Business Problem - Predict the Price of Bangalore House

Using Support Vector Regression - Supervised Machine Learning Algorithm

Load Libraries

```
In [ ]: import pandas as pd
```

Load Data

```
In [ ]: path = r"https://drive.google.com/uc?export=download&id=1xxDtrZKfuWQfl-6KA9XEd_eatitNPnkB"
        df = pd.read_csv(path)
```

```
In [ ]: df.head()
```

Out[3]:

	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 [ ]:

```python
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 [ ]:

```python
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 [ ]:

```python
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
sc.fit(X_train)
X_train = sc.transform(X_train)
X_test = sc.transform(X_test)
```

Support Vector Regression - ML Model Training

```
In [ ]: from sklearn.svm import SVR
```

```
In [ ]: svr_rbf = SVR(kernel='rbf')
        svr_rbf.fit(X_train, y_train)
        svr_rbf.score(X_test, y_test)

Out[9]: 0.2063803584082815
```

```
In [ ]: svr_linear = SVR(kernel='linear')
        svr_linear.fit(X_train, y_train)
        svr_linear.score(X_test, y_test)

Out[10]: 0.796272530851755
```

```
In [ ]: svr_poly = SVR(kernel='poly',degree=2,)
        svr_poly.fit(X_train, y_train)
        svr_poly.score(X_test, y_test)

Out[11]: 0.18266215211874626
```

Predict the value of Home and Test

```
In [ ]: X_test[0]

Out[12]: array([ 0.71301986,  0.0112734 ,  0.30202307,  0.65677518, -0.48064341,
                -1.7385623 ,  2.11587407, -0.25430867,  0.51007548, -0.18373025,
                -0.16389438, -0.1473229 , -0.13023539, -0.12812824, -0.12598816,
                -0.12454231, -0.12953656, -0.12381344, -0.12010681, -0.11551113,
                -0.10992018, -0.10909925, -0.10660036, -0.11234866, -0.09315135,
                -0.08618799, -0.08923672, -0.09023078, -0.08721571, -0.09023078,
                -0.08721571, -0.08195215, -0.08195215, -0.07633675, -0.0751646 ,
                -0.08085949, -0.0739743 , -0.07975227, -0.07153563, -0.0751646 ,
                -0.0677166 , -0.08085949, -0.07153563, -0.07862985, -0.0751646 ,
                -0.07862985, -0.06504853, -0.0751646 , -0.06901264, -0.0751646 ,
                -0.06901264, -0.07028523, -0.07276497, -0.07028523, -0.06367332,
                -0.06226825, -0.06226825, -0.06639573, -0.06504853, -0.05935999,
                -0.06083125, -0.06639573, -0.06639573, -0.06226825, -0.06367332,
                -0.05935999, -0.06639573, -0.06367332, -0.06226825, -0.06226825,
                -0.05935999, -0.05935999, -0.05935999, -0.05630391, -0.05935999,
                -0.05785186, -0.05935999, -0.05935999, -0.06083125, -0.06083125,
                -0.05471275, -0.06083125, -0.06226825, -0.05935999, -0.05935999,
                -0.06226825, -0.06226825, -0.05785186, -0.06504853, -0.06226825,
                -0.06083125, -0.05935999, -0.05307449, -0.05630391, -0.06226825,
                -0.05471275, -0.05935999, -0.05471275, -0.05471275, -0.05138463,
                -0.05307449, -0.05307449, -0.05471275, -0.05471275, -0.05630391,
                -0.05630391, -0.05138463])
```

```
In [ ]: svr_linear.predict([X_test[0]])

Out[13]: array([86.53673464])
```

```
In [ ]: y_pred = svr_linear.predict(X_test)
        y_pred

Out[14]: array([ 86.53673464,  33.70390806, 122.48356838, ...,  26.23978753,
                 64.52634655, 176.17532793])
```

In [ ]:

```
y_test
```

Out[15]:

```
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 [ ]:

```
from sklearn.metrics import mean_squared_error
import numpy as np

mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

print('MSE = ', mse)
print('RMSE = ', rmse)
```

```
MSE =  4093.4831852475745
RMSE =  63.9803343633618
```

Ab milenge next tutorial me,Tab tak ke liye SIKHATE SIKHATE kuch IMPLEMENT karte raho, Thank You.....-:)