

## # Business Problem - Predict the Price of House using K-NN algorithms (Regression)

### # Load Libraries

In [1]:

```
import pandas as pd
```

### # Load Data

In [9]:

```
dataset = pd.read_csv('bangalore house price.csv')
```

In [10]:

```
dataset
```

Out[10]:

	bath	balcony	price	total_sqft_int	bhk	price_per_sqft
0	3	2	150.00	1672	3	8971.291866
1	3	3	149.00	1750	3	8514.285714
2	3	2	150.00	1750	3	8571.428571
3	2	2	40.00	1250	2	3200.000000
4	2	2	83.00	1200	2	6916.666667
5	2	2	40.00	1170	2	3418.803419
6	2	2	65.00	1425	3	4561.403509
7	2	2	43.00	947	2	4540.654699
8	2	2	36.00	1130	2	3185.840708
9	2	2	76.00	1417	3	5363.443896
10	2	1	58.00	1230	2	4715.447154
11	3	2	75.00	1705	3	4398.826979
12	2	1	47.00	1116	2	4211.469534
13	2	2	40.00	991	2	4036.326942
14	1	1	27.00	550	1	4909.090909
15	2	2	70.00	1450	3	4827.586207
16	1	1	25.50	700	1	3642.857143
17	1	0	28.00	440	1	6363.636364
18	1	0	25.25	510	1	4950.980392
19	1	1	25.25	510	1	4950.980392
20	2	0	480.00	2400	3	20000.000000
21	3	2	85.00	1540	3	5519.480519
22	1	0	45.00	600	1	7500.000000
23	4	2	150.00	3150	4	4761.904762
24	2	1	67.00	1250	2	5360.000000
25	4	1	428.00	2850	4	15017.543860
26	3	2	194.00	1630	3	11901.840490
27	2	3	235.00	1875	3	12533.333330
28	2	0	130.00	1200	3	10833.333330

In [12]:

```
x = dataset.drop('price', axis=1)
```

In [13]:

```
x
```

Out[13]:

	bath	balcony	total_sqft_int	bhk	price_per_sqft
0	3	2	1672	3	8971.291866
1	3	3	1750	3	8514.285714
2	3	2	1750	3	8571.428571
3	2	2	1250	2	3200.000000
4	2	2	1200	2	6916.666667
5	2	2	1170	2	3418.803419
6	2	2	1425	3	4561.403509
7	2	2	947	2	4540.654699
8	2	2	1130	2	3185.840708
9	2	2	1417	3	5363.443896
10	2	1	1230	2	4715.447154
11	3	2	1705	3	4398.826979
12	2	1	1116	2	4211.469534
13	2	2	991	2	4036.326942
14	1	1	550	1	4909.090909
15	2	2	1450	3	4827.586207
16	1	1	700	1	3642.857143
17	1	0	440	1	6363.636364
18	1	0	510	1	4950.980392
19	1	1	510	1	4950.980392
20	2	0	2400	3	20000.000000
21	3	2	1540	3	5519.480519
22	1	0	600	1	7500.000000
23	4	2	3150	4	4761.904762
24	2	1	1250	2	5360.000000
25	4	1	2850	4	15017.543860
26	3	2	1630	3	11901.840490
27	2	3	1875	3	12533.333330
28	2	0	1200	3	10833.333330

In [14]:

```
y=dataset['price']
```

In [15]:

```
y
```

Out[15]:

```
0    150.00
1    149.00
2    150.00
3     40.00
4     83.00
5     40.00
6     65.00
7     43.00
8     36.00
9     76.00
10    58.00
11    75.00
12    47.00
13    40.00
14    27.00
15    70.00
16    25.50
17    28.00
18    25.25
19    25.25
20   480.00
21    85.00
22    45.00
23   150.00
24    67.00
25   428.00
26   194.00
27   235.00
28   130.00
```

Name: price, dtype: float64

In [16]:

```
print('Shape of X = ', x.shape)
print('Shape of y = ', y.shape)
```

Shape of X = (29, 5)

Shape of y = (29,)

## # Split Dataset into to training set and test dataset

In [17]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=51)
```

In [18]:

```
print('Shape of X_train = ', X_train.shape)
print('Shape of y_train = ', y_train.shape)
print('Shape of X_test = ', X_test.shape)
print('Shape of y_test = ', y_test.shape)
```

Shape of X\_train = (23, 5)

Shape of y\_train = (23,)

Shape of X\_test = (6, 5)

Shape of y\_test = (6,)

## # K Nearest Neighbor Regression - ML Model Training

In [19]:

```
from sklearn.neighbors import KNeighborsRegressor

regressor = KNeighborsRegressor(n_neighbors=3)
regressor.fit(X_train, y_train)
```

Out[19]:

KNeighborsRegressor(n\_neighbors=3)

In [20]:

```
regressor.score(X_test, y_test)
```

Out[20]:

0.8656000870057868

## # Predict the price of Home

In [21]:

X\_test

Out[21]:

	bath	balcony	total_sqft_int	bhk	price_per_sqft
15	2	2	1450	3	4827.586207
8	2	2	1130	2	3185.840708
2	3	2	1750	3	8571.428571
22	1	0	600	1	7500.000000
7	2	2	947	2	4540.654699
26	3	2	1630	3	11901.840490

In [22]:

```
X_test.iloc[-1, :]
```

Out[22]:

```
bath          3.000000
balcony        2.000000
total_sqft_int 1630.000000
bhk           3.000000
price_per_sqft 11901.84049
Name: 26, dtype: float64
```

In [23]:

```
regressor.predict([X_test.iloc[-1, :]])
```

Out[23]:

```
array([171.66666667])
```

In [24]:

```
y_test.iloc[-1]
```

Out[24]:

```
194.0
```

In [25]:

```
X_test.iloc[-2, :]
```

Out[25]:

```
bath          2.000000
balcony        2.000000
total_sqft_int  947.000000
bhk            2.000000
price_per_sqft 4540.654699
Name: 7, dtype: float64
```

In [26]:

```
regressor.predict([X_test.iloc[-2, :]])
```

Out[26]:

```
array([56.66666667])
```

In [27]:

```
y_test.iloc[-2]
```

Out[27]:

```
43.0
```

In [28]:

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

In [29]:

```
y_pred
```

Out[29]:

```
array([ 66.          , 35.16666667, 127.33333333,  86.66666667,  
       56.66666667, 171.66666667])
```

In [30]:

```
y_test
```

Out[30]:

```
15      70.0  
8       36.0  
2      150.0  
22      45.0  
7       43.0  
26     194.0  
Name: price, dtype: float64
```

In [ ]: