**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**---o0o---**



**ĐỒ ÁN 3**

**TÌM HIỂU VỀ THUẬT TOÁN K-NEAREST-NEIGHBOR**

**Nhóm sinh viên thực hiện**

17110220 Nguyễn Văn Tây

17110217 Trần Nguyên Tài

**GVHD: TS.Lê Văn Vinh**

**Tp. Hồ Chí Minh, tháng 12 năm 2020**

**LỜI CAM ĐOAN**

Nhóm xin cam đoan. Đồ án 3 với đề tài “Tìm hiểu thuật toán K-nearest neighbor” là kết quả tìm hiểu, nghiên cứu của nhóm từ những kiến thức được thầy cô giảng dạy truyền đạt trong thời gian qua và một số tài liệu được nhóm chúng em tìm hiểu sưu tập qua các phương tiện internet.

**MỤC LỤC**

[**DANH MỤC HÌNH ẢNH** 4](#_Toc63019287)

[**DANH MỤC BẢNG** 5](#_Toc63019288)

[**LỜI CẢM ƠN** 6](#_Toc63019289)

[**MỞ ĐẦU** 7](#_Toc63019290)

[**1.1.** **Đặt vấn đề.** 7](#_Toc63019291)

[**1.2.** **Mục đích , phạm vi và đối tượng nghiên cứu** 8](#_Toc63019292)

[**1.3.** **Nội dung thực hiện** 8](#_Toc63019293)

[**1.** **CHƯƠNG 1 : CƠ SỞ LÝ THUYẾT** 9](#_Toc63019294)

[**1.1.** **Machine Learning.** 9](#_Toc63019295)

[**1.1.1.** **Định nghĩa** 9](#_Toc63019296)

[**1.1.2.** **Một số phương thức của Machine Learning** 9](#_Toc63019297)

[**1.2.** **Bài toán phân lớp dữ liệu** 11](#_Toc63019298)

[**1.2.1.** **Quá trình phân lớp dữ liệu** 11](#_Toc63019299)

[**2.** **CHƯƠNG 2 : THUẬT TOÁN K-NEREST NEIGHBOR** 13](#_Toc63019300)

[**2.1.** **K-nearest Neighbours Regression** 13](#_Toc63019301)

[**2.1.1.** **Giới thiệu** 13](#_Toc63019302)

[**2.2.** **Thuật toán k-nearest neighbor** 14](#_Toc63019303)

[**2.2.1.** **Định nghĩa** 14](#_Toc63019304)

[**2.2.2.** **Quy trình làm việc của thuật toán K – Nearest Neighbor** 14](#_Toc63019305)

[**2.2.3.** **Ví dụ minh họa** 14](#_Toc63019306)

[**2.2.4.**  **Ví dụ về K – Nearest Neighbor nhiễu** 16](#_Toc63019307)

[**2.2.5.** **Ưu điểm, nhược điểm của thuật toán** 16](#_Toc63019308)

[**2.3.** **Khoảng cách trong không gian vector** 16](#_Toc63019309)

[**2.3.1.** **Định nghĩa** 17](#_Toc63019310)

[**2.3.2.** **Một số norm thường dùng** 17](#_Toc63019311)

[**3.** **CHƯƠNG 3 : THỬ NGHIỆM** 18](#_Toc63019312)

[**3.1.** **Bộ dữ liệu Iris flower dataset** 18](#_Toc63019313)

[**3.1.1.** **Giới thiệu** 18](#_Toc63019314)

[**3.1.2.** **Sử dụng tập dữ liệu** 19](#_Toc63019315)

[**3.1.3.** **Tập dữ liệu** 20](#_Toc63019316)

[**4.** **CHƯƠNG 4 : KẾT LUẬN** 26](#_Toc63019317)

[**TÀI LIỆU THAM KHẢO** 27](#_Toc63019318)

# **DANH MỤC HÌNH ẢNH**

[Hình 1 mối quan hệ giữa deep learning và AI 8](#_Toc63019277)

[Hình 2 vi dụ về phân lớp 10](file:///C:\Users\nguye\Desktop\Project3\BAOCAO_DOAN3.docx#_Toc63019278)

[Hình 3 Đồ thị minh họa 10](#_Toc63019279)

[Hình 4 thước đo khoảng cách 13](#_Toc63019280)

[Hình 5 khoảng cách hamming 14](#_Toc63019281)

[Hình 6 Ví dụ 15](#_Toc63019282)

[Hình 7 K – Nearest Neighbor nhiễu 16](file:///C:\Users\nguye\Desktop\Project3\BAOCAO_DOAN3.docx#_Toc63019283)

[Hình 8 khoản cách 2 vector 18](#_Toc63019284)

[Hình 9 dữ liệu các loài hoa 19](#_Toc63019285)

[Hình 10 Mô hình dataset 20](#_Toc63019286)

# **DANH MỤC BẢNG**

[Bảng 1 dữ liệu hoa setosa 22](#_Toc63019267)

[Bảng 2 dữ liệu hoa Versicolor 24](#_Toc63019268)

[Bảng 3 Dữ liệu hoa virginica 26](#_Toc63019269)

**LỜI CẢM ƠN**

Lời đầu tiên cho phép nhóm em gởi lời cảm ơn sâu sắc đến thầy Lê Văn Vinh, người đã hướng dẫn truyền đạt và chỉ dẫn cho nhóm em nhứng kiến thức, bài học và kinh nghiệm kể từ khi bắt đầu nhận đề tài.

Để hoàn thành đồ án này nhóm em cũng xin gởi lời cảm ơn đến những người bạn đã luôn hỗ trợ chia sẻ những kiến thức luôn ở bên động viên khuyến khích để nhóm em có thể hoàn thành được đồ án này

Sau nữa nhóm xin cảm ơn đến bạn bè vì luôn bên cạnh giúp đỡ, động viên trong suốt thời gian thực hiện đề tài để nhóm em có thể hoàn thành tốt đề tài của mình

Trong quá trình nghiên cứu và báo cáo do năng lực, kiến thức, trình độ bản thân còn hạn hẹp nên không tránh những thiếu sót. Nhóm em mong nhận được sự thông cảm và những ý kiến đóng góp của thầy cô và các bạn

Nhóm xin chân thành cảm ơn!

TP.HCM, ngày 30 tháng 12 năm 2020

Sinh viên thực hiện

Nguyễn Văn Tây

Trần Nguyên Tài

**MỞ ĐẦU**

* 1. **Đặt vấn đề.**

Những năm gần đây , trí tuệ nhân tạo nổi lên như một bằng chứng cho cuộc cách mạng công nghệ 4.0. Trí tuệ nhân tạo có thể định nghĩa như một ngành của khoa học máy tính liên quan đến việc tự động hóa các hành vi thông minh. Trí tuệ nhân tạo là một bộ phận khoa học máy tính và do đó nó phải được đặt trên những nguyên lý lý thuyết vững chắc, có khả năng ứng dụng cao.

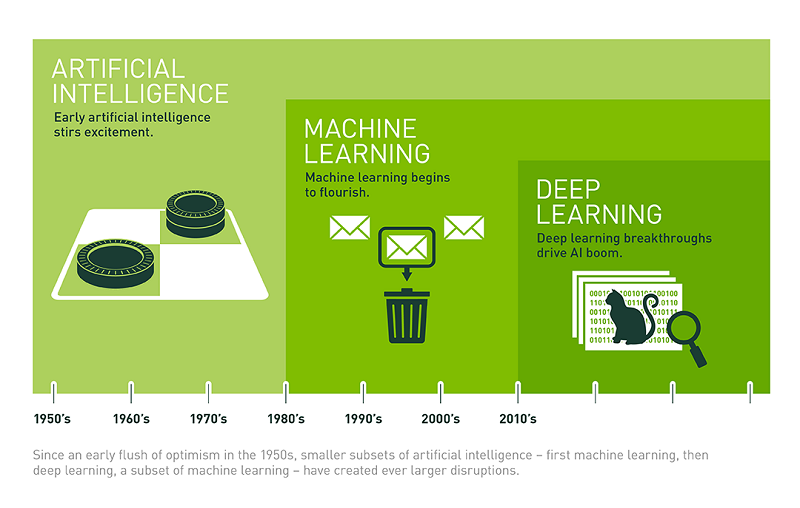
Ở thời điểm hiện tại thuật ngữ này thường được dùng để nói đến các máy tính có mục đích không nhất định và ngành khoa học nghiên cứu về các lý thuyết và các ứng dụng của trí tuệ nhân tạo.

Theo đà phát triển của công nghệ, ứng dụng trí tuệ nhân tạo luôn là xu hướng công nghệ tương lai và các hãng công nghệ trên toàn thế giới đua nhau sáng tạo. Nó là nền tảng cốt lõi của cuộc cách mạng 4.0

Machine Learning là một lĩnh vực của trí tuệ nhân tạo, được sinh ra từ khả năng nhận diện mẫu và từ các lý thuyết các máy tính có thể học mà không cần phải lập trình để xử lý nhiệm vụ cụ thể nào đó.

Hầu hết mọi ngành công nghiệp đang làm việc với lượng dữ liệu lớn đều nhận ra tầm quan trọng của công nghệ Machine Learning. Những cái nhìn sáng suốt từ nguồn dữ liệu này chủ yếu dưới dạng thời gian thực, sẽ giúp các tổ chức vận hành hiệu quả hoặc tạo lợi thế cạch tranh với các đối thủ khác.

Các ứng dụng của Machine Learning đã quá quen thuộc với con người như: Xe tự hành của Google , hệ thông tự tag khuôn mặt trên facebook,hệ thống gợi ý sản phầm của amazone,…. Chỉ là một trong vô vàn những ứng dụng của trí tuệ nhân tạo mang lại



Hình 1 mối quan hệ giữa deep learning và AI

Xu hướng phát triển công nghệ thông tin ngày càng tăng, song song với nó lượng dữ liệu được sinh ra cũng ngày một lớn. vì vậy nhu cầu để xử lý dữ liệu cũng lớn hơn, Machine learning góp phần giải quyết vấn đề này. Một trong những thuật toán thường dùng đó là thuật toán K-nearest neighbor. Ứng dụng của thuật toán này được sử dụng rộng rãi trong các bài toán phân lớp.

* 1. **Mục đích , phạm vi và đối tượng nghiên cứu**
* Mục đích : Nghiên cứu, tìm hiểu thuật toán K-nearest neighbor, đánh giá hiệu quả của thuật toán
* Phạm vi nghiên cứu: Thử nghiệm trên Iris flower dataset
* Đối tượng nghiên cứu: thuật toán K-nearest neighbor và bộ Iris flower dataset.
  1. **Nội dung thực hiện**
* Tìm hiểu thuật toán
* Làm quen bộ dữ liệu Iris Flower dataset
* Sử dụng bộ dữ liệu thử nghiệm và đánh giá

1. **CHƯƠNG 1 : CƠ SỞ LÝ THUYẾT**
   1. **Machine Learning.**
      1. **Định nghĩa**

Là một lĩnh vực của trí tuệ nhân tạo liên quan dến việc nghiên cứu và xây dựng các kỹ thuật cho phép các hệ thống học tự động từ dữ liệu để giải quyết tất cả các vấn đề cụ thể. Ví dụ các máy có thể học cách phân loại thư điện tử có phải thư rác hay không và tự động sắp xếp vào thư mục tương ứng

Machine Learning có liên quan đến thống kê vì cả hai lĩnh vực đều nghiên cứu việc phân tích dữ liệu, nhưng khác với thống kê học máy tập trung vào sự phức tạp của các giải thuật trong việc thực thi tính toán

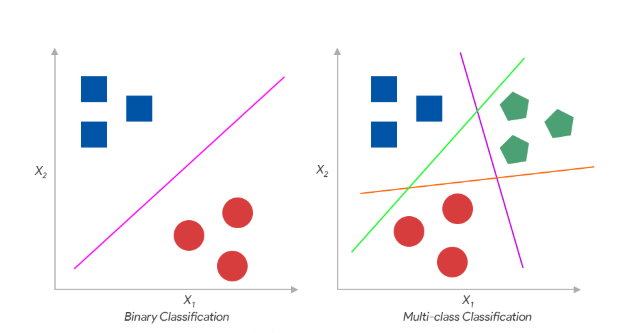
Machine Learning hiện nay áp dụng rộng rãi bao gồm máy truy tìm dữ liệu, máy phân tích thị trường chứng khoán, nhận dạng tiếng nói và chữ viết…

* + 1. **Một số phương thức của Machine Learning**

**Học có giám sát**: Thuật toán dự đoán đầu ra của một dữ liệu mới (new input) dựa trên cặp (input, outcome) đã biết từ trước. Cặp dữ liệu này còn gọi là ( data,label), tức (dữ liệu, nhãn). Supervised learning là nhóm phổ biến nhất trong các thuật toán Machine Learning.

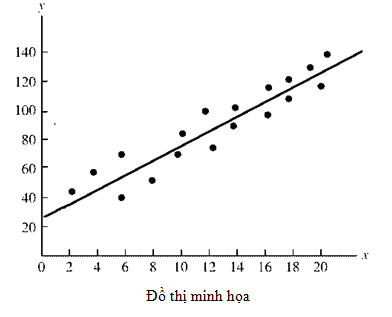
Học có giám sát chia làm hai loại:

* Classification (phân lớp) : Là quá trình phân lớp một đối tượng dữ liệu vào hay nhiều lớp như đã cho trước nhờ một mô hình phân lớp (model). Mô hình này được xây dựng dựa trên tập dữ liệu được xây dựng trước đó có gán nhãn ( tập huấn luyện). Quá trình phân lớp là quá trình gắn nhãn cho đối tượng dữ liệu. Có nhiều nhiều bài toán phân lớp như phân lớp nhị phân , phân lớp đa lớp,phân lớp đa trị… Trong số đó phân lớp nhị phân là một loại phân lớp đặc biệt của phân lớp đa lớp. Ứng dụng của bài toán phân lớp được ứng dụng rất nhiều như nhận dạng khuôn mặt , nhận dạng chữ viết, nhận dạng giọng nói….



Hình 2 vi dụ về phân lớp

* Regression (hồi quy) : Nếu không được chia thành các nhóm mà là giá trị thực cụ thể. Đầu ra của diểm dữ liệu sẽ bằng chính đầu ra của điểm dữ liệu đã biết



Hình 3 Đồ thị minh họa

**Học không giám sát :**  Là một kĩ thuật của máy học nhằm tìm ra một mô hình hay cấu trúc bị ẩn bởi tập dữ liệu không được gán nhãn cho trước. UL khác với SL là không thể xác đị nh trước output từ tập dữ liệu huấn luyện được. Tùy thuộc vào tập huấn luyện kết quả output sẽ khác nhau. Trái ngược với SL, tập dữ liệu huấn luyện của UL không do con người gán nhãn, máy tính sẽ phải tự học hoàn toàn. Có thể nói, học không giám sát thì giá trị đầu ra sẽ phụ thuộc vào thuật toán UL. Ứng dụng lớn phổ biến của học không giám sát là bài toán phân cụm.

**Học bán bám sát**: các bài toán khi có một số lượng lớn dữ liệu lớn nhưng chỉ một phần trong chúng được dán nhãn. Nhưng bài toán này nằm giữa phương thức học giám sát và không giám sát.

* 1. **Bài toán phân lớp dữ liệu**
     1. **Quá trình phân lớp dữ liệu**

Để xây dựng được mô hình phân lớp và đánh giá hiệu quả của mô hình cần phải thực hiện các bước sau:

* + - 1. Chuẩn bị tập dữ liệu huấn luyện và rút trích đặc trưng

Công đoạn này được xem là công đoạn quan trọng nhất của bài toán về machine learning. Vì đây là input cho việc để tìm ra mô hình của bai toán. Chúng ta cần biết phải chọn ra những đặc trung của tập dữ liệu, lược bỏ những đặc trưng không tốt của dữ liệu, gây nhiễu. Ước lượng số chiều của dữ liệu hay nói cách khác là chọn feature. Nếu số lượng dữ liệu nhiễu quá lớn sẽ gây khó khăn cho việc tính toán thì phải giảm số chiều của dữ liệu nhưng vẫn được độ chính xác của dữ liệu đó.

Ở bước này chũng ta cũng chuẩn bị bộ dữ liệu test trên mô hình. Thông thường sẽ sử dụng cross-validation (kiểm tra chéo) để chia tập dataset thành hai phàn, một phần phục vụ cho training và phần còn lại phục vụ cho mục đích testing trên mô hình. Có hai cách thường sử dụng trong cross­validation là splitting và k­fold

* + - 1. Xây dựng mô hình phân lớp

Mục đích của mô hình huấn luyện là tìm ra F(x) và thông qua hàm f tìm được để gán nhãn cho dữ liệu. Bước này thường được gọi là tranning.

F(x) = y.

Trong đó : x là các feature hay input đầu vào của dữ liệu

y là nhãn dán lớp hay output đầu ra.

Thông thường để xây dựng mô hình phân lớp cho bài toán này chúng ta sử dụng các thuật toán học giám sát như: K – nearest neighbor, Support Vector Machine, Decision tree, Navie Bayers.

* + - 1. Kiểm tra dữ liệu với mô hình

Sau khi tìm được mô hình phân lớp ở bước xây dựng mô hình thì bước này ta sẽ đưa dữ liệu mới vào để kiểm tra trên mô hình phân lơp.

* + - 1. Đánh giá mô hình phân lớp và chọn mô hình tốt nhất

Bước cuối cùng chúng ta sẽ dánh giá mức độ lỗi của dữ liệu testing và dữ liệu training thông qua mô hình tìm được. Nếu không đạt được kết quả mong muốn chúng ta phải thay đổi các tham số của thuật toán để tìm ra các mô hình tốt hơn và kiểm tra, đánh giá lại mô hình phân lớp. Cuối cùng chọn ra mô hình phân lớp tốt nhất cho bài toán của chúng ta.

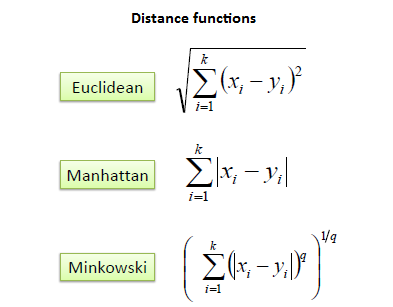
1. **CHƯƠNG 2 : THUẬT TOÁN K-NEREST NEIGHBOR**
   1. **K-nearest Neighbours Regression**
      1. **Giới thiệu**

Hồi quy KNN là một phương pháp phi tham số, theo cách trực quan, xấp xỉ mối quan hệ giữa các biến độc lập và kết quả liên tục bằng cách lấy trung bình các quan sát trong cùng một vùng lân cận . Kích thước của vùng lân cận cần được thiết lập bởi nhà phân tích hoặc có thể được chọn bằng cách sử dụng xác nhận chéo (chúng ta sẽ xem điều này sau) để chọn kích thước giảm thiểu sai số trung bình.

Mặc dù phương pháp này khá hấp dẫn, nhưng nó nhanh chóng trở nên không thực tế khi thứ nguyên tăng lên, tức là khi có nhiều biến độc lập.

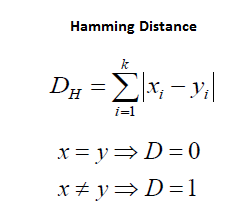
* + 1. **Thuật toán**

Một cách thực hiện đơn giản của hồi quy KNN là tính giá trị trung bình của mục tiêu số của K lân cận gần nhất. Một cách tiếp cận khác sử dụng trung bình có trọng số khoảng cách nghịch đảo của K láng giềng gần nhất. Hồi quy KNN sử dụng các hàm khoảng cách giống như phân loại KNN.



Hình 4 thước đo khoảng cách

Ba thước đo khoảng cách trên chỉ có giá trị đối với các biến liên tục. Trong trường hợp biến phân loại, bạn phải sử dụng khoảng cách Hamming, là thước đo số lượng các trường hợp mà các ký hiệu tương ứng khác nhau trong hai chuỗi có độ dài bằng nhau.



Hình 5 khoảng cách hamming

* 1. **Thuật toán k-nearest neighbor**
     1. **Định nghĩa**

K-nearest neighbor là một trong những thuật toán học có giám sát đơn giản nhất trong Machine Learning . Ý tưởng của K- nearest Neighbor là tìm ra output của kiểu dữ liệu dựa trên thông tin của những dữ liệu training gần nó nhất.

* + 1. **Quy trình làm việc của thuật toán K – Nearest Neighbor**

Bước 1 : Xác định tham số k = số láng giềng gần nhất

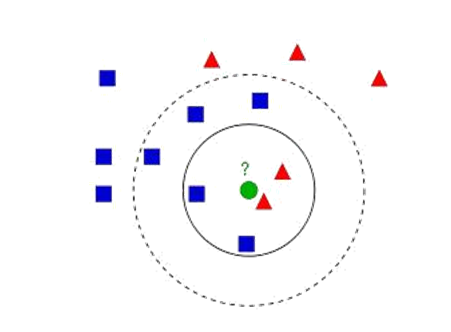
Bước 2: Tính khoảng cách đối tượng cần phân lớp với tất cả các đối tượng trong training data.

Bước 3: Sắp xếp khoảng cách theo thứ tự tăng dần và xác định k láng giềng gần nhất với đối tượng cần phân lớp.

Bước 4: Lấy tất cả các lớp P của K láng giềng gần nhất.

Bước 5: Dựa vào phần lớn lớp của K để xác định lớp cho đối tượng cần phân lớp.

* + 1. **Ví dụ minh họa**



Hình 6 Ví dụ

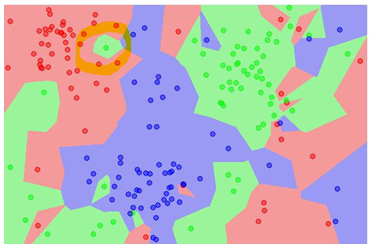
Giả sử bài toán được đặt ra: Mình mới quen một người bạn, tuy nhiên mình là fan của Rap việt, vậy nên mình cần biết người bạn mình là fan của king of rap hay là fan của rap việt. qua thời gian tìm hiểu mình đã thu thập được một số dữ liệu và đã biểu hiện dưới dạng hình vẽ trên

Ta dễ dàng nhận thấy có hai loại hình: hình vuông màu xanh biểu diễn cho những người là fan của king of rap, tam giác màu đỏ biểu diễn cho những người không là fan của King of rap, hình tròn mùa xanh là người bạn của mình mà mình muốn biết có phải là fan của king of rap hay không, khoảng cách giữa chấm tròn màu xanh và các hình còn lại biểu thị cho độ thân thiết của bạn đó và những ngươi khác.

Phương pháp đơn giản nhất là kiểm tra xem bạn đó có chơi thân với người bạn nào nhất, tức là tìm xem chấm xanh thuộc class nào (hình vuông hay là tam giác). Từ hình trên ta dễ dàng nhận thấy điểm gần chấm xanh nhất là hình tam giác màu đỏ, do đó sẽ được phân vào lớp tam giác màu đỏ.

Có một vấn đề ở phương pháp trên là xung quanh chấm xanh xuất hiện rất nhiều hình vuông màu xanh nên việc xét điểm gần nhất là chưa khả thi. Vì vậy ta sẽ xét K điểm gần nhất . Giả sử K = 3 , dựa theo hình ta dễ dàng nhận ra có hai hình tam giác đỏ và một hình vuông xanh có khoảng cách chấm xanh nhất, do đó châm xanh được phân vào lớp tam giác đỏ. Lấy K= 7 ta có năm hình vuông xanh và hai tam giác đỏ, lúc này chấp xanh được xếp vào class hình vuông xanh. Trường hợp K=4 ta thấy sẽ có hai hình vuông xanh và hai hình tam giác đỏ gần với chấm xanh nhất.

Trường hợp này K-Nearest Neighbor sẽ xử lý bằng cách so sánh tổng khoảng cách của hình gần nhất với điểm đang xét. Do xuất hiện điểm bằng nhau , vì vậy người ta thường dùng k số lẻ để xét, đó cũng là ý tưởng của K- Nearest Neighbor.

* + 1.  **Ví dụ về K – Nearest Neighbor nhiễu**

Hình 7 K – Nearest Neighbor nhiễu

Hình trên là bài toán phân lpows với ba lớp: đỏ , lục, lam. Mỗi điểm dữ liệu mới sẽ được gán nhãn dựa theo màu của điểm đó thuộc về. Trong hình này, Vùng khoanh tròn mày vàng, ta nhận thấy rằng điểm mày lục nằm giữa gai vùng lớn với nhiều dữ liệu đỏ và lam , điểm này rất có thể là nhiễu dẫn đến việc dữ liệu test nếu rơi vào vùng này sẽ có khả năng sai lệch cao

* + 1. **Ưu điểm, nhược điểm của thuật toán**

**Ưu điểm :**

Dễ sử dụng và cài đặt

Việc dự đoán kết quả dữ liệu mới dễ dàng

Độ phức tạp tính toán nhỏ.

**Nhược điểm:**

K- Nearest Neighbor nhiễu sẽ đưa kết quả không chính xác

Cần thời gian lưu training set, khi dữ liệu training và test tăng lên nhiều mất nhiều thời gian tính toán.

* 1. **Khoảng cách trong không gian vector**

Trong không gian một chiều, việc đo khoả ng cách giữa hai điểm đã rất quen thuộc: lấy trị tuyệt đối của hiệu giữa hai giá trị đó. Trong không gian hai chiều, tức mặt phẳng, chúng ta thường dùng khoảng cách Euclid để đo khoảng cách giữa hai điểm

Việc đo khoảng cách giữa hia điểm dữ liệu nhiều chiều, tức hai vector, là rất cần thiết trong Machine Learning. Chúng ta cần đánh giá xem điểm nào là điểm gần nhất của một điểm khác; chúng ta cũng cần đánh giá xem độ chính xác của việc ước lượng và trong rất nhiều ví dụ khác . Đó là lý do mà khái niệm norm ra đơi. Có nhiều loại norm khác nhau

* + 1. **Định nghĩa**

Một số hàm f() ánh xạ một điểm x từ không gian n chiều sang tập số thực một chiều được gọi là norm nếu nó thỏa mãn ba điểu kiện sau:

F(x) >= 0 . dấu bằng chỉ xảy ra khi x=0

F(a,x) = |a|f(x),

F() +f() >= f(+ ),

* + 1. **Một số norm thường dùng**

Giả sử các vector X = [x1; x2; …..; xn], y = [y1; y2;….; yn]

Nhân thấy khoản cách Euclid chính là một norm, norm này thường gòi là norm 2:

||X||2 = (1)

Với p là một số nhỏ hơn 1 bất kỳ, hàm số sau đây:

||X||p =(|x1|p +|x­2|p +…+| xn |p)1/p  (2)

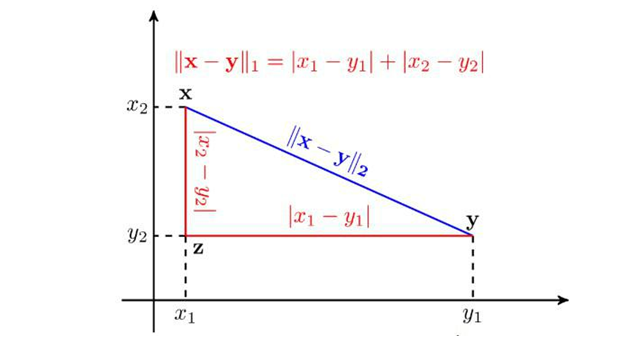
Được chứng minh thỏa mản điều kiện trên , và được gọi là norm p. nhận thấy rằng khi p 🡪0 thì biểu thức bên trên trờ thành phần tử khác 0 của x. hàm số (2) khi p=0 được gọi là giá trị chuẩn (Pseudo -norm) . Nó không phải là norm vù nó không thỏa mãn điều kiện 2 và 3 của norm. giá chuẩn này thường được ký hiệu là ||X||0, khá quan trọng trong machine learning vì nhiều bài toán ta cần “sparse” , một số lượng của x nhỏ, một vài giá trị thường được dùng

Khi P = 2 thường có norm 2

Khi P = 1 chú ta có

||x||1 = |x1| + |x2| +|x3|+ …+|xn| (3)

Là tổng các giá trị tuyệt đối của từng phần tử của x. norm 1 thường được dùng như sấp xỉ norm 0 trong các bài toán có ràng buộc. Dưới đây là một ví dụ so sánh norm 1 và norm 2 trong không gian 2 chiều:



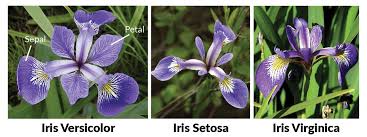
Hình 8 khoản cách 2 vector

Norm 2 (màu xanh) chính là đường chim bay nối giữ vector x và vector y. khoảng cách norm 1 giữa hai điểm (màu đỏ) có thể diễn giải như là đường từ x đến y trong một thành phố mà thành phố được tạo hình bàn cờ, chúng ta chỉ có thể đi dọc theo bàng cờ chứ không thể đi theo đường thẳng.

Khi p 🡪 vô cùng , ta có norm p chính là giá trị tuyệt đối của phần tử lớn nhất vủa vector đó:

1. **CHƯƠNG 3 : THỬ NGHIỆM**
   1. **Bộ dữ liệu Iris flower dataset**
      1. **Giới thiệu**

Tập dữ liệu hoa Iris hoặc tập dữ liệu Iris của Fisher là tập dữ liệu đa biến được giới thiệu bởi nhà thống kê và nhà sinh vật học người Anh Ronald Fisher trong bài báo năm 1936 Việc sử dụng nhiều phép đo trong các vấn đề phân loại như một ví dụ về phân tích phân biệt tuyến tính. Đôi khi nó được gọi là tập dữ liệu Iris của Anderson vì Edgar Anderson đã thu thập dữ liệu để định lượng sự biến đổi hình thái của hoa Iris của ba loài liên quan. Hai trong số ba loài được thu thập ở Bán đảo Gaspé "tất cả từ cùng một đồng cỏ, và được chọn vào cùng một ngày và được đo cùng lúc bởi cùng một người với cùng một bộ máy". Bộ dữ liệu bao gồm 50 mẫu từ mỗi ba loài Iris (Iris setosa, Iris virginica và Iris Verscolor). Bốn đặc điểm được đo từ mỗi mẫu: chiều dài và chiều rộng của đài hoa, chiều dài và chiều rộng cánh hoa, tính bằng centimet. Dựa trên sự kết hợp của bốn tính năng này, Fisher đã phát triển một mô hình phân biệt tuyến tính để phân biệt các loài với nhau.



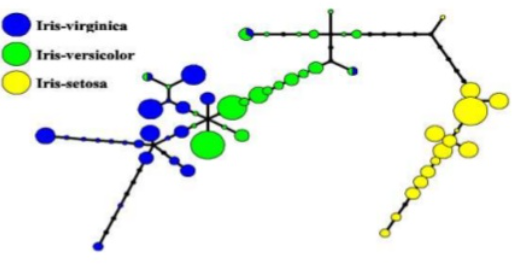
Hình 9 dữ liệu các loài hoa

* + 1. **Sử dụng tập dữ liệu**

Dựa trên mô hình phân biệt tuyến tính của Fisher, bộ dữ liệu này đã trở thành trường hợp thử nghiệm điển hình cho nhiều kỹ thuật phân loại thống kê trong học máy như máy vector hỗ trợ.

Tuy nhiên, việc sử dụng tập dữ liệu này trong phân tích cụm không phổ biến, vì tập dữ liệu chỉ chứa hai cụm có sự phân tách khá rõ ràng. Một trong những cụm chứa Iris setosa, trong khi cụm còn lại chứa cả Iris virginica và Iris Versolor và không thể tách rời nếu không có thông tin về loài mà Fisher sử dụng. Điều này làm cho dữ liệu trở thành một ví dụ tốt để giải thích sự khác biệt giữa các kỹ thuật được giám sát và không giám sát trong khai thác dữ liệu: Mô hình phân biệt tuyến tính của Fisher chỉ có thể thu được khi biết các loài đối tượng: nhãn lớp và cụm không nhất thiết giống nhau.

Tuy nhiên, cả ba loài Iris đều có thể tách rời trong hình chiếu trên thành phần chính phân nhánh phi tuyến. Tập dữ liệu được xấp xỉ bởi cây gần nhất với một số hình phạt cho số lượng nút, uốn cong và kéo dài quá mức. Các điểm dữ liệu được chiếu vào nút gần nhất. Đối với mỗi nút, sơ đồ hình tròn của các điểm được chiếu được chuẩn bị. Diện tích của chiếc bánh tỷ lệ thuận với số lượng điểm được chiếu. Rõ ràng từ sơ đồ (bên dưới) rằng phần lớn tuyệt đối các mẫu của các loài Iris khác nhau thuộc về các nút khác nhau. Chỉ một phần nhỏ Iris-virginica được trộn với Iris- Versolor (các nút màu xanh lam hỗn hợp trong sơ đồ). Do đó, ba loài Iris (Iris setosa, Iris virginica và Iris Verscolor) có thể được phân tách bằng các thủ tục không giám sát trong phân tích thành phần chính phi tuyến. Để phân biệt chúng, chỉ cần chọn các nút tương ứng trên cây chính**.**



Hình 10 Mô hình dataset

* + 1. **Tập dữ liệu**

Bộ dữ liệu gồm 150 bản ghi bao gồm các thuộc tính- sepal.length(chiều dài đài hoa) , sepal.width (chiều rộng cánh hoa), petal.length(chiều dài cánh hoa), petal.width(chiều rộng cánh hoa), variety(tên hoa)

* **Hoa Setosa**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| sepal.length | sepal.width | petal.length | petal.width | variety |
| 5.1 | 3.5 | 1.4 | 0.2 | Setosa |
| 4.9 | 3 | 1.4 | 0.2 | Setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | Setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | Setosa |
| 5 | 3.6 | 1.4 | 0.2 | Setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | Setosa |
| 4.6 | 3.4 | 1.4 | 0.3 | Setosa |
| 5 | 3.4 | 1.5 | 0.2 | Setosa |
| 4.4 | 2.9 | 1.4 | 0.2 | Setosa |
| 4.9 | 3.1 | 1.5 | 0.1 | Setosa |
| 5.4 | 3.7 | 1.5 | 0.2 | Setosa |
| 4.8 | 3.4 | 1.6 | 0.2 | Setosa |
| 4.8 | 3 | 1.4 | 0.1 | Setosa |
| 4.3 | 3 | 1.1 | 0.1 | Setosa |
| 5.8 | 4 | 1.2 | 0.2 | Setosa |
| 5.7 | 4.4 | 1.5 | 0.4 | Setosa |
| 5.4 | 3.9 | 1.3 | 0.4 | Setosa |
| 5.1 | 3.5 | 1.4 | 0.3 | Setosa |
| 5.7 | 3.8 | 1.7 | 0.3 | Setosa |
| 5.1 | 3.8 | 1.5 | 0.3 | Setosa |
| 5.4 | 3.4 | 1.7 | 0.2 | Setosa |
| 5.1 | 3.7 | 1.5 | 0.4 | Setosa |
| 4.6 | 3.6 | 1 | 0.2 | Setosa |
| 5.1 | 3.3 | 1.7 | 0.5 | Setosa |
| 4.8 | 3.4 | 1.9 | 0.2 | Setosa |
| 5 | 3 | 1.6 | 0.2 | Setosa |
| 5 | 3.4 | 1.6 | 0.4 | Setosa |
| 5.2 | 3.5 | 1.5 | 0.2 | Setosa |
| 5.2 | 3.4 | 1.4 | 0.2 | Setosa |
| 4.7 | 3.2 | 1.6 | 0.2 | Setosa |
| 4.8 | 3.1 | 1.6 | 0.2 | Setosa |
| 5.4 | 3.4 | 1.5 | 0.4 | Setosa |
| 5.2 | 4.1 | 1.5 | 0.1 | Setosa |
| 5.5 | 4.2 | 1.4 | 0.2 | Setosa |
| 4.9 | 3.1 | 1.5 | 0.2 | Setosa |
| 5 | 3.2 | 1.2 | 0.2 | Setosa |
| 5.5 | 3.5 | 1.3 | 0.2 | Setosa |
| 4.9 | 3.6 | 1.4 | 0.1 | Setosa |
| 4.4 | 3 | 1.3 | 0.2 | Setosa |
| 5.1 | 3.4 | 1.5 | 0.2 | Setosa |
| 5 | 3.5 | 1.3 | 0.3 | Setosa |
| 4.5 | 2.3 | 1.3 | 0.3 | Setosa |
| 4.4 | 3.2 | 1.3 | 0.2 | Setosa |
| 5 | 3.5 | 1.6 | 0.6 | Setosa |
| 5.1 | 3.8 | 1.9 | 0.4 | Setosa |
| 4.8 | 3 | 1.4 | 0.3 | Setosa |
| 5.1 | 3.8 | 1.6 | 0.2 | Setosa |
| 4.6 | 3.2 | 1.4 | 0.2 | Setosa |
| 5.3 | 3.7 | 1.5 | 0.2 | Setosa |
| 5 | 3.3 | 1.4 | 0.2 | Setosa |

Bảng 1 dữ liệu hoa setosa

* Hoa Versicolor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| sepal.length | sepal.width | petal.length | petal.width | variety |
| 7 | 3.2 | 4.7 | 1.4 | Versicolor |
| 6.4 | 3.2 | 4.5 | 1.5 | Versicolor |
| 6.9 | 3.1 | 4.9 | 1.5 | Versicolor |
| 5.5 | 2.3 | 4 | 1.3 | Versicolor |
| 6.5 | 2.8 | 4.6 | 1.5 | Versicolor |
| 5.7 | 2.8 | 4.5 | 1.3 | Versicolor |
| 6.3 | 3.3 | 4.7 | 1.6 | Versicolor |
| 4.9 | 2.4 | 3.3 | 1 | Versicolor |
| 6.6 | 2.9 | 4.6 | 1.3 | Versicolor |
| 5.2 | 2.7 | 3.9 | 1.4 | Versicolor |
| 5 | 2 | 3.5 | 1 | Versicolor |
| 5.9 | 3 | 4.2 | 1.5 | Versicolor |
| 6 | 2.2 | 4 | 1 | Versicolor |
| 6.1 | 2.9 | 4.7 | 1.4 | Versicolor |
| 5.6 | 2.9 | 3.6 | 1.3 | Versicolor |
| 6.7 | 3.1 | 4.4 | 1.4 | Versicolor |
| 5.6 | 3 | 4.5 | 1.5 | Versicolor |
| 5.8 | 2.7 | 4.1 | 1 | Versicolor |
| 6.2 | 2.2 | 4.5 | 1.5 | Versicolor |
| 5.6 | 2.5 | 3.9 | 1.1 | Versicolor |
| 5.9 | 3.2 | 4.8 | 1.8 | Versicolor |
| 6.1 | 2.8 | 4 | 1.3 | Versicolor |
| 6.3 | 2.5 | 4.9 | 1.5 | Versicolor |
| 6.1 | 2.8 | 4.7 | 1.2 | Versicolor |
| 6.4 | 2.9 | 4.3 | 1.3 | Versicolor |
| 6.6 | 3 | 4.4 | 1.4 | Versicolor |
| 6.8 | 2.8 | 4.8 | 1.4 | Versicolor |
| 6.7 | 3 | 5 | 1.7 | Versicolor |
| 6 | 2.9 | 4.5 | 1.5 | Versicolor |
| 5.7 | 2.6 | 3.5 | 1 | Versicolor |
| 5.5 | 2.4 | 3.8 | 1.1 | Versicolor |
| 5.5 | 2.4 | 3.7 | 1 | Versicolor |
| 5.8 | 2.7 | 3.9 | 1.2 | Versicolor |
| 6 | 2.7 | 5.1 | 1.6 | Versicolor |
| 5.4 | 3 | 4.5 | 1.5 | Versicolor |
| 6 | 3.4 | 4.5 | 1.6 | Versicolor |
| 6.7 | 3.1 | 4.7 | 1.5 | Versicolor |
| 6.3 | 2.3 | 4.4 | 1.3 | Versicolor |
| 5.6 | 3 | 4.1 | 1.3 | Versicolor |
| 5.5 | 2.5 | 4 | 1.3 | Versicolor |
| 5.5 | 2.6 | 4.4 | 1.2 | Versicolor |
| 6.1 | 3 | 4.6 | 1.4 | Versicolor |
| 5.8 | 2.6 | 4 | 1.2 | Versicolor |
| 5 | 2.3 | 3.3 | 1 | Versicolor |
| 5.6 | 2.7 | 4.2 | 1.3 | Versicolor |
| 5.7 | 3 | 4.2 | 1.2 | Versicolor |
| 5.7 | 2.9 | 4.2 | 1.3 | Versicolor |
| 6.2 | 2.9 | 4.3 | 1.3 | Versicolor |
| 5.1 | 2.5 | 3 | 1.1 | Versicolor |
| 5.7 | 2.8 | 4.1 | 1.3 | Versicolor |

Bảng 2 dữ liệu hoa Versicolor

* Hoa Virginica

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| sepal.length | sepal.width | petal.length | petal.width | variety |
| 6.3 | 3.3 | 6 | 2.5 | Virginica |
| 5.8 | 2.7 | 5.1 | 1.9 | Virginica |
| 7.1 | 3 | 5.9 | 2.1 | Virginica |
| 6.3 | 2.9 | 5.6 | 1.8 | Virginica |
| 6.5 | 3 | 5.8 | 2.2 | Virginica |
| 7.6 | 3 | 6.6 | 2.1 | Virginica |
| 4.9 | 2.5 | 4.5 | 1.7 | Virginica |
| 7.3 | 2.9 | 6.3 | 1.8 | Virginica |
| 6.7 | 2.5 | 5.8 | 1.8 | Virginica |
| 7.2 | 3.6 | 6.1 | 2.5 | Virginica |
| 6.5 | 3.2 | 5.1 | 2 | Virginica |
| 6.4 | 2.7 | 5.3 | 1.9 | Virginica |
| 6.8 | 3 | 5.5 | 2.1 | Virginica |
| 5.7 | 2.5 | 5 | 2 | Virginica |
| 5.8 | 2.8 | 5.1 | 2.4 | Virginica |
| 6.4 | 3.2 | 5.3 | 2.3 | Virginica |
| 6.5 | 3 | 5.5 | 1.8 | Virginica |
| 7.7 | 3.8 | 6.7 | 2.2 | Virginica |
| 7.7 | 2.6 | 6.9 | 2.3 | Virginica |
| 6 | 2.2 | 5 | 1.5 | Virginica |
| 6.9 | 3.2 | 5.7 | 2.3 | Virginica |
| 5.6 | 2.8 | 4.9 | 2 | Virginica |
| 7.7 | 2.8 | 6.7 | 2 | Virginica |
| 6.3 | 2.7 | 4.9 | 1.8 | Virginica |
| 6.7 | 3.3 | 5.7 | 2.1 | Virginica |
| 7.2 | 3.2 | 6 | 1.8 | Virginica |
| 6.2 | 2.8 | 4.8 | 1.8 | Virginica |
| 6.1 | 3 | 4.9 | 1.8 | Virginica |
| 6.4 | 2.8 | 5.6 | 2.1 | Virginica |
| 7.2 | 3 | 5.8 | 1.6 | Virginica |
| 7.4 | 2.8 | 6.1 | 1.9 | Virginica |
| 7.9 | 3.8 | 6.4 | 2 | Virginica |
| 6.4 | 2.8 | 5.6 | 2.2 | Virginica |
| 6.3 | 2.8 | 5.1 | 1.5 | Virginica |
| 6.1 | 2.6 | 5.6 | 1.4 | Virginica |
| 7.7 | 3 | 6.1 | 2.3 | Virginica |
| 6.3 | 3.4 | 5.6 | 2.4 | Virginica |
| 6.4 | 3.1 | 5.5 | 1.8 | Virginica |
| 6 | 3 | 4.8 | 1.8 | Virginica |
| 6.9 | 3.1 | 5.4 | 2.1 | Virginica |
| 6.7 | 3.1 | 5.6 | 2.4 | Virginica |
| 6.9 | 3.1 | 5.1 | 2.3 | Virginica |
| 5.8 | 2.7 | 5.1 | 1.9 | Virginica |
| 6.8 | 3.2 | 5.9 | 2.3 | Virginica |
| 6.7 | 3.3 | 5.7 | 2.5 | Virginica |
| 6.7 | 3 | 5.2 | 2.3 | Virginica |
| 6.3 | 2.5 | 5 | 1.9 | Virginica |
| 6.5 | 3 | 5.2 | 2 | Virginica |
| 6.2 | 3.4 | 5.4 | 2.3 | Virginica |

Bảng 3 Dữ liệu hoa virginica

1. **CHƯƠNG 4 : KẾT LUẬN**

Trong quá trình thực hiện đồ án tốt nghiệp, em đã cố gắng hết sức để tìm hiểu và học hỏi nhưng vì khả năng còn giới hạn không tránh khỏi những sai sót, nên có theerchuwa giải quyết được tất cả những vấn đề, đặt ra. Em rất mong nhận được sự thông cảm của quý thầy cô và các bạn. Em xin chân thành cảm ơn.

Những kết quả đạt được:

* Sự hiểu biết về thuật toán KNN cơ bản tương đối tốt.
* Làm quen với Iris flower dataset.
* Từ những gì đã làm được, từ đó hiểu biết thêm về AI, ứng dụng của của ML vào đời sống công nghệ hiện đại.

Làm quen ngôn ngữ lập trình Python Những hạn chế:

* Thuật toán phụ thuộc nhiều vào hệ số K.
* Kết quả đưa ra có sự thay đổi (vì các điểm xét lấy ngẫu nhiên).
* Chưa thự sự hiểu hết về bài toán

# **TÀI LIỆU THAM KHẢO**

K-nearest neighbor cơ bản

<https://machinelearningcoban.com/2017/01/08/knn/#chuan-hoa-du-lieu>

Thuật toán K-nearest neighbor

<https://codelearn.io/sharing/thuat-toan-k-nearest-neighbors-knn>

Iris dataset

<https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html>