

SUPPORT VECTOR REGRESSION

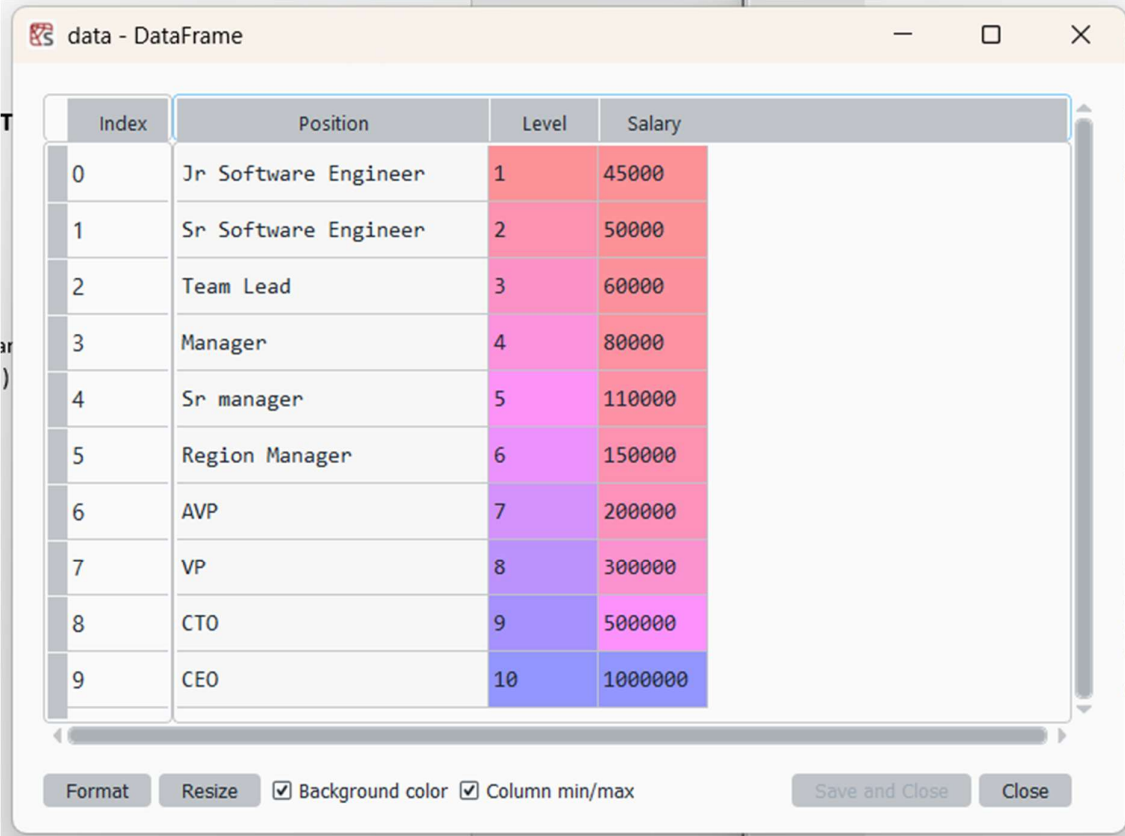
Importing Libraries

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

let's import the dataset

```
data=pd.read_csv(r"C:\Users\TharunMahendra\NIT\6.Algorithms\1.  
Regression\Employee-Salary.csv")
```



Index	Position	Level	Salary
0	Jr Software Engineer	1	45000
1	Sr Software Engineer	2	50000
2	Team Lead	3	60000
3	Manager	4	80000
4	Sr manager	5	110000
5	Region Manager	6	150000
6	AVP	7	200000
7	VP	8	300000
8	CTO	9	500000
9	CEO	10	1000000

#lets divide the dataset into independent and dependent variables

```
x=data.iloc[:,1:2].values
```

```
y=data.iloc[:,2].values
```

The image shows two NumPy array viewer windows. The first window, titled 'x - NumPy object array', displays a 1D array of integers from 1 to 10. The second window, titled 'y - NumPy object array', displays a 1D array of values: 45000, 50000, 60000, 80000, 110000, 150000, 200000, 300000, 500000, 1000000.

	0
0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9
9	10

	0
0	45000
1	50000
2	60000
3	80000
4	110000
5	150000
6	200000
7	300000
8	500000
9	1000000

#fitting the SVR model to the dataset

from sklearn.svm import SVR

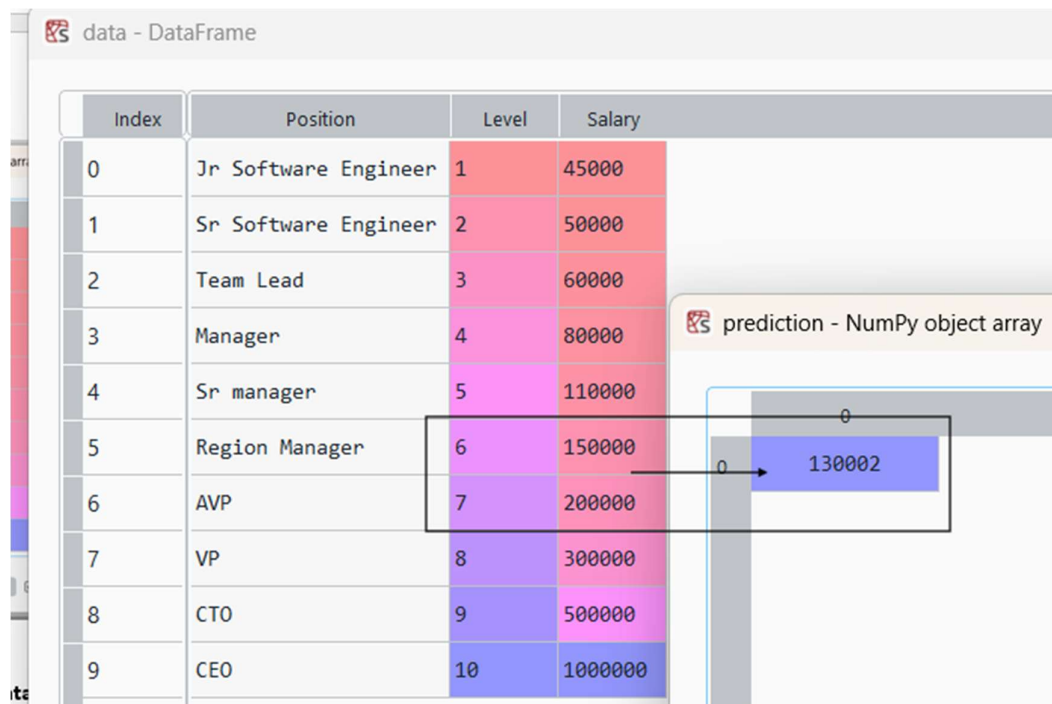
model=SVR() **Parameter Tuning**

model.fit(x,y)

#predicting a new result

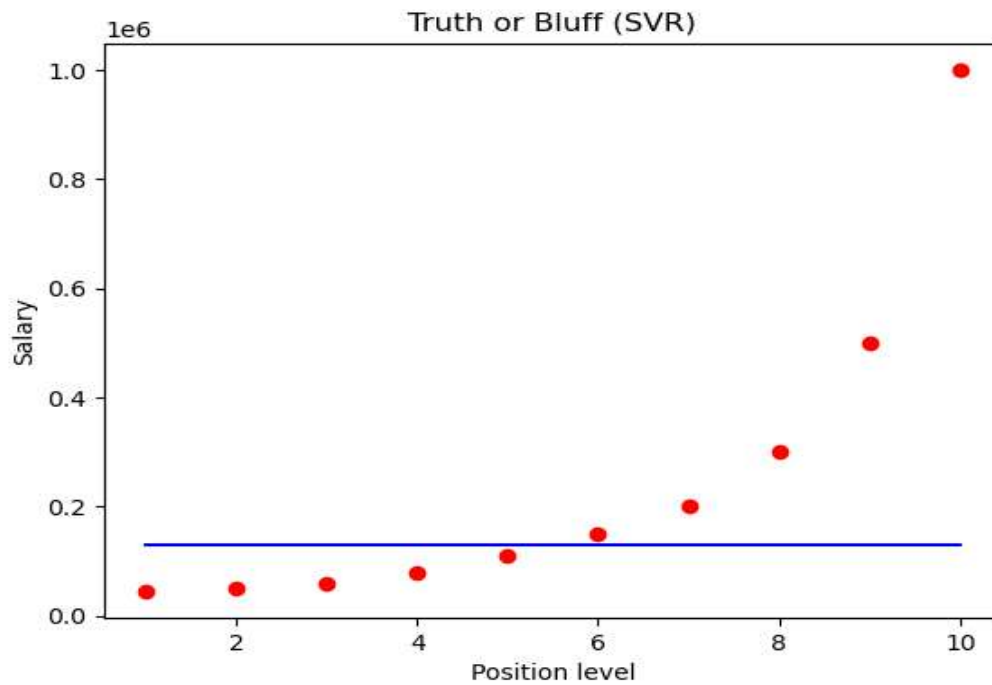
prediction=model.predict([[6.5]])

print(prediction)



Visualising the SVR results

```
plt.scatter(x, y, color = 'red')  
plt.plot(x, model.predict(x), color = 'blue')  
plt.title('Truth or Bluff (SVR)')  
plt.xlabel('Position level')  
plt.ylabel('Salary')  
plt.show()
```



hyperParameter tuning

```
model=SVR(kernel='poly',degree=4,gamma='auto')
```

data - DataFrame

Index	Position	Level	Salary
0	Jr Software Engineer	1	45000
1	Sr Software Engineer	2	50000
2	Team Lead	3	60000
3	Manager	4	80000
4	Sr manager	5	110000
5	Region Manager	6	150000
6	AVP	7	200000
7	VP	8	300000
8	CTO	9	500000
9	CEO	10	1000000

prediction - NumPy object array

0 → 175708

