

Support Vector Regression (SVR)

Importing the libraries

```
In [2]: import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Importing the dataset

```
In [3]: dataset = pd.read_csv('Position_Salaries.csv')
x = dataset.iloc[:, 1:-1].values
y = dataset.iloc[:, -1].values
```

```
In [4]: print(x)
```

```
[[ 1]
 [ 2]
 [ 3]
 [ 4]
 [ 5]
 [ 6]
 [ 7]
 [ 8]
 [ 9]
[10]]
```

```
In [5]: print(y)
```

```
[ 45000  50000  60000  80000 110000 150000 200000 300000 500000
1000000]
```

```
In [6]: y = y.reshape(len(y), 1)
```

```
In [7]: print(y)
```

```
[[ 45000]
 [ 50000]
 [ 60000]
 [ 80000]
 [110000]
 [150000]
 [200000]
 [300000]
 [500000]
[1000000]]
```

Feature Scaling

```
In [8]: from sklearn.preprocessing import StandardScaler
x_scale = StandardScaler()
y_scale = StandardScaler()
x = x_scale.fit_transform(x)
y = y_scale.fit_transform(y)
```

```
In [9]: print(x)
```

```
[[-1.5666989 ]
 [-1.21854359]
 [-0.87038828]
 [-0.52223297]
 [-0.17407766]
 [ 0.17407766]
 [ 0.52223297]
 [ 0.87038828]
 [ 1.21854359]
 [ 1.5666989 ]]
```

```
In [10]: print(y)
```

```
[[-0.72004253]
 [-0.70243757]
 [-0.66722767]
 [-0.59680786]
 [-0.49117815]
 [-0.35033854]
 [-0.17428902]
 [ 0.17781001]
 [ 0.88200808]
 [ 2.64250325]]
```

Training the SVR model on the whole dataset

```
In [11]: from sklearn.svm import SVR
model = SVR(kernel='rbf')
model.fit(x, y)
```

C:\Users\nirma_000\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

```
Out[11]: SVR()
```

Predicting a new result

```
In [12]: y_scale.inverse_transform([model.predict(x_scale.transform([[6.5]]))])
```

```
Out[12]: array([[170370.0204065]])
```

Visualising the SVR results

```
In [13]: plt.scatter(x_scale.inverse_transform(x), y_scale.inverse_transform(y), color = 'red')
dummy = y_scale.inverse_transform([model.predict(x)])
dummy = dummy.reshape(len(y), 1)
plt.plot(x_scale.inverse_transform(x), dummy, color = 'blue')
plt.title('Truth or Bluff (SVR)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```

