# Một số thuật toán Machine Learning

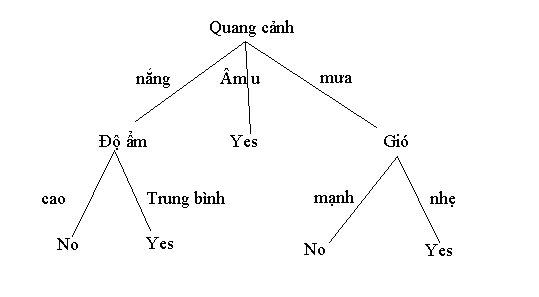
* Cây quyết định (Decision Trees)
* Phân loại bayes
* SVM (Support Vector Machine)
* KNN (K-near-neighbors)

## Thuật toán Cây quyết định (Decision Trees)

Đây là một trong những thuật toán phổ biến nhất được sử dụng hiện nay. Cây quyết định là thuật toán học có giám sát, dùng để phân loại các vấn đề. Thuật toán có thể thực hiện cho cả biến phân loại và biến liên tục. Trong thuật toán này, ta chia dữ liệu thành 2 hoặc nhiều lớp dựa trên phân loại theo các thuộc tính/biến quan trọng.

Đứng dưới góc nhìn thực tế, cây quyết định là một danh sách tối thiểu các câu hỏi dạng yes/no mà người ta phải hỏi, để đánh giá xác suất đưa ra quyết định đúng đắn.

Dưới đây là mô hình ví dụ về cây quyết định:



Ví dụ như ví dụ trên: Cây quyết định dữ liệu Quang cảnh.

Nhìn từ cây quyết định trên, ta có thể rút ra được kết luận: Nếu trời nắng & độ ẩm trung bình thì người chơi sẽ quyết định “Yes”.

Tuy nhiên, ta cần chú ý tới vấn đề [Overfitting](https://trituenhantao.io/tu-dien-thuat-ngu/overfitting/) trong thuật toán này. Để giảm hiện tượng này, ta thường dùng phương pháp cắt tỉa cây.

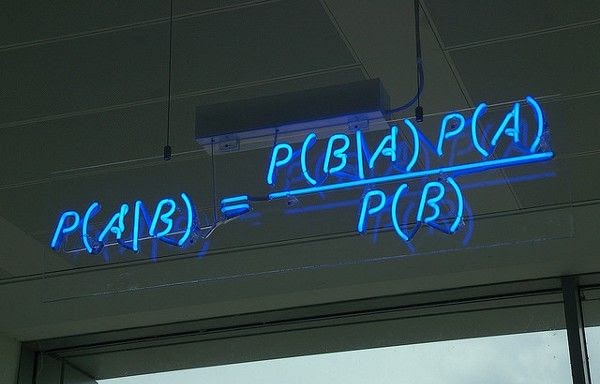
Với [Overfitting](https://trituenhantao.io/tu-dien-thuat-ngu/overfitting/) là hiện tượng mô hình ghi nhớ quá tốt dữ liệu huấn luyện và phụ thuộc vào nó, việc này khiến cho mô hình không thể tổng quát hóa các quy luật để hoạt động với dữ liệu chưa từng được chứng kiến.

* **Ưu điểm:**
* Mô hình sinh ra các quy tắc dễ hiểu cho người đọc, tạo ra bộ luật với mỗi nhánh lá là một luật của cây.
* Dữ liệu đầu vào có thể là là dữ liệu missing, không cần chuẩn hóa hoặc tạo biến giả**.**
* Có thể làm việc với cả dữ liệu số và dữ liệu phân loại**.**
* Có thể xác thực mô hình bằng cách sử dụng các kiểm tra thống kê.
* Có khả năng là việc với dữ liệu lớn.
* **Nhược điểm:**
* Mô hình cây quyết định phụ thuộc rất lớn vào dữ liệu. Thậm chí, với một sự thay đổi nhỏ trong bộ dữ liệu, cấu trúc mô hình cây quyết định có thể thay đổi hoàn toàn.
* Cây quyết định hay gặp vấn đề overfitting.

## Phân loại Bayes (Naive Bayes)

Thuật toán phân loại Naive Bayes là một nhóm các phân loại xác suất đơn giản dựa trên định lý Bayes giả định về việc độc lập giữa các thuộc tính. Ngay cả khi, các thuộc tính này có sự tương quan với nhau thì phương pháp này vẫn xem các thuộc tính là độc lập với nhau.

Định lý Bayes cho phép tính xác suất xảy ra của một sự kiện ngẫu nhiên A khi biết sự kiện liên quan B đã xảy ra. Xác suất này được ký hiệu là P(A|B), và đọc là “xác suất của A nếu có B”. Đại lượng này được gọi xác suất có điều kiện hay xác suất hậu nghiệm vì nó được rút ra từ giá trị được cho của B hoặc phụ thuộc vào giá trị đó.



Theo định lí Bayes, P(A|B) sẽ phụ thuộc vào 3 yếu tố:

- Xác suất xảy ra A của riêng nó, không quan tâm đến B. Kí hiệu là P(A).

- Xác suất xảy ra B của riêng nó, không quan tâm đến A. Kí hiệu là P(B).

- Xác suất xảy ra B khi biết A xảy ra. Kí hiệu là P(B|A). Đại lượng này gọi là khả năng xảy ra B khi biết A đã xảy ra.

Thuật toán này được áp dụng trong một số bài toán như:

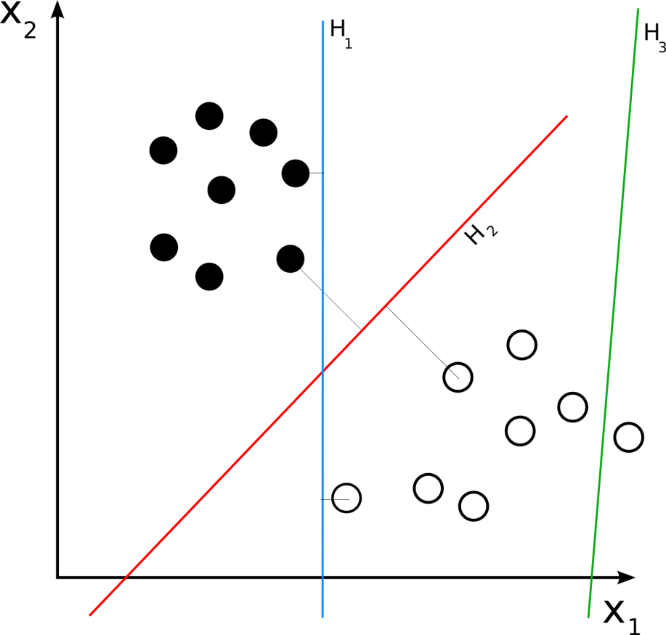
* Đánh dấu một email là spam hay không.
* Phân loại bài viết tin tức thuộc lĩnh vực công nghệ, chính trị hay thể thao.
* Kiểm tra một đoạn văn bản mang cảm xúc tích cực hay tiêu cực.
* Sử dụng cho các phần mềm nhận diện khuôn mặt.
* **Ưu điểm:**
* Giả định độc lập: hoạt động tốt cho nhiều bài toán/miền dữ liệu và ứng dụng.Đơn giản nhưng đủ tốt để giải quyết nhiều bài toán như phân lớp văn bản, lọc spam,…
* Cho phép kết hợp tri thức tiền nghiệm (prior knowledge) và dữ liệu quan sát được (observed data). Tốt khi có sự chênh lệch số lượng giữa các lớp phân loại.
* Huấn luyện mô hình (ước lượng tham số) dễ và nhanh.
* **Nhược điểm:**
* Giả định độc lập (ưu điểm cũng chính là nhược điểm): hầu hết các trường hợp thực tế trong đó có các thuộc tính trong các đối tượng thường phụ thuộc lẫn nhau.
* Vấn đề zero (đã nêu cách giải quyết ở phía trên).
* Mô hình không được huấn luyện bằng phương pháp tối ưu mạnh và chặt chẽ. Tham số của mô hình là các ước lượng xác suất điều kiện đơn lẻ. Không tính đến sự tương tác giữa các ước lượng này.

## Thuật toán SVM (Support Vector Machine)

SVM là phương pháp phân loại trong đó dữ liệu thô của bạn sẽ được biểu diễn trên không gian n chiều (n-số thuộc tính). Thông qua không gì biểu diễn diễn dự liệu đó, ta có thể thực hiện phân loại dữ liệu.

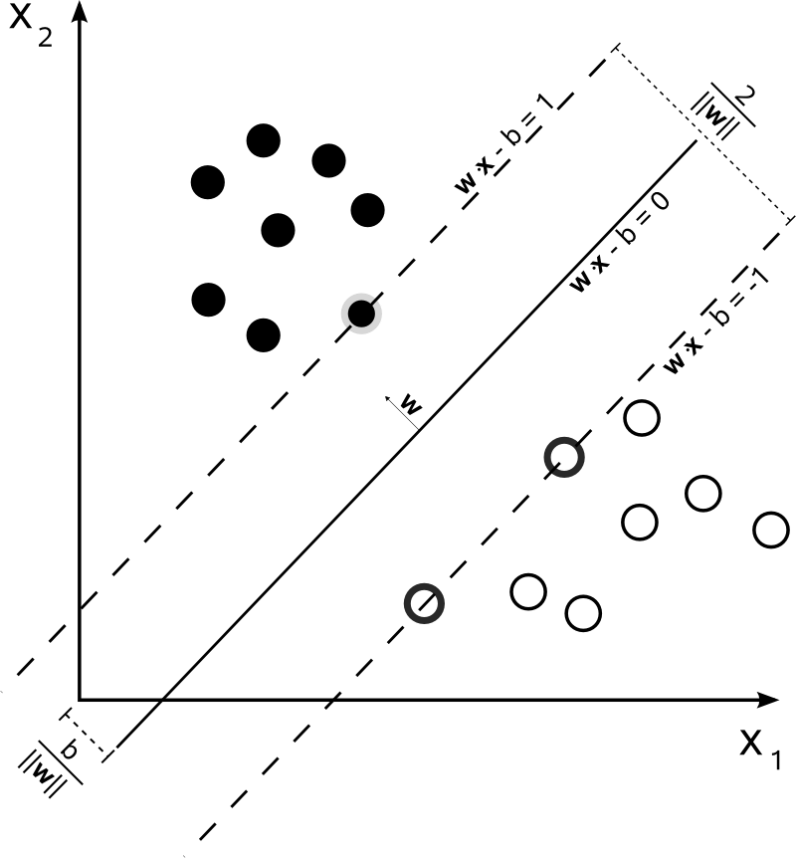
**Ý tưởng của SVM**: Tìm một siêu phẳng (hyper lane) phân chia "tốt nhất" tập dữ liệu. Siêu phẳng này sẽ chia không gian thành các miền khác nhau và mỗi miền sẽ chứa một loại dữ liệu.

Ví dụ: cho một tập các điểm thuộc 2 loại trong môi trường N chiều, SVM cố gắng tìm ra N-1 mặt phẳng để phân tách các điểm đó thành 2 nhóm.

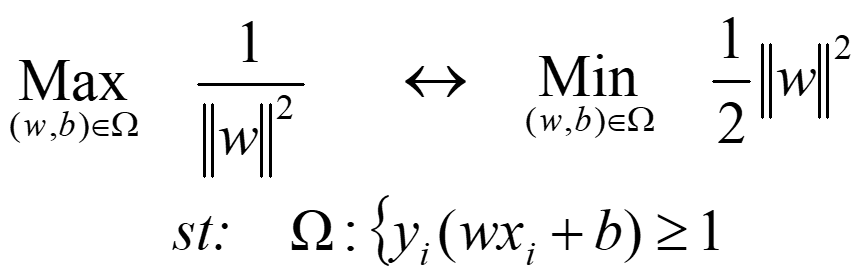


**Siêu phẳng** được biểu diễn bằng hàm số <W.X> = b (W và X là các vector <W.X> là tích vô) Hay W^T=b (W^T là ma trận chuyễn vị)

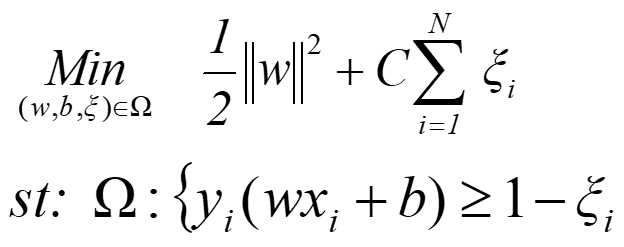
**Phân chia tốt nhất** là khoảng cách từ các đối tượng gần nhất tới siêu phẳng là cực đại (còn gọi là khoảng cách lề cực đại)



Mô hình SVM lề cứng:



Mô hình SVM lề mềm:



SVM thực hiện giải quyết rất nhiều dạng bài toán như hiển thị quảng cáo, phát hiện giới tính dựa trên hình ảnh, phân loại hình ảnh có quy mô lớn …

* **Ưu điểm:**
* Xử lý trên không gian số chiều cao: SVM là một công cụ tính toán hiệu quả trong không gian chiều cao, trong đó đặc biệt áp dụng cho các bài toán phân loại văn bản và phân tích quan điểm nơi chiều có thể cực kỳ lớn.
* Tiết kiệm bộ nhớ: Do chỉ có một tập hợp con của các điểm được sử dụng trong quá trình huấn luyện và ra quyết định thực tế cho các điểm dữ liệu mới nên chỉ có những điểm cần thiết mới được lưu trữ trong bộ nhớ khi ra quyết định.
* Tính linh hoạt - phân lớp thường là phi tuyến tính. Khả năng áp dụng Kernel mới cho phép linh động giữa các phương pháp tuyến tính và phi tuyến tính từ đó khiến cho hiệu suất phân loại lớn hơn.
* **Nhược điểm:**
* Bài toán số chiều cao: Trong trường hợp số lượng thuộc tính (p) của tập dữ liệu lớn hơn rất nhiều so với số lượng dữ liệu (n) thì SVM cho kết quả khá tồi.
* Chưa thể hiện rõ tính xác suất: Việc phân lớp của SVM chỉ là việc cố gắng tách các đối tượng vào hai lớp được phân tách bởi siêu phẳng SVM. Điều này chưa giải thích được xác suất xuất hiện của một thành viên trong một nhóm là như thế nào.

## Thuật toán KNN (K-near-neighbors)

K-nearest neighbors là thuật toán học máy có giám sát, đơn giản và dễ triển khai. Thường được dùng trong các bài toán phân loại và hồi quy.

**Bài toán đặt ra:** Giả sử bạn có điểm của một môn học nhưng bạn không biết thuộc loại nào (Giỏi, khá, trung bình, yếu) và không biết bất kỳ quy tắc nào để phân loại cả.

Có một cách giải quyết là bạn phải đi khảo sát những người xung quanh. Để biết điểm của mình thuộc loại nào thì bạn phải đi hỏi những người có điểm gần số điểm của mình nhất. Giả sử trong lớp có 50 người, bạn khảo sát 5 người gần điểm mình nhất và được dữ liệu như sau:

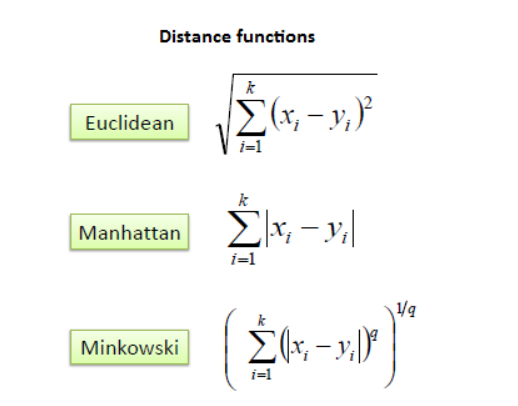
Điểm của bạn: 7

Điểm bạn của bạn:

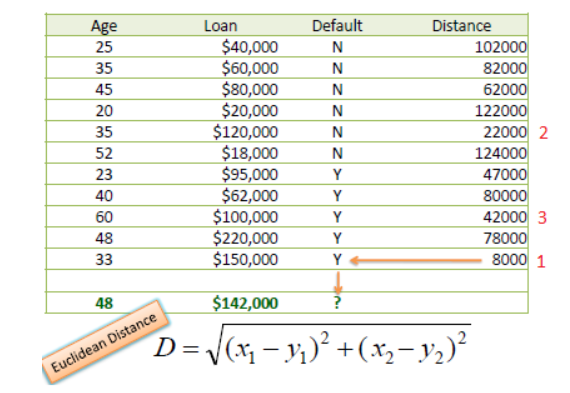
* 7.1=>Khá
* 7.2=>Khá
* 6.7=>Khá
* 6.6=>Khá
* 6.4=>Trung bình

Qua kết quả trên thì bạn sẽ đoán được là mình thuộc loại nào. Với cách này chúng ta có thể phân biệt loại dữ liệu 1 chiều bằng. Nếu ta khảo sát càng nhiều, càng rộng thì dự đoán đưa ra càng chính xác.

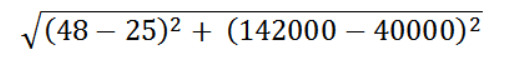
**Ý tưởng của KNN**: Thuật toán KNN cho rằng dữ liệu tương quan nhau sẽ tồn tại gần nhau tròn một không gian, từ đó công việc của chúng ta là sẽ tìm k điểm gần với dữ liệu cần kiểm tra nhất. Việc tìm khoảng cách giữa hai điểm cũng có nhiều công thức có thể sử dụng tùy trường hợp mà ta lựa chọn cho phù hợp. Đây là 3 cách cơ bản để tính khoảng cách 2 điểm dữ liệu x, y có k thuộc tính:



Ví dụ chúng ta có dữ liệu là tuổi, khoản vay và khả năng vỡ nợ như hình sau:



Dữ liệu cần phân loại của chúng ta là {age: 48, loan:142000}. Đây là dữ liệu 2 chiều và chúng ta cần dự đoán người này thuộc nguy cơ vỡ nợ hay không. Chúng ta sé dùng một cách khá phổ biến để tính khoảng cách và Euclidean. Ví dụ ở hàng đầu tiên khoảng cách sẽ được tính:



Thực hiện tương tự, ta sẽ tính được khoảng cahcs ở cột Distance, từ đó chọn ra k = 3 khoảng cách nhỏ nhất (gần với dữ liệu nào nhất). Với 3 khoảng cách này chúng ta nhận được 3 label là (Yes, No, Yes). Trong 3 label này Yes xuất hiện nhiều hơn nên chúng ta sẽ đưa ra dự đoán người này có khả năng vỡ nợ.

* **Ưu điểm:**
* Là một thuật toán đơn giản để hiểu và giải thích.
* Rất hữu ích cho dữ liệu phi tuyến vì không có giả định về dữ liệu trong thuật toán này.
* Là một thuật toán linh hoạt vì chúng ta có thể sử dụng nó để phân loại cũng như hồi quy.
* Có độ chính xác tương đối cao nhưng có nhiều mô hình học có giám sát tốt hơn KNN.
* **Nhược điểm:**
* Về mặt tính toán, nó là một thuật toán hơi tốn kém vì nó lưu trữ tất cả các dữ liệu huấn luyện.
* Yêu cầu bộ nhớ lưu trữ cao so với các thuật toán học có giám sát khác.
* Dự đoán chậm trong trường hợp N lớn.
* Rất nhạy cảm với quy mô dữ liệu cũng như các tính năng không liên quan.
* **Các ứng dụng của KNN:**
* Hệ thống ngân hàng: KNN có thể được sử dụng trong hệ thống ngân hàng để dự đoán một cá nhân có phù hợp để phê duyệt khoản vay không? Cá nhân đó có các đặc điểm giống với cá nhân mặc định không.
* Tính toán xếp hạng tín dụng: Các thuật toán KNN có thể được sử dụng để xếp hạng tín dụng của một cá nhân bằng cách so sánh với những người có đặc điểm tương tự .
* Chính trị: Với sự trợ giúp của thuật toán KN, chúng ta có thể phân loại cử tri tiềm năng thành nhiều lớp khác nhau như “Sẽ bỏ phiếu”, “Sẽ không bỏ phiếu”, “Sẽ bỏ phiếu cho Đảng ‘Đại hội’,” Sẽ bỏ phiếu cho Đảng ‘BJP’.
* Các lĩnh vực khác mà thuật toán KNN có thể được sử dụng là nhận dạng giọng nói, phát hiện chữ viết tay, nhận dạng hình ảnh và nhận dạng video.