In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

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

In [3]:

```
df
```

Out[3]:

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

545 rows × 13 columns

In [4]:

```
df.isnull().sum()
```

Out[4]:

```
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]:

```
from sklearn.preprocessing import OneHotEncoder
```

In [21]:

```
ohe = OneHotEncoder(sparse_output = False,handle_unknown = 'ignore',drop='firs
```

In [22]:
```
categorical_cols = ['mainroad','guestroom','basement','hotwaterheating','aircc
```

In [23]:
```
encoded_cols = ohe.fit_transform(df[categorical_cols])
```

In [26]:
```
encoded_df = pd.DataFrame(
    encoded_cols,
    columns=ohe.get_feature_names_out(categorical_cols)
)
```

In [28]:
```
df_numeric = df.drop(columns=categorical_cols).reset_index(drop=True)
```

In [29]:
```
df_final = pd.concat([df_numeric,encoded_df.reset_index(drop=True)],axis=1)
```

In [30]:
```
df_final.head()
```

Out[30]:

	price	area	bedrooms	bathrooms	stories	parking	mainroad_yes	gues
0	13300000	7420	4	2	3	2	1.0	
1	12250000	8960	4	4	4	3	1.0	
2	12250000	9960	3	2	2	2	1.0	
3	12215000	7500	4	2	2	3	1.0	
4	11410000	7420	4	1	2	2	1.0	

In [32]:
```
from sklearn.model_selection import train_test_split
```

In [50]:
```
X_train,X_test,y_train,y_test = train_test_split(df_final.iloc[:,1:14],df_fina
```

In [98]:
```
from sklearn.linear_model import LinearRegression
```

In [99]:
```
model = LinearRegression()
```

In [108…
```
model.fit(X_train,y_train)
```

Out[108…

▾ LinearRegression

Parameters

	fit_intercept	True
	copy_X	True
	tol	1e-06
	n_jobs	None
	positive	False

In [109…

```
#y_pred = model.predict(X_test.iloc[:5])
y_pred = model.predict(X_test)
```

In [110…

```
X_test.head()
```

Out[110…

	area	bedrooms	bathrooms	stories	parking	mainroad_yes	guestroom_y
316	5900	4	2	2	1	0.0	0
77	6500	3	2	3	0	1.0	0
360	4040	2	1	1	0	1.0	0
90	5000	3	1	2	0	1.0	0
493	3960	3	1	1	0	1.0	0

In [111…

```
y_test.head()
```

Out[111…

```
316    4060000
77     6650000
360    3710000
90     6440000
493    2800000
Name: price, dtype: int64
```

In [112…

```
from sklearn.metrics import r2_score
```

In [113…

```
r2 = r2_score(y_test, y_pred)
```

In [114…

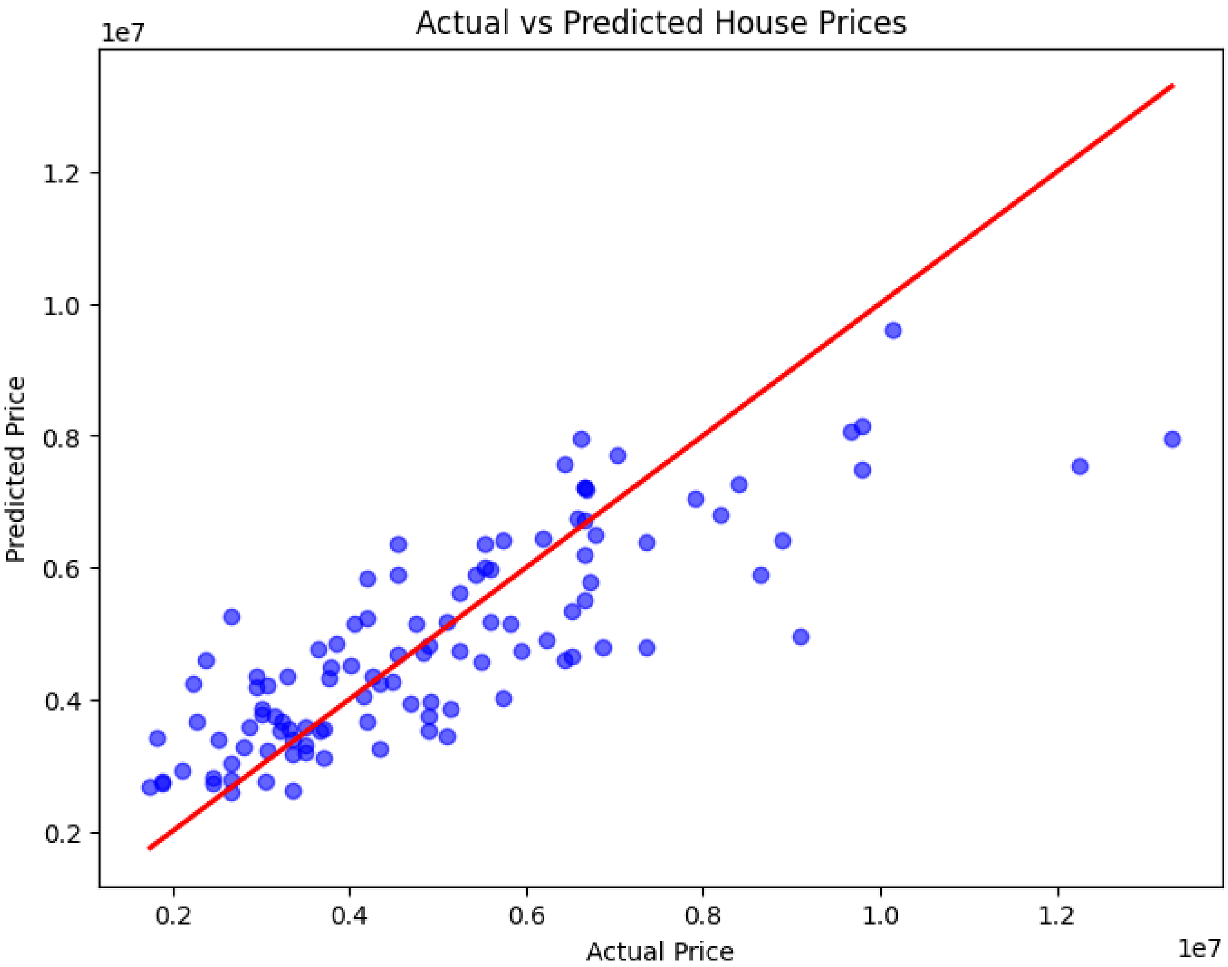
```
r2
```

Out[114…

```
0.6529242642153184
```

In [118…

```
plt.figure(figsize=(8,6))
plt.scatter(y_test, y_pred, color='blue', alpha=0.6)
plt.plot([y_test.min(), y_test.max()],
         [y_test.min(), y_test.max()],
         color='red', linewidth=2)
plt.xlabel("Actual Price")
plt.ylabel("Predicted Price")
plt.title("Actual vs Predicted House Prices")
plt.show()
```


Actual vs Predicted House Prices
1e7
1.2
1.0
0.8
0.6
0.4
0.2
Predicted Price
0.2
0.4
0.6
0.8
1.0
1.2
Actual Price
1e7