SUPPORT VECTOR REGRESSION

- Nguyễn Hoàng Yến Như
- Nguyễn Trần Phúc Nghi
- Nguyễn Trần Phúc An
- Nguyễn Đức Anh Phúc
- Trịnh Thị Thanh Trúc
- KS. Cao Bá Kiệt
- KS. Quan Chí Khánh An
- KS. Lê Ngọc Huy
- CN. Bùi Cao Doanh
- CN. Nguyễn Trọng Thuận
- KS. Phan Vĩnh Long
- KS. Nguyễn Cường Phát
- ThS. Nguyễn Hoàng Ngân
- KS. Hồ Thái Ngọc
- ThS. Đỗ Văn Tiến
- ThS. Nguyễn Hoàn Mỹ
- ThS. Dương Phi Long
- ThS. Trương Quốc Dũng
- ThS. Nguyễn Thành Hiệp
- ThS. Nguyễn Võ Đăng Khoa
- ThS. Võ Duy Nguyên
- TS. Nguyễn Văn Tâm
- ThS. Trần Việt Thu Phương
- TS. Nguyễn Tấn Trần Minh Khang

DATASET

Dataset

- Tên tập dữ liệu: Position Salaries.
- Nguồn: https://www.superdatascience.com/pages/machine-learning.
- Tập dữ liệu gồm 10 điểm dữ liệu, mỗi điểm dữ liệu gồm 3 thuộc tính, gồm:
  + Vị trí công việc (Position): mô tả tên một công việc.
  + Cấp bậc (Level): là một số nguyên trong khoảng 1 – 10, tương ứng với vị trí cao hay thấp trong một công ty.
  + Mức lương (Salary): là một số thực dương.

Dataset

Position	Level	Salary
Business Analyst	1	45,000
Junior Consultant	2	50,000
Senior Consultant	3	60,000
Manager	4	80,000
Country Manager	5	110,000

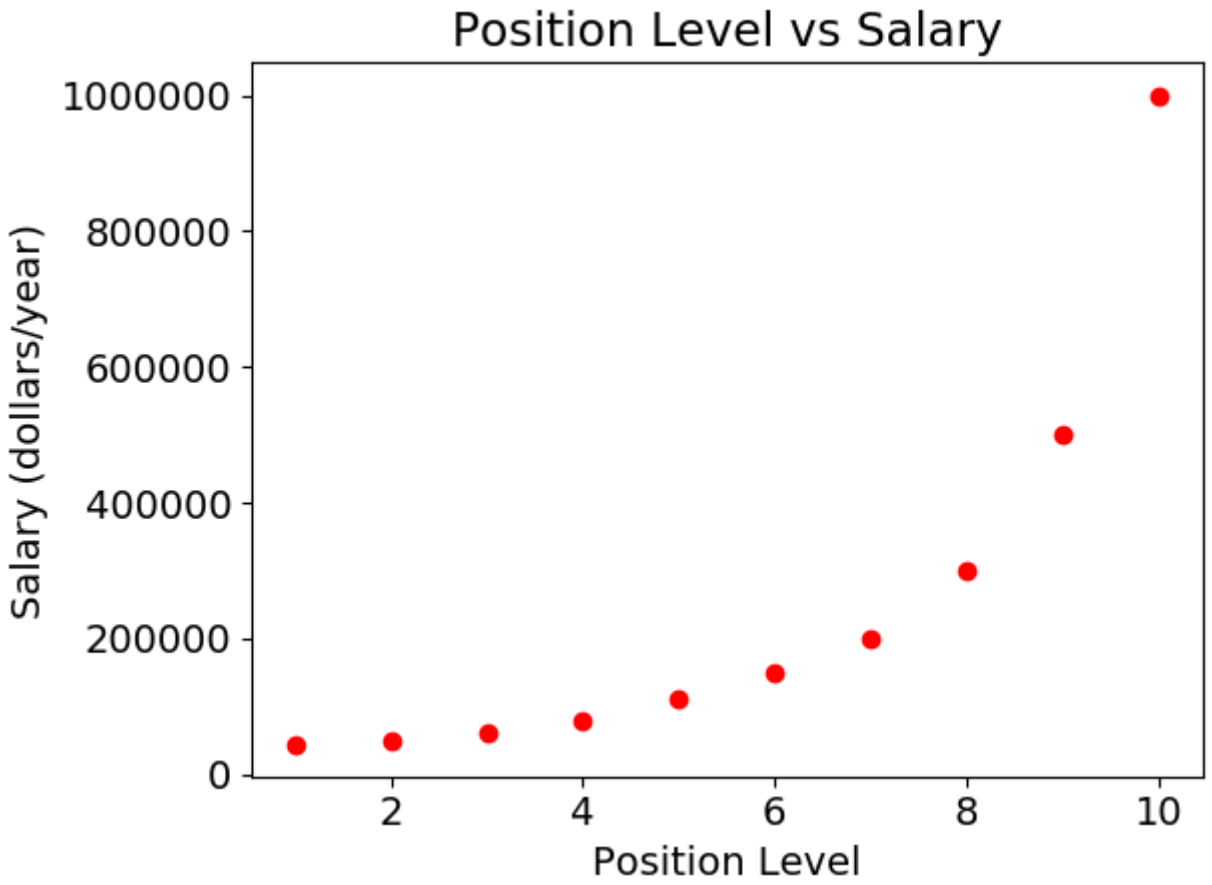
Position	Level	Salary
Region Manager	6	150,000
Partner	7	200,000
Senior Partner	8	300,000
C-level	9	500,000
CEO	10	1,000,000

Dataset

- Bài toán: Dự đoán mức lương của một người khi biết được cấp độ (vị trí) công việc của người đó.
- Ta sẽ sử dụng mô hình Support Vector Regression để giải quyết bài toán này.

TRỰC QUAN HÓA DỮ LIỆU

Trực quan hóa dữ liệu



Trực quan hóa dữ liệu

– Đọc dữ liệu từ file csv và phân tách các giá trị

+ Giá trị đầu vào – ký hiệu là X

+ Giá trị đầu ra – ký hiệu là Y.

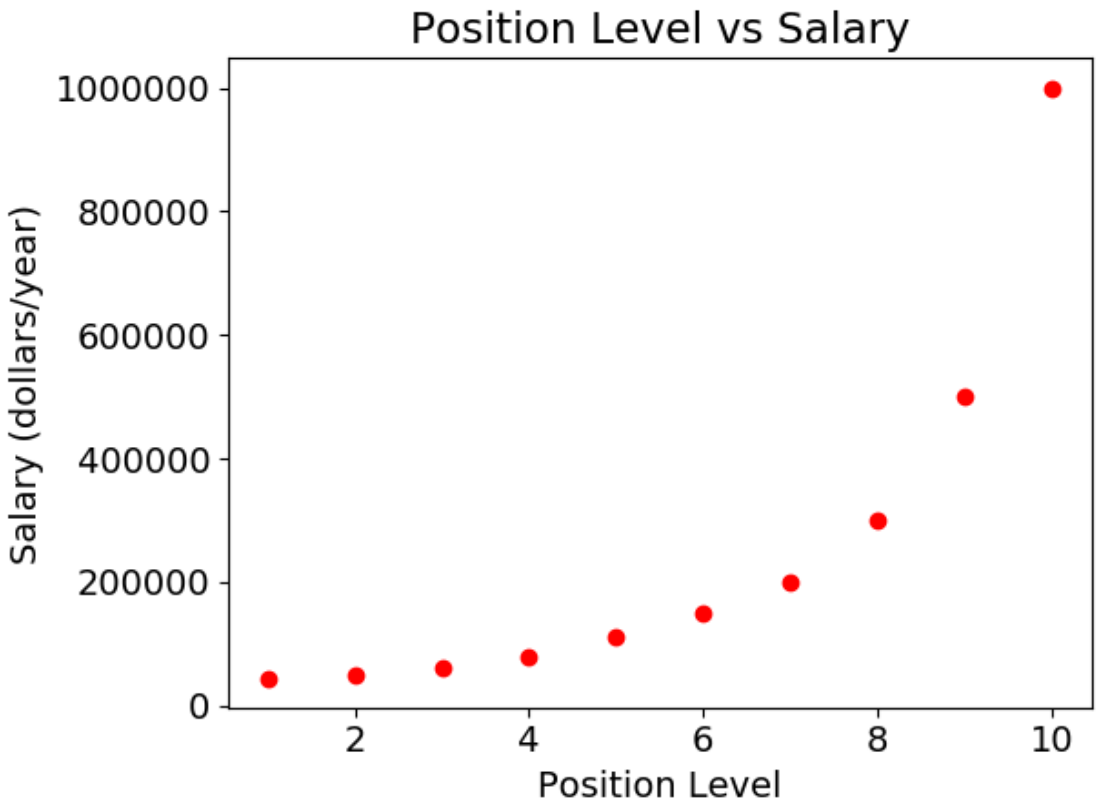
```
1. import pandas as pd
2. dataset = pd.read_csv("Position_Salaries.csv")
3. X = dataset.iloc[:, 1:-1].values
4. Y = dataset.iloc[:, -1].values.reshape(-1,1)
```

Trực quan hóa dữ liệu

- Ta vẽ các điểm `(level, salary)` lên mặt phẳng tọa độ để xem xét sự tương quan giữa cấp độ công việc và mức lương.

```
5. import matplotlib.pyplot as plt
6. plt.scatter(X, Y, color = "red")
7. plt.title("Position Level vs Salary")
8. plt.xlabel("Position Level")
9. plt.ylabel("Salary (dollars/year)")
10.plt.show()
```

Trực quan hóa dữ liệu



- Tập dữ liệu này không có dạng một đường thẳng.
- Do đó, Linear Regression sẽ không hoạt động tốt trên tập dữ liệu này.

TIỀN XỬ LÝ DỮ LIỆU

Tiền xử lý dữ liệu

– Trong thuật toán SVR, dữ liệu nên thỏa mãn 2 điều kiện sau:

+ Kỳ vọng bằng 0.

+ Phương sai bằng 1.

Do đó, ta cần chuẩn hóa dữ liệu trước khi huấn luyện mô hình.

Tiền xử lý dữ liệu

– Lớp `StandardScaler` trong module `sklearn.preprocessing` đã được xây dựng sẵn để chuẩn hóa dữ liệu về dạng trên.

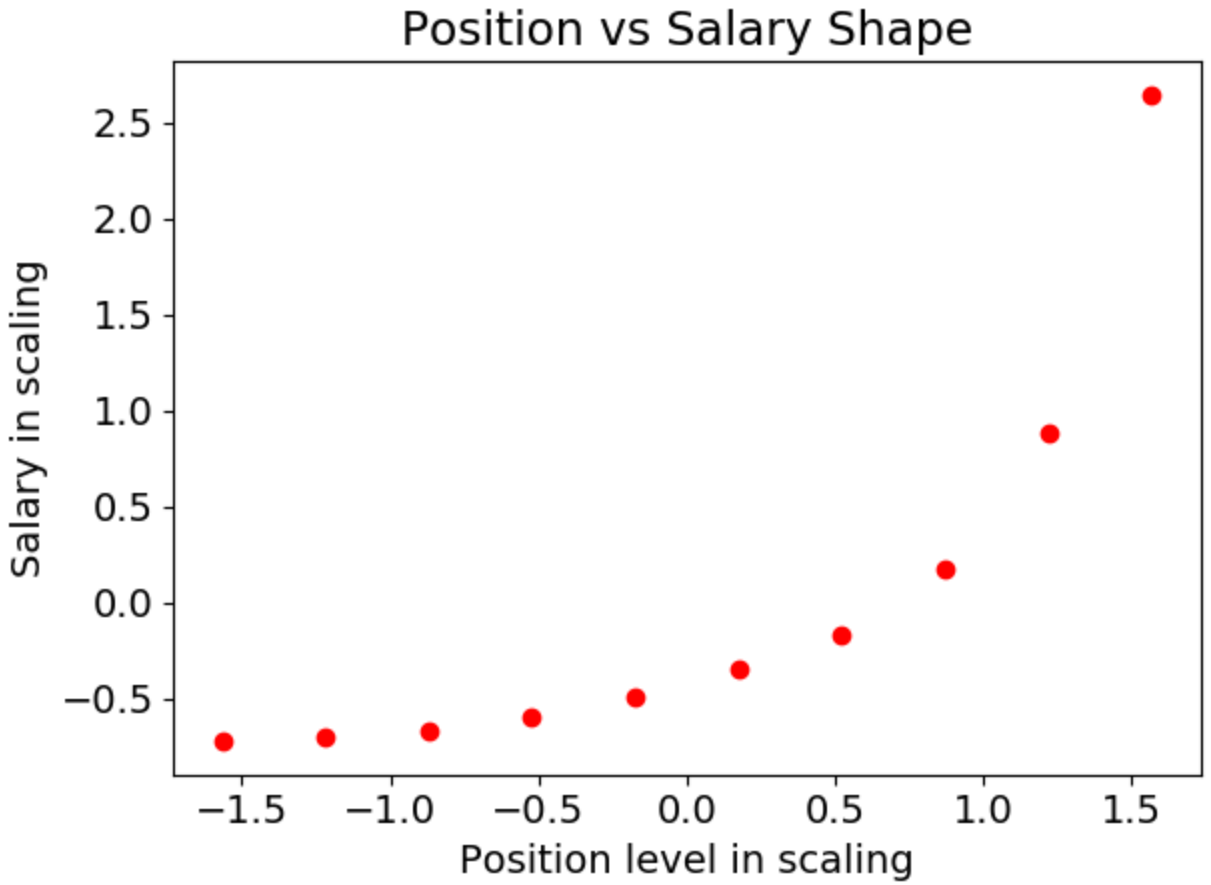
```
11.from sklearn.preprocessing import StandardScaler
12.SC_X = StandardScaler()
13.SC_Y = StandardScaler()
14.X_trans = SC_X.fit_transform(X)
15.Y_trans = SC_Y.fit_transform(Y)
```

Tiền xử lý dữ liệu

– Trực quan hóa dữ liệu đã chuẩn hóa.

```
16.plt.scatter(X_trans, Y_trans, color = "red")
17.plt.title("Position vs Salary Shape")
18.plt.xlabel("Position level in scaling")
19.plt.ylabel("Salary in scaling")
20.plt.show()
```

Tiền xử lý dữ liệu



SUPPORT VECTOR REGRESSION

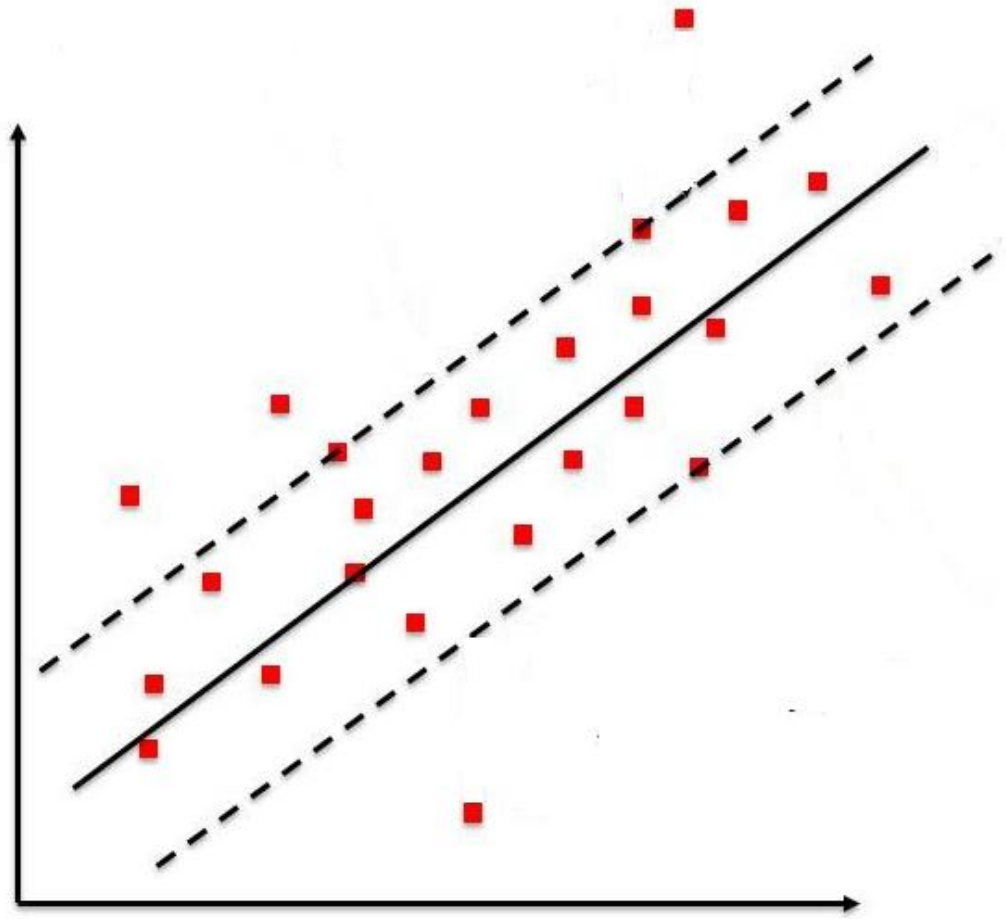
Support Vector Regression

- Support Vector Regression (SVR) là một thuật toán Regression (hồi quy) dựa trên Support Vector Machine.
- Thuật toán này được sử dụng cho cả 2 loại dữ liệu:
  + Dữ liệu có phân phối tuyến tính.
  + Dữ liệu không có phân phối tuyến tính (phi tuyến).

Support Vector Regression

- Thuật toán SVR (hay SVM) sẽ tìm một số vector đặc biệt (gọi là support vectors).
- Mô hình (Model) dự đoán (predict) kết quả đầu ra của những điểm dữ liệu mới dựa trên các vector đặc biệt (support vectors) này.

Support Vector Regression



Support Vector Regression

- Thuật toán SVR chuẩn, chỉ có thể dự đoán trên tập dữ liệu có phân phối tuyến tính.
- Tuy nhiên, các thuật toán cải tiến của SVR, gọi là kernel-SVR, có thể hoạt động tốt trên cả những dữ liệu phi tuyến.

Support Vector Regression

– Một vài loại kernel-SVR thường được sử dụng:

+ Linear kernel-SVR: kernel mặc định của SVR, chỉ sử dụng được cho tập dữ liệu có phân phối tuyến tính.

+ Polynomial kernel-SVR.

+ Sigmoid kernel-SVR.

+ Radial Basis Function kernel-SVR.

HUẤN LUYỆN MÔ HÌNH

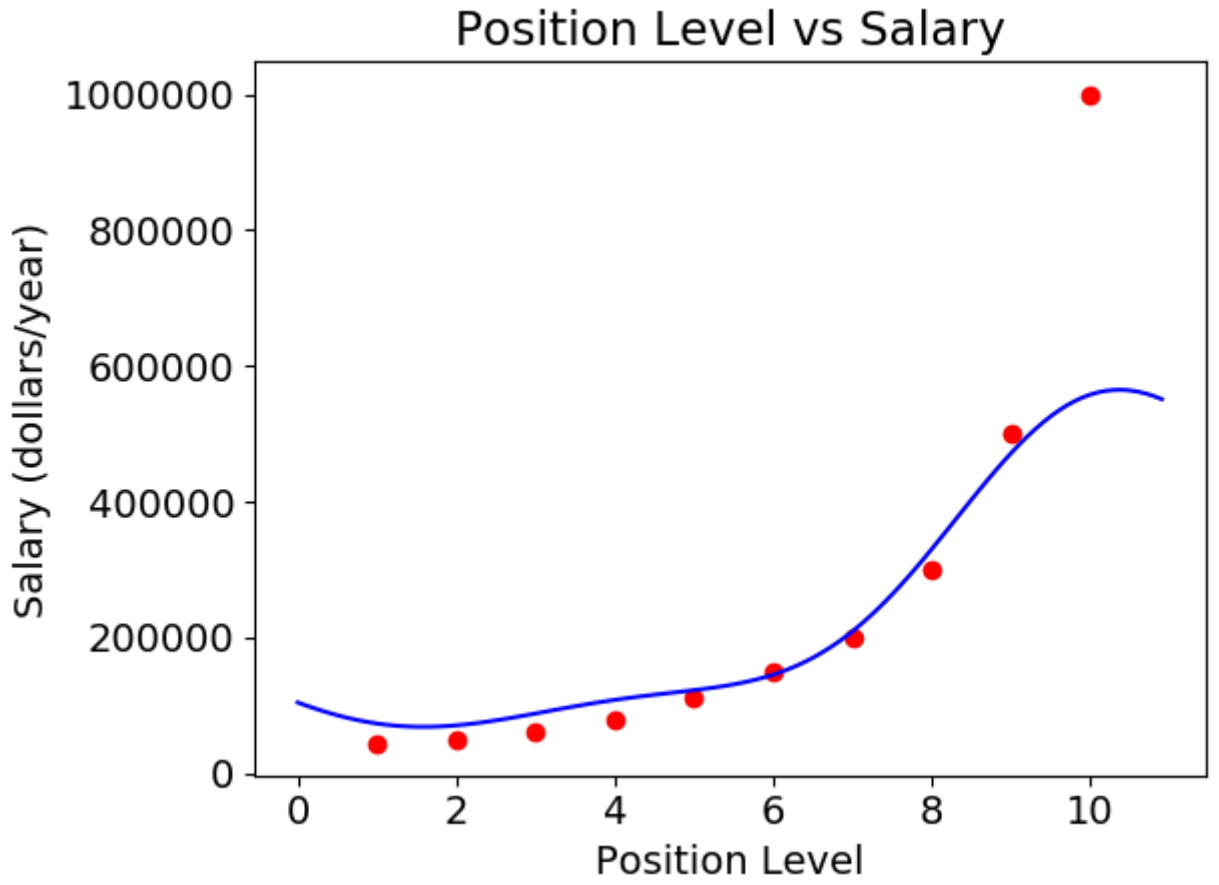
Huấn luyện mô hình

- Ta sử dụng Radius Basis Function kernel-SVR cho bài toán này.
- Lớp SVR của module sklearn.svm đã được xây dựng sẵn để huấn luyện các mô hình kernel SVR.

```
21.from sklearn.svm import SVR
22.svr = SVR(kernel = "rbf")
23.svr.fit(X_trans, Y_trans)
```

TRỰC QUAN HÓA KẾT QUẢ MÔ HÌNH

Trực quan hóa kết quả mô hình



Trực quan hóa kết quả mô hình

- Vì dữ liệu huấn luyện của chúng ta đã được chuẩn hóa, nên ta cần định nghĩa lại dữ liệu cho phù hợp khi dự đoán.

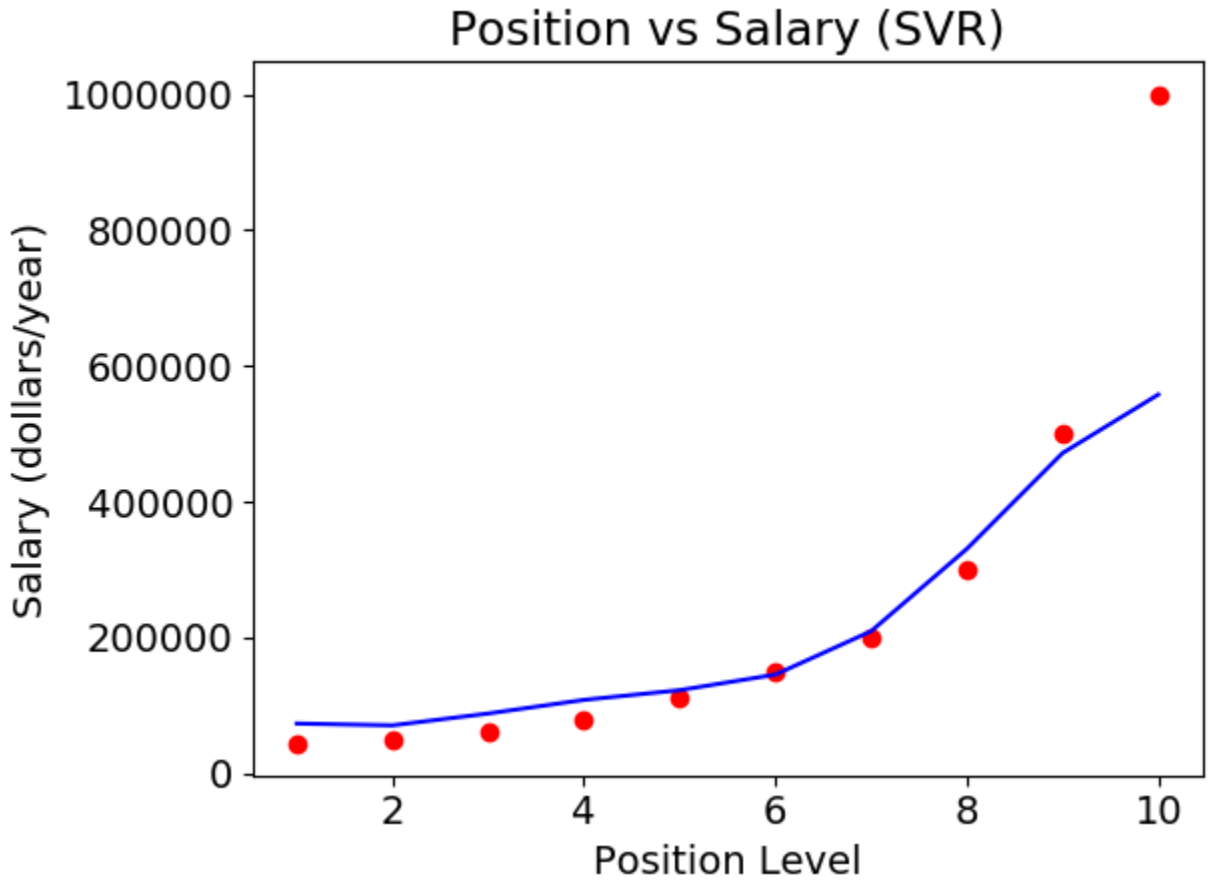
```
24.def predict(model, X, SC_X, SC_Y):
25.    X_trans = SC_X.transform(X)
26.    Y_trans_pred = model.predict(X_trans)
27.    Y_pred = SC_Y.inverse_transform(Y_trans_pred)
28.    return Y_pred
29.Y_pred = predict(svr, X, SC_X, SC_Y)
```

Trực quan hóa kết quả mô hình

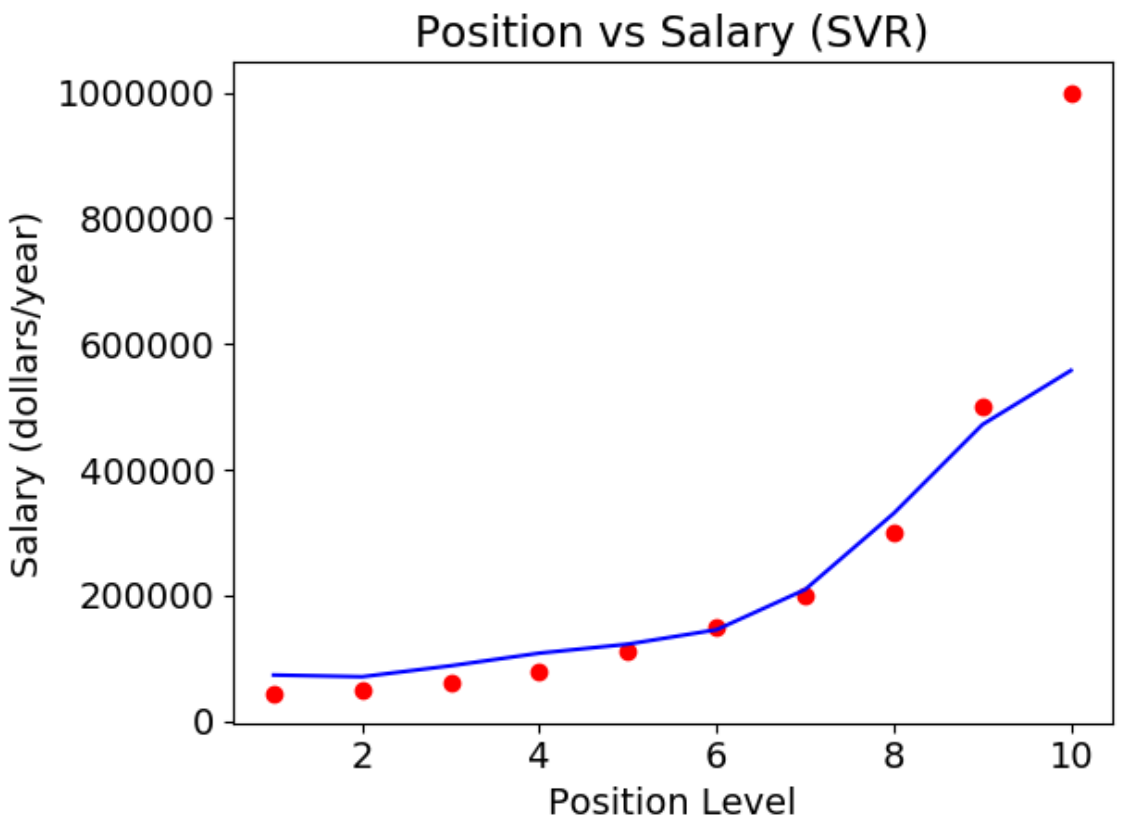
– Trực quan hóa kết quả trên mặt phẳng tọa độ.

```
30.plt.scatter(X, Y, color = "red")
31.plt.plot(X, Y_pred, color = "blue")
32.plt.title("Position vs Salary")
33.plt.xlabel("Position Level")
34.plt.ylabel("Salary (dollars/year)")
35.plt.show()
```

Trực quan hóa kết quả mô hình



Trực quan hóa kết quả mô hình



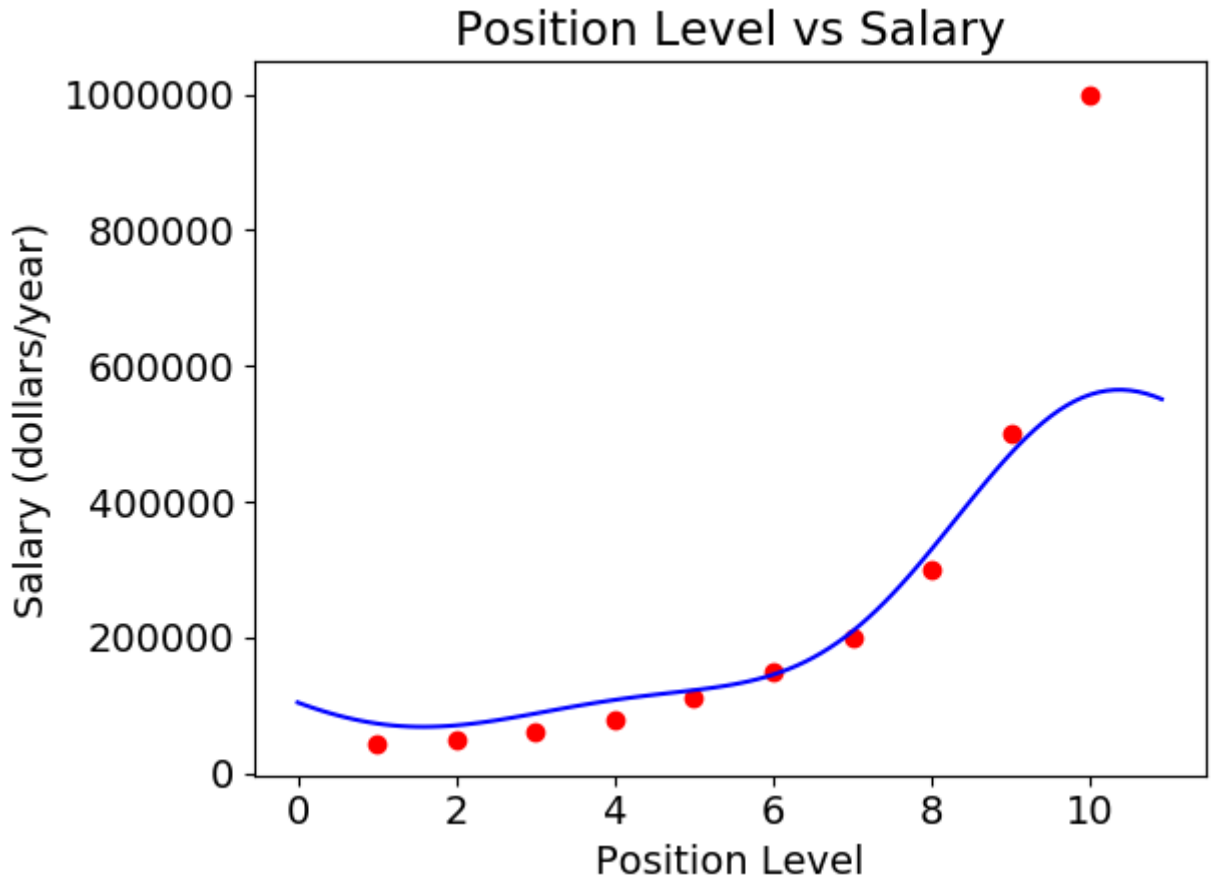
- Mô hình SVR dự đoán không chính xác điểm dữ liệu cuối cùng (level=10).
- Các điểm còn lại dự đoán khá chính xác.

Trực quan hóa kết quả mô hình

– Vẽ lại đồ thị kết quả của mô hình.

```
36.import numpy as np
37.X_dummy = np.arange(0, 10, 0.1).reshape(-1, 1)
38.Y_dummy_pred = predict(svr, X_dummy, SC_X, SC_Y)
39.plt.scatter(X, Y, color = "red")
40.plt.plot(X_dummy, Y_dummy_pred, color = "blue")
41.plt.title("Position Level vs Salary")
42.plt.xlabel("Position Level")
43.plt.ylabel("Salary (dollars/year)")
44.plt.show()
```

Trực quan hóa kết quả mô hình



Trực quan hóa kết quả mô hình

– Xây dựng hàm so sánh kết quả trên một điểm dữ liệu trong tập traning.

```
45.def compare(i_example):
46.     x = X[i_example : i_example + 1]
47.     y = Y[i_example]
48.     y_pred = predict(svr, x, SC_X, SC_Y)
49.     print(x, y, y_pred)
```

Trực quan hóa kết quả mô hình

– Gọi thực hiện hàm so sánh kết quả trên toàn bộ tập training.

```
50.for i in range(len(X)):
51.     compare(i)
```

Trực quan hóa kết quả

Position	Level	Salary	Predicted Salary
Business Analyst	1	45,000	73,474
Junior Consultant	2	50,000	70,786
Senior Consultant	3	60,000	88,213
Manager	4	80,000	108,254
Country Manager	5	110,000	122,574

Trực quan hóa kết quả

Position	Level	Salary	Predicted Salary
Region Manager	6	150,000	145,503
Partner	7	200,000	209,410
Senior Partner	8	300,000	330,606
C-level	9	500,000	471,671
CEO	10	1,000,000	557,821

Chúc các bạn học tốt

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN TP.HCM

Nhóm UIT-Together

Nguyễn Tấn Trần Minh Khang