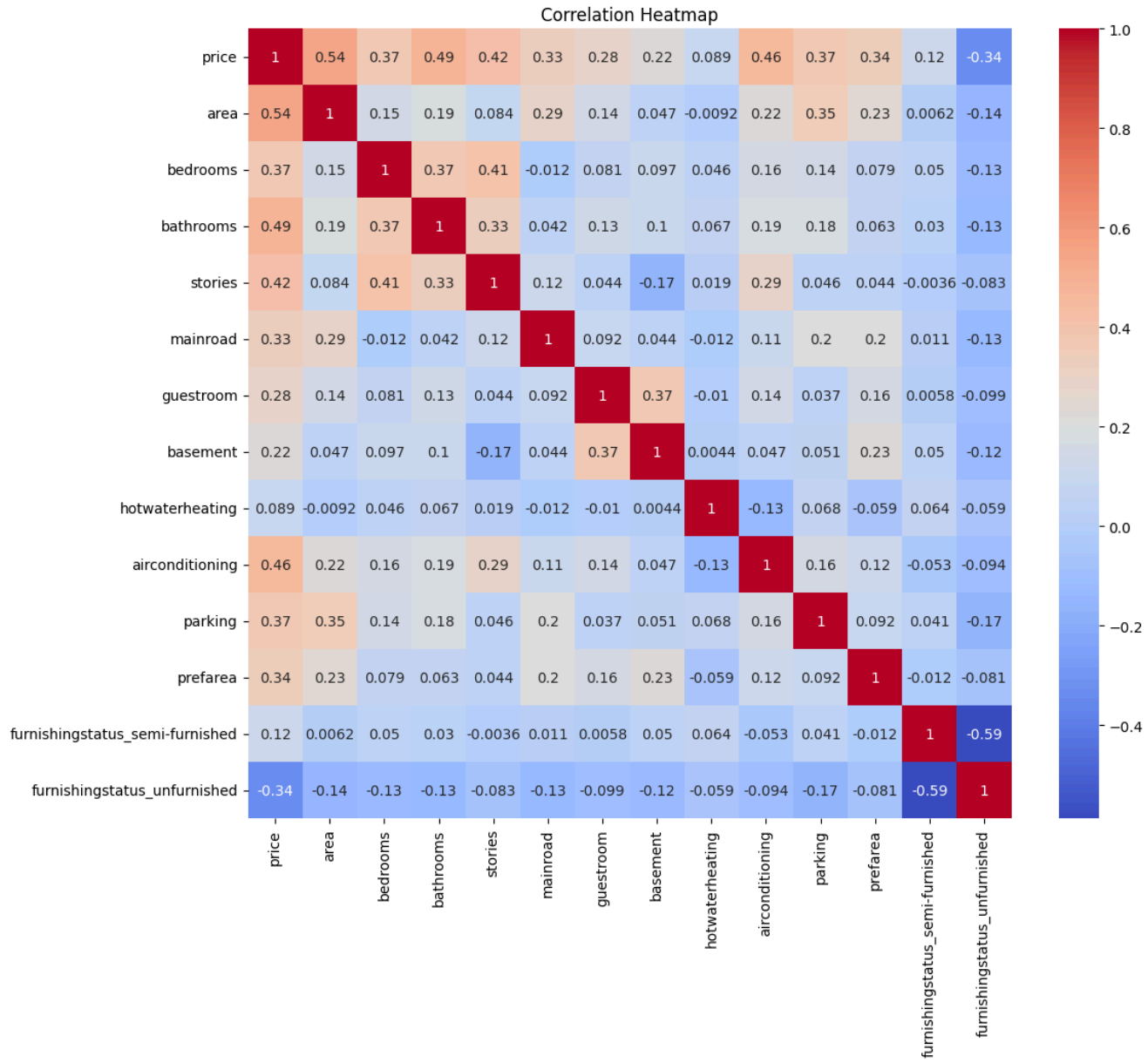
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score


df=pd.read_csv('/bin/Housing.csv')
print(df.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
```

```
binary_map = {'yes': 1, 'no': 0}
df['mainroad'] = df['mainroad'].map(binary_map)
df['guestroom'] = df['guestroom'].map(binary_map)
df['basement'] = df['basement'].map(binary_map)
df['hotwaterheating'] = df['hotwaterheating'].map(binary_map)
df['airconditioning'] = df['airconditioning'].map(binary_map)
df['prefarea'] = df['prefarea'].map(binary_map)
df = pd.get_dummies(df, columns=['furnishingstatus'], drop_first=True)
df['price'] = np.log1p(df['price'])


plt.figure(figsize=(12, 10))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```




```
X = df.drop('price', axis=1)
y = df['price']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
lr = LinearRegression()
lr.fit(X_train, y_train)
```

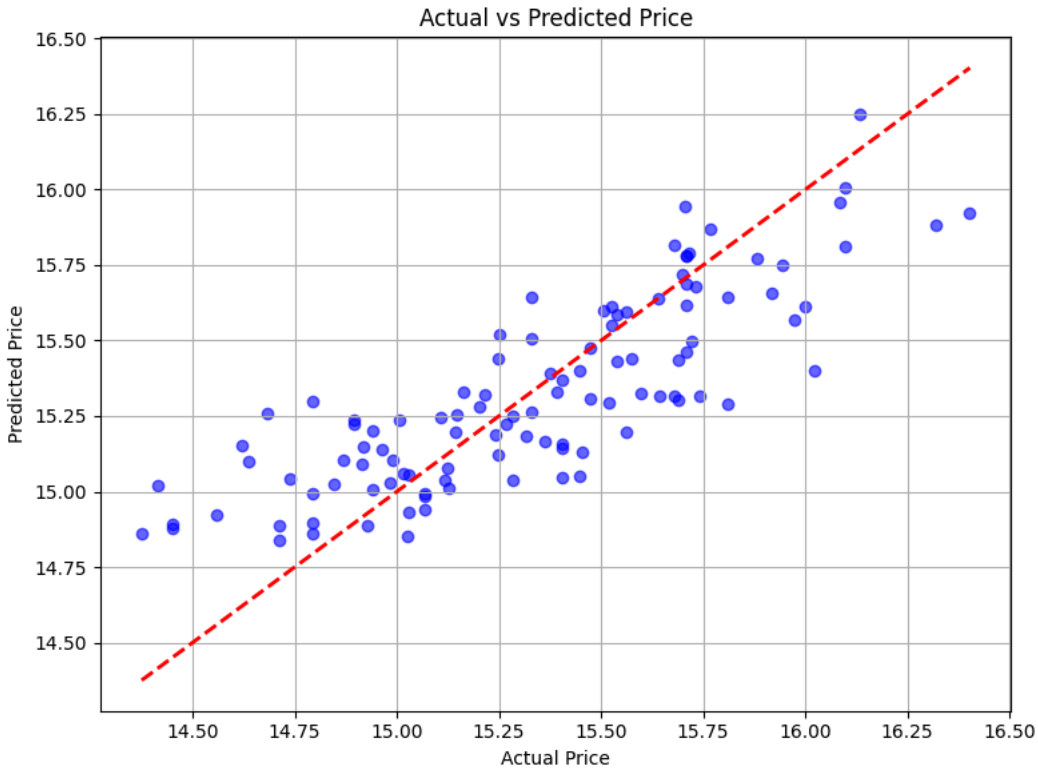
```
▾ LinearRegression
LinearRegression()
```

```
y_pred = lr.predict(X_test)
```

```
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print("\nEvaluation Metrics:")
print(f"Mean Absolute Error (MAE): {mae:.2f}")
print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"R² Score: {r2:.4f}")
```

```
Evaluation Metrics:
Mean Absolute Error (MAE): 0.20
Mean Squared Error (MSE): 0.06
R² Score: 0.6722
```

```
plt.figure(figsize=(8,6))
plt.scatter(y_test, y_pred, alpha=0.6, color='blue')
plt.plot([y.min(), y.max()], [y.min(), y.max()], 'r--', lw=2)
plt.xlabel('Actual Price')
plt.ylabel('Predicted Price')
plt.title('Actual vs Predicted Price')
plt.grid(True)
plt.tight_layout()
plt.show()
```




```
coeff_df = pd.DataFrame(lr.coef_, index=X.columns, columns=['Coefficient'])
print("\nModel Coefficients:\n", coeff_df)
```