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

```
housing = pd.read_csv('Housing.csv')
```

```
housing
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking
0	13300000	7420	4	2	3	yes	no	no	no	yes	2
1	12250000	8960	4	4	4	yes	no	no	no	yes	3
2	12250000	9960	3	2	2	yes	no	yes	no	no	2
3	12215000	7500	4	2	2	yes	no	yes	no	yes	3
4	11410000	7420	4	1	2	yes	yes	yes	no	yes	2
...	...	...	...	...	...	...	...	...	...	...	...
540	1820000	3000	2	1	1	yes	no	yes	no	no	2
541	1767150	2400	3	1	1	no	no	no	no	no	0
542	1750000	3620	2	1	1	yes	no	no	no	no	0
543	1750000	2910	3	1	1	no	no	no	no	no	0
544	1750000	3850	3	1	2	yes	no	no	no	no	0

545 rows × 13 columns

```
housing.isnull().sum()
```

	0
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

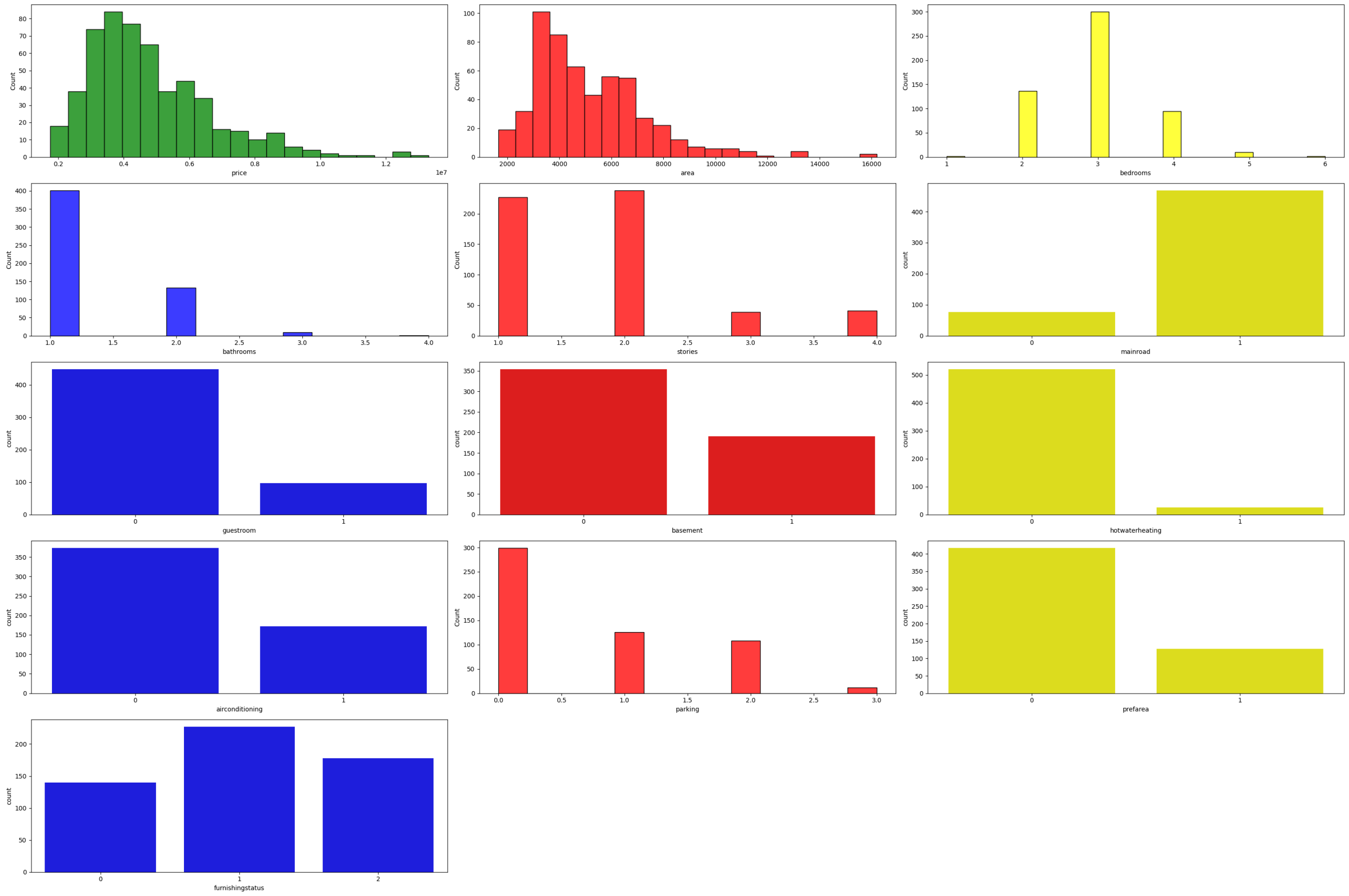
dtype: int64

```
housing.duplicated().sum()
```

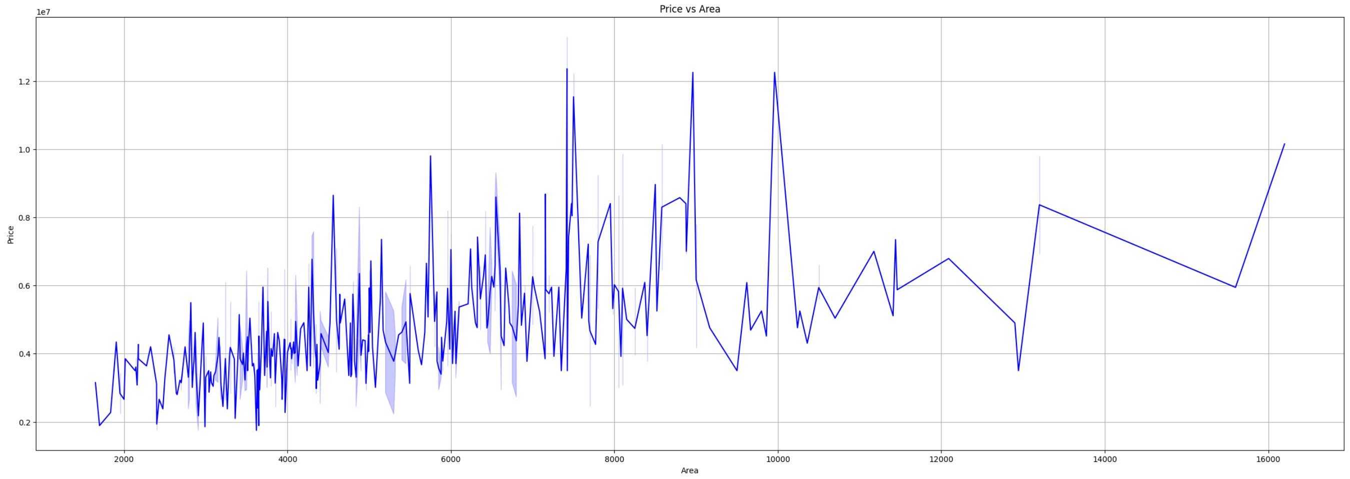
```
np.int64(0)
```

```
housing.columns = housing.columns.str.strip()
```

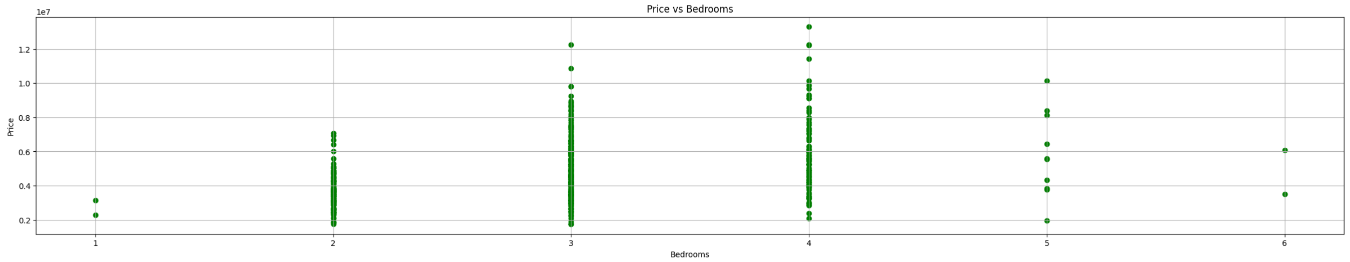
```
# Set up the figure
fig, axes = plt.subplots(nrows=5, ncols=3, figsize=(30, 20))
axes = axes.flatten()

sns.histplot(housing['price'], ax=axes[0],color='green')
sns.histplot(housing['area'], ax=axes[1],color='red')

sns.histplot(housing['bedrooms'], ax=axes[2],color='yellow')

sns.histplot(housing['bathrooms'], ax=axes[3],color='blue')

sns.histplot(housing['stories'], ax=axes[4],color='red')

sns.countplot(x=housing['mainroad'], ax=axes[5],color='yellow')


sns.countplot(x=housing['guestroom'], ax=axes[6],color='blue')


sns.countplot(x=housing['basement'], ax=axes[7],color='red')


sns.countplot(x=housing['hotwaterheating'], ax=axes[8],color='yellow')

sns.countplot(x=housing['airconditioning'], ax=axes[9],color='blue')

sns.histplot(housing['parking'], ax=axes[10],color="red")

sns.countplot(x=housing['prefarea'], ax=axes[11],color="yellow")


sns.countplot(x=housing['furnishingstatus'], ax=axes[12],color="blue")

for j in range(13, len(axes)):
    fig.delaxes(axes[j])
plt.tight_layout()
plt.show()
```



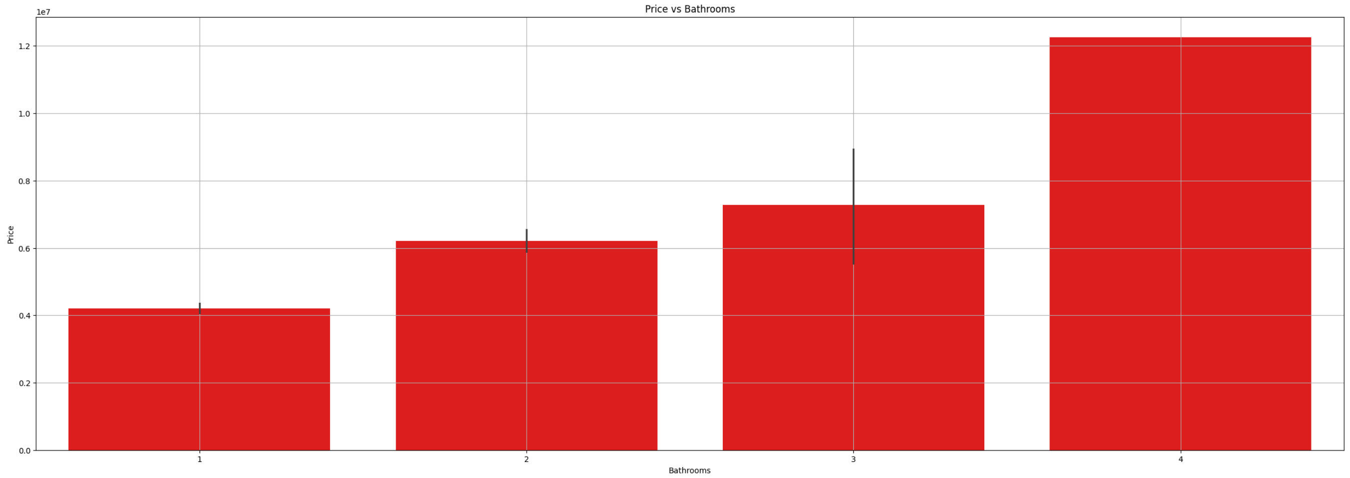
```
plt.figure(figsize=(30, 10))
sns.lineplot(x=housing['area'], y=housing['price'], color='blue')
plt.title('Price vs Area')
plt.xlabel('Area')
plt.ylabel('Price')
plt.grid(True)
plt.show()
```



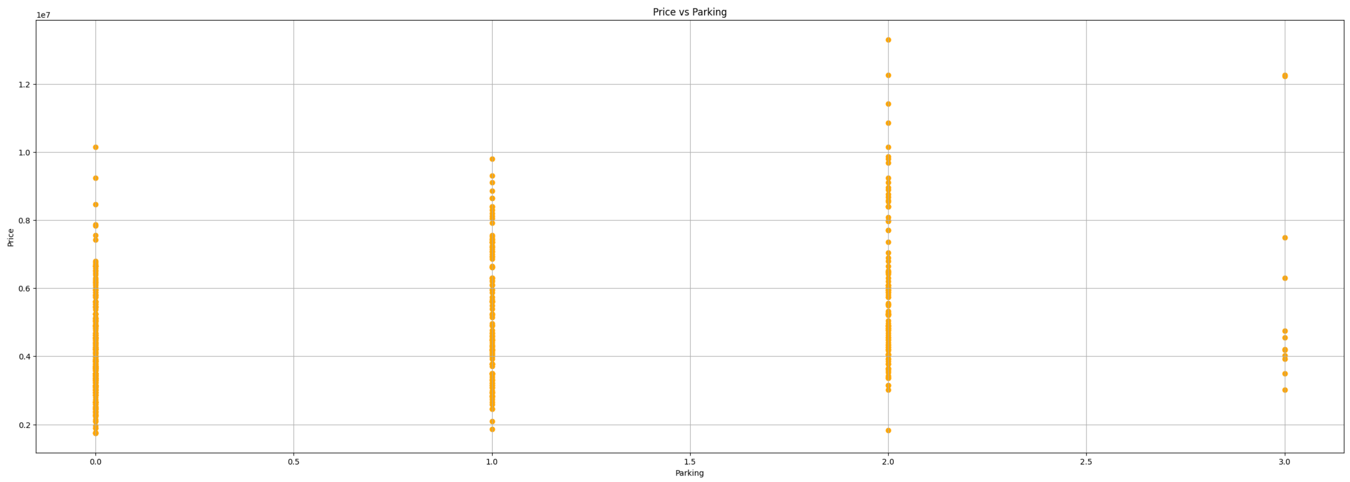
```
plt.figure(figsize=(30, 5))
plt.scatter(housing['bedrooms'], housing['price'], color='green')
plt.title('Price vs Bedrooms')
plt.xlabel('Bedrooms')
plt.ylabel('Price')
plt.grid(True)
plt.show()
```



```
plt.figure(figsize=(30, 10))
sns.barplot(x=housing['bathrooms'], y=housing['price'], color='red')
plt.title('Price vs Bathrooms')
plt.xlabel('Bathrooms')
plt.ylabel('Price')
plt.grid(True)
plt.show()
```



```
plt.figure(figsize=(30, 10))
plt.scatter(housing['parking'], housing['price'], color='orange')
plt.title('Price vs Parking')
plt.xlabel('Parking')
plt.ylabel('Price')
plt.grid(True)
plt.show()
```



```
#Encoding categorical features
le = LabelEncoder()
housing['mainroad'] = le.fit_transform(housing['mainroad'])
housing['guestroom'] = le.fit_transform(housing['guestroom'])
housing['basement'] = le.fit_transform(housing['basement'])
housing['hotwaterheating'] = le.fit_transform(housing['hotwaterheating'])
housing['airconditioning'] = le.fit_transform(housing['airconditioning'])
housing['prefarea'] = le.fit_transform(housing['prefarea'])
housing['furnishingstatus'] = le.fit_transform(housing['furnishingstatus'])
#creating new features
housing['total_rooms'] = housing['bedrooms'] + housing['bathrooms']
housing['price_per_room'] = housing['price'] / housing['total_rooms']
housing['price_per_area'] = housing['price'] / housing['area']
housing['price_per_bathroom'] = housing['price'] / housing['bathrooms']
housing['area_bedrooms'] = housing['area'] * housing['bedrooms']
housing['area_bathrooms'] = housing['area'] * housing['bathrooms']
housing['area_furnishingstatus'] = housing['area'] * housing['furnishingstatus']
```

```
# deal with skewed price and area
housing['log_price'] = np.log(housing['price'])
housing['log_area'] = np.log(housing['area'])
```

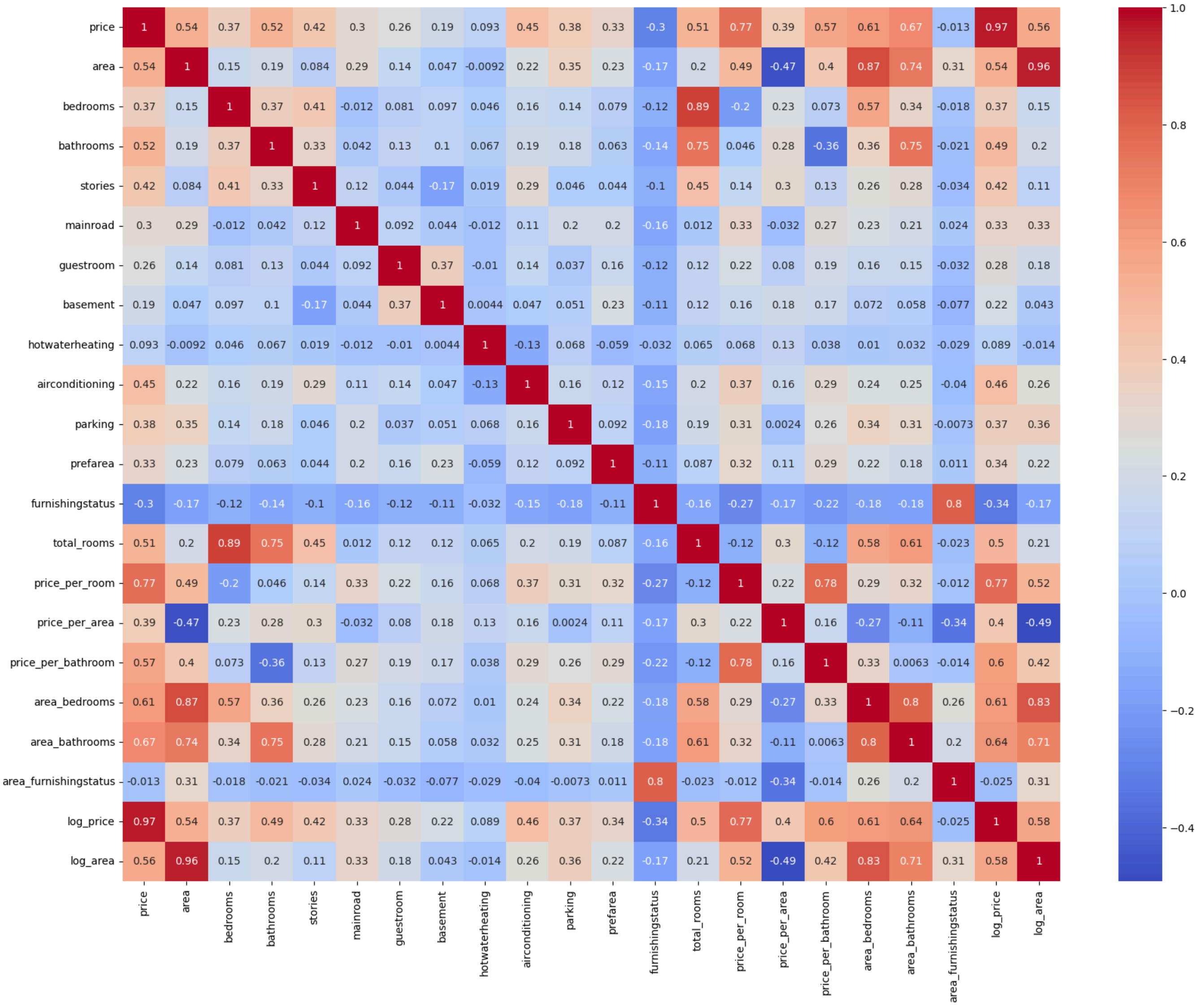
```
housing
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	...	fur
0	13300000	7420	4	2	3	1	0	0	0	1	...	
1	12250000	8960	4	4	4	1	0	0	0	1	...	
2	12250000	9960	3	2	2	1	0	1	0	0	...	
3	12215000	7500	4	2	2	1	0	1	0	1	...	
4	11410000	7420	4	1	2	1	1	1	0	1	...	
...	...	...	...	...	...	...	...	...	...	...	...	
540	1820000	3000	2	1	1	1	0	1	0	0	...	
541	1767150	2400	3	1	1	0	0	0	0	0	...	
542	1750000	3620	2	1	1	1	0	0	0	0	...	
543	1750000	2910	3	1	1	0	0	0	0	0	...	
544	1750000	3850	3	1	2	1	0	0	0	0	...	

545 rows × 22 columns

```
correlation = housing.corr()['price'].sort_values(ascending=False)
print(correlation)
fig, ax = plt.subplots(figsize=(20, 15))
sns.heatmap(housing.corr(), annot=True, cmap='coolwarm')
plt.show()
```

```
price                    1.000000
log_price                0.969010
price_per_room           0.768193
area_bathrooms           0.674764
area_bedrooms            0.611196
price_per_bathroom       0.570693
log_area                 0.561231
area                     0.535997
bathrooms                0.517545
total_rooms              0.511809
airconditioning          0.452954
stories                  0.420712
price_per_area           0.392921
parking                  0.384394
bedrooms                 0.366494
prefarea                 0.329777
mainroad                 0.296898
guestroom                0.255517
basement                 0.187057
hotwaterheating          0.093073
area_furnishingstatus   -0.013093
furnishingstatus        -0.304721
Name: price, dtype: float64
```



	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus	total_rooms	price_per_room	price_per_area	price_per_bathroom	area_bedrooms	area_bathrooms	area_furnishingstatus	log_price	log_area
price	1	0.54	0.37	0.52	0.42	0.3	0.26	0.19	0.093	0.45	0.38	0.33	-0.3	0.51	0.77	0.39	0.57	0.61	0.67	-0.013	0.97	0.56
area	0.54	1	0.15	0.19	0.084	0.29	0.14	0.047	-0.0092	0.22	0.35	0.23	-0.17	0.2	0.49	-0.47	0.4	0.87	0.74	0.31	0.54	0.96
bedrooms	0.37	0.15	1	0.37	0.41	-0.012	0.081	0.097	0.046	0.16	0.14	0.079	-0.12	0.89	-0.2	0.23	0.073	0.57	0.34	-0.018	0.37	0.15
bathrooms	0.52	0.19	0.37	1	0.33	0.042	0.13	0.1	0.067	0.19	0.18	0.063	-0.14	0.75	0.046	0.28	-0.36	0.36	0.75	-0.021	0.49	0.2
stories	0.42	0.084	0.41	0.33	1	0.12	0.044	-0.17	0.019	0.29	0.046	0.044	-0.1	0.45	0.14	0.3	0.13	0.26	0.28	-0.034	0.42	0.11
mainroad	0.3	0.29	-0.012	0.042	0.12	1	0.092	0.044	-0.012	0.11	0.2	0.2	-0.16	0.012	0.33	-0.032	0.27	0.23	0.21	0.024	0.33	0.33
guestroom	0.26	0.14	0.081	0.13	0.044	0.092	1	0.37	-0.01	0.14	0.037	0.16	-0.12	0.12	0.22	0.08	0.19	0.16	0.15	-0.032	0.28	0.18
basement	0.19	0.047	0.097	0.1	-0.17	0.044	0.37	1	0.0044	0.047	0.051	0.23	-0.11	0.12	0.16	0.18	0.17	0.072	0.058	-0.077	0.22	0.043
hotwaterheating	0.093	-0.0092	0.046	0.067	0.019	-0.012	-0.01	0.0044	1	-0.13	0.068	-0.059	-0.032	0.065	0.068	0.13	0.038	0.01	0.032	-0.029	0.089	-0.014
airconditioning	0.45	0.22	0.16	0.19	0.29	0.11	0.14	0.047	-0.13	1	0.16	0.12	-0.15	0.2	0.37	0.16	0.29	0.24	0.25	-0.04	0.46	0.26
parking	0.38	0.35	0.14	0.18	0.046	0.2	0.037	0.051	0.068	0.16	1	0.092	-0.18	0.19	0.31	0.0024	0.26	0.34	0.31	-0.0073	0.37	0.36
prefarea	0.33	0.23	0.079	0.063	0.044	0.2	0.16	0.23	-0.059	0.12	0.092	1	-0.11	0.087	0.32	0.11	0.29	0.22	0.18	0.011	0.34	0.22
furnishingstatus	-0.3	-0.17	-0.12	-0.14	-0.1	-0.16	-0.12	-0.11	-0.032	-0.15	-0.18	-0.11	1	-0.16	-0.27	-0.17	-0.22	-0.18	-0.18	0.8	-0.34	-0.17
total_rooms	0.51	0.2	0.89	0.75	0.45	0.012	0.12	0.12	0.065	0.2	0.19	0.087	-0.16	1	-0.12	0.3	-0.12	0.58	0.61	-0.023	0.5	0.21
price_per_room	0.77	0.49	-0.2	0.046	0.14	0.33	0.22	0.16	0.068	0.37	0.31	0.32	-0.27	-0.12	1	0.22	0.78	0.29	0.32	-0.012	0.77	0.52
price_per_area	0.39	-0.47	0.23	0.28	0.3	-0.032	0.08	0.18	0.13	0.16	0.0024	0.11	-0.17	0.3	0.22	1	0.16	-0.27	-0.11	-0.34	0.4	-0.49
price_per_bathroom	0.57	0.4	0.073	-0.36	0.13	0.27	0.19	0.17	0.038	0.29	0.26	0.29	-0.22	-0.12	0.78	0.16	1	0.33	0.0063	-0.014	0.6	0.42
area_bedrooms	0.61	0.87	0.57	0.36	0.26	0.23	0.16	0.072	0.01	0.24	0.34	0.22	-0.18	0.58	0.29	-0.27	0.33	1	0.8	0.26	0.61	0.83
area_bathrooms	0.67	0.74	0.34	0.75	0.28	0.21	0.15	0.058	0.032	0.25	0.31	0.18	-0.18	0.61	0.32	-0.11	0.0063	0.8	1	0.2	0.64	0.71
area_furnishingstatus	-0.013	0.31	-0.018	-0.021	-0.034	0.024	-0.032	-0.077	-0.029	-0.04	-0.0073	0.011	0.8	-0.023	-0.012	-0.34	-0.014	0.26	0.2	1	-0.025	0.31
log_price	0.97	0.54	0.37	0.49	0.42	0.33	0.28	0.22	0.089	0.46	0.37	0.34	-0.34	0.5	0.77	0.4	0.6	0.61	0.64	-0.025	1	0.58
log_area	0.56	0.96	0.15	0.2	0.11	0.33	0.18	0.043	-0.014	0.26	0.36	0.22	-0.17	0.21	0.52	-0.49	0.42	0.83	0.71	0.31	0.58	1

```
#split features and target
X = housing.drop(['price','log_price'],axis=1)
y = housing['log_price']


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


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

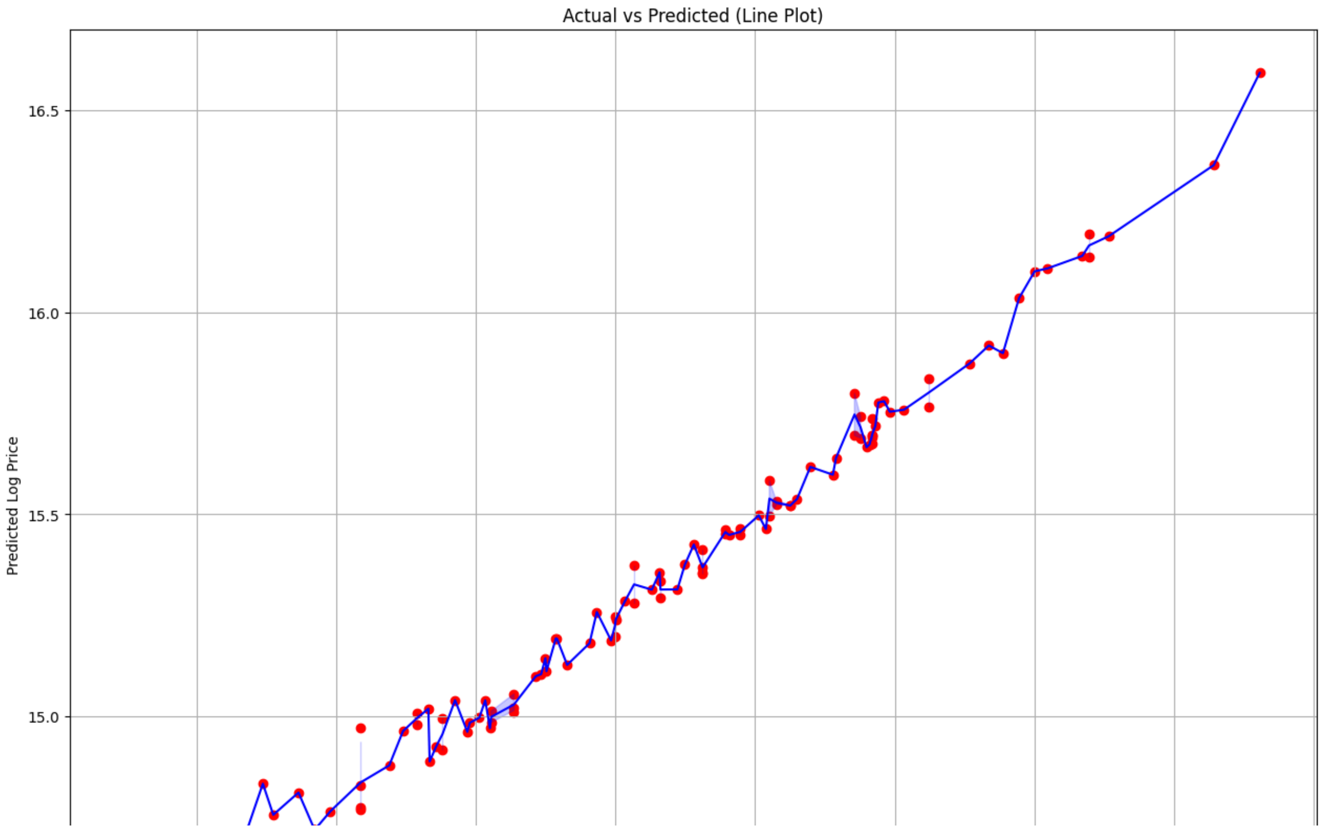
```
▾ LinearRegression
LinearRegression()
```

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

```
accuracy_score=model.score(X_test,y_test)
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
print('-------------------- Mohamed--------------------')
print(' ')
print(accuracy_score*100)
print(' ')
print('-------------------Ayman------------------------')
print(' ')
print(f"Mean Squared Error (MSE): {mse:.4f}")
print(' ')
print('------------------Fathy-------------------------')
print(' ')
print(f"Root Mean Squared Error (RMSE): {rmse:.4f}")
print(' ')
print('--------------EL-Halmoushy---------------------')
print(' ')
print(f"Mean Absolute Error (MAE): {mae:.4f}")
print(' ')
print('------------------------------------------------')

print(f"R² Score: {r2:.4f}")
```

```
-------------------- Mohamed--------------------

98.11790745806447

-------------------Ayman------------------------

Mean Squared Error (MSE): 0.0036

------------------Fathy-------------------------

Root Mean Squared Error (RMSE): 0.0603

--------------EL-Halmoushy---------------------

Mean Absolute Error (MAE): 0.0429

------------------------------------------------
R² Score: 0.9812
```

```
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(25, 10))

axes[0].scatter(y_test, y_pred, color='red')
axes[0].set_xlabel("Actual Log Price")
axes[0].set_ylabel("Predicted Log Price")
axes[0].set_title("Actual vs Predicted (Scatter Plot)")
axes[0].grid(True)
sns.lineplot(x=y_test, y=y_pred, color='blue', ax=axes[0])
axes[0].set_xlabel("Actual Log Price")
axes[0].set_ylabel("Predicted Log Price")
axes[0].set_title("Actual vs Predicted (Line Plot)")
axes[0].grid(True)
for j in range(1, len(axes)):
    fig.delaxes(axes[j])

plt.tight_layout()
plt.show()
```



```
lasso = Lasso(alpha=0.01)
lasso.fit(X_train, y_train)
y_pred_lasso = lasso.predict(X_test)
mse = mean_squared_error(y_test, y_pred_lasso)
mae = mean_absolute_error(y_test, y_pred_lasso)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred_lasso)
acc_lasso=lasso.score(X_test,y_test)
print(acc_lasso*100)
print(f"Lasso Regression Results:")
print(f"R² Score: {r2:.4f}")
print(f"RMSE: {rmse:.4f}")
print(f"MAE: {mae:.4f}")
```

```
95.75936142598209
Lasso Regression Results:
R² Score: 0.9576
RMSE: 0.0905
MAE: 0.0636
```

Start coding or generate with AI.