

**Giảng viên: Ths.Lê Thị Thủy**

**Lớp: 20221IT6043011**

**Nhóm: 7**

**Thành viên:**

1. **Đoàn Đức Duy - 2020605731**
2. **Nguyễn Quang Huy - 2020606068**
3. **Lê Ngọc Trường – 2020601391**

**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN: TRÍ TUỆ NHÂN TẠO**

**ĐỀ TÀI: Dự báo thời tiết sử dụng thuật toán KNN**

*Hà Nội, năm 2022*

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

MỤC LỤC

[PHỤ LỤC 3](#_Toc122310537)

[LỜI MỞ ĐẦU 4](#_Toc122310538)

[CHƯƠNG 1: MỘT SỐ THUẬT TOÁN MACHINE LEARNING 6](#_Toc122310539)

[1.1 Thuật toán Cây quyết định (Decision Trees) 6](#_Toc122310540)

[1.2 Phân loại Bayes (Naive Bayes) 7](#_Toc122310541)

[1.3 Thuật toán SVM (Support Vector Machine) 9](#_Toc122310542)

[1.4 Thuật toán KNN (K-near-neighbors) 13](#_Toc122310543)

[CHƯƠNG 2: XÂY DỰNG ỨNG DỤNG 17](#_Toc122310544)

[2.1. Phân tích đề tài 17](#_Toc122310545)

[2.2. Thuật toán áp dụng 17](#_Toc122310546)

[2.3: Xây dựng bộ dữ liệu nhỏ và giải bằng tay 17](#_Toc122310547)

[2.4. Bộ dữ liệu dành cho máy 21](#_Toc122310548)

[2.5. Cài đặt thuật toán 28](#_Toc122310549)

[2.6. Kết quả 33](#_Toc122310550)

[Kết luận 36](#_Toc122310551)

[1. Kết quả đạt được 36](#_Toc122310552)

[2. Hạn chế 36](#_Toc122310553)

[Tài liệu tham khảo 37](#_Toc122310554)

# PHỤ LỤC

[Hình 1:Ví dụ minh hoạ về mô hình cây quyết định 6](#_Toc122310458)

[Hình 2: Định lý Bayes 8](#_Toc122310459)

[Hình 3: Biểu đồ minh hoạ SVM 10](#_Toc122310460)

[Hình 4:Biểu đồ minh hoạ SVM 11](#_Toc122310461)

[Hình 5: Một số cách tính khoảng cách 2 điểm dữ liệu x, y có k thuộc tính 14](#_Toc122310462)

[Hình 6: Minh hoạ cách tính khoảng cách Euclidean 15](#_Toc122310463)

[Hình 7: Bảng dữ liệu nhỏ minh hoạ việc giải bằng tay 17](#_Toc122310464)

[Hình 8: Bảng dữ liệu 100 bản ghi đầu 27](#_Toc122310465)

[Hình 9: Mô hình thuật toán KNN áp dụng cho bài toán 28](#_Toc122310466)

[Hình 10: Khai báo các thư viện cho chương trình 28](#_Toc122310467)

[Hình 11: Đoạn mã lấy dữ liệu từ file csv 29](#_Toc122310468)

[Hình 12: Minh hoạ kết quả trả về từ bộ trainSet và testSet 29](#_Toc122310469)

[Hình 13: Xây dựng hàm tính khoảng cách 30](#_Toc122310470)

[Hình 14: Minh hoạ khoảng cách tính được giữa 2 bản ghi dữ liệu 30](#_Toc122310471)

[Hình 15: Xây dựng hàm tìm ra các nhãn có khoảng cách nhỏ nhất 30](#_Toc122310472)

[Hình 16: Kết quả trả về k(5) nhãn có khoảng cách nhỏ nhất 31](#_Toc122310473)

[Hình 17: Xây dựng hàm tìm trạng thái thời tiết xuất hiện nhiều nhất trong k trạng thái thời tiết tìm được 31](#_Toc122310474)

[Hình 18: Hàm main của chương trình 32](#_Toc122310475)

[Hình 19: Ví dụ kết quả trả về độ chính xác 32](#_Toc122310476)

[Hình 20: Nhập dữ liệu từ bàn phím 32](#_Toc122310477)

[Hình 21:Tạo hàm đọc dữ liệu từ file csv 33](#_Toc122310478)

[Hình 22: Đưa ra dự đoán của bộ dữ liệu vừa nhập 33](#_Toc122310479)

[Hình 23: So sánh labels và predicted trên bộ testSet 34](#_Toc122310480)

[Hình 24: Ví dụ dữ liệu đã nhập 35](#_Toc122310481)

[Hình 25: Minh hoạ kết quả bài toán 35](#_Toc122310482)

# LỜI MỞ ĐẦU

Trí tuệ nhân tạo tiếng anh là **Artificial intelligence** – viết tắt là **AI**. Chúng ta có thể hiểu đó như là một ngành của khoa học máy tính. Con người lập trình nên nó với mục đích để giúp máy tính có thể thực hiện các hành vi thông minh và xử lý các yêu cầu giống như con người.

Những năm gần đây, AI nổi lên như một bằng chứng của cuộc cách mạng công nghiệp lần thứ tư. Theo đà phát triển của công nghệ,

ứng dụng trí tuệ nhân tạo luôn là xu hướng công nghệ tương lai mà các hãng công nghệ trên toàn thế giới đua nhau sáng tạo, nó là nền tảng cốt lõi của cuốc cách mạng công nghệ 4.0. Hầu hết mọi nghành công nghiệp đang làm việc với hàm lượng lớn dữ liệu đều nhận ra tầm quan trọng của công nghệ ML. Những cái nhìn sáng suốt từ nguồn dữ liệu này – chủ yếu dạng thời gian thực – sẽ giúp các tổ chức vận hành hiệu quả hơn hoặc tạo lợi thế cạnh tranh so với các đối thủ. Các ứng dụng của ML đã quá quen thuộc với con người: xe tự hành của Google và Tesla, hệ thống tự tag khuôn mặt trên Facebook, hệ thống gợi ý sản phẩm của Amazon, hệ thống gợi ý phim của Netflix…, chỉ là một vài trong vô vàn những ứng dụng của trí tuệ nhân tạo và cụ thể là ML.

Nhằm vận dụng những kiến thức đã học để áp dụng vào thực tế, trong bài tập lớn này chúng em đã nghiên cứu ứng dụng của trí tuệ nhân tạo trong việc dự báo thời tiết và thuật toán được sử dụng đó là thuật toán KNN- (K-Nearest Neighbors) được viết bằng ngôn ngữ Python.

Với sự hướng dẫn, động viên tận tình của cô Lê Thị Thủy, chúng em đã hiểu được cách thức hoạt động của thuật toán và hoàn thành bài báo cáo này. Vì lượng kiến thức và trình độ chuyên môn có hạn nên trong tài liệu không tránh khỏi những sai sót, nhóm sinh viên chúng em rất mong nhận được sự góp ý từ cô và bạn đọc để đề tài của chúng em được hoàn thiện hơn.

Nhóm chúng em xin chân thành cảm ơn.

# CHƯƠNG 1: MỘT SỐ THUẬT TOÁN MACHINE LEARNING

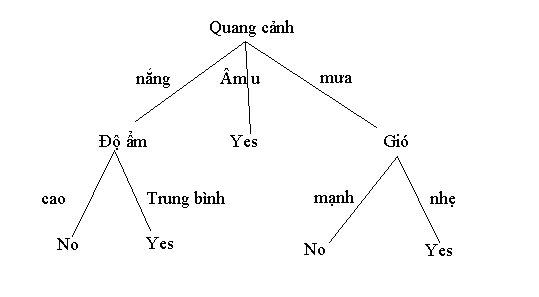
* Cây quyết định (Decision Trees)
* Phân loại bayes
* SVM (Support Vector Machine)
* KNN (K-near-neighbors)

## 1.1 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:



Hình 1:Ví dụ minh hoạ về mô hình 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.

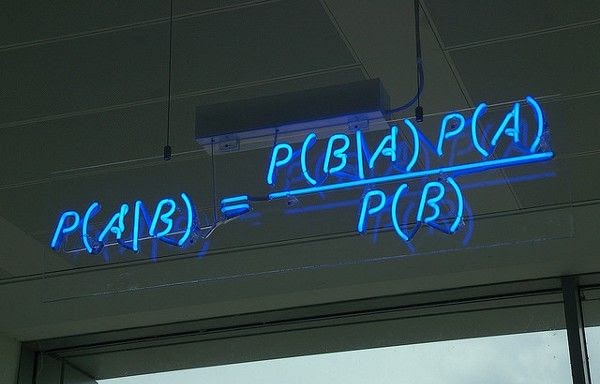
Với [Overfitting](https://trituenhantao.io/tu-dien-thuat-ngu/overfitting/) là hiện tượng mô hình ghi nhớ quá tốt dữ liệu huấn luyện và phụ thuộc vào nó, việc này khiến cho mô hình không thể tổng quát hóa các quy luật để hoạt động với dữ liệu chưa từng được chứng kiến.

* **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.

## 1.2 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.



Hình 2: Định lý Bayes

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:

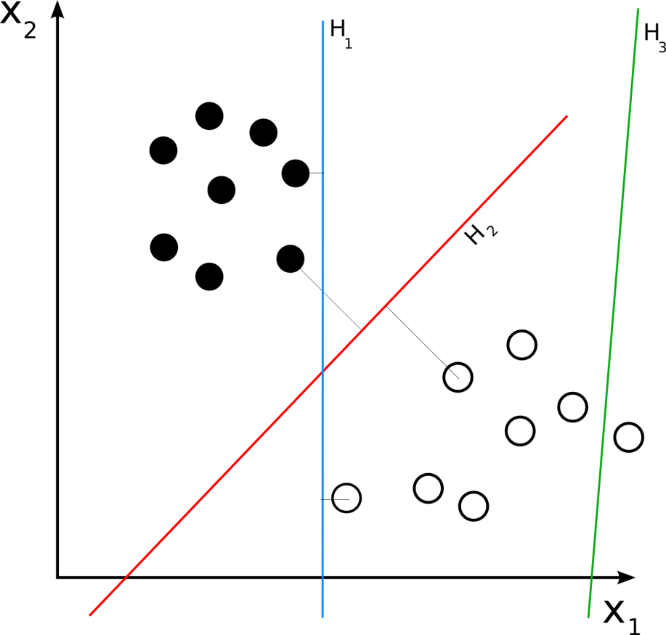
* Đánh dấu một email là spam hay không.
* Phân loại bài viết tin tức thuộc lĩnh vực công nghệ, chính trị hay thể thao.
* Kiểm tra một đoạn văn bản mang cảm xúc tích cực hay tiêu cực.
* Sử dụng cho các phần mềm nhận diện khuôn mặt.
* **Ưu điểm:**
* Giả định độc lập: hoạt động tốt cho nhiều bài toán/miền 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.

## 1.3 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.

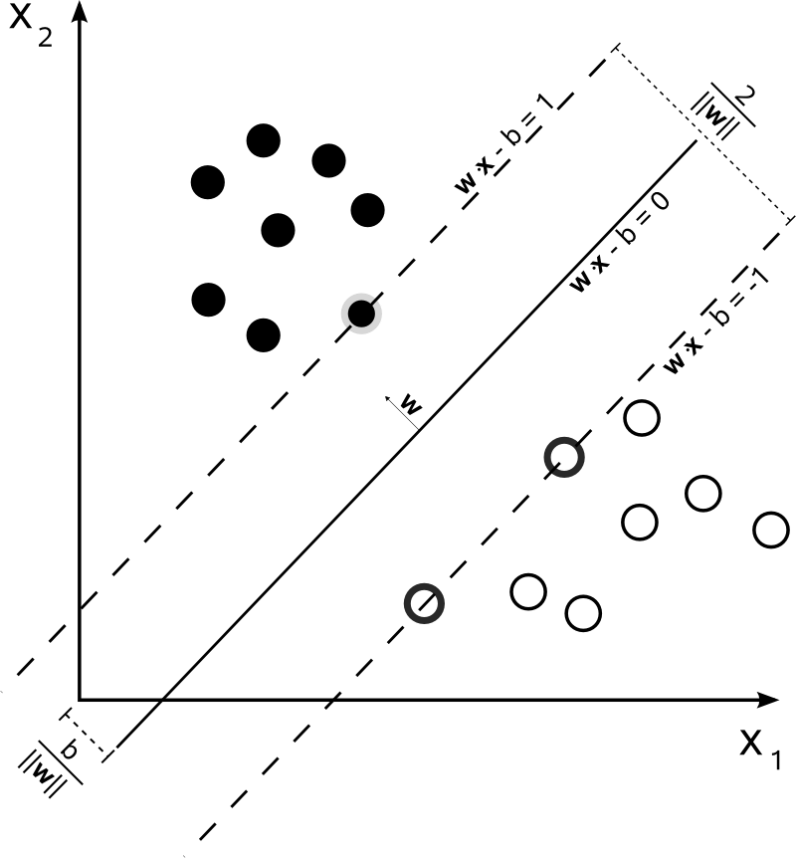
Ví dụ: cho một tập các điểm thuộc 2 loại trong môi trường N chiều, SVM cố gắng tìm ra N-1 mặt phẳng để phân tách các điểm đó thành 2 nhóm.



Hình 3: Biểu đồ minh hoạ SVM

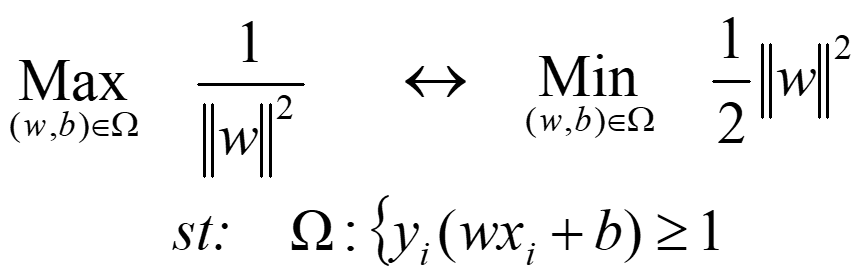
**Siêu phẳng** được biểu diễn bằng hàm số <W.X> = b (W và X là các vector <W.X> là tích vô) Hay W^T=b (W^T là ma trận chuyễn vị)

**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)

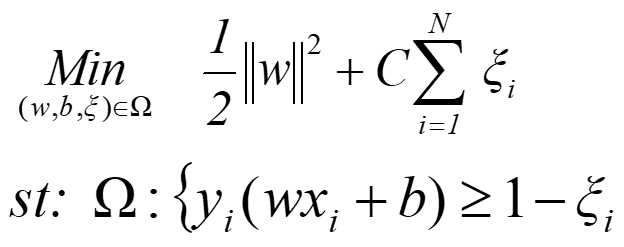


Hình 4:Biểu đồ minh hoạ SVM

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.

## 1.4 Thuật toán KNN (K-nearest-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:

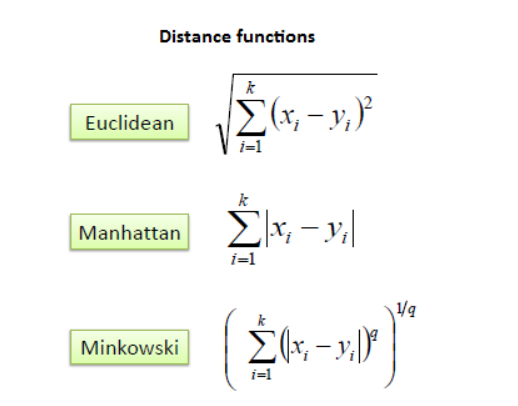
Điểm của bạn: 7

Điểm bạn của bạn:

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

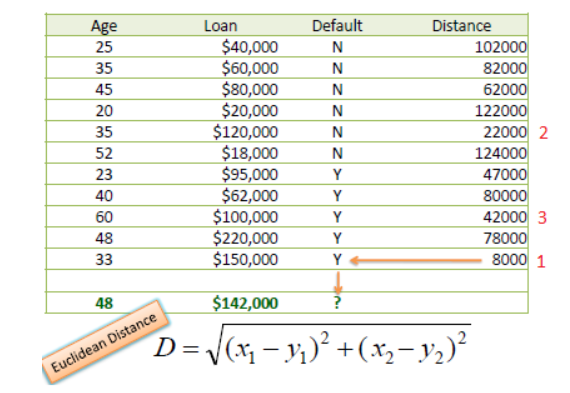
Qua kết quả trên thì bạn sẽ đoán được là mình thuộc loại nào. Với cách này chúng ta có thể phân biệt loại dữ liệu 1 chiều bằng. Nếu ta khảo sát càng nhiều, càng rộng thì dự đoán đưa ra càng chính xác.

**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:



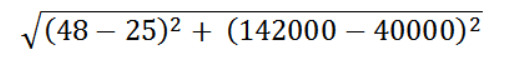
Hình 5: Một số cách tính khoảng cách 2 điểm dữ liệu x, y có k thuộc tính

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



Hình 6: Minh hoạ cách tính khoảng cách Euclidean

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.

# CHƯƠNG 2: XÂY DỰNG ỨNG DỤNG

## 2.1. Phân tích đề tài

Ngày nay, nhu cầu đi lại, tổ chức sự kiện của con người ngày càng tăng cao. Vì vậy, nếu ta nắm bắt được tình hình thời tiết như lượng mưa, độ ẩm, nhiệt độ,…thì sẽ dễ dàng sắp xếp cũng như tổ chức sự kiện, cuộc vui chơi một cách tốt nhất. Hiểu được nhu cầu cần thiết đó, nhóm em đã quyết định xây dựng bài toán “Dự báo thời tiết” để giúp người dùng có thể dễ dàng có những quyết định tốt nhất cho những dự định sắp tới của mình.

## 2.2. Thuật toán áp dụng

Bước 1: Tính khoảng cách từ X tới tất cả các đối tượng trong bộ dữ liệu huấn luyện

Bước 2: Chọn ra k đối tượng gần với X nhất (khoảng cách tới X nhỏ nhất)

Bước 3: Phân lớp X vào lớp xuất hiện phổ biến nhất trong k đối tượng.

## 2.3: Xây dựng bộ dữ liệu nhỏ và giải bằng tay

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ngày** | **Nhiệt độ (°C)** | **Độ ẩm (%)** | **Tốc độ gió (km/h)** | **Mật độ mây (%)** | **Thời tiết** |
| D01 | 26 | 79 | 8 | 75 | Nhiều mây |
| D02 | 23 | 86 | 9 | 100 | Mưa |
| D03 | 26 | 76 | 7 | 45 | Nắng |
| D04 | 24 | 84 | 13 | 100 | Mưa |
| D05 | 29 | 72 | 13 | 15 | Nắng |
| D06 | 32 | 66 | 11 | 4 | Nắng |
| D07 | 27 | 88 | 11 | 73 | Nhiều mây |
| D08 | 20 | 86 | 13 | 100 | Mưa |
| D09 | 22 | 97 | 10 | 99 | Nhiều mây |
| D10 | 25 | 95 | 12 | 77 | ? |
| D11 | 15 | 68 | 15 | 100 | ? |
| D12 | 28 | 71 | 13 | 78 | ? |
| D13 | 22 | 91 | 7 | 100 | ? |

Hình 7: Bảng dữ liệu nhỏ minh hoạ việc giải bằng tay

**Giải**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x | 25 | 95 | 12 | 77 | ? |

**Áp dụng công thức tính khoảng cách Euclidean:**



Ta có:

d(x, D01) = 16.64

d(x, D02) = 24.96

d(x, D03) = 37.56

d(x, D04) = 25.53

d(x, D05) = 66.26

d(x, D06) = 78.87

d(x, D07) = 8.37

d(x, D08) = 25.22

d(x, D09) = 22.38

Chọn ra 3 láng giềng gần x nhất: D07, D01, D09

Ta xác định được lớp của x là Nhiều mây, là lớp của 3 láng giềng gần nhất của x.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| y | 15 | 68 | 15 | 100 |

**Áp dụng công thức tính khoảng cách Euclidean:**



Ta có:

d(y, D01) = 30.26

d(y, D02) = 20.59

d(y, D03) = 57.22

d(y, D04) = 18.47

d(y, D05) = 86.26

d(y, D06) = 97.59

d(y, D07) = 35.9

d(y, D08) = 18.79

d(y, D09) = 30.27

Chọn ra 4 láng giềng gần y nhất: D04, D08, D02, D01

Ta xác định được lớp của y là Mưa, là lớp của 3 trong 4 láng giềng gần nhất của y.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| z | 28 | 71 | 13 | 78 |

**Áp dụng công thức tính khoảng cách Euclidean:**



Ta có:

d(z, D01) = 10.10

d(z, D02) = 27.39

d(z, D03) = 33.97

d(z, D04) = 25.87

d(z, D05) = 63.02

d(z, D06) = 74.30

d(z, D07) =17.86

d(z, D08) = 27.80

d(z, D09) = 34.09

Chọn ra 3 láng giềng gần z nhất: D01, D07, D04

Ta xác định được lớp của z là Nhiều mây, là lớp của 2 trong 3 láng giềng gần nhất của z.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| t | 30 | 91 | 8 | 20 |

**Áp dụng công thức tính khoảng cách Euclidean:**



Ta có:

d(t, D01) = 56.44

d(t, D02) = 80.47

d(t, D03) = 29.44

d(t, D04) = 80.68

d(t, D05) = 20.29

d(t, D06) = 29.89

d(t, D07) = 53.25

d(t, D08) = 80.93

d(t, D09) = 79.66

Chọn ra 5 láng giềng gần t nhất: D05, D03, D06, D07, D01

Ta xác định được lớp của t là Nắng, là lớp của 3 trong 5 láng giềng gần nhất của t.

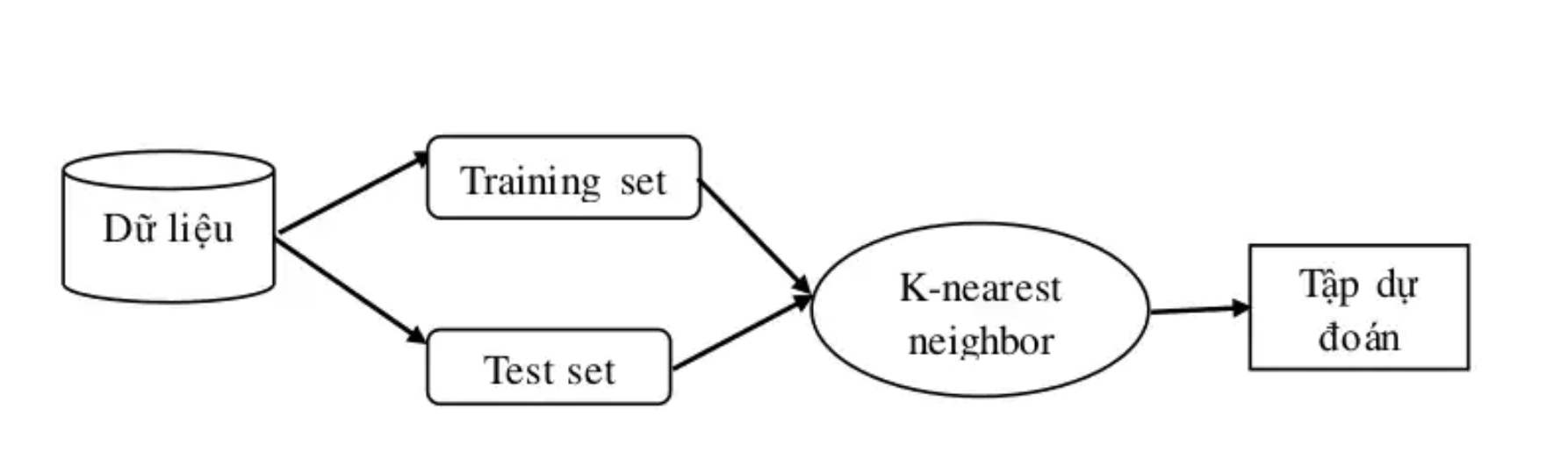
## 2.4. Bộ dữ liệu dành cho máy

* Dữ liệu thu thập được gồm 427 bản ghi (tương đương với dữ liệu thời tiết tại Hà Nội từ ngày 01/09/2020 đến 01/11/2021)
* 5 cột đầu tiên lần lượt là các thuộc tính: Nhiệt độ lớn nhất đo được trong ngày, nhiệt độ nhỏ nhất đo được trong ngày, tốc độ gió, mây che phủ, độ ẩm (không tính cột index).
* Cột cuối cùng: Conditions – Trạng thái thời tiết

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Index | Max Temperature (o C) | Min Temperature (oC) | Wind Speed (km/h) | Cloud Cover (%) | Relative Humidity (%) | Conditions |
| 1 | 35.9 | 25 | 12.9 | 11.6 | 67.11 | Rain |
| 2 | 36 | 27.5 | 12.1 | 11.8 | 68.15 | Rain |
| 3 | 35.5 | 28 | 11.2 | 21.7 | 73.05 | Rain |
| 4 | 35 | 28 | 19.4 | 27.7 | 71.29 | Rain, Partially cloudy |
| 5 | 36 | 28 | 15.7 | 22.1 | 69.52 | Clear |
| 6 | 35.7 | 26.5 | 25.9 | 37.2 | 72.42 | Rain, Partially cloudy |
| 7 | 28.1 | 25 | 12.1 | 53.2 | 85.11 | Rain, Partially cloudy |
| 8 | 31.2 | 25 | 14.8 | 49 | 81.74 | Rain, Partially cloudy |
| 9 | 31 | 26 | 19.4 | 67.6 | 80.83 | Rain, Partially cloudy |
| 10 | 33.7 | 26.5 | 14.8 | 55.4 | 80.11 | Rain, Partially cloudy |
| 11 | 33.2 | 26 | 14.3 | 30.1 | 79.21 | Rain, Partially cloudy |
| 12 | 33.5 | 25.5 | 14.8 | 45.1 | 82.51 | Rain, Partially cloudy |
| 13 | 32.7 | 25 | 18.4 | 55.3 | 81.14 | Rain, Partially cloudy |
| 14 | 33.5 | 26 | 15.8 | 47.6 | 79.8 | Rain, Partially cloudy |
| 15 | 33.5 | 26 | 18.4 | 41.4 | 78.08 | Rain, Partially cloudy |
| 16 | 34.5 | 27 | 23.2 | 32 | 76.82 | Rain, Partially cloudy |
| 17 | 34.5 | 27 | 15.7 | 32.5 | 72.4 | Partially cloudy |
| 18 | 29 | 26 | 23.2 | 55.3 | 67.26 | Rain, Partially cloudy |
| 19 | 25.5 | 24 | 22.3 | 75.6 | 81.14 | Rain, Overcast |
| 20 | 26.7 | 24 | 10.3 | 65.7 | 87.66 | Rain, Partially cloudy |
| 21 | 32.5 | 25 | 17.5 | 57 | 80.66 | Partially cloudy |
| 22 | 31.8 | 25 | 15.7 | 53.9 | 85.55 | Rain, Partially cloudy |
| 23 | 30.9 | 25 | 18.4 | 56.2 | 85.23 | Rain, Partially cloudy |
| 24 | 28.5 | 23 | 18.4 | 66.9 | 80.03 | Rain, Partially cloudy |
| 25 | 33 | 25 | 16.6 | 53.2 | 75.99 | Rain, Partially cloudy |
| 26 | 32.7 | 25 | 16.7 | 56.6 | 79.28 | Rain, Partially cloudy |
| 27 | 28.3 | 24 | 21.4 | 67 | 83.47 | Rain, Partially cloudy |
| 28 | 30 | 24 | 13 | 65.2 | 81.15 | Rain, Partially cloudy |
| 29 | 33 | 25.9 | 16.6 | 44.7 | 77.18 | Rain, Partially cloudy |
| 30 | 34.1 | 26 | 16.6 | 37.7 | 75.58 | Rain, Partially cloudy |
| 31 | 31 | 25 | 16.6 | 50.9 | 81.67 | Rain, Partially cloudy |
| 32 | 32 | 26 | 13.9 | 48 | 79.83 | Rain, Partially cloudy |
| 33 | 31 | 25.5 | 13.9 | 48.2 | 87.3 | Rain, Partially cloudy |
| 34 | 33 | 25 | 13.9 | 50.8 | 80.45 | Rain, Partially cloudy |
| 35 | 32.5 | 26 | 18.5 | 46.2 | 82.36 | Rain, Partially cloudy |
| 36 | 29.2 | 24 | 18.5 | 39 | 72.02 | Rain, Partially cloudy |
| 37 | 28.1 | 23 | 20.5 | 56.5 | 67.45 | Partially cloudy |
| 38 | 27.4 | 22 | 24.1 | 39 | 61.22 | Partially cloudy |
| 39 | 27.4 | 21 | 21.4 | 22.7 | 60.35 | Clear |
| 40 | 27.1 | 20 | 24.1 | 24.9 | 61.17 | Clear |
| 41 | 27.5 | 20.5 | 17.5 | 27.6 | 59.47 | Partially cloudy |
| 42 | 29.3 | 21.5 | 19.4 | 19.9 | 62.86 | Clear |
| 43 | 30.5 | 25 | 21.4 | 28.8 | 56.39 | Partially cloudy |
| 44 | 27 | 22.5 | 23.2 | 93.8 | 83.82 | Rain, Overcast |
| 45 | 25 | 22.7 | 19.4 | 93.2 | 85.94 | Rain, Overcast |
| 46 | 27 | 22.8 | 25 | 68 | 72.69 | Rain, Partially cloudy |
| 47 | 24.5 | 21 | 21.4 | 34 | 60.1 | Rain, Partially cloudy |
| 48 | 24 | 20.5 | 23.2 | 52.9 | 69.63 | Rain, Partially cloudy |
| 49 | 23 | 21 | 18.4 | 32 | 66.07 | Rain, Partially cloudy |
| 50 | 24.5 | 21.4 | 15.7 | 26.3 | 57.91 | Rain, Partially cloudy |
| 51 | 27.5 | 21.5 | 21.2 | 18.1 | 54.08 | Clear |
| 52 | 29 | 18 | 20.5 | 7.2 | 51.14 | Clear |
| 53 | 29 | 17 | 19.4 | 1.7 | 54.61 | Clear |
| 54 | 28.5 | 16.5 | 13 | 5.8 | 59.82 | Clear |
| 55 | 24.2 | 19 | 14.8 | 71.1 | 71.02 | Rain, Partially cloudy |
| 56 | 24 | 22 | 12.1 | 86.9 | 80.24 | Rain, Overcast |
| 57 | 29.8 | 22 | 12.1 | 34.2 | 80.55 | Rain, Partially cloudy |
| 58 | 30.5 | 22.5 | 20.3 | 46.9 | 74.48 | Rain, Partially cloudy |
| 59 | 24 | 22 | 23.2 | 86.3 | 76.04 | Rain, Overcast |
| 60 | 22 | 20 | 12.1 | 95.5 | 91.87 | Rain, Overcast |
| 61 | 24.5 | 20.7 | 9.4 | 72.6 | 86.18 | Rain, Partially cloudy |
| 62 | 28.4 | 21 | 11.2 | 58.2 | 81.98 | Partially cloudy |
| 63 | 28 | 22 | 19.4 | 39.8 | 78.98 | Rain, Partially cloudy |
| 64 | 24.7 | 21 | 15.7 | 36.4 | 59.71 | Rain, Partially cloudy |
| 65 | 24.8 | 20.5 | 15.7 | 31 | 59.81 | Partially cloudy |
| 66 | 26.7 | 19 | 10.3 | 33.7 | 70.67 | Partially cloudy |
| 67 | 26 | 22 | 10.1 | 25.4 | 74.95 | Partially cloudy |
| 68 | 28.2 | 22.1 | 20.3 | 20.8 | 61.08 | Clear |
| 69 | 29 | 21.5 | 17.6 | 5.6 | 53.88 | Clear |
| 70 | 28 | 20.3 | 19.4 | 0.1 | 49.5 | Clear |
| 71 | 27.2 | 17 | 17.5 | 13 | 48.97 | Clear |
| 72 | 27 | 18.5 | 12.1 | 23 | 51.92 | Clear |
| 73 | 28.3 | 19 | 13.9 | 5.2 | 52.17 | Clear |
| 74 | 29 | 18.5 | 16.6 | 1 | 51.91 | Clear |
| 75 | 28 | 15.5 | 18.5 | 11.8 | 52.47 | Clear |
| 76 | 19.5 | 18.7 | 18.5 | 96.2 | 90.15 | Rain, Overcast |
| 77 | 21 | 19 | 8.5 | 93.8 | 94.97 | Rain, Overcast |
| 78 | 27.1 | 20.7 | 18.5 | 77.7 | 89.35 | Rain, Overcast |
| 79 | 28.7 | 22.8 | 13.9 | 66.5 | 81.48 | Partially cloudy |
| 80 | 28 | 22.8 | 12.1 | 68.7 | 83.51 | Rain, Partially cloudy |
| 81 | 29 | 23 | 16.6 | 72.4 | 81.42 | Partially cloudy |
| 82 | 30 | 24 | 19.4 | 66.8 | 81.35 | Partially cloudy |
| 83 | 29 | 23 | 14.8 | 63.2 | 81.67 | Partially cloudy |
| 84 | 29 | 23.5 | 16.6 | 52.8 | 76.66 | Partially cloudy |
| 85 | 26.5 | 23 | 15.7 | 68.8 | 75.58 | Partially cloudy |
| 86 | 27 | 22 | 13.9 | 53.4 | 76.99 | Partially cloudy |
| 87 | 28.5 | 22 | 12.1 | 53.4 | 73.49 | Partially cloudy |
| 88 | 27 | 19 | 19.4 | 35.3 | 71.01 | Partially cloudy |
| 89 | 22.5 | 18 | 20.5 | 64.5 | 65.73 | Partially cloudy |
| 90 | 22.1 | 17 | 21.4 | 22 | 52.63 | Clear |
| 91 | 23.2 | 16 | 22.3 | 14.6 | 48.87 | Clear |
| 92 | 21 | 17.5 | 18.4 | 36.8 | 52.27 | Partially cloudy |
| 93 | 21 | 16 | 14.8 | 62.4 | 64.18 | Partially cloudy |
| 94 | 22 | 18 | 18.4 | 33.9 | 55.18 | Partially cloudy |
| 95 | 20.3 | 14.5 | 18.5 | 43.8 | 54.88 | Partially cloudy |
| 96 | 20 | 13 | 20.5 | 21.3 | 57.21 | Clear |
| 97 | 22 | 11 | 10.3 | 0.1 | 64.98 | Clear |
| 98 | 23 | 15.1 | 15.7 | 41.6 | 63.4 | Partially cloudy |
| 99 | 22 | 15.5 | 16.6 | 8.8 | 63.72 | Clear |
| 100 | 23.2 | 13 | 13 | 25.6 | 69.2 | Partially cloudy |

Hình 8: Bảng dữ liệu 100 bản ghi đầu

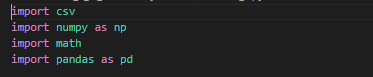
## 2.5. Cài đặt thuật toán

******

Hình 9: Mô hình thuật toán KNN áp dụng cho bài toán

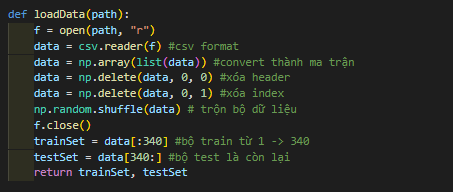
Trong phần này, ta sẽ tách bộ dữ liệu thành hai phần, gọi là training test và test set. Thuật toán sẽ dựa vào thông tin ở training test để dự đoán thời tiết trong ngày của mỗi dữ liệu trong test set. Dữ liệu được chuẩn hóa sẽ đối chiếu với dự đoán thời tiết thật của mỗi dữ liệu trong test set để đánh giá hiệu quả, độ chính xác của KNN.

**Bước 1**: Khai báo các thư viện cần thiết



Hình 10: Khai báo các thư viện cho chương trình

**Bước 2**: Tạo hàm đọc dữ liệu vào từ file csv



Hình 11: Đoạn mã lấy dữ liệu từ file csv

Dùng module csv để định dạng dữ liệu đọc vào, sau đó chuyển qua ma trận bằng numpy để dễ dàng xử lí.

Một số thao tác tiền xử lí gồm: Xóa đi hàng đầu tiên chứa tiêu đề, xóa đi cột Index, sau đó chúng ta sẽ sử dụng phương thức shuffle của numpy.random để trộn dữ liệu. Lí do là để sau khi trộn chúng ta sẽ lấy 340 hàng đầu làm dữ liệu train và 87 hàng cuối để làm dữ liệu test.

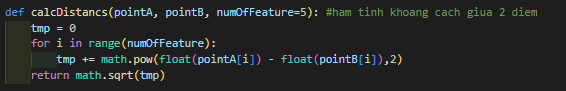
Hàm này dữ liệu trả về là bộ trainSet và testSet , ví dụ trainSet lấy 5 dòng đầu và testSet lấy 5 dòng tiếp từ bộ dữ liệu

Text

Description automatically generated

Hình 12: Minh hoạ kết quả trả về từ bộ trainSet và testSet

**Bước 3**: Xây dựng hàm tính khoảng cách



Hình 13: Xây dựng hàm tính khoảng cách

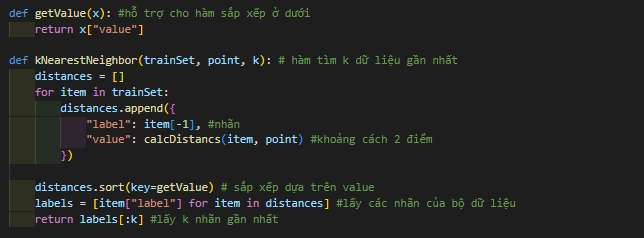
Hàm thực hiện việc tính toán dữ liệu 2 điểm truyền vào bằng công thức Euclidean. Đơn giản là duyệt qua tất cả các thuộc tính tương ứng mỗi điểm, tính tổng của hiệu bình phương mỗi thuộc tính, cuối cùng là trả về căn bậc 2 của tổng đó.  
 Ví dụ hàm trả về khoảng cách của 2 dòng dữ liệu:

A black screen with white text

Description automatically generated with low confidence

Hình 14: Minh hoạ khoảng cách tính được giữa 2 bản ghi dữ liệu

**Bước 4**: Xây dựng hàm tìm k điểm dữ liệu gần nhất



Hình 15: Xây dựng hàm tìm ra các nhãn có khoảng cách nhỏ nhất

Hàm này sẽ duyệt qua tất cả các giá trị trong trainSet, tính khoảng cách giữa điểm truyền vào với những điểm trong tập dữ liệu ban đầu. Kết quả của vòng lặp này là một list các dictionary gồm tên nhãn (dự đoán thời tiết) và khoảng cách đến điểm đó. Tiếp theo chúng ta sẽ sắp xếp tăng dần list này với giá trị so sánh là khoảng cách. Vì kết quả chúng ta chỉ cần biết là thời tiết trong ngày nên chúng ta sẽ thêm 1 vòng lặp để tạo một list các nhãn có cùng thứ tự. Cuối cùng là trả về k điểm dữ liệu đầu tiên của list (nhỏ nhất)

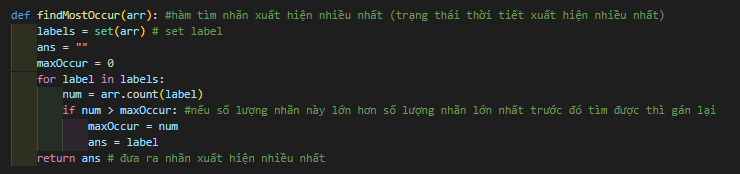
Mình hoạ hàm sắp xếp, tìm 5 nhãn có khoảng cách đến điểm đó là nhỏ nhất rồi trả về 5 nhãn đó:





Hình 16: Kết quả trả về k(5) nhãn có khoảng cách nhỏ nhất

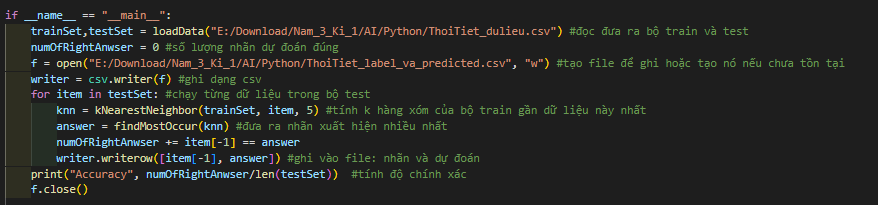
**Bước 5**: Xây dựng hàm tìm trạng thái thời tiết xuất hiện nhiều nhất trong k trạng thái thời tiết tìm được



Hình 17: Xây dựng hàm tìm trạng thái thời tiết xuất hiện nhiều nhất trong k trạng thái thời tiết tìm được

Hàm này sẽ duyệt qua từng nhãn có trong list labels - tập hợp các nhãn để tìm ra nhãn xuất hiện nhiều nhất.

**Bước 6**: Duyệt qua các giá trị trong bộ dữ liệu test để kiểm tra độ chính xác của thuật toán



Hình 18: Hàm main của chương trình

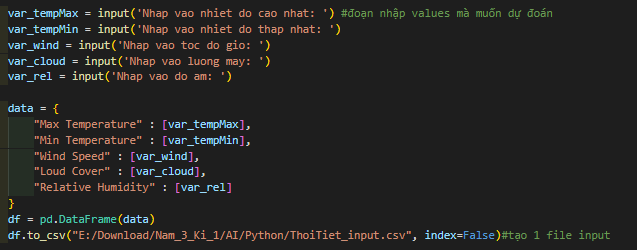
Để đánh giá độ chính xác của thuật toán KNN, chúng ta sẽ xem có bao nhiêu điểm trong dữ liệu test được dự đoán đúng. Lấy số lượng đó chia cho tổng số lượng trong tập dữ liệu test sẽ ra độ chính xác.



Hình 19: Ví dụ kết quả trả về độ chính xác

Ta có thể thấy độ chính xác được phân tích từ dữ liệu trong bộ testSet đạt ~0.71 tương đương 71% tổng số bản test

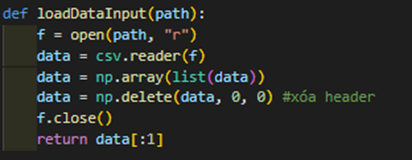
**Bước 7**: Nhập dữ liệu từ bàn phím



Hình 20: Nhập dữ liệu từ bàn phím

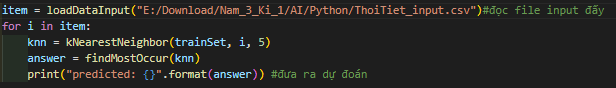
Nhập dữ liệu thời tiết vào từ bàn phím, sử dụng thư viện pandas để tạo file csv chứa dữ liệu vừa nhập vào.

**Bước 8**: Tạo hàm đọc dữ liệu từ file csv vừa tạo ở bước 6



Hình 21:Tạo hàm đọc dữ liệu từ file csv

**Bước 9:** Đưa ra dự đoán của bộ dữ liệu vừa nhập





Hình 22: Đưa ra dự đoán của bộ dữ liệu vừa nhập

## 2.6. Kết quả

Lấy ví dụ k = 11

Chương trình tính toán và sinh ra file predict so sánh labels và predicted trên bộ testSet(87/428):

Text

Description automatically generated

Hình 23: So sánh labels và predicted trên bộ testSet

Từ đó đưa ra độ chính xác của thuật toán trên bộ testSet(87/428):



Chương trình sinh ra file input mà ta đã nhập dữ liệu:

Ví dụ nhập:

Nhiệt độ cao nhất :31

Nhiệt độ thấp nhất :26

Tốc độ gió :17

Mật độ mây che phủ: 20

Độ ẩm:74

A picture containing text

Description automatically generated

Hình 24: Ví dụ dữ liệu đã nhập

Cuối cùng tính toán và đưa ra dự đoán dựa trên bộ trainSet(340/428):

Text

Description automatically generated

Hình 25: Minh hoạ kết quả bài toán

Với số liệu input như trên chương trình dự đoán là thời tiết trong lành, quang đãng

# Kết luận

# 1. Kết quả đạt được

- Sự hiểu biết về thuật toán KNN cơ bản tương đối tốt.

- Hiểu biết thêm về AI, ứng dụng của của Machine Learning vào đời sống công nghệ hiện đại.

- Làm quen ngôn ngữ lập trình Python

- Áp dụng được kiến thức đã học vào bài toán thực tiễn (dự báo thời tiết)

# 2. Hạn chế

- Thuật toán KNN còn phụ thuộc nhiều vào hệ số K.

- Kết quả đưa ra có sự thay đổi (vì các điểm xét lấy ngẫu nhiên).

# Tài liệu tham khảo

[1]. Giáo trình trí tuệ nhân tạo - Trường Đại học Công nghiệp Hà Nội.

[2]. [Machine learning là gì? Tổng quan về machine learning (nguyenvanhieu.vn)](https://nguyenvanhieu.vn/machine-learning-la-gi/#:~:text=Machine%20learning%20l%C3%A0%20g%C3%AC%3F%20Machine%20learning%20l%C3%A0%20m%E1%BB%99t,t%C3%ADnh%20h%E1%BB%8Dc%20%C4%91%E1%BB%83%20l%C3%A0m%20c%C3%B4ng%20vi%E1%BB%87c%20g%C3%AC%20%C4%91%C3%B3%21).

[3]. [Thuật Toán K-Nearest Neighbors (KNN) Siêu Cơ Bản (codelearn.io)](https://codelearn.io/sharing/thuat-toan-k-nearest-neighbors-knn).

[4]. Tài liệu điện tử trên internet.