Decision Tree

Là một **thuật toán rất mạnh**, phổ biến trong thực tiễn (ML công nghiệp, các cuộc thi Kaggle), **dù ít được chú ý trong học thuật**.

Khái niệm Decision Tree

- Là **một mô hình có cấu trúc cây**, với:
  - **Nút gốc (root node)** ở trên cùng.
  - **Nút quyết định (decision node)**: kiểm tra một đặc trưng và rẽ trái/phải theo giá trị.
  - **Lá (leaf node)**: nơi đưa ra dự đoán đầu ra (cat/not-cat).
- Có nhiều **cây khác nhau có thể xây dựng**, tất cả đều phân loại được dữ liệu.
- Một số cây sẽ **tổng quát tốt hơn** (generalize) cho tập kiểm tra hoặc kiểm định chéo.
- Nhiệm vụ của **thuật toán học cây quyết định** là:
  - Chọn ra cây phù hợp nhất với **dữ liệu huấn luyện**, **tổng quát tốt**, tránh quá khớp.

Quy trình xây dựng cây quyết định

1. Chọn đặc trưng tại nút gốc (root node)

- Đầu tiên, ta chọn một đặc trưng để **phân tách (split)** tập dữ liệu ban đầu tại nút gốc.

2. Tiếp tục chia nhỏ các nhánh

- Tập trung vào từng nhánh (trái/phải), chọn tiếp đặc trưng để chia nhỏ tiếp:
- Lặp lại quy trình cho nhánh phải

Mục tiêu của việc chia tách là tạo ra các node "thuần" (pure)

- Mỗi node lý tưởng chỉ chứa **toàn bộ là 1 lớp duy nhất** (toàn cat hoặc không cat).

Các quyết định quan trọng trong thuật toán học cây

1. Chọn đặc trưng nào để chia (feature selection)

- Mục tiêu: **Tối đa hóa độ thuần khiết (purity)** của node sau khi chia.

2. Khi nào dừng việc chia tách?

Các tiêu chí dừng phổ biến:

a. Node thuần 100%

- Nếu tất cả ví dụ trong node là cùng một lớp (toàn cat hoặc toàn not-cat), thì dừng và tạo node lá.

b. Giới hạn độ sâu của cây (maximum depth)

- Ví dụ: Chỉ cho phép cây sâu tối đa 2 mức → các node ở sâu hơn sẽ không được chia tiếp.
- Lý do:
  - Giảm độ phức tạp của cây
  - **Ngăn overfitting**

c. Cải thiện độ thuần không đáng kể

- Nếu sau khi chia, độ thuần cải thiện rất ít (tức là giảm impurity quá nhỏ) → không đáng chia thêm.

d. Số lượng ví dụ quá ít

- Nếu node chỉ có rất ít ví dụ (ví dụ chỉ còn 3), chia nhỏ hơn có thể không hữu ích và dễ overfit.