Business Problem - Predict the Price of Bangalore House

Using Linear Regression - Supervised Machine Learning Algorithm

Load Libraries

In [14]:

```
import pandas as pd
```

Load Data

In [19]:

```
path = r"https://docs.google.com/uc?export=download&id=1vV2oTGhe8FUm1GROMIFTsOHUJ5lry4XC"
df = pd.read_csv(path)
```

In [11]:

```

# df = pd.read_csv("bangalore house price prediction OHE-data.csv")
```

In [20]:

```
df.head()
```

Out[20]:

	bath	balcony	price	total_sqft_int	bhk	price_per_sqft	area_typeSuper built-up Area	area_typeBuilt-up Area	area_typePlot Area	avail
0	3.0	2.0	150.0	1672.0	3	8971.291866	1	0	0	
1	3.0	3.0	149.0	1750.0	3	8514.285714	0	1	0	
2	3.0	2.0	150.0	1750.0	3	8571.428571	1	0	0	
3	2.0	2.0	40.0	1250.0	2	3200.000000	1	0	0	
4	2.0	2.0	83.0	1200.0	2	6916.666667	0	0	1	

5 rows × 108 columns

Split Data

In [21]:

```
X = df.drop('price', axis=1)
y = df['price']

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

```
Shape of X =  (7120, 107)
Shape of y =  (7120,)
```

In [22]:
```
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)

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 =  (5696, 107)
Shape of y_train =  (5696,)
Shape of X_test =  (1424, 107)
Shape of y_test =  (1424,)
```

Feature Scaling

In [23]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
sc.fit(X_train)
X_train = sc.transform(X_train)
X_test = sc.transform(X_test)
```

Polynomial Regressio - ML Model Training

In [24]:
```
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
```

In [25]:
```
poly_reg=PolynomialFeatures(degree=2)
poly_reg.fit(X_train)
```

Out[25]:
```
PolynomialFeatures(degree=2, include_bias=True, interaction_only=False,
                   order='C')
```

In [26]:
```
X_train_poly=poly_reg.transform(X_train)
X_test_poly=poly_reg.transform(X_test)
```

In [27]:
```
X_train_poly.shape,X_test_poly.shape
```

Out[27]:
```
((5696, 5886), (1424, 5886))
```

In [29]:
```
lr=LinearRegression()
```

In [31]:
```
lr.fit(X_train_poly,y_train)
```

Out[31]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [32]:
```
lr.score(X_test_poly,y_test)
```

Out[32]:
```
0.9973664450488681
```

In [33]:
```
lr.predict([X_test_poly[0,:]])
```

Out[33]:
```
array([80.])
```

In [40]:
```
y_test
```

Out[40]:
```
2435     80.00
3113     40.00
426     120.00
1124     79.00
1161     45.00
         ...  
2078     28.34
6855     84.00
4381     32.00
3862     63.00
43      180.00
Name: price, Length: 1424, dtype: float64
```

In [44]:

```
y_predict=lr.predict(X_test_poly)
```

In [41]:

```
y_test
```

Out[41]:

```
2435     80.00
3113     40.00
426     120.00
1124     79.00
1161     45.00
          ... 
2078     28.34
6855     84.00
4381     32.00
3862     63.00
43      180.00
Name: price, Length: 1424, dtype: float64
```

In [46]:

```
from sklearn.metrics import mean_squared_error
import numpy  as np
mse=mean_squared_error(y_test,y_predict)
rmse=np.sqrt(mse)
mse, rmse
```

Out[46]:

```
(52.91585349268419, 7.274328387740287)
```