**Quy trình tìm K của thuật toán KNN (The process of finding K of the KNN algorithm )**

Bài toán đặt ra là phân loại dữ liệu dựa trên các thuộc tính như :

1. Giới tính, tuổi, huyết áp, bệnh tim, tình trạng hôn nhân, loại công việc, loại cư trú, mức độ đường huyết trung bình, chỉ số BMI, tình trạng hút thuốc và tình trạng đột quỵ.
2. Chúng ta có thể sử dụng thuật toán KNN để phân loại dữ liệu dựa trên các thuộc tính này.

Ý tưởng của KNN:

1. KNN cho rằng những dữ liệu tương tự nhau sẽ tồn tại gần nhau trong không gian.
2. Chúng ta cần tìm k điểm gần nhất với dữ liệu cần kiểm tra để đưa ra dự đoán.
3. Có nhiều cách để tính khoảng cách giữa hai điểm dữ liệu, như Euclidean.

Thực hành với ví dụ thực tế:

Dựa trên dữ liệu, chúng ta có thể áp dụng thuật toán KNN để phân loại dữ liệu.

Ví dụ bài toán cụ thể : Chúng ta có một dữ liệu với các thuộc tính như giới tính, tuổi, huyết áp, bệnh tim, tình trạng hôn nhân, loại công việc, loại cư trú, mức độ đường huyết trung bình, chỉ số BMI, tình trạng hút thuốc và tình trạng đột quỵ.

Chúng ta có thể tính khoảng cách giữa dữ liệu cần phân loại và các điểm dữ liệu khác trong tập dữ liệu.

Dựa vào k điểm gần nhất, chúng ta có thể đưa ra dự đoán về loại dữ liệu của dữ liệu cần phân loại.

Các bước làm :

Chuẩn bị dữ liệu:

1. Tiền xử lý dữ liệu bằng cách mã hóa các giá trị của các thuộc tính như đã được mô tả trước đó.
2. Chia tập dữ liệu thành tập huấn luyện và tập kiểm tra.

Xác định số lượng k điểm gần nhất:

1. Chọn một giá trị k, đại diện cho số lượng điểm gần nhất mà chúng ta muốn sử dụng để đưa ra dự đoán. Giá trị k có thể được chọn dựa trên các phương pháp như kiểm định chéo hoặc tối ưu hóa.

Tính khoảng cách:

1. Sử dụng một phương pháp tính khoảng cách như khoảng cách Euclidean để tính toán khoảng cách giữa dữ liệu cần phân loại và các điểm dữ liệu trong tập huấn luyện.

Tìm k điểm gần nhất:

1. Chọn k điểm gần nhất dựa trên khoảng cách tính được.

Đưa ra dự đoán:

1. Dựa vào loại dữ liệu của k điểm gần nhất, đưa ra dự đoán về loại dữ liệu của dữ liệu cần phân loại.

Ví dụ: Nếu hầu hết trong k điểm gần nhất là những người có đột quỵ, chúng ta có thể dự đoán rằng dữ liệu cần phân loại cũng thuộc loại đột quỵ.

Lưu và đánh giá mô hình:

1. Đánh giá mô hình bằng cách so sánh dự đoán với nhãn thực tế của dữ liệu kiểm tra.
2. Sử dụng các độ đo như độ chính xác (accuracy), độ chính xác dương tính (precision) để đánh giá hiệu suất của mô hình.

Giải :

1. Đầu tiên tôi sẽ chia dữ liệu thành tỷ lệ 80-20, tức là 80% dữ liệu cho tập huấn luyện và 20% dữ liệu cho tập kiểm tra. Tôi sẽ được tập dữ liệu như sau :   
   A screenshot of a computer program

   Description automatically generated

Sau đó in ra dữ liệu sẽ bị cắt ra và không theo thứ tự

1. Tiếp theo tôi sẽ lấy ra chọn ra một dòng dữ liệu mới sau :

X = [0, 80, 0, 1, 0 , 2 ,1, 206, 32, 3]

Tôi sẽ lấy điểm đầu tiên trong dữ liệu training làm ví dụ:

Y = [0, 79, 0, 1, 0, 2, 1, 205.33, 31, 3, 1]

Trong đó ta thấy :

Target (hoặc Labels):

* stroke = 1 có nghĩa là đột quỵ

Các Features (đặc trưng):

* gender = 0
* age = 79
* hypertension = 0
* heart\_disease = 1
* ever\_married = 0
* work\_type = 2
* Residence\_type = 1
* avg\_glucose\_level = 205.33
* bmi = 31
* smoking\_status = 3

Tôi sẽ lấy điểm dữ liệu đó tính khoảng cách eclude giữa 2 điểm :

dist = sqrt((x1 - y1)^2 + (x2 - y2)^2 + ... + (x10 - y10)^2)

Khi đó ta thay số vào sẽ được :

Cách tính khoảng cách Euclid giữa X và Y sử dụng công thức :

dist = sqrt((0 - 0)^2 + (80 - 79)^2 + (0 - 0)^2 + (1 - 1)^2 + (0 - 0)^2 + (2 - 2)^2 + (1 - 1)^2 + (206 - 205.33)^2 + (32 - 31)^2 + (3 - 3)^2)

dist = sqrt(0 + 1 + 0 + 0 + 0 + 0 + 0 + 0.4489 + 1 + 0)

dist = sqrt(2.4489)

dist ≈ 1.564

Vậy, khoảng cách Euclidean giữi vectơ X và Y là khoảng cách xấp xỉ 1.564.

Tiếp tục chúng ta tính tương tự với tất cả các điểm còn lại trong data training, tôi thu được danh sách các khoảng cách:

26.664510121132917, 19.74882469632649, 100.10123076166447, 46.66597154244194, 32.974753979370334,...

Sau đó, tôi sẽ sắp xếp các khoảng cách đã tính được từ bé đến lớn, và lấy ra top K điểm gần nhất với điểm dữ liệu mới.

Khi đó tôi quan sát ở đây tôi chọn K = 5 bởi đây là giá trị phổ biến và phù hợp trong KNN. Nếu K quá nhỏ thì kết quả dễ bị ảnh hưởng bởi outlier. Ngược lại nếu K quá lớn thì kết quả sẽ mất tính LOCAL.

Vì vậy K = 5 là giá trị tương đối tốt để cân bằng, tránh overfitting hoặc underfitting.

Như vậy tôi đã giải thích chi tiết từng bước tính toán khoảng cách Euclidean và lý do chọn K = 5.

Phần thực hành :

Xem tui làm !