*Decision Tree(Cây quyết định)*

**Decision Tree** (Cây Quyết Định) là một trong những thuật toán học máy phổ biến, được sử dụng rộng rãi trong cả bài toán phân loại và hồi quy. Mô hình này mô phỏng quá trình ra quyết định của con người thông qua một cấu trúc cây, với mỗi nút trong cây đại diện cho một thuộc tính (feature) và mỗi nhánh đại diện cho một giá trị có thể có của thuộc tính đó.

**Cấu trúc của một Decision Tree:**

* **Nút gốc (Root Node):** Đại diện cho toàn bộ tập dữ liệu.
* **Nút trong (Internal Node):** Đại diện cho một phép kiểm tra trên một thuộc tính.
* **Nhánh (Branch):** Kết nối các nút và đại diện cho kết quả của phép kiểm tra.
* **Nút lá (Leaf Node):** Đại diện cho một lớp (trong bài toán phân loại) hoặc một giá trị (trong bài toán hồi quy).

**Các thuật toán xây dựng Decision Tree phổ biến**

* **ID3 (Iterative Dichotomiser 3):** Sử dụng thông tin tương đối (Information Gain) để chọn thuộc tính tốt nhất tại mỗi nút.

**ID3 (Iterative Dichotomiser 3)**

* Nguyên lý: ID3 sử dụng thông tin tương đối (Information Gain) để chọn thuộc tính tốt nhất tại mỗi nút. Thuộc tính nào làm giảm entropy (độ hỗn loạn) của dữ liệu nhiều nhất sau khi phân chia sẽ được chọn.
* Entropy: Là một thước đo độ không chắc chắn của một tập hợp dữ liệu. Entropy càng cao, dữ liệu càng hỗn loạn.
* Information Gain: Là sự giảm đi của entropy sau khi phân chia dữ liệu theo một thuộc tính.

A screenshot of a computer code

Description automatically generatedTrong đó:

* + SplitInfo(A) là một thước đo độ phân chia của thuộc tính A.

**CART (Classification and Regression Trees)**

* Nguyên lý:
  + Sử dụng chỉ số Gini cho bài toán phân loại và độ giảm phương sai cho bài toán hồi quy để chọn thuộc tính tốt nhất.

A white rectangular object with a white background

Description automatically generated

Chỉ số Gini càng nhỏ, dữ liệu càng thuần nhất.

Độ giảm phương sai: Đo sự giảm phương sai của biến mục tiêu sau khi phân chia dữ liệu.

Ưu điểm: Linh hoạt, có thể xử lý cả bài toán phân loại và hồi quy.

* **C4.5:** Là một phiên bản cải tiến của ID3, xử lý được các thuộc tính có giá trị liên tục và các dữ liệu bị thiếu.
* **CART (Classification and Regression Trees):** Sử dụng chỉ số Gini hoặc độ giảm phương sai để chọn thuộc tính tốt nhất.

**Quy trình xây dựng Decision Tree**

1. **Chọn thuộc tính gốc:** Chọn thuộc tính có khả năng phân chia dữ liệu tốt nhất tại nút gốc.
2. **Tạo các nhánh:** Tạo một nhánh cho mỗi giá trị có thể có của thuộc tính đã chọn.
3. **Phân chia dữ liệu:** Phân chia tập dữ liệu thành các tập con dựa trên các nhánh.
4. **Lặp lại:** Áp dụng các bước trên cho mỗi tập con cho đến khi đạt được điều kiện dừng (ví dụ: tất cả các dữ liệu tại một nút thuộc cùng một lớp, độ sâu của cây đạt đến giới hạn, ...).

**Ưu điểm của Decision Tree**

* Dễ hiểu và giải thích: Quy tắc quyết định được thể hiện rõ ràng qua cấu trúc cây.
* Xử lý được nhiều loại dữ liệu: Cả dữ liệu số và danh mục.
* Không yêu cầu chuẩn hóa dữ liệu.
* Có thể xử lý dữ liệu thiếu.

**Nhược điểm của Decision Tree**

* Dễ bị overfitting: Cây quá phức tạp có thể học quá kỹ dữ liệu huấn luyện, dẫn đến hiệu quả kém trên dữ liệu kiểm tra.
* Không ổn định: Một thay đổi nhỏ trong dữ liệu huấn luyện có thể dẫn đến một cây quyết định hoàn toàn khác.
* Khó xử lý các quan hệ tương tác phức tạp giữa các thuộc tính.

**Các kỹ thuật cải tiến Decision Tree**

* Pruning: Cắt tỉa các nhánh không cần thiết để giảm overfitting.
* Bagging: Tạo nhiều cây quyết định từ các mẫu dữ liệu khác nhau và lấy trung bình hoặc biểu quyết để đưa ra quyết định cuối cùng.
* Random Forest: Là một mở rộng của Bagging, kết hợp với việc chọn ngẫu nhiên các thuộc tính tại mỗi nút.
* Boosting: Tạo tuần tự các cây quyết định, với mỗi cây tập trung vào các mẫu dữ liệu mà các cây trước đó dự đoán sai.

**Ứng dụng của Decision Tree**

* Phân loại: Dự đoán lớp của một đối tượng mới (ví dụ: spam/không spam, bệnh nhân mắc bệnh/không mắc bệnh).
* Hồi quy: Dự đoán giá trị liên tục của một biến (ví dụ: giá nhà, doanh số).
* Phân tích rủi ro: Đánh giá khả năng xảy ra các sự kiện trong tương lai.

*Thuật toán SVM (Support Vector Machine) - Máy Hỗ Trợ Vecto*

SVM là một thuật toán học máy giám sát mạnh mẽ, thường được sử dụng cho cả bài toán phân loại và hồi quy. Tuy nhiên, nó nổi bật nhất trong các bài toán phân loại. Ý tưởng chính của SVM là tìm một siêu phẳng (hyperplane) tốt nhất để phân chia dữ liệu thành các lớp khác nhau.

* **Support Vector:** Là những điểm dữ liệu nằm gần nhất với siêu phẳng. Chúng đóng vai trò quan trọng trong việc xác định vị trí của siêu phẳng.
* **Machine:** Ám chỉ việc sử dụng máy tính để thực hiện các phép tính phức tạp trong quá trình tìm kiếm siêu phẳng.

**Nguyên lý hoạt động cơ bản**

1. **Tìm siêu phẳng tối ưu:** Mục tiêu của SVM là tìm một siêu phẳng sao cho khoảng cách từ siêu phẳng đến các điểm dữ liệu gần nhất (support vectors) là lớn nhất. Khoảng cách này được gọi là margin.
2. **Kernel trick:** Để xử lý dữ liệu không tuyến tính, SVM sử dụng kernel trick. Thay vì trực tiếp tìm siêu phẳng trong không gian gốc, SVM ánh xạ dữ liệu vào một không gian đặc trưng có chiều cao hơn, nơi dữ liệu có thể tách tuyến tính.

**Ưu điểm của SVM**

* Hiệu quả với dữ liệu có chiều cao: SVM hoạt động tốt với dữ liệu có nhiều đặc trưng.
* Khả năng tổng quát hóa tốt: SVM thường có khả năng dự đoán chính xác trên dữ liệu mới chưa từng thấy.
* Xử lý dữ liệu không tuyến tính: Nhờ kernel trick, SVM có thể giải quyết các bài toán phân loại phức tạp.
* Ít bị ảnh hưởng bởi hiệu ứng chiều dữ liệu: SVM không bị ảnh hưởng nhiều bởi "lời nguyền của chiều dữ liệu".

**Nhược điểm của SVM**

* Thời gian huấn luyện có thể lâu: Đặc biệt với tập dữ liệu lớn.
* Việc lựa chọn kernel và các tham số điều chỉnh có thể phức tạp.
* Khó giải thích: Mô hình SVM thường khó giải thích so với các mô hình khác như cây quyết định.

**Các loại SVM**

* Linear SVM: Sử dụng siêu phẳng tuyến tính để phân chia dữ liệu.
* Non-linear SVM: Sử dụng kernel trick để ánh xạ dữ liệu vào một không gian đặc trưng cao hơn, nơi dữ liệu có thể tách tuyến tính.
* One-class SVM: Dùng để phát hiện dị biệt.
* Nu-SVM: Là một biến thể của SVM, cho phép kiểm soát số lượng điểm dữ liệu nằm trong margin.

**Ứng dụng của SVM**

* Phân loại văn bản: Phân loại email, tin tức, ...
* Nhận dạng hình ảnh: Phân loại ảnh, phát hiện đối tượng.
* Phân tích gen: Phân loại gen, dự đoán protein.
* Dự báo tài chính: Dự báo giá cổ phiếu, phát hiện gian lận.

**Các tham số quan trọng trong SVM**

* C: Tham số điều chỉnh độ phạt cho các điểm dữ liệu nằm sai phía của siêu phẳng.
* Kernel: Hàm kernel (linear, polynomial, RBF, ...)
* Gamma: Tham số điều chỉnh độ rộng của hàm kernel RBF.

*SVM là một thuật toán mạnh mẽ và linh hoạt, được sử dụng rộng rãi trong nhiều lĩnh vực. Tuy nhiên, để sử dụng hiệu quả SVM, cần hiểu rõ về nguyên lý hoạt động và cách điều chỉnh các tham số của nó.*