Lab 3 Assignment: Multi Linear Regression

Aim: Write a script to implement following for the given Dataset Bengaluru Housing Dataset/ Boston Housing Dataset.

Perform the following: Exercise 1: Draw a scatter plot for the data mentioned for given attributes.

Exercise 2: Perform Data pre-processing.

Exercise 3: Performs gradient descent to learn `theta` . (using the library and without using the library). Compare the values of ‘theta’ in both cases.

Exercise 4: Splitting data into the training and testing, 60:40, 70:30, ND 80:20.

Exercise 5: Train multilinear regression model and test USING Gradient Descent and using the library. Compare your results with Simple Linear Regression.

Exercise 6: Did you expect ridge regression to outperform the lasso, or vice versa? Which predictors turned out to be important in the final model(s)?

Exercise 1:

Draw a scatter plot for the data mentioned for given attributes.

In [1]:

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

In [2]:

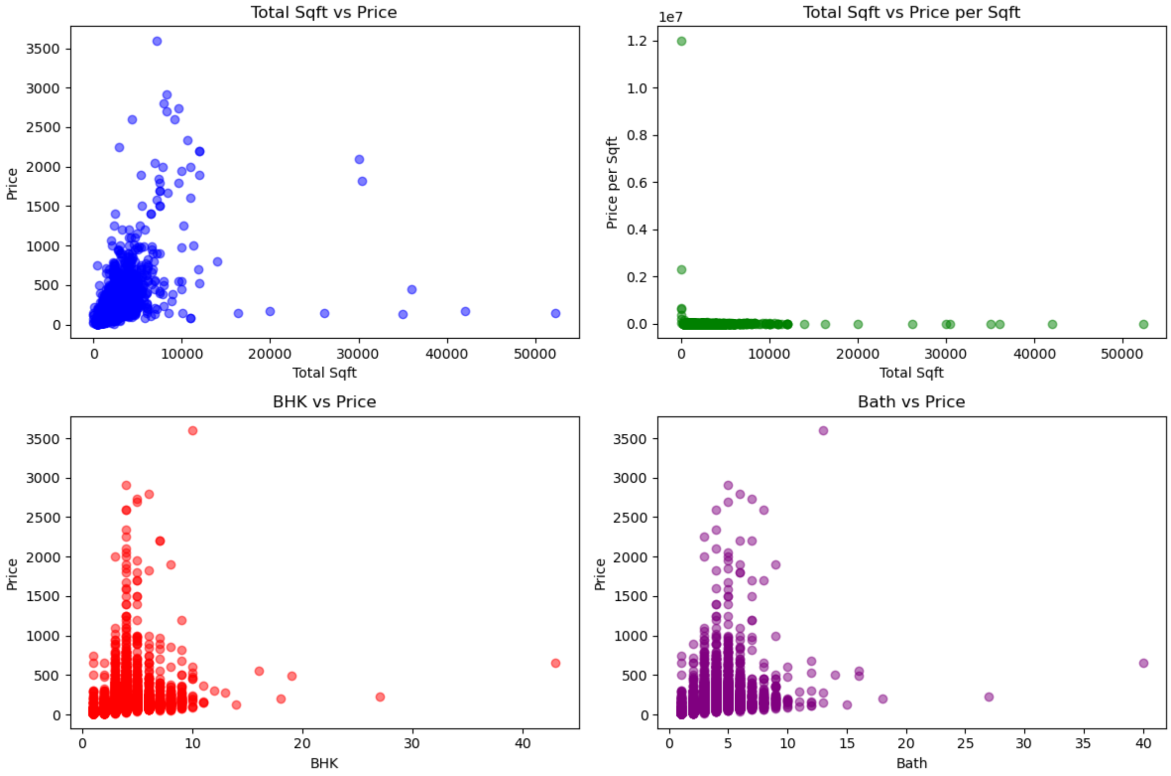
```
df = pd.read_csv(r"C:\Users\raval\jupyter_notebook\pdeu_data_science\prml_lab\bhp.csv")
df
```

Out[2]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.07	2	3699.810606
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4	4615.384615
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3	4305.555556
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.00	3	6245.890861
4	Kothanur	2 BHK	1200.0	2.0	51.00	2	4250.000000
...	...	...	...	...	...	...	...
13195	Whitefield	5 Bedroom	3453.0	4.0	231.00	5	6689.834926
13196	Richards Town	4 BHK	3600.0	5.0	400.00	4	11111.111111
13197	Raja Rajeshwari Nagar	2 BHK	1141.0	2.0	60.00	2	5258.545136
13198	Padmanabhanagar	4 BHK	4689.0	4.0	488.00	4	10407.336319
13199	Doddathoguru	1 BHK	550.0	1.0	17.00	1	3090.909091

13200 rows × 7 columns

```
In [3]: # Create subplots
        fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(12, 8))

        # Scatter plot 1 - Total Sqft vs Price
        axes[0, 0].scatter(df['total_sqft'], df['price'], c='blue', alpha=0.5)
        axes[0, 0].set_title('Total Sqft vs Price')
        axes[0, 0].set_xlabel('Total Sqft')
        axes[0, 0].set_ylabel('Price')

        # Scatter plot 2 - Total Sqft vs Price per Sqft
        axes[0, 1].scatter(df['total_sqft'], df['price_per_sqft'], c='green', alpha=0.5)
        axes[0, 1].set_title('Total Sqft vs Price per Sqft')
        axes[0, 1].set_xlabel('Total Sqft')
        axes[0, 1].set_ylabel('Price per Sqft')

        # Scatter plot 3 - BHK vs Price
        axes[1, 0].scatter(df['bhk'], df['price'], c='red', alpha=0.5)
        axes[1, 0].set_title('BHK vs Price')
        axes[1, 0].set_xlabel('BHK')
        axes[1, 0].set_ylabel('Price')

        # Scatter plot 4 - Bath vs Price
        axes[1, 1].scatter(df['bath'], df['price'], c='purple', alpha=0.5)
        axes[1, 1].set_title('Bath vs Price')
        axes[1, 1].set_xlabel('Bath')
        axes[1, 1].set_ylabel('Price')

        # Adjust layout
        plt.tight_layout()

        # Show the plots
        plt.show()
```



Exercise 2:

Perform Data pre-processing.

In [4]:

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

Out[4]:

```
location          0
size              0
total_sqft        0
bath              0
price             0
bhk               0
price_per_sqft    0
dtype: int64
```

data is pre-processed already

Exercise 3:

Performs gradient descent to learn `theta` . (using the library and without using the library). Compare the values of ‘theta’ in both cases.

In [5]:

```
x = df[['bhk']].values
y = df['price_per_sqft'].values
Theta = np.zeros(x.shape[1])
Theta_0 = 0
learning_rate = 0.001
epochs = 10000
n = float(len(x))
for i in range(epochs):

    y_pred = np.dot(x, Theta) + Theta_0
    d_theta = (-2/n) * np.dot(x.T, (y - y_pred))
    d_theta_0 = (-2/n) * np.sum(y - y_pred)
    Theta = Theta - learning_rate * d_theta
    Theta_0 = Theta_0 - learning_rate * d_theta_0
print("Thetas without using a library:", Theta)
print("Intercept (Theta_0):", Theta_0)
```

```
Thetas without using a library: [2555.38725194]
Intercept (Theta_0): 763.5548130675422
```

In [6]:

```
x = df[['bhk']]
y = df['price_per_sqft']

model = LinearRegression()
model.fit(x, y)
theta_values = model.coef_
theta_0 = model.intercept_

print("Theta values (Coefficients):", theta_values)
print("Theta_0 (Intercept):", theta_0)
```

```
Theta values (Coefficients): [2555.40479612]
Theta_0 (Intercept): 763.4962168177162
```

Exercise 4:

Splitting data into the training and testing, 60:40, 70:30 ND 80:20.

In [7]:

```
X= df[['bhk','total_sqft','bath']]
y = df['price_per_sqft']
```

60:40 split

```
In [8]: X_train, X_test, y_train, y_test= train_test_split(X, y, test_size=0.4, random_state=1)
        X_train.shape,X_test.shape
```

```
Out[8]: ((7920, 3), (5280, 3))
```

70:30 split

```
In [9]: X_train, X_test, y_train, y_test= train_test_split(X, y, test_size=0.3, random_state=1)
        X_train.shape,X_test.shape
```

```
Out[9]: ((9240, 3), (3960, 3))
```

80:20 split

```
In [10]: X_train, X_test, y_train, y_test= train_test_split(X, y, test_size=0.2, random_state=1)
         X_train.shape,X_test.shape
```

```
Out[10]: ((10560, 3), (2640, 3))
```

Exercise 5:

Train Multilinear regression model and test USING gradient descent and using the library. Compare your results with simple linear regression

In [23]:

```python
# Performs gradient descent to learn theta, without using the Library

x1, x2, x3 = df['bhk'], df['total_sqft'], df['bath']
y= df["price_per_sqft"]

Theta_0  = 0.6
Theta_1 = 0.2
Theta_2 = 0.3
Theta_3 = 0.7

L = 0.000001
epochs= 100
n = float(len(df))

for i in range(epochs):
    y_pred = Theta_0 + Theta_1*x1 + Theta_2*x2 + Theta_3*x3
    d_theta_0 = (-2/n)*sum(y - y_pred)
    d_theta_1 = (-2/n)*sum(x1 *(y - y_pred))
    d_theta_2 = (-2/n) * sum(x2* (y - y_pred))
    d_theta_3 = (-2/n)* sum(x3* (y - y_pred))

    Theta_0= Theta_0- L *d_theta_0
    Theta_1= Theta_1- L *d_theta_1
    Theta_2= Theta_2- L *d_theta_2
    Theta_3= Theta_3- L *d_theta_3

    print("Theta without using library: ", Theta_0, Theta_1, Theta_2, Theta_3)
```

```
Theta without using library:  0.6149022487058641 0.24994557757968294 20.374742700459706 0.
7485360392003488
Theta without using library:  0.5673593202163109 0.10277101428478422 -118.13476417797173
0.6029021453037219
Theta without using library:  0.9506665381041666 1.3156568605508223 837.533991629799 1.796
9733422831333
Theta without using library:  -1.638748782602545 -6.855421677213238 -5756.261172881948 -6.
252474828469451
Theta without using library:  16.282626110699677 49.71971172090359 39738.71879906596 49.47
5265442449235
Theta without using library:  -107.31358961734628 -340.43235693947344 -274161.4217107906 -
334.8382583248831
Theta without using library:  745.5142200386781 2351.6837991718676 1891644.5220125094 231
6.9857074613533
Theta without using library:  -5138.6570523627115 -16222.819527186166 -13051691.850775259
-15979.523170596345
Theta without using library:  35460.21126300089 111935.23614757853 90052329.35154088 11026
0.4556390197
Theta without using library:  -244658.34119560994 -772312.271234107 -621330920.2087663 -76
0752.9713105297
Theta without using library:  1688066.0034512822 5328699.873925075 4286975485.819262 52489
48.438566756
Theta without using library:  -11647085.26481693 -36766242.44221652 -29578696485.441196 -3
6215980.363229975
Theta without using library:  80360996.84166227 253674775.53206635 204083109276.41473 2498
78155.144988
Theta without using library:  -554463985.4016504 -1750271071.994307 -1408105171476.5745 -1
724075657.761787
Theta without using library:  3825615991.203103 12076284792.186947 9715454556749.955 11895
545202.327057
Theta without using library:  -26395470351.966526 -83322324555.55527 -67033385826604.75 -8
2075281853.88602
Theta without using library:  182119913988.39792 574896161299.0133 462507934048040.3 56629
1983899.1912
Theta without using library:  -1256566472492.9268 -3966591163125.725 -3191149998162203.0 -
3907225217864.912
Theta without using library:  8669888235847.399 27368151876085.62 2.201786728638658e+16 26
958546716558.82
Theta without using library:  -59819328039925.32 -188831091057559.47 -1.5191591749686634e+
17 -186004952503280.75
Theta without using library:  412733349012875.94 1302871348837690.2 1.0481690024167055e+18
1283371938387894.8
Theta without using library:  -2847722015093376.0 -8989376389848560.0 -7.232015418330909e+
18 -8854836981894113.0
Theta without using library:  1.9648329107988436e+16 6.2023689407596984e+16 4.989848668524
388e+19 6.109541250715797e+16
Theta without using library:  -1.355668968704357e+17 -4.2794270491046355e+17 -3.4428286299
9217e+20 -4.215379048820592e+17
Theta without using library:  9.353662301801894e+17 2.952661482011918e+18 2.37543656389062
85e+21 2.90847050474586e+18
Theta without using library:  -6.453714031661346e+18 -2.0372376318883107e+19 -1.6389717512
838906e+22 -2.0067473361247207e+19
Theta without using library:  4.452846752275687e+19 1.4056258037253662e+20 1.1308356713626
24e+23 1.3845885197985124e+20
Theta without using library:  -3.072315274891128e+20 -9.698347748795941e+20 -7.80238777528
9045e+23 -9.553197528394303e+20
Theta without using library:  2.119794745576044e+21 6.691535457537135e+21 5.38338651120070
2e+24 6.591386661923275e+21
Theta without using library:  -1.4625874499586048e+22 -4.61693557905634e+22 -3.71435657437
62008e+25 -4.547836260879909e+22
Theta without using library:  1.0091364049471137e+23 3.1855310752551984e+23 2.562781760682
9333e+26 3.137854857651368e+23
Theta without using library:  -6.962703555390241e+23 -2.19790985983185e+24 -1.768233668839
9576e+27 -2.1650148648449974e+24
Theta without using library:  4.8040324937819586e+24 1.5164842652049692e+25 1.220022069606
901e+28 1.493787819270942e+25
Theta without using library:  -3.3146216864922606e+25 -1.0463234041773686e+26 -8.417744083
022753e+28 -1.0306636158648178e+26
Theta without using library:  2.286978062447577e+26 7.219281407983204e+26 5.80796177483044
6e+29 7.111234108107847e+26
Theta without using library:  -1.577938344949282e+27 -4.981062627441443e+27 -4.00729930076
21e+30 -4.9065136055941926e+27
```

```
Theta without using library:  1.0887246630588494e+28 3.43676655560993e+28 2.76490244055666
35e+31 3.3853302248104342e+28
Theta without using library:  -7.5118359075728995e+28 -2.371253935392933e+29 -1.9076901753
613305e+32 -2.3357645881076617e+29
Theta without using library:  5.182915443815123e+29 1.6360858775636027e+30 1.3162423931447
64e+33 1.6115994153459825e+30
Theta without using library:  -3.5760382452785186e+30 -1.1288445150517404e+31 -9.081632121
858189e+33 -1.1119496753942124e+31
Theta without using library:  2.467346741485967e+31 7.7886494629487e+31 6.2660223091367155
e+34 7.672080722018788e+31
Theta without using library:  -1.7023866986767504e+32 -5.373907535344526e+32 -4.3233457435
58534e+35 -5.293479004281567e+32
Theta without using library:  1.1745898633145426e+33 3.707816398184888e+33 2.9829639117452
61e+36 3.652323402744296e+33
Theta without using library:  -8.104277060398153e+33 -2.5582692579336e+34 -2.058145294539
985e+37 -2.5199809477743774e+34
Theta without using library:  5.591680017257677e+34 1.7651201929228704e+35 1.4200513914225
487e+38 1.7387025399706178e+35
Theta without using library:  -3.858072124432355e+35 -1.2178738753953818e+36 -9.7978794773
66146e+38 -1.1996465787450957e+36
Theta without using library:  2.661940681759892e+36 8.402922261710142e+36 6.76020902009937
7e+39 8.277160013346172e+36
Theta without using library:  -1.836650006705353e+37 -5.797735214035971e+37 -4.66431803953
1262e+40 -5.710963470441643e+37
Theta without using library:  1.2672270536473593e+38 4.000243315975998e+38 3.2182233876516
05e+41 3.940373716121116e+38
Theta without using library:  -8.74344267897004e+38 -2.760034047134952e+39 -2.220466461559
8484e+42 -2.7187260263632657e+39
Theta without using library:  6.032682908749734e+39 1.90433114678785e+40 1.532047565694308
5e+43 1.8758299945469328e+40
Theta without using library:  -4.1623493644046957e+40 -1.3139247758160725e+41 -1.057061560
7951898e+44 -1.2942599343667228e+41
Theta without using library:  2.8718817967760546e+41 9.065641337722904e+41 7.2933710958536
06e+44 8.929960511221833e+41
Theta without using library:  -1.9815023518174053e+42 -6.254989203106147e+42 -5.0321820331
653504e+45 -6.161373972454835e+42
Theta without using library:  1.3671703252778781e+43 4.3157332695451315e+43 3.472037235196
8615e+46 4.2511418925916224e+43
Theta without using library:  -9.433017814014024e+43 -2.9777115593755597e+44 -2.3955895242
14919e+47 -2.933145670388002e+44
Theta without using library:  6.508466679996184e+44 2.054521347139117e+45 1.65287661962611
15e+48 2.0237723748315432e+45
Theta without using library:  -4.49062424770229e+45 -1.4175509889666406e+46 -1.14042956528
70514e+49 -1.3963352268792725e+46
Theta without using library:  3.098380482768933e+46 9.78062753700951e+46 7.868582433424586
e+49 9.63424587701608e+46
Theta without using library:  -2.137778866917106e+47 -6.748305758471996e+47 -5.42905861056
0863e+50 -6.647307310741476e+47
Theta without using library:  1.4749958919678126e+48 4.6561051872699655e+48 3.745868794828
961e+51 4.5864196375608523e+48
Theta without using library:  -1.0176978147694666e+49 -3.2125567943783976e+49 -2.584524138
4535413e+52 -3.1644760966312373e+49
Theta without using library:  7.021774418671881e+49 2.2165567017952816e+50 1.7832351820411
538e+53 2.1833826290427403e+50
Theta without using library:  -4.844789413042407e+50 -1.529349962270233e+51 -1.23037261179
23078e+54 -1.506460962015299e+51
Theta without using library:  3.342742597129367e+51 1.055200304689521e+52 8.48915936099823
5e+54 1.0394076603380622e+52
Theta without using library:  -2.3063805499125125e+52 -7.280529051467029e+52 -5.8572359272
07717e+55 -7.171565089374434e+52
Theta without using library:  1.5913254121279896e+53 5.023321452210034e+53 4.0412968172786
53e+56 4.948139964102753e+53
Theta without using library:  -1.097961291505165e+54 -3.465923730796677e+54 -2.78835958945
78364e+57 -3.414051019438938e+54
Theta without using library:  7.575565553444132e+54 2.3913714107255522e+55 1.9238748232694
624e+58 2.3555809754556126e+55
Theta without using library:  -5.22688676718773e+55 -1.6499662624488737e+56 -1.32740925869
2423e+59 -1.6252720595956873e+56
Theta without using library:  3.6063769872050194e+56 1.1384215162100831e+57 9.158679757906
321e+59 1.1213833424646227e+57
Theta without using library:  -2.4882794583359562e+57 -7.854727566650438e+57 -6.3191826001
35758e+60 -7.73716986847085e+57
```

```
Theta without using library:  1.7168295729324997e+58 5.419499216045198e+58 4.3600245657008
755e+61 5.338388337568748e+58
Theta without using library:  -1.1845549633186322e+59 -3.7392731324532223e+59 -3.008271071
816578e+62 -3.683309340127386e+59
Theta without using library:  8.173032916284323e+59 2.5799733520932765e+60 2.0756063882575
074e+63 2.5413602078354878e+60
Theta without using library:  -5.639119257372896e+60 -1.7800952917135682e+61 -1.4320989618
711127e+64 -1.753453514101481e+61
Theta without using library:  3.8908036128809885e+61 1.2282061925213235e+62 9.881003682562
374e+64 1.209824257354569e+62
Theta without using library:  -2.6845243136530817e+62 -8.474211792872688e+62 -6.8175619404
9948e+65 -8.347382590484334e+62
Theta without using library:  1.8522319571042845e+63 5.8469226053197056e+63 4.703889635682
6785e+66 5.759414699146897e+63
Theta without using library:  -1.2779780780788336e+64 -4.03418097024098e+64 -3.24552646500
22215e+67 -3.9738034428376056e+64
Theta without using library:  8.817621150449449e+64 2.7834498930852215e+65 2.2393046714203
534e+68 2.741791419299475e+65
Theta without using library:  -6.083863572193335e+65 -1.9204872970419396e+66 -1.5450453002
057363e+69 -1.8917443439468188e+66
Theta without using library:  4.1976623097687554e+66 1.3250719789359485e+67 1.066029562727
434e+70 1.3052403029873968e+67
Theta without using library:  -2.896246547569988e+67 -9.142553309597695e+67 -7.35524730865
5209e+70 -9.005721380871897e+67
Theta without using library:  1.9983132146648553e+68 6.308055890364562e+68 5.0748745497325
e+71 6.2136464376920185e+68
Theta without using library:  -1.378769258181601e+69 -4.3523475082382036e+69 -3.5014936432
139914e+72 -4.287208144663254e+69
Theta without using library:  9.513046570257102e+69 3.00297415902777e+70 2.415913460188719
e+73 2.9580301776060716e+70
Theta without using library:  -6.5636838441867e+70 -2.0719516956583301e+71 -1.666899455446
0095e+74 -2.0409418522214778e+71
Theta without using library:  4.528722243527658e+71 1.4295773462570646e+72 1.1501048528241
367e+75 1.4081815918187796e+72
Theta without using library:  -3.1246668251986558e+72 -9.863605378512628e+72 -7.9353386802
5489e+75 -9.71598183151996e+72
Theta without using library:  2.1559155637003188e+73 6.805557692821083e+73 5.4751181873306
15e+76 6.703702384612445e+73
Theta without using library:  -1.4875096059271771e+74 -4.695607106424807e+74 -3.7776483617
301794e+77 -4.625330351654759e+74
Theta without using library:  1.0263318587152539e+75 3.2398117969326135e+75 2.606450976328
6635e+78 3.1913231874740967e+75
Theta without using library:  -7.081346433103066e+75 -2.2353617416547034e+76 -1.7983639665
4269e+79 -2.2019062234692892e+76
Theta without using library:  4.885892109828162e+76 1.5423248105906295e+77 1.2408109669166
5e+80 1.5192416224036745e+77
Theta without using library:  -3.3711020826898527e+77 -1.0641525159156887e+78 -8.561180518
871915e+80 -1.0482258883883098e+78
Theta without using library:  2.3259476460922796e+78 7.342296313680123e+78 5.9069281164432
8e+81 7.2324079125020474e+78

Theta without using library:  -1.6048260538124782e+79 -5.065938796516543e+79 -4.0755827652
40592e+82 -4.990119476370354e+79
Theta without using library:  1.1072762825604548e+80 3.4953282724690217e+80 2.812015746405
881e+83 3.4430154783451415e+80
Theta without using library:  -7.639835875098354e+80 -2.4116595606567755e+81 -1.9401967800
714904e+84 -2.3755654830025396e+81
```

In [14]:

```python
# # Create a DataFrame with your features
# data = pd.DataFrame({'x1': x1, 'x2': x2, 'x3': x3})

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize the Linear Regression model
model = LinearRegression()

# Fit the model to the training data
model.fit(X_train, y_train)

# Get the coefficients and intercept
coefficients = model.coef_
intercept = model.intercept_

# Make predictions on the test set
y_pred = model.predict(X_test)

# Calculate the Mean Squared Error (MSE) to evaluate the model
mse = mean_squared_error(y_test, y_pred)

print(f"Coefficients: {coefficients}")
print(f"Intercept: {intercept}")
print(f"Mean Squared Error: {mse}")
```

```
Coefficients: [1491.71288993   -1.94961844 1933.72142831]
Intercept: 1717.623832424195
Mean Squared Error: 1986674652.4347043
```

The Value of Thetas for without library and with library are different to each other , the without library falling short by a very large numbers at 100 epochs and 0.0001 learning rate

In [13]:

```python
X= df[['bhk','total_sqft','bath']]
```

Exercise 6

Did you expect ridge regression to outperform the lasso ,or vice versa ?

In [24]:

```python
from sklearn.linear_model import Lasso, Ridge
from sklearn.metrics import mean_squared_error, r2_score
```

In [25]:

```python
X= df[['bhk','total_sqft','bath']]
y = df['price']
```

In [26]:

```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=3)
```

Lasso

In [27]:

```python
lasso_model = Lasso(alpha=1.0)
lasso_model.fit(X_train, y_train)
lasso_pred = lasso_model.predict(X_test)
l_mse = mean_squared_error(y_test, lasso_pred)
l_r2 = r2_score(y_test, lasso_pred)
```

In [28]:

```python
print("Mean Square Error:", l_mse)
print("R-2 :", l_r2)
```

```
Mean Square Error: 11553.307662056648
R-2 : 0.46703493251013517
```

Ridge

In [29]:
```
ridge_model = Ridge(alpha=1.0)
ridge_model.fit(X_train, y_train)
ridge_pred = ridge_model.predict(X_test)
r_mse = mean_squared_error(y_test, ridge_pred)
r_r2 = r2_score(y_test, ridge_pred)
```

In [30]:
```
print("Mean Square Error:", r_mse)
print("R-2:", r_r2)
```

```
Mean Square Error: 11582.80239988878
R-2: 0.46567431220994837
```

Expected Ridge to perform better due to regularization factor/ bias being squared allowing for faster regularization but here

Exercise 7

Which predictors turned out to be important in the final models(s)?

"bath" appears to be the most important feature for price prediction in your model, as it has the largest positive coefficient. This suggests that the number of bathrooms has a significant impact on the price.

"bhk" is also important, but it has a slightly smaller positive coefficient, indicating that the number of bedrooms (bhk) is another influential factor in price prediction.

"total_sqft" has a negative coefficient, which suggests that an increase in total square footage is associated with a decrease in price in your model. This may be counterintuitive and could indicate an issue with the model or data. You should further investigate this feature and its relationship with the target variable.

In [ ]:
```
```