# Support Vector Regression (SVR)

## Importing the libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

## Importing the dataset

```
dataset = pd.read_csv('Data.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
```

```
y = y.reshape(len(y),1)
```

## Splitting the dataset into the Training set and Test set

```
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_
```

```
print(X_train)
```

```
[[  11.22   43.13 1017.24   80.9 ]
 [  13.67   54.3  1015.92   75.42]
 [  32.84   77.95 1014.68   45.8 ]
 ...
 [  16.81   38.52 1018.26   75.21]
 [  12.8    41.16 1022.43   86.19]
 [  32.32   67.9  1006.08   37.93]]
```

```
print(y_train)
```

```
[[473.93]
 [467.87]
 [431.97]
 ...
 [459.01]
 [462.72]
 [428.12]]
```

```
print(X_test)
```

```
[[  28.66   77.95 1009.56   69.07]
 [  17.48   49.39 1021.51   84.53]
 [  14.86   43.14 1019.21   99.14]
 ...
 [  12.24   44.92 1023.74   88.21]
 [  27.28   47.93 1003.46   59.22]
 [  17.28   39.99 1007.09   74.25]]
```

```
print(y_test)
```

```
[[431.23]
 [460.01]
 [461.14]
 ...
 [473.26]
 [438.  ]
 [463.28]]
```

## Feature Scaling

```
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
sc_y = StandardScaler()
X_train = sc_X.fit_transform(X_train)
y_train = sc_y.fit_transform(y_train)
```

## Training the SVR model on the Training set

```
from sklearn.svm import SVR
regressor = SVR(kernel = 'rbf')
regressor.fit(X_train, y_train)
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:1408: DataConve
  y = column_or_1d(y, warn=True)
```

```
▾ SVR  ⓘ ?

SVR()
```

## Predicting the Test set results

```
y_pred = sc_y.inverse_transform(regressor.predict(sc_X.transform(X_test)).reshape(
np.set_printoptions(precision=2)
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)
```

```
[[434.05 431.23]
 [457.94 460.01]
 [461.03 461.14]
 ...
 [470.6  473.26]
 [439.42 438.  ]
 [460.92 463.28]]
```

## Evaluating the Model Performance

```
from sklearn.metrics import r2_score
r2_score(y_test, y_pred)
```

```
0.9480784049986258
```