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


dataset = pd.read_csv('https://raw.githubusercontent.com/amppmann/Machine-Learning-SourceCodes/main/Position_Salaries.csv')
x = dataset.iloc[:, 1:-1].values
y = dataset.iloc[:, -1].values

print("\nx : ",x)
print("\ny : ",y)



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


print("\nReshaped y : ",y)



from sklearn.preprocessing import StandardScaler
sc_x = StandardScaler()
sc_y = StandardScaler()

X = sc_x.fit_transform(x)

Y = sc_y.fit_transform(y)


print("Transformed X : ",X)
print("Transformed Y : ",Y)



from sklearn.svm import SVR
regressor = SVR(kernel = 'rbf')
regressor.fit(X, Y)






print("New Value")
print(sc_y.inverse_transform(regressor.predict(sc_x.transform([[6.5]])).reshape(-1,1)))
# Visualising the SVR results
plt.scatter(sc_x.inverse_transform(X), sc_y.inverse_transform(y).reshape(-1,1), color = 'red')
plt.plot(sc_x.inverse_transform(X), sc_y.inverse_transform(regressor.predict(X).reshape(-1,1)), color = 'blue')
plt.title('Truth or Bluff (SVR)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()

# Visualising the SVR results (for higher resolution and smoother curve)


X_grid = np.arange(min(sc_x.inverse_transform(X)), max(sc_x.inverse_transform(X)), 0.1)
X_grid = X_grid.reshape((len(X_grid), 1))
plt.scatter(sc_x.inverse_transform(X), sc_y.inverse_transform(y).reshape(-1,1), color = 'red')
plt.plot(X_grid, sc_y.inverse_transform(regressor.predict(sc_x.transform(X_grid)).reshape(-1,1)), color = 'blue')
plt.title('Truth or Bluff (SVR)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```

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

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

Reshaped y :  [[ 45000]
 [ 50000]
 [ 60000]
 [ 80000]
 [110000]
 [150000]
 [200000]
 [300000]
 [500000]
 [1000000]]
Transformed X :  [[-1.5666989 ]
 [-1.21854359]
 [-0.87038828]
 [-0.52223297]
 [-0.17407766]
 [ 0.17407766]
 [ 0.52223297]
 [ 0.87038828]
 [ 1.21854359]
 [ 1.5666989 ]]
Transformed Y :  [[-0.72004253]
 [-0.70243757]
 [-0.66722767]
 [-0.59680786]
 [-0.49117815]
 [-0.35033854]
 [-0.17428902]
 [ 0.17781001]
 [ 0.88200808]
 [ 2.64250325]]
New Value
/usr/local/lib/python3.9/dist-packages/sklearn/utils/validation.py:1143: DataConversionWarning: A column-vector y wa
  y = column_or_1d(y, warn=True)
[[170370.0204065]]
```