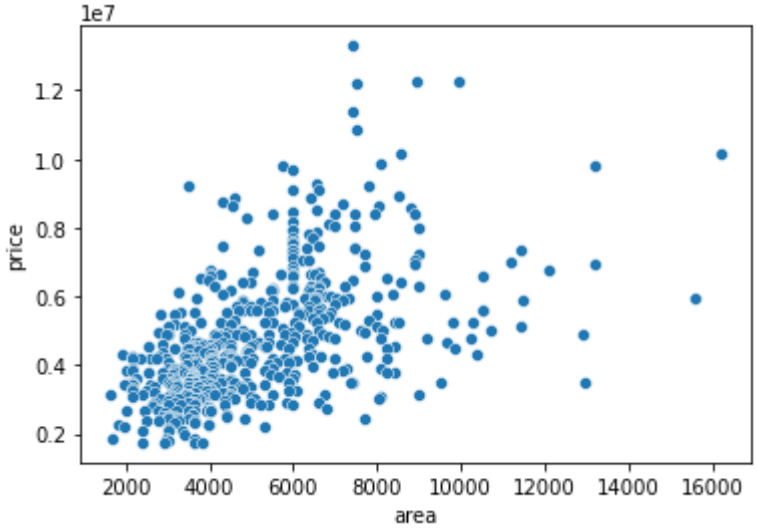
In [4]:

```
# Step 3: Perform visualizations
# Univariate Analysis
sns.histplot(df['price'], kde=True)
plt.show()
```



In [5]:

```
# Bi-Variate Analysis
sns.scatterplot(x='area', y='price', data=df)
plt.show()
```



In [6]:

```
# Multi-Variate Analysis
sns.pairplot(df)
plt.show()
```


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

           parking
count   545.000000
mean      0.693578
std       0.861586
```

In [7]:

```
# Step 4: Perform descriptive statistics
print(df.describe())
```

```
min      0.000000
25%      0.000000
50%      0.000000
75%      1.000000
max      3.000000  In
```

[8]:

```
# Step 5: Check for Missing values and deal with them
print(df.isnull().sum())
```

```
price               0
area                0
bedrooms            0
bathrooms           0
stories             0
mainroad            0
guestroom           0
basement            0
hotwaterheating     0
airconditioning     0
parking             0
furnishingstatus    0
dtype: int64 In [10]:
```

```


# Step 6: Find and replace outliers
# Identify outliers using statistical methods (e.g., Z-score, IQR)
# Replace outliers with appropriate values (e.g., mean, median, trimmed mean)

# Step 7: Check for Categorical columns and perform encoding
categorical_columns = ['mainroad', 'guestroom', 'basement', 'hotwaterheating', 'aircondit
df_encoded = pd.get_dummies(df, columns=categorical_columns, drop_first=True)
```

In [11]:

```
# Step 8: Split the data into dependent and independent variables
X = df_encoded.drop('price', axis=1)
y = df_encoded['price']
```

In [12]:

```
# Step 9: Scale the independent variables
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

In [13]:

```
# Step 10: Split the data into training and testing
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_st
```

In [14]:

```
# Step 11: Build the model
model = LinearRegression()
```

In [15]:

```
# Step 12: Train the model
model.fit(X_train, y_train)
```

Out[15]:

```
LinearRegression()
```

In [16]:

```
# Step 13: Test the model
y_pred = model.predict(X_test)
```

In [18]:

In

```
# Step 14: Measure the performance using metrics
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)


print('Mean Squared Error:', mse)
print('R-squared:', r2)
```

```
Mean Squared Error: 1837637189871.7068
R-squared: 0.6364404686639462
```