RANDOM FOREST REGRESSION

- Nguyễn Hoàng Yến Như
- Nguyễn Trần Phúc Nghi
- Nguyễn Trần Phúc An
- Nguyễn Đức Anh Phúc
- Trịnh Thị Thanh Trúc
- ThS. Nguyễn Hữu Lợi
- 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 bằng cách sử dụng thuật toán Rừng ngẫu nhiên – Random Forest Regression.

TIỀN XỬ LÝ DỮ LIỆU

Tiền xử lý dữ liệu

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

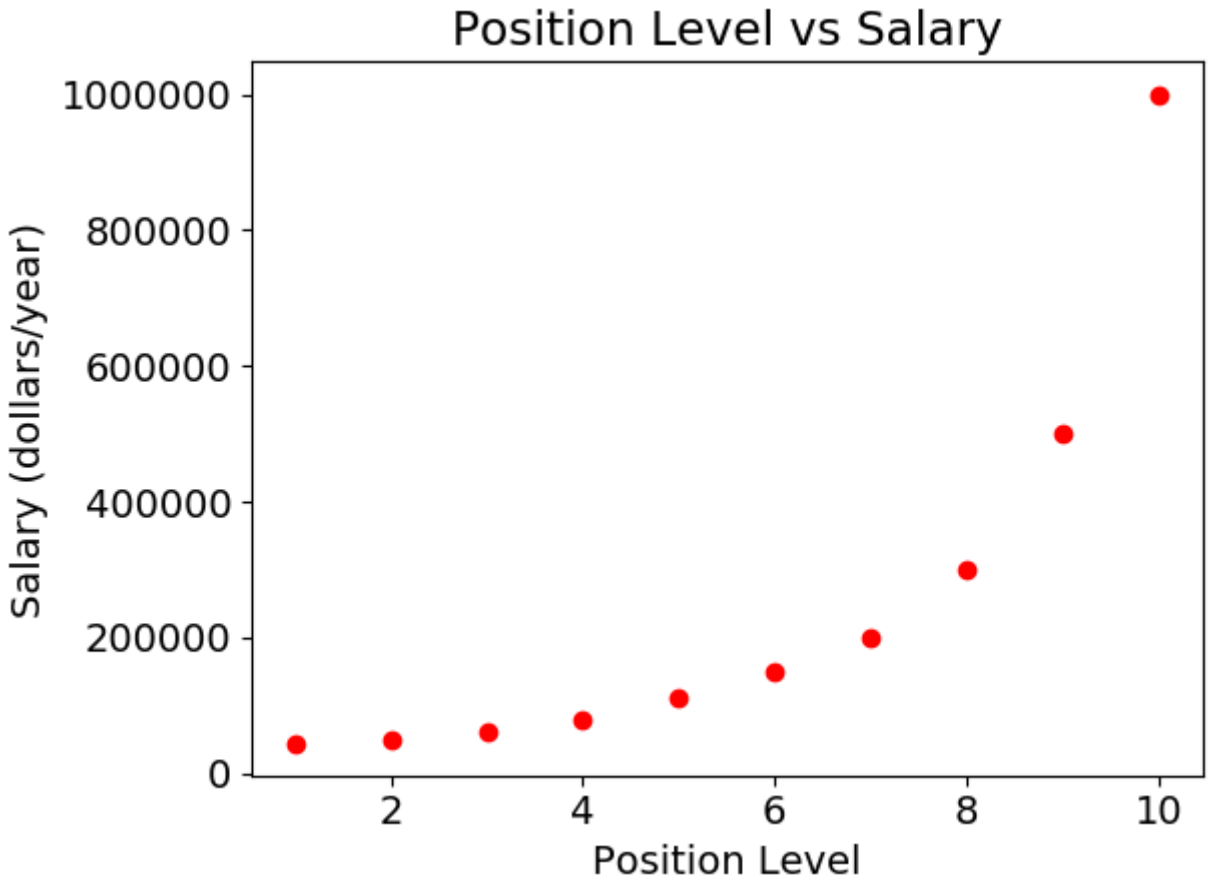
+ Giá đầu vào – ký hiệu là X.

+ 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

Trực quan hóa dữ liệu

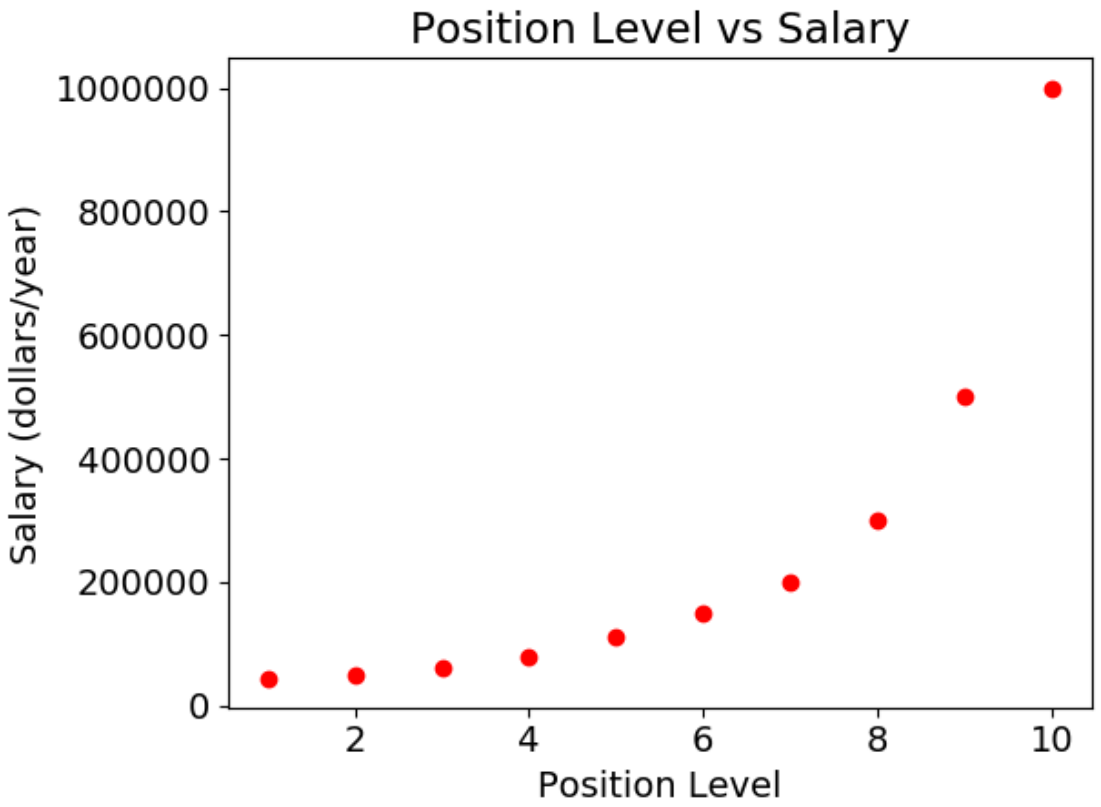


Trực quan hóa dữ liệu

– 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, thuật toán hồi qui tuyến tính (Linear Regression) sẽ không hoạt động tốt trên tập dữ liệu này.

RANDOM FOREST

Random Forest

- Với thuật toán Random Forest, trong mỗi tập dữ liệu, ta có thể xây dựng được nhiều cây quyết định (Decision Tree) khác nhau.
- *Random Forest* sẽ kết hợp các cây quyết định khác nhau đó để tạo ra một mô hình mới.
- Kết quả đầu ra của mô hình Random Forest được tổng hợp từ kết quả của các cây quyết định mà thuật toán tạo ra.

Random Forest

- Bước 1: Chọn số lượng cây quyết định muốn tạo, gọi là n.
- Bước 2: Xây dựng n cây quyết định, với mỗi cây:
  - + Bước 2.1: Chọn K điểm dữ liệu ngẫu nhiên trong tập dữ liệu.
  - + Bước 2.2: Xây dựng cây quyết định dựa trên K điểm dữ liệu được chọn.
- Bước 3: Đối với một điểm dữ liệu mới, ta thực hiện dự đoán trên tất cả cây quyết định xây dựng được. Kết quả đầu ra của điểm dữ liệu này có thể được lấy là trung bình cộng dự đoán của tất cả các cây quyết định.

HUẤN LUYỆN MÔ HÌNH

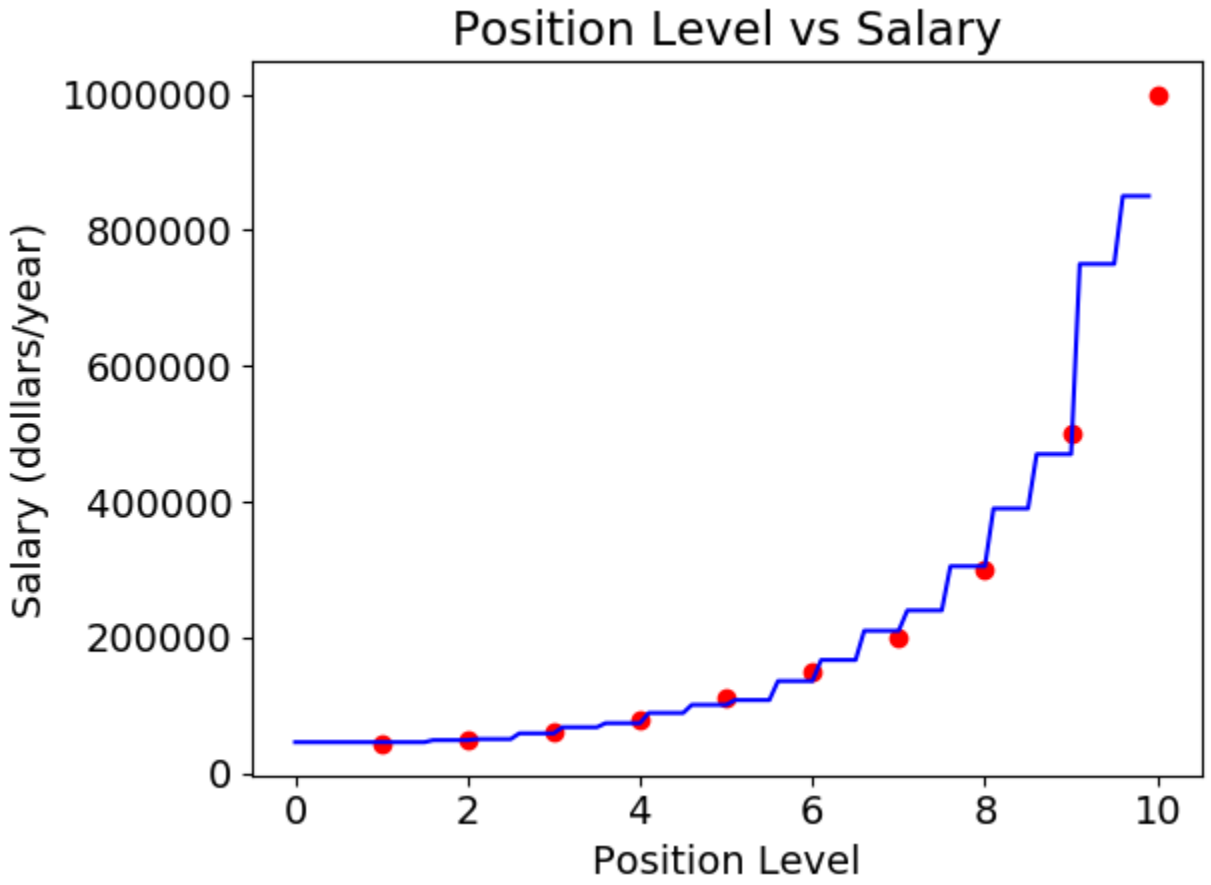
Huấn luyện mô hình

- Ta huấn luyện mô hình Random Forest Regression với lớp `RandomForestRegressor` trong module `sklearn.ensemble`.
- Số lượng cây quyết định ta sử dụng trong bài này là 10.

```
11.from sklearn.ensemble import RandomForestRegressor
12.regressor = RandomForestRegressor(n_estimators = 10)
13.regressor.fit(X, Y)
```

TRỰC QUAN HÓA KẾT QUẢ MÔ HÌNH

Trực quan hóa kết quả mô hình

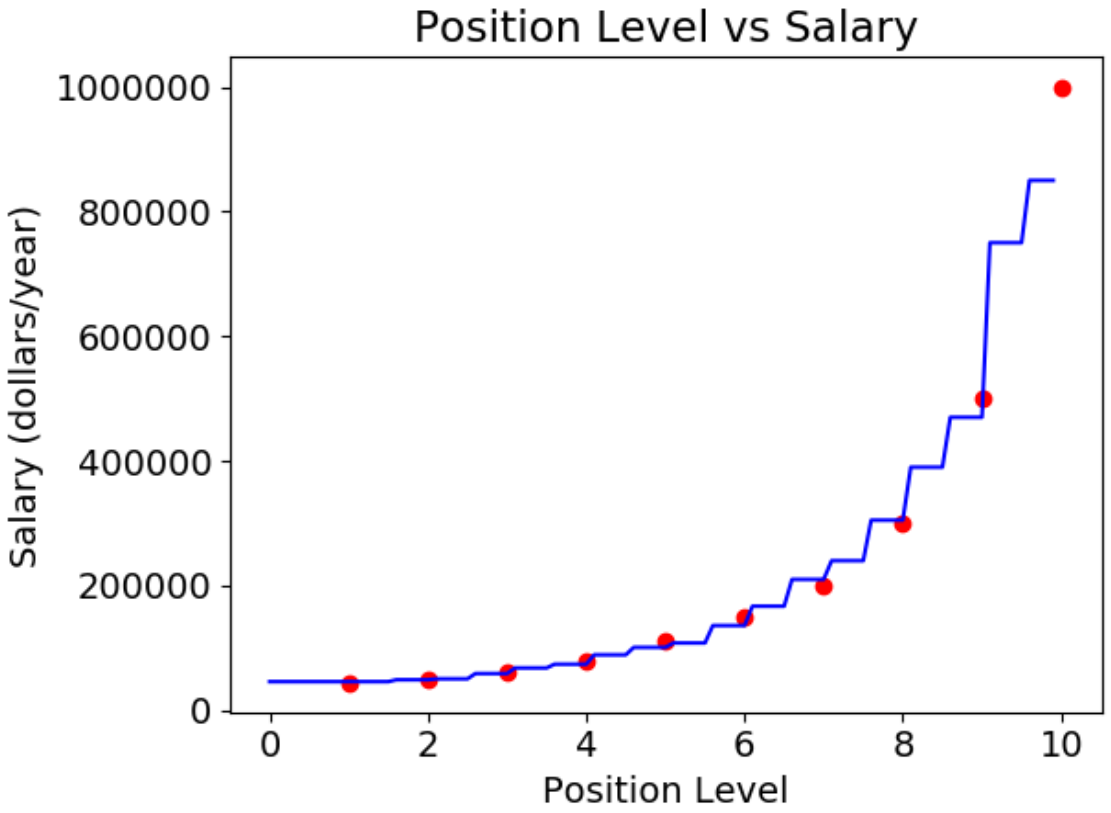


Trực quan hóa kết quả mô hình

— Vẽ kết quả dự đoán được trên mặt phẳng tọa độ.

```
14.import numpy as np
15.X_dummy = np.arange(0, 10, 0.1).reshape(-1, 1)
16.Y_dummy_pred = regressor.predict(X_dummy)
17.plt.scatter(X, Y, color = "red")
18.plt.plot(X_dummy, Y_dummy_pred, color = "blue")
19.plt.title("Position Level vs Salary")
20.plt.xlabel("Position Level")
21.plt.ylabel("Salary (dollars/year)")
22.plt.show()
```

Trực quan hóa kết quả mô hình



- Nhận xét kết quả:
  + Mô hình dạng bậc thang là một đặc trưng trong thuật toán cây quyết định và thuật toán rừng ngẫu nhiên.

Trực quan hóa kết quả mô hình

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

```
23.def compare(i_example):
24.     x = X[i_example : i_example + 1]
25.     y = Y[i_example]
26.     y_pred = regressor.predict(x)
27.     print(x, y, y_pred)
```

Trực quan hóa kết quả mô hình

– Gọi thực hiện hàm so sánh kết quả trên mọi điểm dữ liệu trong tập training.

```
28.for i in range(len(X)):
29.     compare(i)
```

Trực quan hóa kết quả

Position	Level	Salary	Predicted Salary
Business Analyst	1	45,000	46,000
Junior Consultant	2	50,000	49,000
Senior Consultant	3	60,000	59,000
Manager	4	80,000	74,000
Country Manager	5	110,000	101,000

Trực quan hóa kết quả

Position	Level	Salary	Predicted Salary
Region Manager	6	150,000	136,000
Partner	7	200,000	210,000
Senior Partner	8	300,000	305,000
C-level	9	500,000	470,000
CEO	10	1,000,000	850,000

Cảm ơn quí vị đã lắng nghe

Nhóm tác giả
Hồ Thái Ngọc
ThS. Võ Duy Nguyên
TS. Nguyễn Tấn Trần Minh Khang