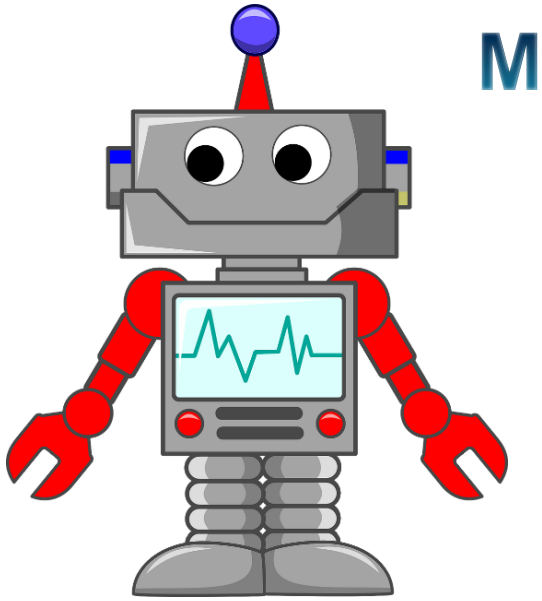
ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

CS116 – LẬP TRÌNH PYTHON CHO MÁY HỌC

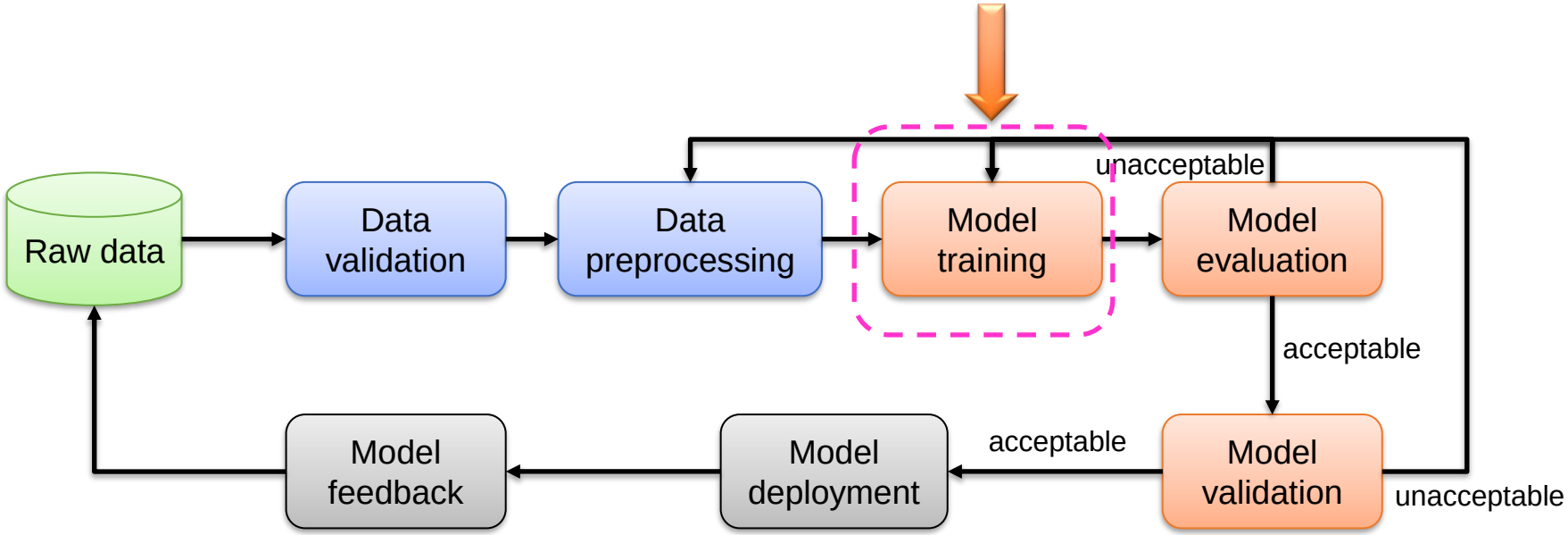
BÀI 07

HỌC CÓ GIÁM SÁT – MÔ HÌNH PHÂN LỚP (CLASSIFICATION)



TS. Nguyễn Vinh Tiệp

Vị trí của bài học



NỘI DUNG

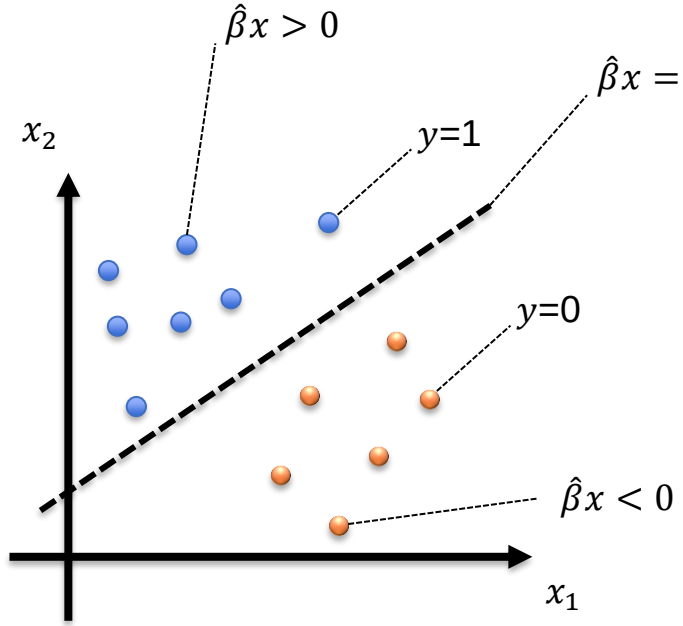
1. MÔ HÌNH LOGITIC REGRESSION

2. MÔ HÌNH VỚI DỮ LIỆU CÓ QUAN HỆ PHI TUYẾN

Mô hình Logistic Regression

- Cho hai tập điểm “xanh” và “cam” được biểu diễn với “y=1 hoặc y=0”
- Hàm mô hình dự đoán: $\hat{y} = \sigma(\hat{\beta}x)$ hàm sigmoid giúp ép [0,1], do xanh và cam chỉ là 0,1
- Hàm độ lỗi dự đoán: $\mathrm{BCE} = \frac{1}{n}\sum_{i=1}^{n} y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)$

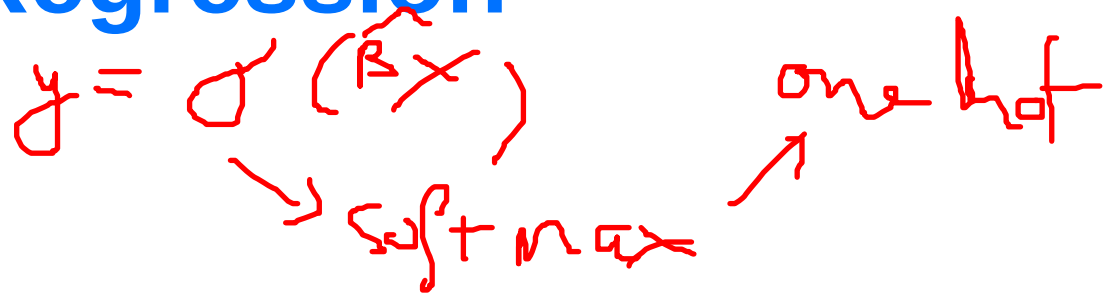
binary cos entrophy



sai số rất lớn giúp mô hình chạy nhanh

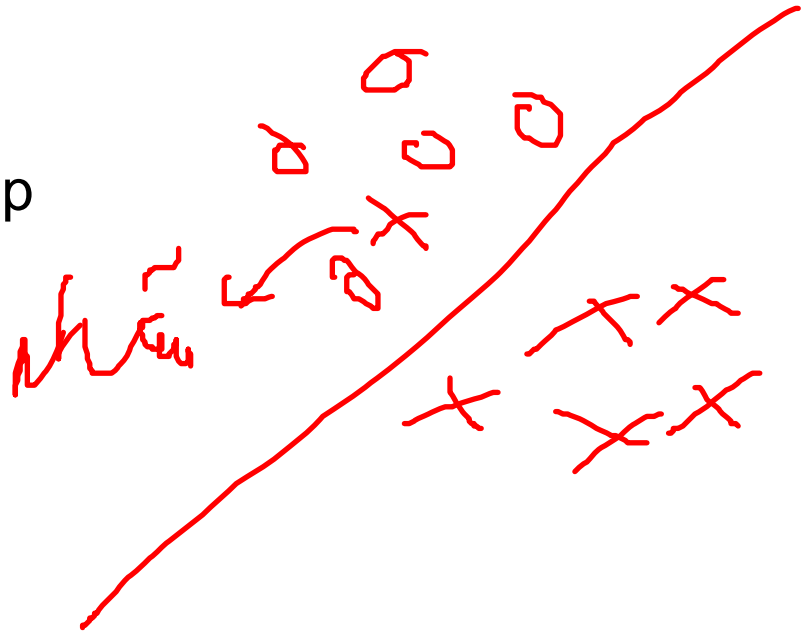
đường thẳng ở đây để phân lớp

Mô hình Logistic Regression



- **Ưu điểm:**
  - Đơn giản, dễ cài đặt
  - Có thể mở rộng sang bài toán phân loại nhiều lớp

- **Khuyết điểm:**
  - Không giải quyết được với dữ liệu phức tạp
  - Dễ bị ảnh hưởng bởi dữ liệu nhiễu



NỘI DUNG

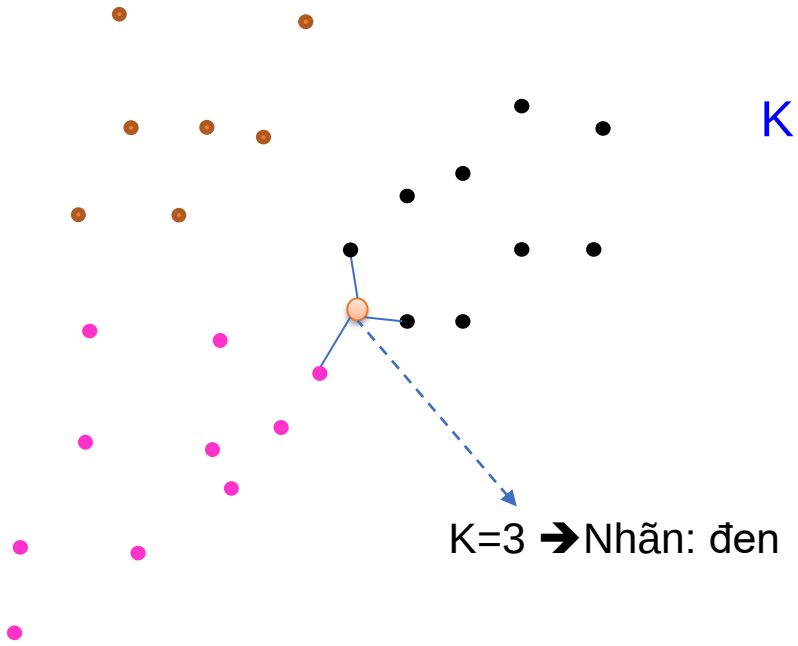
1. MÔ HÌNH LOGITIC REGRESSION

2. MÔ HÌNH VỚI DỮ LIỆU CÓ QUAN HỆ PHI TUYẾN

Mô hình K-Nearest Neighbor (KNN)

- **KNNClassifier**: Là một mô hình phân loại dựa trên nhãn của K đặc trưng có khoảng cách gần nhất
- Đây là mô hình không tham số

lazy learning -> không có tham số gì hết nhưng phải buộc nhớ tất cả input đầu vào => tốn bộ nhớ



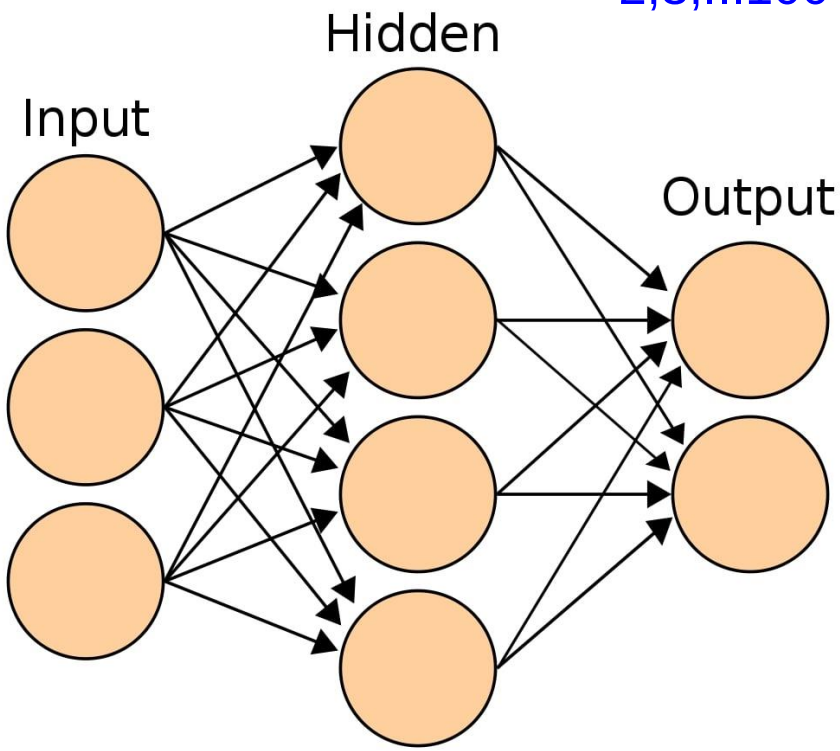
K được gọi là siêu tham số,

còn tham số đưa ra quyết định với 1 đặc trưng đầu vào

Multi-Layer Perceptron (MLP)

- **Multi Layer Perceptron** (MLPClassifer): sử dụng thêm các lớp ẩn (hidden layer) để học các đặc trưng trung gian, giúp việc ra quyết định dễ hơn

2,3,...100

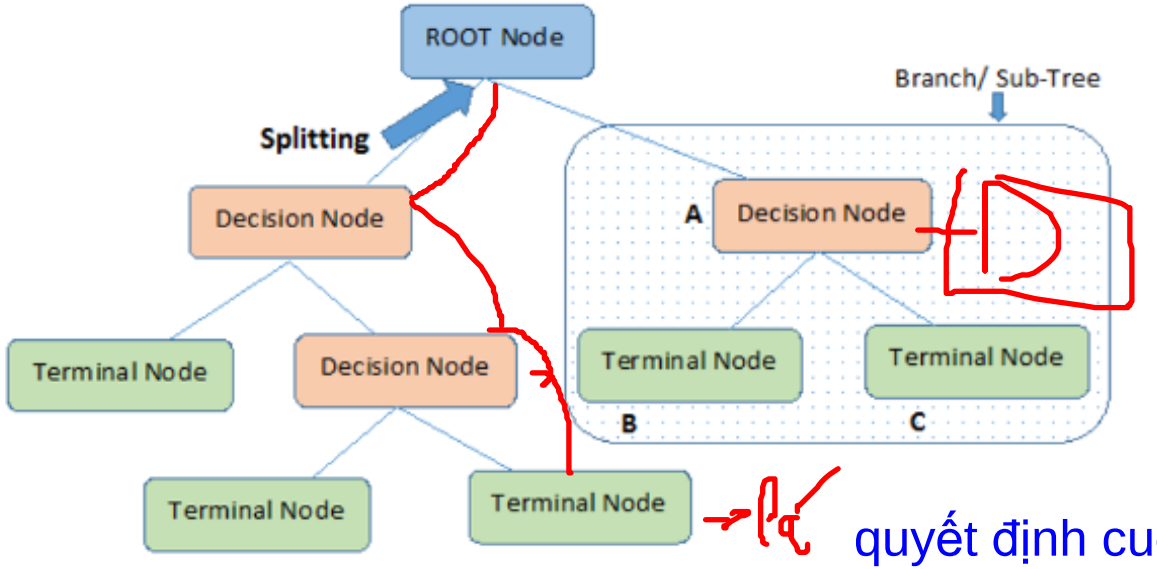


nếu k =5 thì output sẽ ra 5 neuron

Cây quyết định – Decision Tree

tiến hành node này nhận giá trị từ bao nhiêu đến bao nhiêu để chia trường hợp

- **Decision Tree**: là một cấu trúc phân cấp không tham số, trong đó:
  - mỗi nút đại diện cho một thuộc tính (hay đặc trưng)
  - mỗi nhánh từ nút đó tương ứng với một trong số khả năng của thuộc tính đó
  - mỗi nút lá (nút cuối cùng) đưa ra quyết định phân loại cuối cùng dựa trên các thuộc tính đã duyệt trước đó

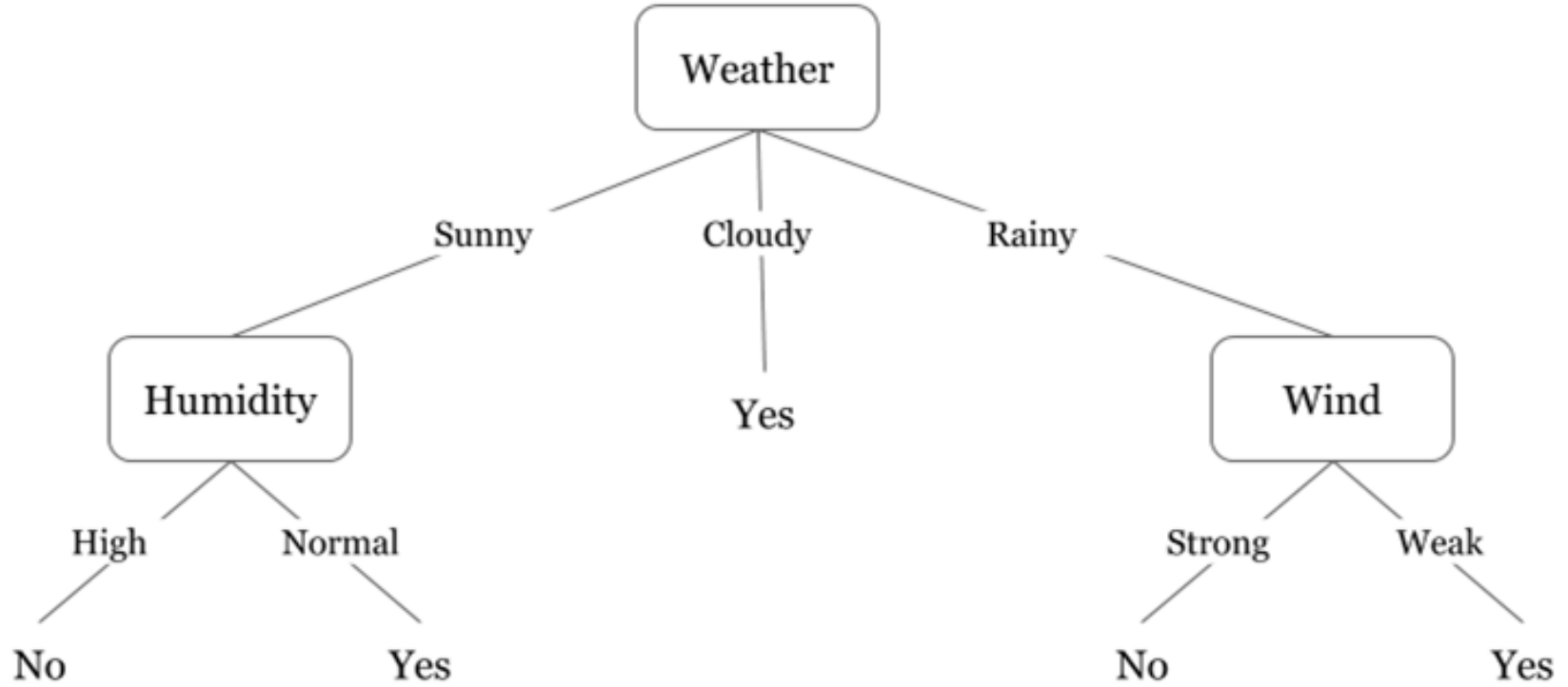


quyết định cuối cùng dựa trên các node trước

Nguồn: https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/

Cây quyết định – Decision Tree

- Ví dụ: Hôm nay trời nắng, độ ẩm cao, gió yếu. Chúng ta có nên chơi cầu lông không?



Nguồn: https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/ml-decision-tree/tutorial/

Cây quyết định – Decision Tree

- Làm sao để xây dựng một cây quyết định từ bộ dữ liệu huấn luyện

Day	Weather	Temperature	Humidity	Wind	Play?
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak	Yes
10	Rainy	Mild	High	Strong	No

Cây quyết định – Decision Tree

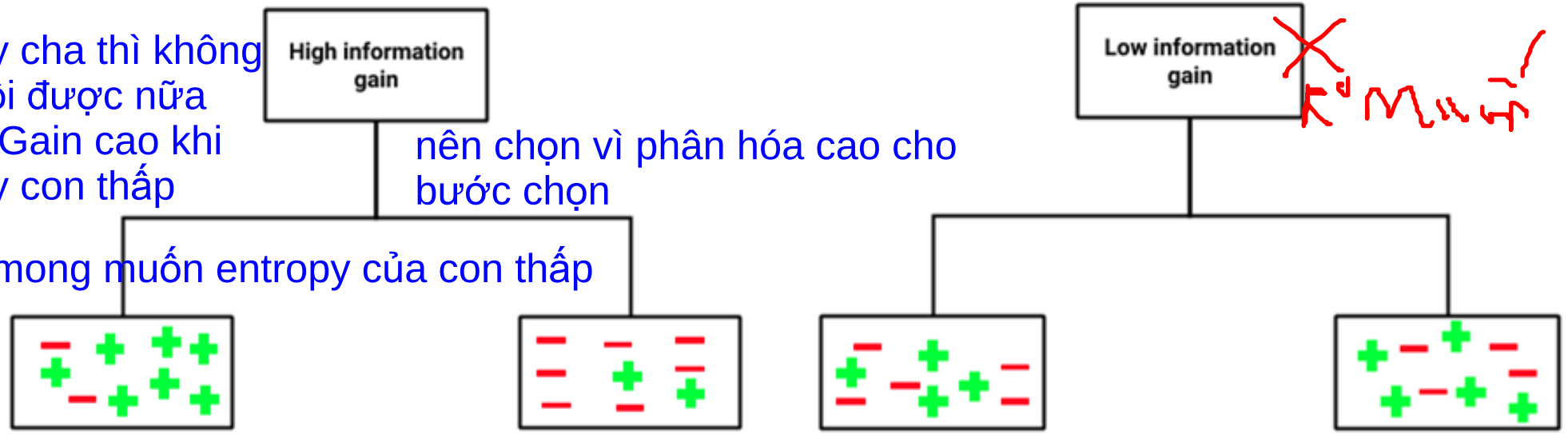
- Dựa trên độ đo Information Gain để quyết định chọn đặc trưng nào để phân loại cho nút tiếp theo
- Entropy thể hiện mức độ đa dạng của các loại đối tượng trong tập S

sl đối tượng

$$\text{Entropy}(S) = -\sum_{i=1}^{C} p_i \log p_i \qquad \text{InfoGain} = \text{Entropy}(Parent) - \text{E}[\text{Entropy}(children)]$$

với S là tập đối tượng gồm C nhãn, p_i với $i = 1..C$ là tỉ lệ của nhãn thứ i trong S

entropy cha thì không thay đổi được nữa => infoGain cao khi entropy con thấp

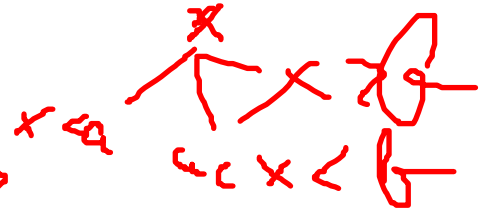


nên chọn vì phân hóa cao cho bước chọn

luôn mong muốn entropy của con thấp

Cây quyết định – Decision Tree

- Nhận xét:



có thể có siêu tham số để giới hạn độ sâu

Ưu điểm	Khuyết điểm
- **Tính giải thích mô hình**: nhờ khả năng trực quan hóa với cấu trúc cây	**- Overfitting**: dễ bị overfitting với dữ liệu nếu cây không được cắt tỉa (pruning) hoặc cây quá sâu (mô hình quá phức tạp)
- Có thể làm việc được trên cả đặc trưng dạng số lẫn phân loại	**- Không ổn định**: chỉ cần thay đổi dữ liệu liệu một phần nhỏ, cây có thể thay đổi hoàn toàn → dùng kỹ thuật ensemble với Random Forest.
- Phi tham số: không đưa ra giả định về sự phân bố của các biến và mối quan hệ giữa các đặc trưng với output	**- Khó tối ưu**: Việc tìm cây tối ưu tốn nhiều chi phí tính toán, trong khi dùng heuristic thì không đảm bảo được cây tối ưu
- Lựa chọn đặc trưng: ưu tiên các đặc trưng nhiều thông tin để phân thành nhánh mới	**- Bias**: có xu hướng bị ảnh hưởng bởi các đặc trưng có nhiều giá trị (biến phân loại) hoặc phạm vi lớn hơn (biến số), vì sẽ giúp tạo nhiều nhánh

không có tính toán x và y

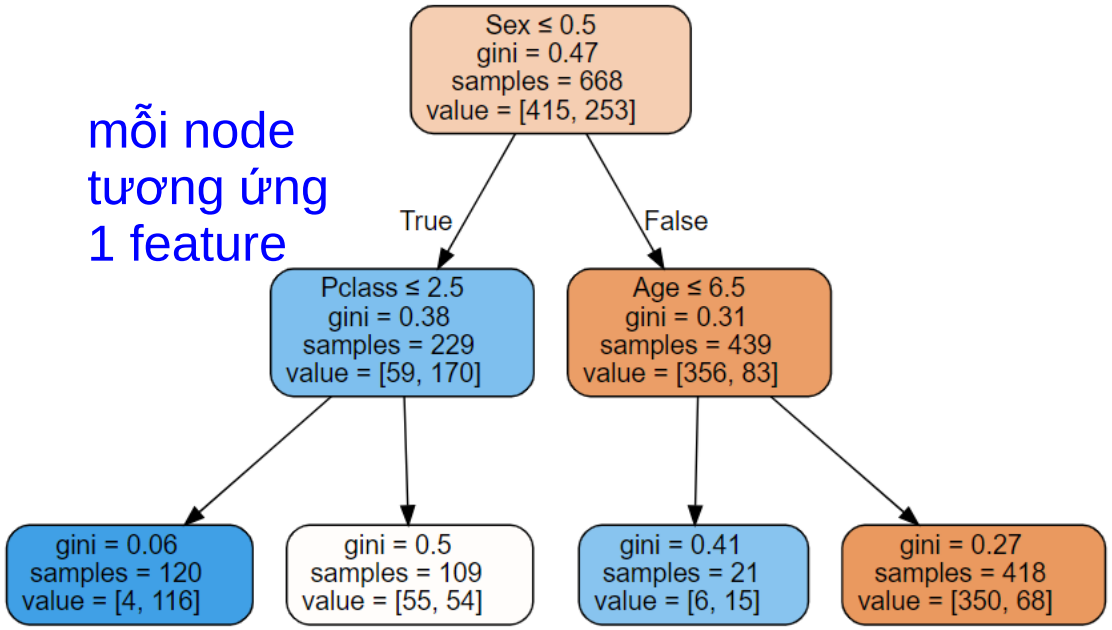
có thể dùng cho lựa chọn đặc trưng

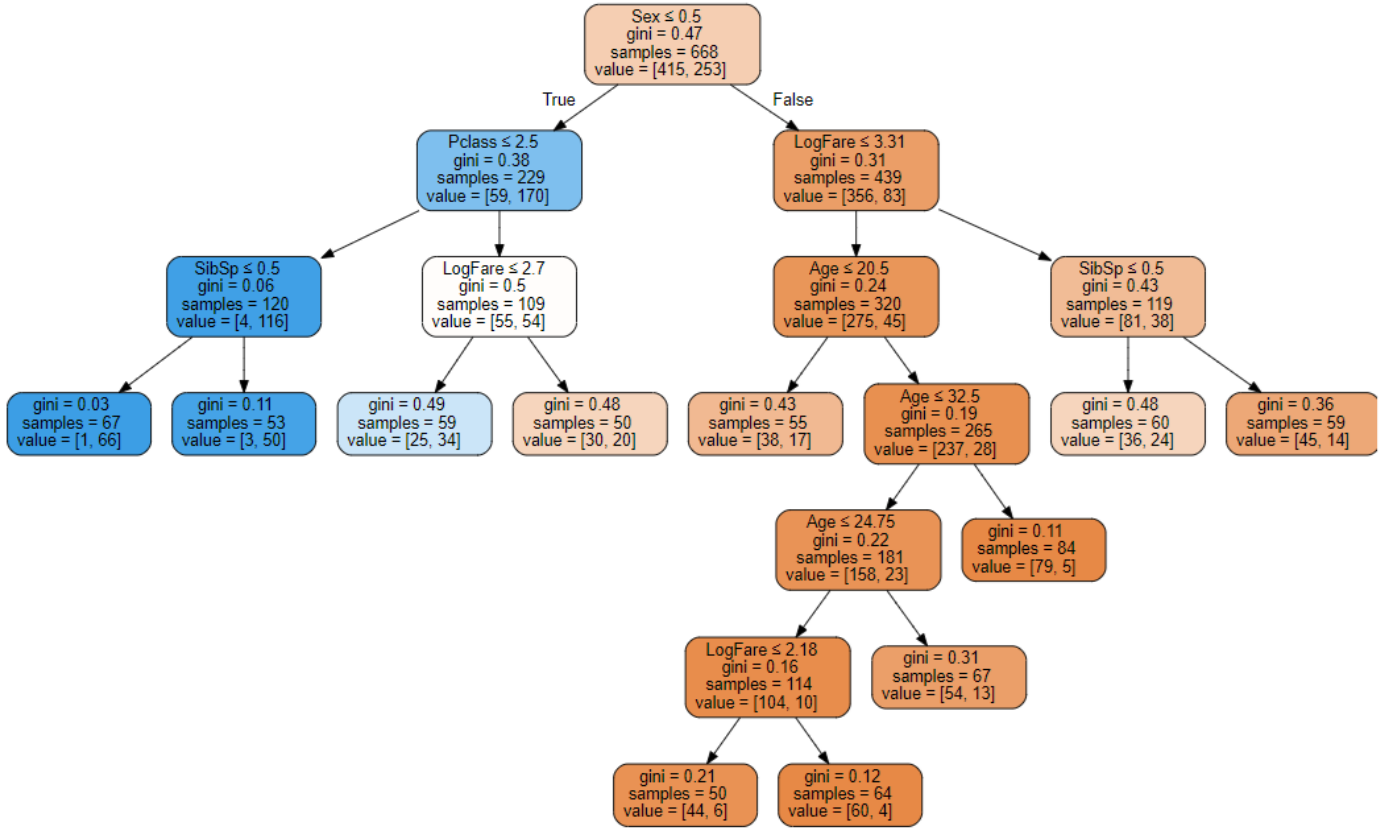
dễ bias dẫn đến overfitting luôn
ví dụ x1 có rất nhiều dạng giá trị thì dễ bias vào x1 hơn, dù feature khác có ít giá trị

Cây quyết định – Decision Tree

- Ví dụ với Titanic Dataset

mỗi node tương ứng 1 feature





Một số mô hình phân lớp khác

- Các mô hình phân lớp:
  - Support Vector Machine
  - Naïve Bayes Classifier lý thuyết thống kê
  - Random Forest
  - Gradient Boost, XGBoost, LightGBM, CatBoost Classifier

BÀI QUIZ VÀ HỎI ĐÁP