

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

#assign data
data = pd.read_csv("C:\\Users\\Sarthak Tyagi\\Downloads\\Housing.csv")

#display a few rows of the data
data.head(3)
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom
0	13300000	7420	4	2	3	yes	no
1	12250000	8960	4	4	4	yes	no
2	12250000	9960	3	2	2	yes	no

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

```
#statistics for the data
data.describe()
```

	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

```
count    parking
545.000000
mean      0.693578
std       0.861586
min       0.000000
25%       0.000000
50%       0.000000
75%       1.000000
max       3.000000
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 545 entries, 0 to 544
```

```
Data columns (total 13 columns):
```

#	Column	Non-Null Count	Dtype
0	price	545 non-null	int64
1	area	545 non-null	int64
2	bedrooms	545 non-null	int64
3	bathrooms	545 non-null	int64
4	stories	545 non-null	int64
5	mainroad	545 non-null	object
6	guestroom	545 non-null	object
7	basement	545 non-null	object
8	hotwaterheating	545 non-null	object
9	airconditioning	545 non-null	object
10	parking	545 non-null	int64
11	prefarea	545 non-null	object
12	furnishingstatus	545 non-null	object

```
dtypes: int64(6), object(7)
```

```
memory usage: 55.5+ KB
```

```
#check for missing values
```

```
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
```

```
#assigning dummy variables
```

```
data = pd.get_dummies(data, drop_first=True)
data
```

	price	area	bedrooms	bathrooms	stories	parking
mainroad_yes \						
0	13300000	7420	4	2	3	2
True						
1	12250000	8960	4	4	4	3
True						
2	12250000	9960	3	2	2	2
True						
3	12215000	7500	4	2	2	3
True						
4	11410000	7420	4	1	2	2
True						
..
..						
540	1820000	3000	2	1	1	2
True						
541	1767150	2400	3	1	1	0
False						
542	1750000	3620	2	1	1	0
True						
543	1750000	2910	3	1	1	0
False						
544	1750000	3850	3	1	2	0
True						
guestroom_yes						
airconditioning_yes \						
0	False	False	False	False		
True						
1	False	False	False	False		
True						
2	False	True	False	False		
False						
3	False	True	False	False		
True						
4	True	True	False	False		
True						
..		
..						
540	False	True	False	False		
False						
541	False	False	False	False		
False						

542	False	False	False
False			
543	False	False	False
False			
544	False	False	False
False			

	prefarea_yes	furnishingstatus_semi-furnished	\
0	True	False	
1	False	False	
2	True	True	
3	True	False	
4	False	False	
..	
540	False	False	
541	False	True	
542	False	False	
543	False	False	
544	False	False	

	furnishingstatus_unfurnished
0	False
1	False
2	False
3	False
4	False
..	...
540	True
541	False
542	True
543	False
544	True

[545 rows x 14 columns]

#splitting the data

X = data.drop('price', axis=1)

y = data['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()

#get mean squared error

y_pred= model.predict(X_test)

```
mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error:{mse}')
```

Mean Squared Error:1754318687330.6643

```
#get accuracy of model
model.score(X_test, y_test)
```

0.6529242642153184

```
#visualize the data
plt.figure(figsize=(10, 6))
sns.scatterplot(x=y_test, y=y_pred, alpha=0.6)
sns.lineplot(x=[y_test.min(), y_test.max()], y=[y_test.min(),
y_test.max()], color='red')
plt.title('Actual vs Predicted Prices')
plt.xlabel('Actual Prices')
plt.ylabel('Predicted Prices')
plt.show()
```

