DECISION TREE 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 cây quyết định hồi quy – decision tree regression.

TIỀN XỬ LÝ DỮ LIỆU

Tiền xử lý dữ liệu

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

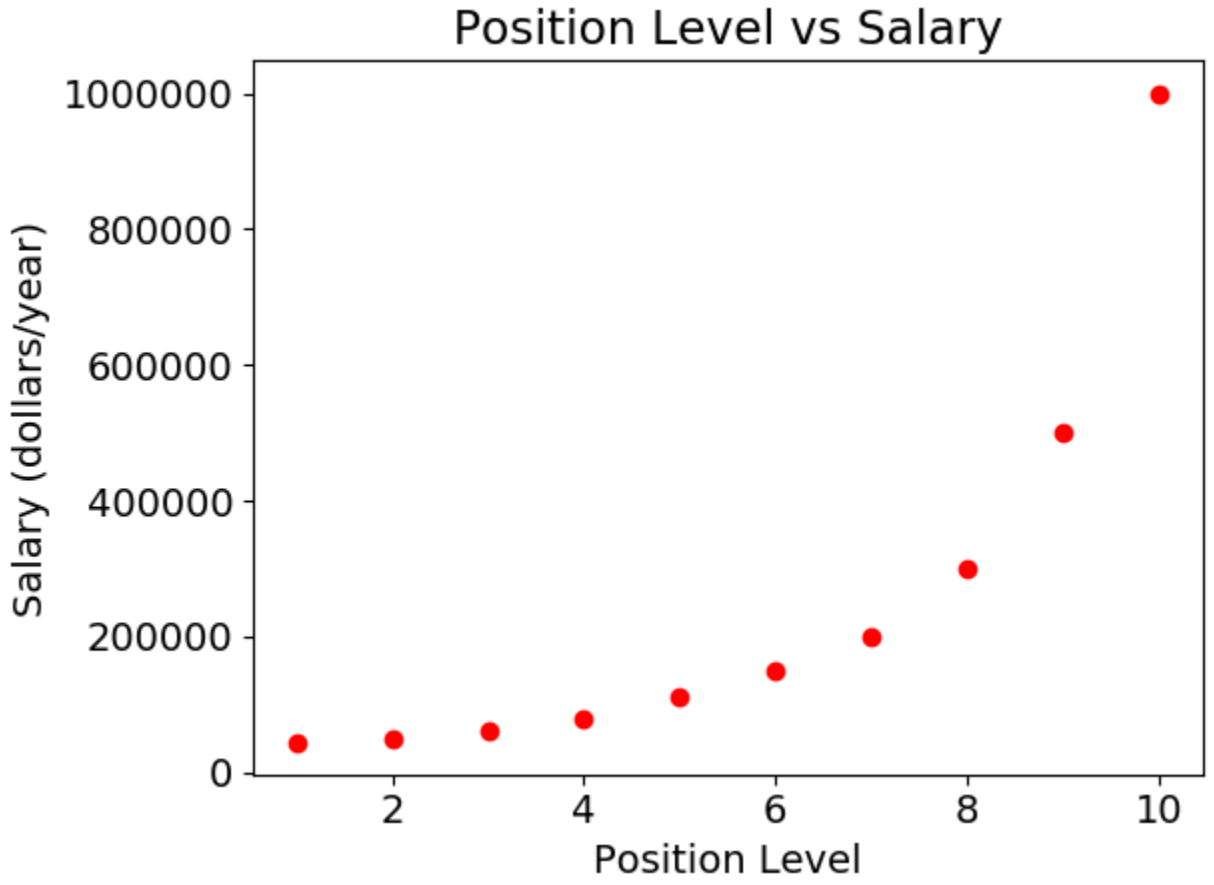
+ Giá trị đầu vào – ký hiệu là X.

+ 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

Trực quan hóa dữ liệu

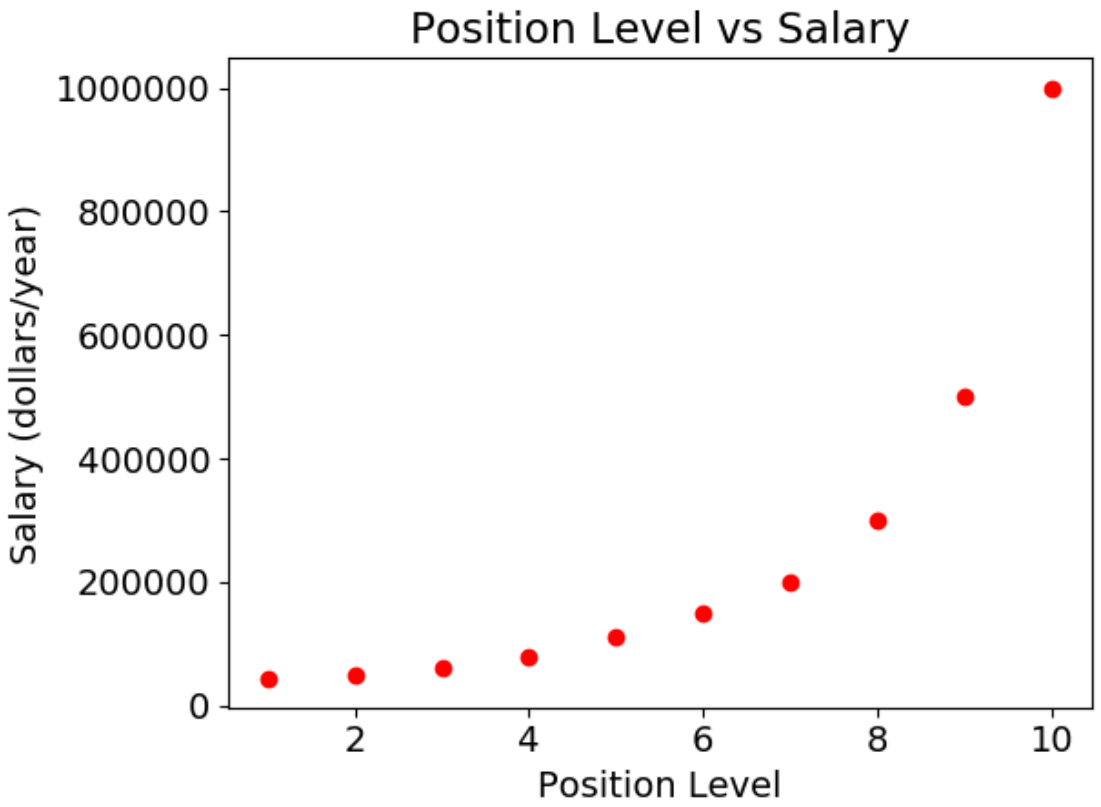


Trực quan hóa dữ liệu

– 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



- Nhận xét 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.

DECISION TREE

Decision Tree

– Bài toán mở đầu: Sắp đến kỳ thi, một cậu sinh viên tự đặt ra quy tắc *học* hay *chơi* của mình như sau:

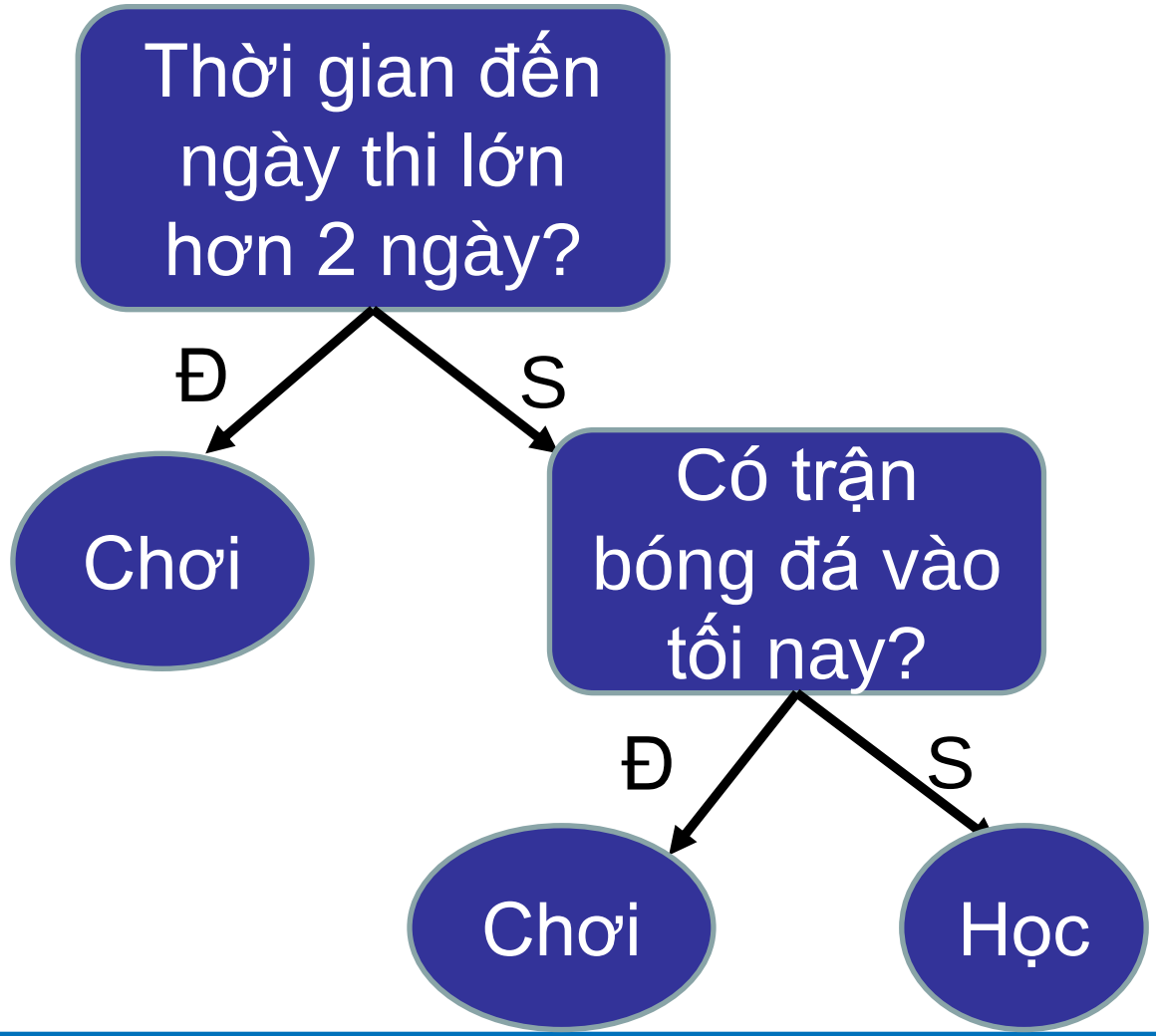
+ Nếu còn nhiều hơn hai ngày tới ngày thi, cậu ta sẽ đi chơi.

+ Nếu còn không quá hai ngày và đêm hôm đó có một trận bóng đá, cậu sẽ sang nhà bạn chơi và cùng xem trận bóng đêm đó.

+ Cậu sẽ chỉ học trong các trường hợp còn lại.

Decision Tree

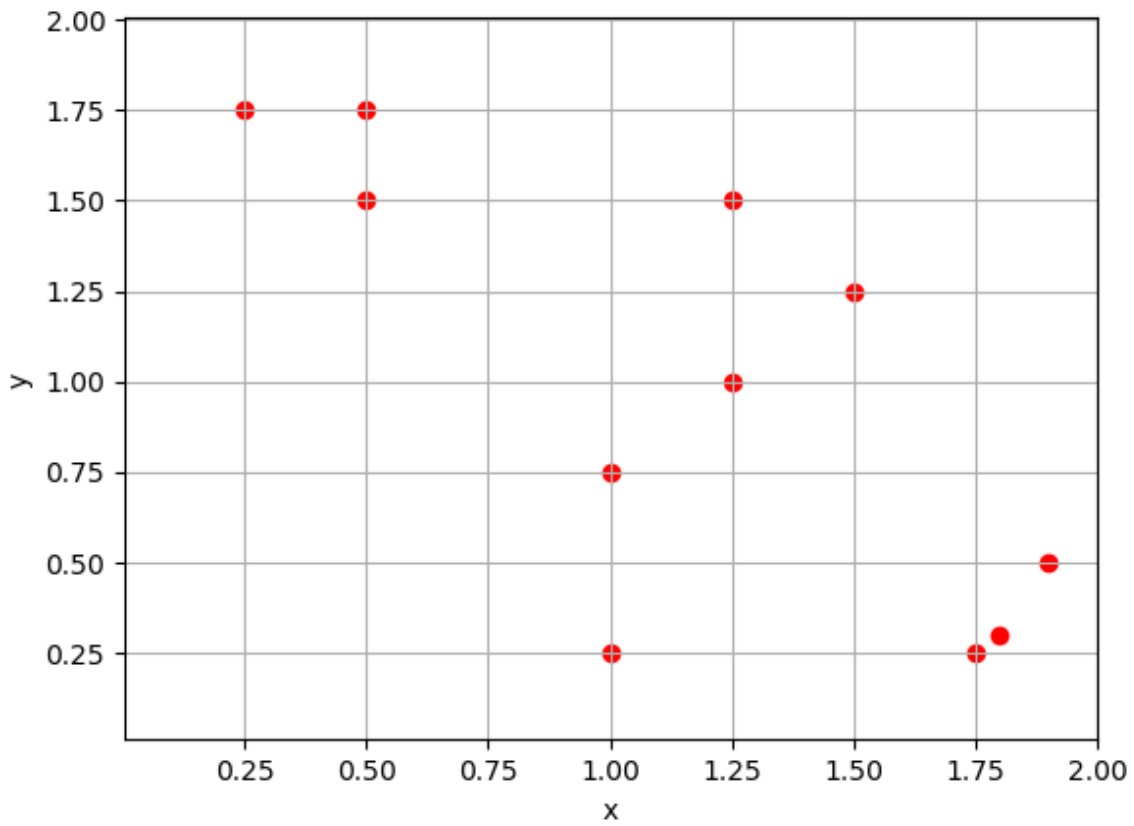
- Việc ra quyết định của cậu sinh viên này có thể được mô tả trên sơ đồ sau.
- Sơ đồ trong hình được gọi là một cây quyết định. Cụ thể hơn là cây quyết định phân loại.



DECISION TREE REGRESSION

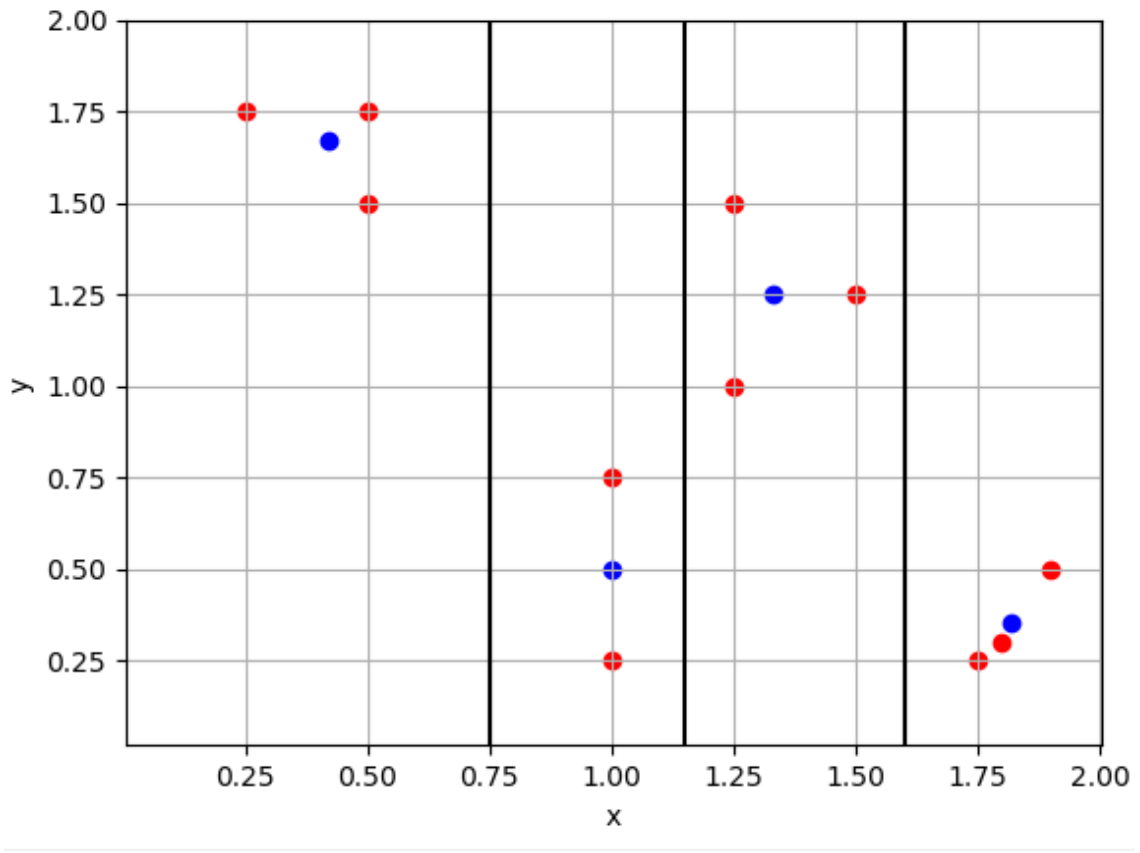
Decision Tree Regression

- Cây quyết định hồi quy có phần phức tạp hơn cây quyết định phân loại.
- Xét sự phân phối của của các điểm dữ liệu (x, y) bên đây.
- Bài toán yêu cầu dựa trên x, dự đoán giá trị của y.

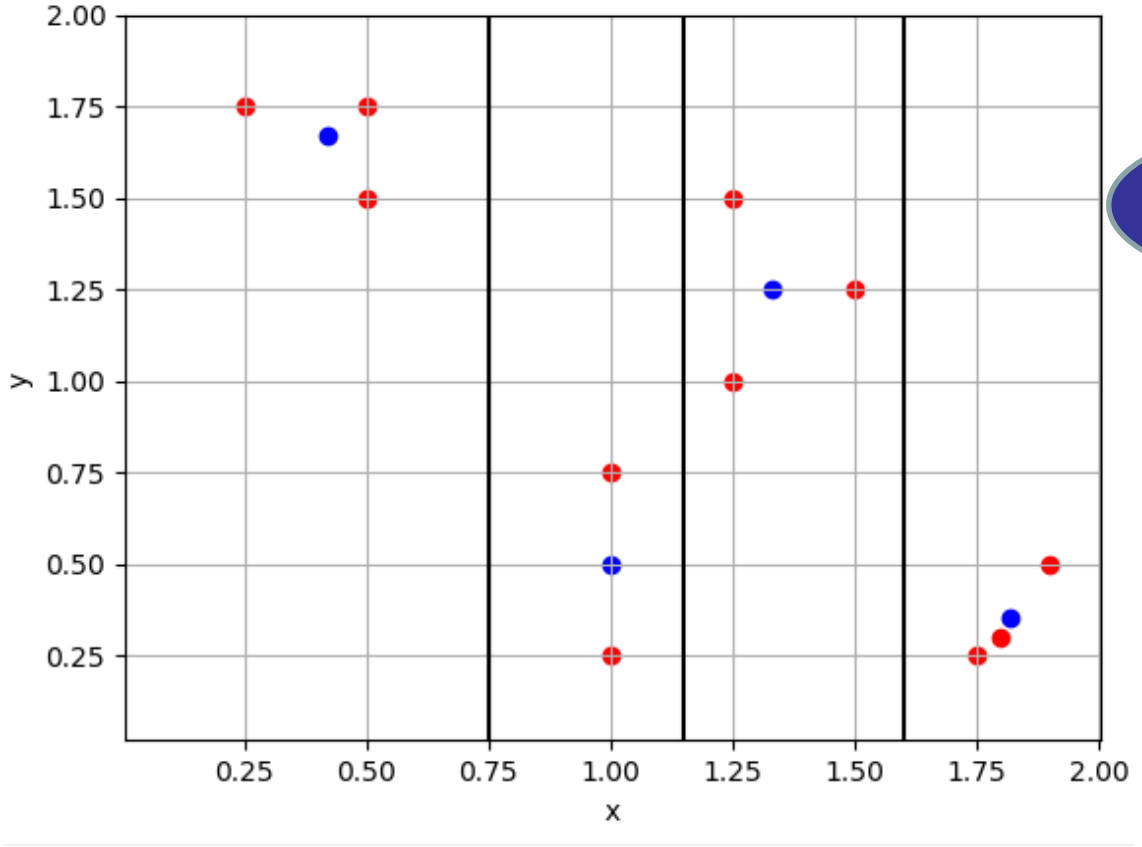


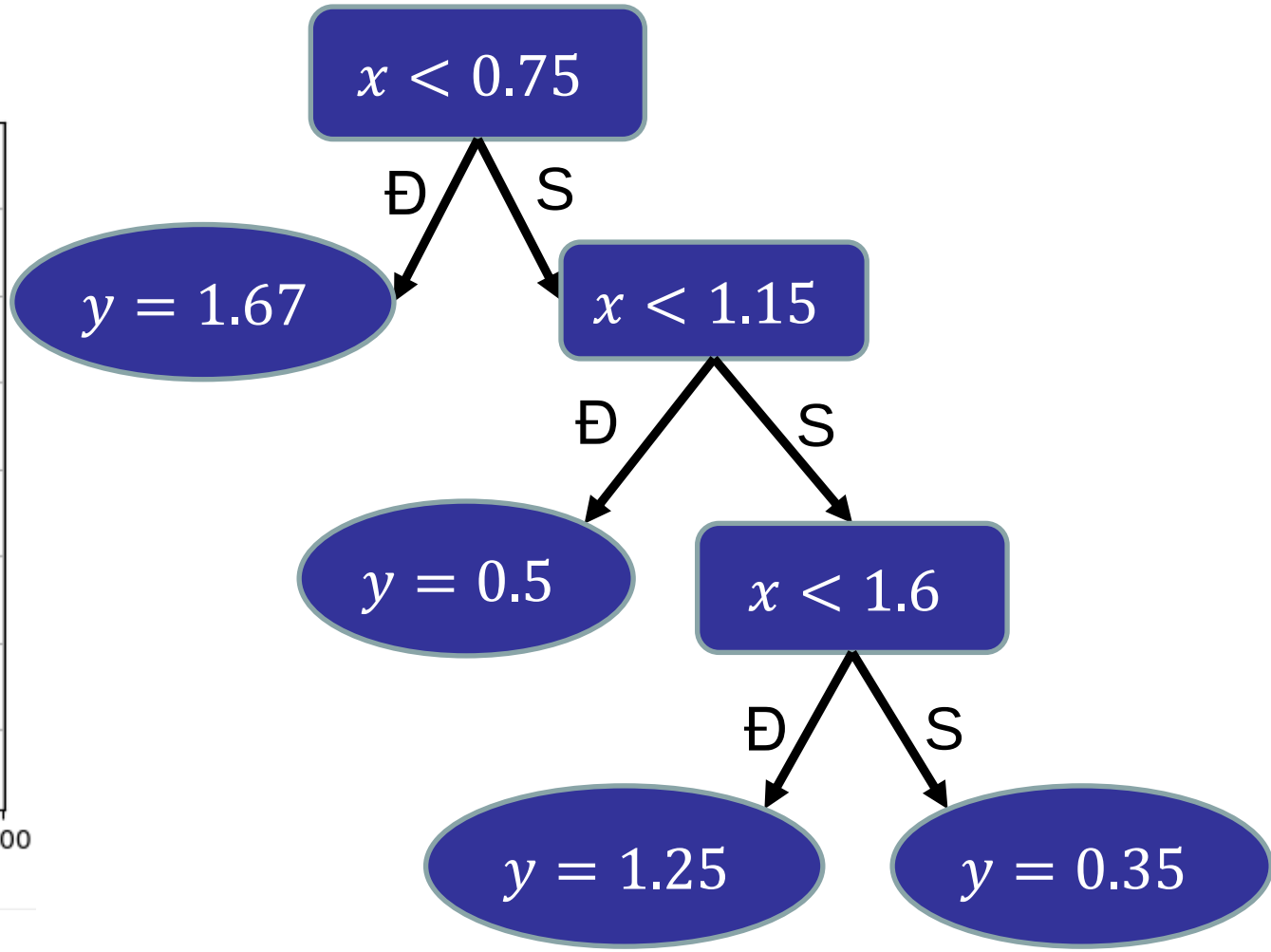
Decision Tree Regression

- Ta có thể giải quyết bài toán bằng cây quyết định hồi quy như hình bên.
- Chia trục hoành thành nhiều đoạn.
- Nếu điểm dữ liệu mới x thuộc một trong những đoạn trên, ta sẽ dự đoán y là giá trị trung bình của tất cả những giá trị y trong đoạn đó.



Decision Tree Regression





HUẤN LUYỆN MÔ HÌNH

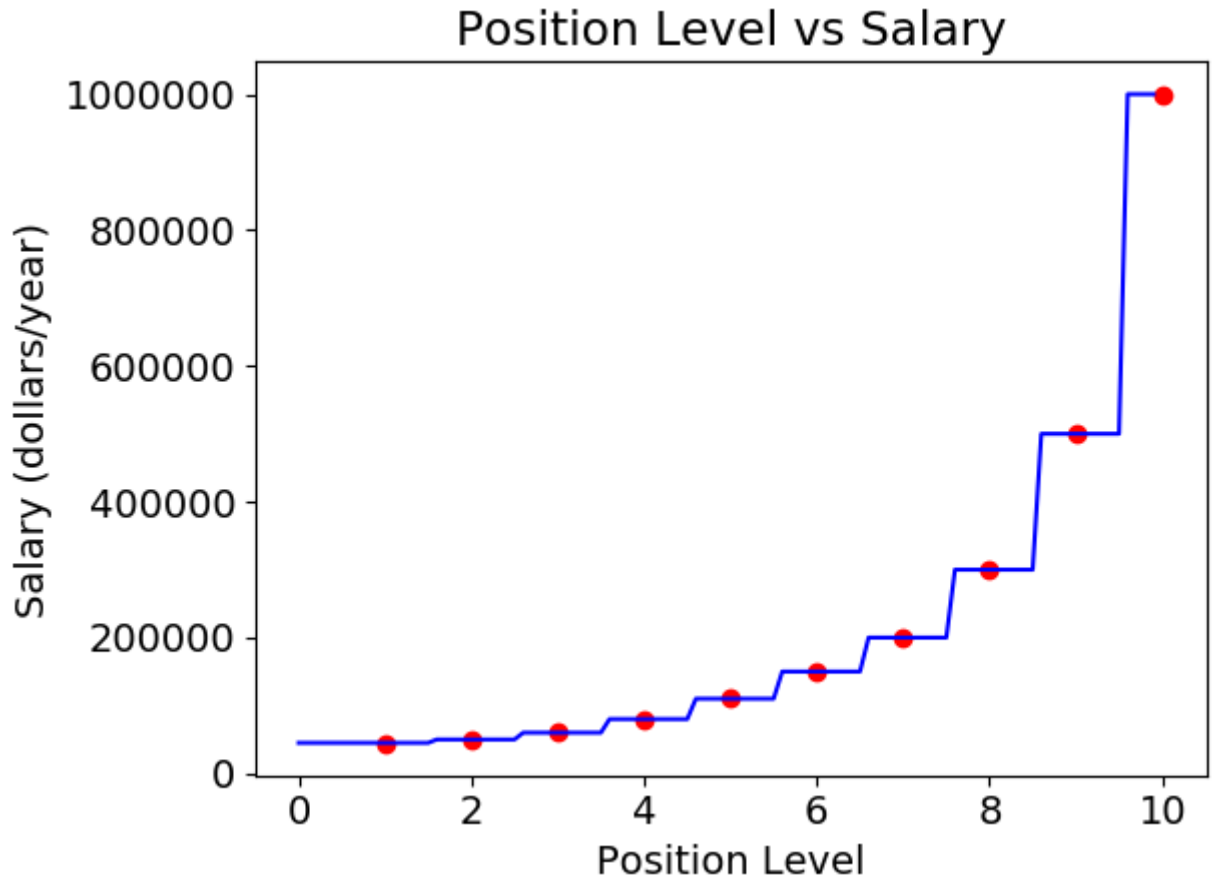
Huấn luyện mô hình

- Ta sẽ bắt đầu huấn luyện mô hình với lớp `DecisionTreeRegressor` trong module `sklearn.tree`.

```
11.from sklearn.tree import DecisionTreeRegressor
12.regressor = DecisionTreeRegressor()
13.regressor.fit(X, Y)
```

TRỰC QUAN HÓA KẾT QUẢ MÔ HÌNH

Trực quan hóa kết quả mô hình

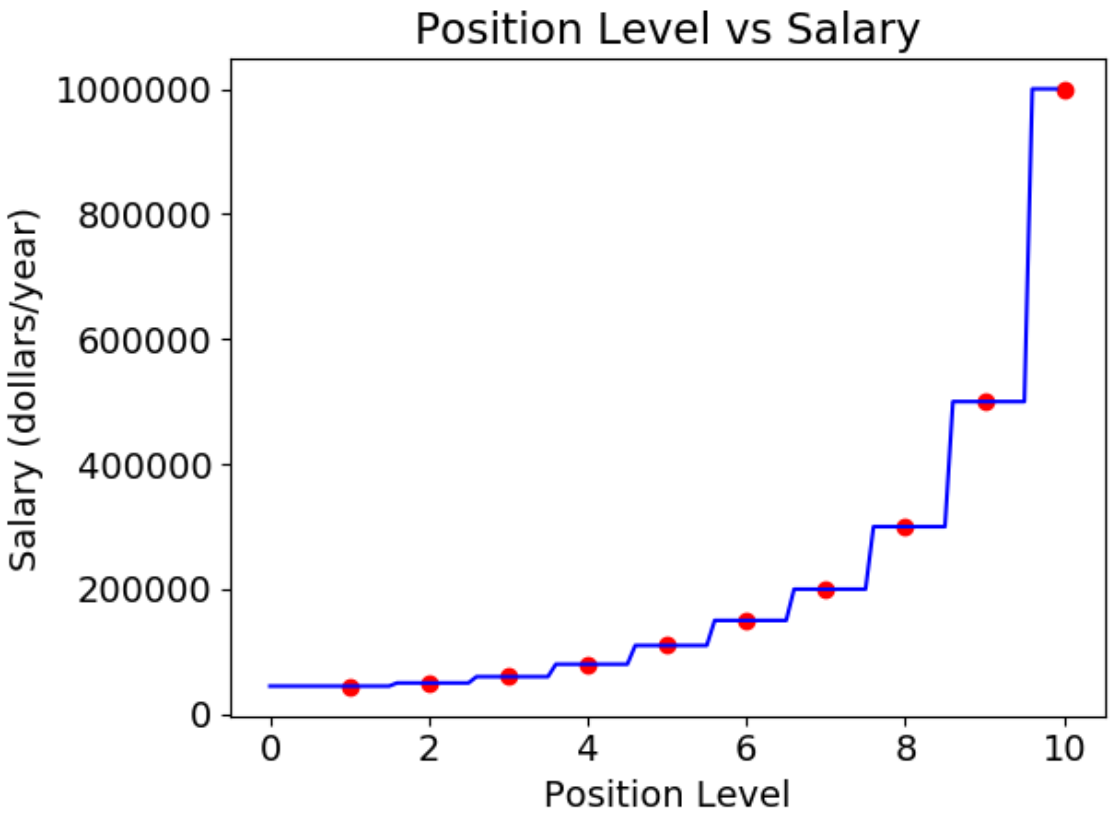


Trực quan hóa kết quả mô hình

– 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 có dạng bậc thang.
  - + Đây là một đặc trưng của cây quyết định hồi quy.

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 thuộc 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	53,356
Junior Consultant	2	50,000	31,759
Senior Consultant	3	60,000	94,632
Manager	4	80,000	121,724
Country Manager	5	110,000	143,275

Trực quan hóa kết quả

Position	Level	Salary	Predicted Salary
Region Manager	6	150,000	184,003
Partner	7	200,000	184,003
Senior Partner	8	300,000	289,994
C-level	9	500,000	528,694
CEO	10	1,000,000	988,916

Cảm ơn quí vị đã lắng nghe

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