**Thuật Toán K-Nearest Neighbors (KNN) Siêu Cơ Bản**

**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. Trong bài viết hôm nay, mình và các bạn sẽ cùng tìm hiểu và đi qua một ví dụ đơn giản để hiểu rõ hơn về KNN nhé. Chúng ta sẽ đi qua các phần:**

1. Ví dụ đơn giản nhất
2. Ý tưởng của KNN
3. Thực hành với ví dụ thực tế

Ok bắt đầu thôi nào.

**1. Ví dụ đơn giản nhất**

**Bài toán đặt ra**: Bạn có điểm của một môn học nhưng bạn không biết thuộc loại nào (Giỏi, khá, trung bình, yếu). Giả sử bạn không biết bất kì 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 đứa có điểm gần số điểm mình nhất. Giả sử trong lớp 50 đứa, mình khảo sát 5 đứa gần điểm mình nhất và được dữ liệu như sau:

Điểm của tôi: 7

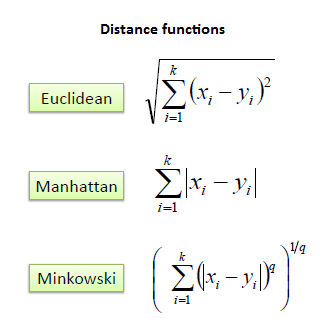
Điểm của bạn tôi:

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

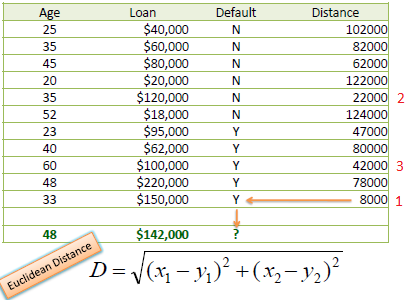
Qua kết quả trên thì mình sẽ mạnh dạng đoán là mình loại khá đúng không nào? Với cách này chúng ta có thể phân loại dữ liệu 1 chiều (1 feature) bằng cách làm khá đơn giản. Và các bạn có nhận thấy rằng dữ liệu mình khảo sát càng nhiều, càng rộng thì dự đoán đưa ra càng chính xác (Giả sử lớp bạn không có ai loại khá ngoài bạn thì cho dù bạn lấy bao nhiêu người gần điểm bạn nhất củng sẽ ra kết quả sai). Okey mình sẽ tiếp tới phần tiếp theo để cùng tìm hiểu tổng quát ý tưởng của thuật toán KNN là gì nhé?

**2. Ý tưởng của KNN**

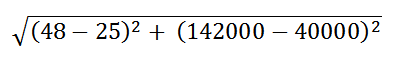
Thuật toán KNN cho rằng những dữ liệu tương tự nhau sẽ tồn tại **gần nhau** trong một không gian, từ đó công việc của chúng ta là sẽ tìm k điểm gần với dữ liệu cần kiểm tra nhất. Việc tìm khoảng cách giữa 2 điểm củng có nhiều công thức có thể sử dụng, tùy trường hợp mà chúng ta lựa chọn cho phù hợp. 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 vở nợ như hình:

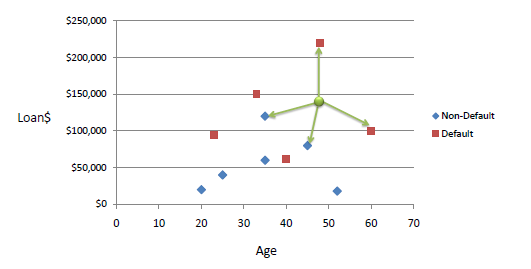


Dữ liệu cần phân loại của chúng ta là {age: 48, loan: 142000}. Đây dữ liệu 2 chiều và chúng ta cần dự đoán người này thuộc nguy cơ vở nợ hay không. Chúng ta sẽ dùng một cách khá phổ biến để tính khoảng cách là Euclidean. Ví dụ ở hàng đầu tiên khoảng cách sẽ được tính:



Thực hiện tương tự, ta sẽ tính được khoảng cách ở cột Distance, từ đó chọn ra k = 3 khoảng cách nhỏ nhất (gần với dữ liệu vào nhất). Với 3 khoảng cách này chúng ra 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 vở nợ.

Vì đây là dử liệu 2 chiều nên chúng ta củng có thể biểu diễn dữ liệu trong hệ tọa độ như hình:



Trên hệ tọa độ này chúng ta thể dễ dàng nhận thấy cách chúng ta chọn k điểm gần nhất. Nhưng với dữ liệu lớn, nhiều chiều thì việc biểu diễn dữ liệu trên một không gian là không hề dễ dàng.

Ưu điểm của KNN

1. Độ phức tạp tính toán của quá trình training là bằng 0.
2. Việc dự đoán kết quả của dữ liệu mới rất đơn giản.
3. Không cần giả sử gì về phân phối của các class.

Nhược điểm của KNN

1. KNN rất nhạy cảm với nhiễu khi K nhỏ.
2. Như đã nói, KNN là một thuật toán mà mọi tính toán đều nằm ở khâu test. Trong đó việc tính khoảng cách tới *từng* điểm dữ liệu trong training set sẽ tốn rất nhiều thời gian, đặc biệt là với các cơ sở dữ liệu có số chiều lớn và có nhiều điểm dữ liệu. Với K càng lớn thì độ phức tạp cũng sẽ tăng lên. Ngoài ra, việc lưu toàn bộ dữ liệu trong bộ nhớ cũng ảnh hưởng tới hiệu năng của KNN.