IDS PROJECT

We Have used **SVR** (Support Vector Regression) Machine Learning technique to apply regression on the Dataset of Position and Salaries of employees on the basis of the Level of job in the company.

**SVR** uses the same basic idea as **Support Vector Machine (SVM**), a classification algorithm, but applies it to predict real values rather than a class. SVR acknowledges the presence of non-linearity in the data and provides a proficient prediction model.

Lets start the Code Explanation:

**Step 1:**

**Importing Libraries** of machine learning which are “Numpy” , “Pandas” and “Matplotlib.pyplot”

|  |
| --- |
| *import numpy as np*  *import pandas as pd*  *import matplotlib.pyplot as plt* |

**Step 2:**

**Importing** **Dataset** and Splitting dataset into **Dependent and Independent Variables.** we have used pandas library for splitting the dataset into dependent and independent variables

|  |
| --- |
| *dataset = pd.read\_csv("Position\_Salaries.csv")*  *print(dataset)*  *X = dataset.iloc[:,1:-1].values*  *y = dataset.iloc[:,-1].values* |

we did not break dataset into train data and test data because Data set is small and we will provide test values randomly after the model training

**Step 3:**

**Reshaping** the dependent variable into 2D array by using python’s reshape() method for further processing.

|  |
| --- |
| *y = y.reshape(len(y), 1)* |

**Step 4:**

**Feature Scaling:** we have used scikitlearn library for the feature scaling and transforming the values of the salary class into range of -3 to +3 by using the StandardScaler class

|  |
| --- |
| *from sklearn.preprocessing import StandardScaler*  *sc\_X = StandardScaler()*  *sc\_y = StandardScaler()*  *X\_transformed = sc\_X.fit\_transform(X)*  *y\_transformed = sc\_y.fit\_transform(y)* |

**Step 5:**

**Training the SVR model:** we have used SVR class of svm module of scikitlearn library and will train the model

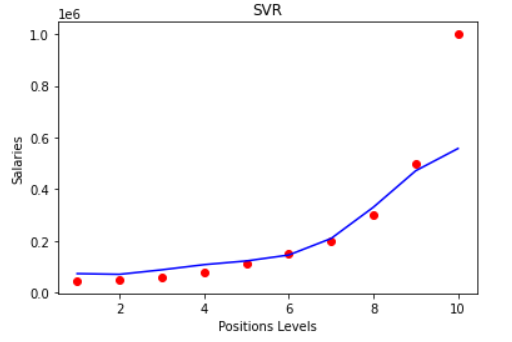
|  |
| --- |
| *from sklearn.svm import SVR*  *regressor = SVR(kernel = 'rbf')*  *regressor.fit(X\_transformed,y\_transformed)* |

**Step 6:**

**Predicting:** we will use 6.5 value to get the prediction from our model

|  |
| --- |
| *sc\_y.inverse\_transform(regressor.predict(sc\_X.transform([[6.5]])))* |

**Step 7:**

**Visualizing the Results:**

|  |
| --- |
| *plt.scatter(X,y, color='r')*  *plt.plot(X,sc\_y.inverse\_transform(regressor.predict(sc\_X.transform(X))) , color = 'b')*  *plt.title("SVR")*  *plt.xlabel("Positions Levels")*  *plt.ylabel("Salaries")*  *plt.show()* |

**Step 8:**

**Visualizing the Result with more smooth Curve:**

|  |
| --- |
| *X\_grid = np.arange(min(X), max(X), 0.1 )*  *print(X\_grid)*  *X\_grid = X\_grid.reshape((len(X\_grid), 1))*  *print("AFTER")*  *print(X\_grid)*  *plt.scatter(X,y, color='r')*  *plt.plot(X\_grid,sc\_y.inverse\_transform(regressor.predict(sc\_X.transform(X\_grid))) , color = 'b')*  *plt.title("SVR")*  *plt.xlabel("Positions Levels")*  *plt.ylabel("Salaries")*  *plt.show()* |

