**TRƯỜNG ĐẠI HỌC QUY NHƠN**

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----

A blue and white logo

Description automatically generated with low confidence

**Môn học: THỰC HÀNH LÀM VIỆC NHÓM**

**ĐỀ TÀI: K-NEAREST NEIGHBOR (KNN)**

**Giảng viên hướng dẫn: Th.S Lê Thị Xinh**

**Nhóm sinh viên thực hiện:**

4451050605 - CNTT44D - Lê Anh Tú

4451051032 - CNTT44D - Nguyễn Sơn Tùng

4451050690 - CNTT44D - Huỳnh Thịnh Phát

4451051021 - CNTT44D - Nguyễn Văn Trường

4451050965 - CNTT44D - Nguyễn Văn Thanh

**Mã lớp học phần:** 231105013615

***Bình Định, 09/2023***

**MỤC LỤC**

[I. Thành viên nhóm: 3](#_Toc9690)

[II. Đề tài, phân chia công việc và đề cương chi tiết: 3](#_Toc9049)

[1. Đề tài: 3](#_Toc20954)

[2. Kế hoạch phân chia công việc: 3](#_Toc3402)

[3. Đề cương chi tiết: 4](#_Toc4683)

[*3.1. Tên đề tài:* 4](#_Toc30922)

[*3.2. Đặt vấn đề:* 4](#_Toc21117)

[*3.3. Mục đích và mục tiêu nghiên cứu:* 4](#_Toc5312)

[*3.4. Đối tượng cần nghiên cứu:* 4](#_Toc15119)

[*3.5. Phương pháp nghiên cứu:* 4](#_Toc12104)

[3.6. *Nội dung nghiên cứu:* 4](#_Toc20957)

1. **Thành viên nhóm:**
2. 4451050605 - CNTT44D - Lê Anh Tú
3. 4451051032 - CNTT44D - Nguyễn Sơn Tùng
4. 4451050690 - CNTT44D - Huỳnh Thịnh Phát
5. 4451051021 - CNTT44D - Nguyễn Văn Trường
6. 4451050965 - CNTT44D - Nguyễn Văn Thanh
7. **Đề tài, phân chia công việc và đề cương chi tiết:**
8. ***Đề tài:***

K-Nearest Neighbor (KNN)

1. ***Phân chia công việc:***

* Tuần 1: - Tìm hiểu tổng quan về học máy (tất cả thành viên)
* Tuần 2: - Tìm hiểu sơ lược về KNN (Nguyễn Sơn Tùng)
* Tuần 3: - Nguyên lý hoạt động của thuật toán KNN. (Lê Anh Tú, Nguyễn Văn Trường)

- Ưu nhược điểm của thuật toán KNN. (Nguyễn Văn Thanh, Huỳnh Thịnh Phát)

* Tuần 4: - Các loại khoảng cách trong KNN . (Lê Anh Tú)

- Cách chọn giá trị k trong KNN. (Nguyễn Sơn Tùng)

- Các phương pháp xử lý dữ liệu trước khi sử dụng thuật toán KNN. (Huỳnh Thịnh Phát, Nguyễn Văn Trường)

* Tuần 5: - Tìm hiểu công cụ quản lý code github. (tất cả thành viên)
* Tuần 6 + 7: - Áp dụng KNN cho bài toán phân loại. (Huỳnh Thịnh Phát, Nguyễn Văn Thanh)

- Áp dụng KNN cho bài toán hồi quy. (Nguyễn Sơn Tùng, Nguyễn Văn Trường)

* Tuần 8: - Thử nghiệm một số phương pháp cải thiện hiệu quả của thuật toán KNN. (tất cả thành viên)
* Tuần 9 + 10: - Phân tích kết quả phân loại của thuật toán KNN. ( Huỳnh Thịnh Phát, Nguyễn Văn Trường)

- Phân tích hiệu quả của thuật toán KNN. (Nguyễn Sơn Tùng, Nguyễn Văn Thanh)

* Tuần 11: - Thảo luận về các ứng dụng của thuật toán KNN. (tất cả thành viên)
* Tuần 12: - Hoàn thành bài tập nhóm và viết báo cáo. (tất cả thành viên)

1. ***Đề cương chi tiết:***
   1. **Tên đề tài:**

Tìm hiểu và ứng dụng thuật toán K-Nearest Neighbor.

* 1. **Đặt vấn đề:**

- Thuật toán k-nearest neighbors (KNN) là một thuật toán học máy có giám sát được sử dụng để phân loại và hồi quy. Thuật toán này hoạt động bằng cách tìm các điểm dữ liệu gần nhất với dữ liệu mới được phân loại và sử dụng nhãn của các điểm dữ liệu gần nhất để phân loại dữ liệu mới.

-KNN là một thuật toán đơn giản và hiệu quả, có thể được sử dụng trong nhiều lĩnh vực khác nhau, bao gồm phân loại hình ảnh, phân loại văn bản và phân loại dữ liệu thời gian thực.

* 1. **Mục đích và mục tiêu nghiên cứu:**

Mục đích:

- Hệ thống hóa lý luận về thuật toán k-nearest neighbors.

- Phân tích ưu điểm, nhược điểm và các ứng dụng của thuật toán k-nearest neighbors.

- Đề xuất một số giải pháp nhằm cải thiện hiệu quả của thuật toán k-nearest neighbors.

Mục tiêu:

- Khái quát được khái niệm, nguyên lý hoạt động, ưu điểm, nhược điểm và các ứng dụng của thuật toán k-nearest neighbors.

- Hiểu được cách cải thiện hiệu quả của thuật toán k-nearest neighbors.

* 1. **Đối tượng cần nghiên cứu:**

- Lý luận về thuật toán k-nearest neighbors.

- Ưu điểm, nhược điểm và các ứng dụng của thuật toán k-nearest neighbors.

- Giải pháp cải thiện hiệu quả của thuật toán k-nearest neighbors.

* 1. **Phương pháp nghiên cứu:**

- Phương pháp nghiên cứu lý luận: phân tích, tổng hợp, hệ thống hóa các tài liệu, luận cứ liên quan đến đề tài nghiên cứu.

- Phương pháp nghiên cứu thực tiễn: khảo sát, điều tra, phân tích số liệu thực tế để đánh giá hiệu quả của thuật toán k-nearest neighbors.

- Phương pháp so sánh, đối chiếu: so sánh thuật toán k-nearest neighbors với các thuật toán học máy khác.

**3.6. Nội dung nghiên cứu:**

- Chương 1: Tổng quan về học máy.

- Chương 2: Tổng quan về thuật toán k-nearest neighbors. Ưu điểm, nhược điểm và các ứng dụng của thuật toán k-nearest neighbors.

- Chương 3: Giải pháp cải thiện hiệu quả của thuật toán k-nearest neighbors.

**Tuần 1: TỔNG QUAN VỀ HỌC MÁY**

Học máy (machine learning) là một ngành khoa học nghiên cứu các thuật toán cho phép máy tính có thể học được các khái niệm.

**Phân loại: có 2 loại học máy chính**

* Phương pháp quy nạp: Máy sẽ học được các khái niệm dựa trên dữ liệu đã thu thập được trước đó. Phương pháp này cho phép tận dụng được nguồn dữ liệu rất nhiếu và sẵn có
* Phương pháp suy diễn: Máy sẽ học được các khái niệm dựa vào các nguyên tắc. Phương pháp này cho phép sử dụng các kiến thức chuyên ngành để hỗ trợ máy tính.

**Các dạng học máy :**

* Học có giám sát: Máy tính được cung cấp dữ liệu đầu vào và đầu ra mong muốn. Sau khi học xong máy tính quan sát một đầu vào mới và cho ra kết quả đầu ra.
* Học không giám sát: Máy tính học từ dữ liệu đầu vào mà không có đầu ra, sau đó nó tự phát triển kiến thức về dữ liệu .
* Học bán giám sát: Cách học của học bán giám sát là sử dụng dữ liệu có nhãn và dự đoán từ mô hình ban đầu để tạo dữ liệu có nhãn giả, sau đó huấn luyện lại mô hình để cải thiện hiệu suất.
* Học tăng cương: Máy tính đưa ra hành động và nhận kết quả phản hồi từ môi trường. Sau đó máy tính học từ kết quả của hành động nó thực hiện.

**Ứng dụng của học máy:**

Có rất nhiều ứng dụng thực tế khác nhau của học máy. Hai lĩnh vực ứng dụng lớn nhất của học máy là khai phá dữ liệu (data mining) và nhận dạng mẫu (pattern recognition).

- KHAI PHÁ DỮ LIỆU là ứng dụng kỹ thuật học máy vào các cơ sở dữ liệu hoặc các tập dữ liệu lớn để phát hiện quy luật hay tri thức trong dữ liệu đó hoặc để dự đoán các thông tin quan tâm trong tương lai.

- NHẬN DẠNG MẪU là ứng dụng các kỹ thuật học máy để phát hiện các mẫu có tính quy luật trong dữ liệu, thường là dữ liệu hình ảnh, âm thanh. Bài toán nhận dạng mẫu cụ thể thường là xác định nhãn cho đầu vào cụ thể, ví dụ cho ảnh chụp mặt người, cần xác định đó là ai.

\* Xử lý ngôn ngữ tự nhiên (Natural Language Processing): xử lý văn bản, giao tiếp người - máy,...

\* Nhận dạng (Pattern Recognition): nhận dạng tiếng nói, chữ viết tay, vân tay, thị giác máy (Computer Vision),...

\* Tìm kiếm (Search Engine)

\* Chẩn đoán trong y tế: phân tích ảnh X-quang, các hệ chuyên gia chẩn đoán tự động.

\* Tin sinh học: phân loại chuỗi gene, quá trình hình thành gene/protein

\* Vật lý: phân tích ảnh văn thiên, tác động giữa các hạt,...

\* Phát hiện gian lận tài chính (financial fraud): gian lận thẻ tín dụng

\* Phân tích thị trường chứng khoán (stock market analysis)

\* Chơi trò chơi: tự động chơi cờ, hành động của các nhân vật ảo

Röbốt: là tông hợp của rất nhiều ngành khoa học, trong đó học máy tạo nên hệ thần kinh/bộ não của người máy.

**Lý do cần tới học máy:**

Học máy là một nhánh nghiên cứu rất quan trọng của trí tuệ nhân tạo với khá nhiều ứng dụng thành công trong thực tế. Có một số lý do giải thích cho sự cần thiết và phát triển của học máy:

+ Thứ nhất, rất khó xây dựng hệ thống thông minh có thể thực hiện các công việc liên quan đến trí tuệ như thị giác máy, xử lý ngôn ngữ tự nhiên mà không sử dụng tới kinh nghiệm và quá trình học. Thông thường, khi viết chương trình, cần có thuật toán rõ ràng để chuyển đổi đầu vào thành đầu ra. Tuy nhiên, trong nhiều bài toán, rất khó để xây dựng được thuật toán như vậy.

+ Thứ hai, nhiều ứng dụng đòi hỏi chương trình máy tính phải có khả năng thích nghi. Ví dụ, hành vi mua sắm của khách hàng có thể thay đổi theo thời điểm cụ thể trong ngày, trong năm, hoặc theo tuổi tác. Việc xây dựng thuật toán cố định cho những ứng dụng cần thích nghi và thay đổi là không phù hợp. Học máy mang lại khả năng thích nghi nhờ phân tích dữ liệu thu thập được.

+ Thứ ba, việc tìm được chuyên gia và thu thập được tri thức cần thiết cho việc thiết kế thuật toán để giải quyết các vấn đề tương đối khó, trong khi dữ liệu ngày càng nhiều và có thể thu thập dễ dàng hơn. Khả năng lưu trữ và tính toán của máy tính cũng ngày càng tăng, cho phép thực hiện thuật toán học máy trên dữ liệu có kích thước lớn.

+ Cuối cùng, khả năng học là một hoạt động trí tuệ quan trọng của con người, do vậy học máy luôn thu hút được sự quan tâm khi xây dựng hệ thống thông minh.

**Tuần 2: Sơ lược về KNN**

**K-nearest neighbors (KNN) là một thuật toán học máy có giám sát, được sử dụng để phân loại và hồi quy. KNN hoạt động bằng cách tìm các điểm dữ liệu gần nhất với điểm dữ liệu mới cần phân loại hoặc dự đoán.**

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.

* Trong phân loại, KNN sẽ gán điểm dữ liệu mới cho lớp mà các điểm dữ liệu gần nhất thuộc về. Số lượng điểm dữ liệu gần nhất được sử dụng là một tham số được gọi là K.
* Trong hồi quy, KNN sẽ sử dụng các điểm dữ liệu gần nhất để ước tính giá trị của điểm dữ liệu mới.

**Các loại khoảng cách được sử dụng trong KNN:**

* Khoảng cách Euclide
* Khoảng cách Manhattan
* Khoảng cách Hamming
* Khoảng cách Minkowski

**Các tham số của KNN:**

* K: Số lượng điểm dữ liệu gần nhất được sử dụng để phân loại hoặc dự đoán.
* Khoảng cách: Loại khoảng cách được sử dụng để tính toán khoảng cách giữa các điểm dữ liệu.
* Trọng số: Trọng số được gán cho các điểm dữ liệu gần nhất.

**Cách thức hoạt động của KNN:**

Để phân loại một điểm dữ liệu mới, KNN sẽ thực hiện các bước sau:

* Tính toán khoảng cách giữa điểm dữ liệu mới và tất cả các điểm dữ liệu trong tập huấn luyện.
* Sắp xếp các điểm dữ liệu theo khoảng cách, từ gần nhất đến xa nhất.
* Chọn K điểm dữ liệu gần nhất.
* Gán điểm dữ liệu mới cho lớp mà các điểm dữ liệu gần nhất thuộc về.

**Ví dụ:**

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).

**Ưu điểm của KNN:**

Đơn giản và dễ hiểu

Không cần tính toán phức tạp

Có thể áp dụng cho nhiều loại dữ liệu

Có thể xử lý dữ liệu nhiễu

**Nhược điểm của KNN:**

Có thể không chính xác cho các tập dữ liệu lớn

Có thể không hiệu quả cho các tập dữ liệu có nhiều chiều

**KNN được sử dụng trong nhiều ứng dụng khác nhau, bao gồm:**

* Sử dụng KNN để phân loại hình ảnh hoa.
* Sử dụng KNN để phân loại văn bản spam.
* Sử dụng KNN để phân loại âm thanh nhạc rock và nhạc pop.
* Sử dụng KNN để phân loại đối tượng trong video.
* Sử dụng KNN để dự đoán giá cổ phiếu.
* Sử dụng KNN để dự đoán doanh số bán hàng.
* Sử dụng KNN để dự đoán thời tiết.

**Tuần 3: Nguyên lý hoạt động và ưu nhược điểm của KNN**

1. **Nguyên lý hoạt động KNN**

Nguyên lý hoạt động của thuật toán K-nearest neighbors (KNN) là dựa trên ý tưởng rằng các điểm dữ liệu tương tự nhau sẽ nằm gần nhau trong không gian. Để phân loại một điểm dữ liệu mới, KNN sẽ tìm K điểm dữ liệu gần nhất với điểm dữ liệu mới đó, sau đó gán điểm dữ liệu mới cho lớp mà K điểm dữ liệu gần nhất thuộc về.

* **Các bước hoạt động của KNN**

Để phân loại một điểm dữ liệu mới, KNN sẽ thực hiện các bước sau:

* Tính toán khoảng cách giữa điểm dữ liệu mới và tất cả các điểm dữ liệu trong tập huấn luyện.
* Sắp xếp các điểm dữ liệu theo khoảng cách, từ gần nhất đến xa nhất.
* Chọn K điểm dữ liệu gần nhất.
* Gán điểm dữ liệu mới cho lớp mà K điểm dữ liệu gần nhất thuộc về.

**1. Tính toán khoảng cách giữa điểm dữ liệu mới và tất cả các điểm dữ liệu trong tập huấn luyện**

Đầu tiên, KNN sẽ tính toán khoảng cách giữa điểm dữ liệu mới và tất cả các điểm dữ liệu trong tập huấn luyện. Khoảng cách có thể được tính toán bằng nhiều cách khác nhau, chẳng hạn như khoảng cách Euclide, khoảng cách Manhattan, khoảng cách Hamming, hoặc khoảng cách Minkowski.

**2. Sắp xếp các điểm dữ liệu theo khoảng cách, từ gần nhất đến xa nhất**

Sau khi tính toán khoảng cách, KNN sẽ sắp xếp các điểm dữ liệu theo khoảng cách, từ gần nhất đến xa nhất.

**3. Chọn K điểm dữ liệu gần nhất**

Tiếp theo, KNN sẽ chọn K điểm dữ liệu gần nhất. Số lượng điểm dữ liệu gần nhất được chọn là một tham số được gọi là K.

**4. Gán điểm dữ liệu mới cho lớp mà K điểm dữ liệu gần nhất thuộc về**

Cuối cùng, KNN sẽ gán điểm dữ liệu mới cho lớp mà K điểm dữ liệu gần nhất thuộc về.

1. **Ưu Nhược điểm KNN:**

* **Ưu điểm của thuật toán KNN**
* Đơn giản và dễ hiểu: KNN là một thuật toán đơn giản và dễ hiểu. Không cần giả định phân phối dữ liệu hay học một mô hình phức tạp.
* Khả năng xử lý dữ liệu phi cấu trúc: KNN có khả năng xử lý dữ liệu phi cấu trúc, không yêu cầu các giả định về cấu trúc dữ liệu. Điều này cho phép nó áp dụng cho nhiều loại dữ liệu, bao gồm cả dữ liệu văn bản, hình ảnh, và âm thanh.
* Hiệu suất tốt đối với tập dữ liệu nhỏ: KNN hoạt động tốt trên các tập dữ liệu nhỏ với số lượng mẫu ít. Nó không đòi hỏi quá nhiều tính toán trước khi thực hiện dự đoán.
* Dễ dàng tinh chỉnh tham số: K là tham số quan trọng trong thuật toán KNN. Việc điều chỉnh giá trị K có thể ảnh hưởng đến hiệu suất của thuật toán. Tuy nhiên, việc tinh chỉnh K là khá dễ dàng và có thể được thực hiện thông qua quá trình thử và sai.
* **Nhược điểm của thuật toán KNN**
* Phụ thuộc vào kích thước dữ liệu: KNN có hiệu suất giảm khi tập dữ liệu lớn vì việc tính toán khoảng cách giữa các điểm dữ liệu mất nhiều thời gian. KNN không phải là thuật toán phù hợp cho việc xử lý các tập dữ liệu lớn.
* Nhạy cảm với nhiễu và dữ liệu không đồng nhất: KNN dễ bị ảnh hưởng bởi các nhiễu trong dữ liệu và các điểm dữ liệu nằm trong các lớp khác nhau gần nhau. Điều này có thể dẫn đến việc phân loại không chính xác hoặc không ổn định.
* Cần xử lý các biến số và mất cân bằng dữ liệu: KNN không xử lý được các biến số khác nhau và cần sự cân bằng dữ liệu trong các lớp khác nhau. Nếu một lớp có số lượng mẫu nhiều hơn so với lớp khác, KNN có thể dễ dàng bị thiên vị và cho ra kết quả không chính xác.
* Yêu cầu lưu trữ toàn bộ dữ liệu huấn luyện: KNN yêu cầu lưu trữ toàn bộ dữ liệu huấn luyện trong bộ nhớ để tính toán khoảng cách và tìm láng giềng gần nhất. Điều này có thể là một vấn đề khi làm việc với các tập dữ liệu lớn và yêu cầu nhiều tài nguyên.

**Tuần 4: CÁC LOẠI KHOẢNG CÁCH TRONG KNN, CÁCH CHỌN THAM SỐ K VÀ CÁC PHƯƠNG PHÁP XỬ LÝ DỮ LIỆU TRƯỚC KHI SỬ DỤNG THUẬT TOÁN KNN**

* **Các loại khoảng cách được sử dụng trong KNN:**

Khoảng cách là một phép đo mức độ gần gũi giữa hai điểm dữ liệu. Có nhiều loại khoảng cách khác nhau có thể được sử dụng trong KNN, bao gồm:

* Khoảng cách Euclide: Khoảng cách Euclide là khoảng cách tiêu chuẩn giữa hai điểm dữ liệu trong không gian Euclide. Nó được tính bằng công thức sau:

d(x, y) = √(x1 - y1)^2 + (x2 - y2)^2 + ... + (xn - yn)^2

* Khoảng cách Manhattan: Khoảng cách Manhattan là khoảng cách tổng của các khoảng cách theo trục cho hai điểm dữ liệu. Nó được tính bằng công thức sau:

d(x, y) = |x1 - y1| + |x2 - y2| + ... + |xn - yn|

* Khoảng cách Hamming: Khoảng cách Hamming là số lượng bit khác nhau giữa hai điểm dữ liệu. Nó được tính bằng công thức sau:

d(x, y) = |x1 ⊕ y1| + |x2 ⊕ y2| + ... + |xn ⊕ yn|

* Khoảng cách Minkowski: Khoảng cách Minkowski là một tổng có trọng số của các khoảng cách theo trục cho hai điểm dữ liệu. Nó được tính bằng công thức sau:

d(x, y) = (|x1 - y1|^p + |x2 - y2|^p + ... + |xn - yn|^p)^(1/p)

* **Cách chọn tham số K**
* K là một tham số quan trọng của thuật toán KNN, quyết định mức độ ảnh hưởng của các điểm dữ liệu gần nhất đến quyết định phân loại hoặc dự đoán. K càng lớn thì quyết định của KNN sẽ càng dựa trên ý kiến của nhiều điểm dữ liệu.
* Cách chọn tham số K phụ thuộc vào dữ liệu và nhiệm vụ cụ thể. Một số cách chọn tham số K phổ biến bao gồm:

+ Sử dụng phương pháp cross-validation: Phương pháp cross-validation sẽ chia tập dữ liệu thành các tập con, sau đó sử dụng mỗi tập con làm tập kiểm tra để đánh giá hiệu quả của KNN với các giá trị K khác nhau.

+ Sử dụng phương pháp grid search: Phương pháp grid search sẽ thử nghiệm tất cả các giá trị K trong một tập hợp các giá trị K xác định, sau đó chọn giá trị K có hiệu quả nhất.

+ Sử dụng phương pháp dựa trên kiến thức: Phương pháp này sẽ sử dụng kiến thức về dữ liệu và nhiệm vụ cụ thể để chọn tham số K.

* **Trước khi sử dụng thuật toán KNN, cần phải xử lý dữ liệu để đảm bảo rằng dữ liệu phù hợp với thuật toán.**

Một số phương pháp xử lý dữ liệu trước khi sử dụng thuật toán KNN bao gồm:

Loại bỏ các điểm dữ liệu ngoại lệ: Các điểm dữ liệu ngoại lệ là các điểm dữ liệu nằm cách xa các điểm dữ liệu khác trong không gian. Các điểm dữ liệu ngoại lệ có thể làm giảm hiệu quả của KNN.

Tiền xử lý dữ liệu nhiễu: Dữ liệu nhiễu là dữ liệu sai lệch hoặc không chính xác. Dữ liệu nhiễu có thể làm giảm hiệu quả của KNN.

Chuẩn hóa dữ liệu: Chuẩn hóa dữ liệu sẽ đưa tất cả các thuộc tính dữ liệu về cùng một phạm vi. Chuẩn hóa dữ liệu có thể giúp KNN hoạt động hiệu quả hơn.

Tính toán trọng số cho các điểm dữ liệu: Trọng số cho các điểm dữ liệu có thể được sử dụng để điều chỉnh mức độ ảnh hưởng của các điểm dữ liệu gần nhất đến quyết định phân loại hoặc dự đoán.

Dưới đây là mô tả chi tiết về từng phương pháp:

**Loại bỏ các điểm dữ liệu ngoại lệ:**

Xác định các điểm dữ liệu ngoại lệ dựa trên các giá trị thống kê: Các điểm dữ liệu có giá trị nằm ngoài một số ngưỡng thống kê nhất định có thể được coi là các điểm dữ liệu ngoại lệ.

Xác định các điểm dữ liệu ngoại lệ dựa trên các thuộc tính: Các điểm dữ liệu có các thuộc tính nằm ngoài một số phạm vi nhất định có thể được coi là các điểm dữ liệu ngoại lệ.

**Dữ liệu nhiễu có thể được xử lý bằng cách sử dụng các kỹ thuật như:**

Xóa dữ liệu nhiễu: Dữ liệu nhiễu có thể được xóa trực tiếp khỏi tập dữ liệu.

Sửa dữ liệu nhiễu: Dữ liệu nhiễu có thể được sửa chữa bằng cách thay thế các giá trị sai lệch bằng các giá trị chính xác.

Giảm thiểu tác động của dữ liệu nhiễu: Dữ liệu nhiễu có thể được giảm thiểu tác động bằng cách sử dụng các kỹ thuật như bình thường hóa dữ liệu hoặc tính toán trọng số cho các điểm dữ liệu.

**Dữ liệu có thể được chuẩn hóa bằng cách sử dụng các kỹ thuật như:**

Chuẩn hóa min-max: Chuẩn hóa min-max sẽ chuyển đổi tất cả các giá trị dữ liệu về cùng một phạm vi, từ 0 đến 1.

Chuẩn hóa z-score: Chuẩn hóa z-score sẽ chuyển đổi tất cả các giá trị dữ liệu về cùng một giá trị trung bình và độ lệch chuẩn.

**Trọng số cho các điểm dữ liệu có thể được tính toán bằng cách sử dụng các kỹ thuật như:**

Trọng số dựa trên khoảng cách: Trọng số của các điểm dữ liệu gần nhất có thể được tăng lên để tăng cường ảnh hưởng của chúng đến quyết định phân loại hoặc dự đoán.

Trọng số dựa trên độ tin cậy: Trọng số của các điểm dữ liệu có thể được tăng lên nếu chúng có độ tin cậy cao.

**Tuần 5: TÌM HIỂU CÔNG CỤ QUẢN LÝ GITHUB**

GitHub là một dịch vụ lưu trữ mã nguồn và cộng tác mã nguồn mã nguồn mở. Nó cho phép người dùng lưu trữ, chia sẻ và cộng tác trên mã nguồn. GitHub được sử dụng bởi các nhà phát triển phần mềm từ khắp nơi trên thế giới để phát triển các dự án phần mềm.

* **GitHub có một số chức năng chính, bao gồm:**
* Lưu trữ mã nguồn: GitHub cho phép người dùng lưu trữ mã nguồn của họ trên một máy chủ từ xa. Điều này giúp bảo vệ mã nguồn và cho phép người dùng truy cập mã từ bất kỳ đâu.
* Chia sẻ mã nguồn: GitHub cho phép người dùng chia sẻ mã nguồn của họ với những người khác. Điều này có thể được thực hiện bằng cách tạo kho lưu trữ công khai hoặc riêng tư.
* Cộng tác trên mã nguồn: GitHub cho phép người dùng cộng tác trên mã nguồn của nhau. Điều này có thể được thực hiện bằng cách tạo nhánh và pull request.
* GitHub cũng cung cấp một số tính năng bổ sung, chẳng hạn như:
* Issue tracker: Issue tracker cho phép người dùng theo dõi các vấn đề và yêu cầu trong dự án.
* Wiki: Wiki cho phép người dùng tạo và lưu trữ tài liệu về dự án.
* Giao tiếp: GitHub cung cấp một số tính năng giao tiếp, chẳng hạn như các cuộc trò chuyện, nhóm và email.

GitHub là một công cụ quan trọng cho các nhà phát triển phần mềm. Nó cung cấp một cách dễ dàng để lưu trữ, chia sẻ và cộng tác trên mã nguồn.

* **Một số câu lệnh thường dùng trong git mà em tìm hiểu được:**
* git init: tạo 1 repo trong máy(nếu chưa có)
* git status: trạng thái các file
* git add namefile.phần\_mở\_rộng
* git commit -m "nội dung commit" (ghi chú thích để biết làm cái gì. Vd update index.html file)
* git log: xem file đã commit vào local chưa
* git remote add origin http://đường dẫn web github repo của bạn.git
* git push -u origin master: add code branch master từ máy lên repo github
* Tất cả các bước trên cần thực hiện theo thứ tự
* git add . : add toàn bộ file trong máy lên github
* git pull: đồng bộ code trên github về máy
* **Dưới đây là một số lợi ích của việc sử dụng GitHub:**
* Lưu trữ mã nguồn an toàn: GitHub sử dụng mã hóa SSL để bảo vệ mã nguồn của bạn.
* Chia sẻ mã nguồn dễ dàng: GitHub cho phép bạn chia sẻ mã nguồn của mình với những người khác bằng cách tạo kho lưu trữ công khai hoặc riêng tư.
* Cộng tác trên mã nguồn hiệu quả: GitHub cho phép bạn cộng tác trên mã nguồn của nhau bằng cách tạo nhánh và pull request.
* Theo dõi tiến độ dự án: GitHub cung cấp một số tính năng để theo dõi tiến độ dự án, chẳng hạn như issue tracker và wiki.
* Tìm hiểu từ cộng đồng: GitHub có một cộng đồng lớn và tích cực của các nhà phát triển phần mềm. Bạn có thể học hỏi từ những người khác và tìm kiếm sự giúp đỡ khi cần thiết.

**Tuần 6+7 ÁP DỤNG KNN CHO BÀI TOÁN PHÂN LOẠI VÀ BÀI TOÁN HỒI QUY**

Thuật toán KNN (K-Nearest Neighbors) là một thuật toán học máy có giám sát, được sử dụng rộng rãi trong các bài toán phân loại và hồi quy. Thuật toán này dựa trên giả định rằng các dữ liệu tương tự nhau sẽ có cùng nhãn.

* **Ứng dụng thuật toán KNN cho bài toán phân loại**

Trong bài toán phân loại, thuật toán KNN sẽ phân loại một điểm dữ liệu mới dựa trên nhãn của các điểm dữ liệu gần nó nhất. Số lượng điểm dữ liệu gần nhất được gọi là K.

Cách hoạt động của thuật toán KNN

Thuật toán KNN có thể được chia thành các bước sau:

Tập dữ liệu huấn luyện: Tập dữ liệu huấn luyện gồm các điểm dữ liệu đã biết nhãn.

Chọn K: Chọn số lượng điểm dữ liệu gần nhất (K).

Tính khoảng cách: Tính khoảng cách giữa điểm dữ liệu mới và các điểm dữ liệu trong tập dữ liệu huấn luyện.

Lựa chọn nhãn: Lựa chọn nhãn của điểm dữ liệu mới dựa trên nhãn của các điểm dữ liệu gần nhất.

Đặc điểm của thuật toán KNN

Ưu điểm:

Đơn giản và dễ hiểu.

Dễ triển khai và có thể áp dụng cho nhiều tập dữ liệu khác nhau.

Có thể giải quyết các bài toán phân loại phức tạp.

Nhược điểm:

Độ chính xác của mô hình phụ thuộc vào giá trị K.

Có thể gặp phải hiện tượng overfitting nếu K quá lớn.

Một số ví dụ về ứng dụng thuật toán KNN cho bài toán phân loại

Phân loại hình ảnh: Thuật toán KNN có thể được sử dụng để phân loại hình ảnh dựa trên các đặc điểm của hình ảnh. Ví dụ, thuật toán KNN có thể được sử dụng để phân loại ảnh người, ảnh động vật hoặc ảnh đồ vật.

Phân loại văn bản: Thuật toán KNN có thể được sử dụng để phân loại văn bản dựa trên nội dung của văn bản. Ví dụ, thuật toán KNN có thể được sử dụng để phân loại email là spam hay không spam, hoặc phân loại bài báo là tin tức hay quảng cáo.

Phân loại khách hàng: Thuật toán KNN có thể được sử dụng để phân loại khách hàng dựa trên các đặc điểm của khách hàng. Ví dụ, thuật toán KNN có thể được sử dụng để phân loại khách hàng là tiềm năng hay không tiềm năng, hoặc phân loại khách hàng là trung thành hay không trung thành.

Ví dụ:

Giả sử chúng ta có tập dữ liệu về màu sắc của hoa, với hai lớp: hoa hồng và hoa cúc. Tập dữ liệu bao gồm các đặc điểm sau:

Chiều dài cánh hoa: Số lượng pixel trong hình ảnh của cánh hoa.

Chiều rộng cánh hoa: Số lượng pixel trong hình ảnh của cánh hoa.

Chúng ta có thể sử dụng thuật toán KNN để xây dựng một mô hình phân loại hoa hồng và hoa cúc. Để làm điều này, chúng ta cần thu thập dữ liệu về hoa hồng và hoa cúc, chuẩn hóa dữ liệu và phân chia dữ liệu thành tập huấn luyện và tập kiểm tra.

Tiếp theo, chúng ta cần chọn giá trị k. Chúng ta có thể bắt đầu với giá trị k = 3 và tăng giá trị k cho đến khi chúng ta đạt được độ chính xác mong muốn.

Cuối cùng, chúng ta có thể sử dụng tập kiểm tra để đánh giá mô hình KNN. Nếu mô hình đạt được độ chính xác cao, chúng ta có thể sử dụng mô hình để phân loại hoa hồng và hoa cúc.

Ví dụ

Giả sử chúng ta có tập dữ liệu huấn luyện gồm 10 điểm dữ liệu, mỗi điểm dữ liệu có 2 thuộc tính:

Thuộc tính 1 Thuộc tính 2 Lớp

0 0 A

1 1 A

2 2 A

3 3 A

4 4 B

5 5 B

6 6 B

7 7 B

8 8 B

Để phân loại điểm dữ liệu mới có thuộc tính 1 = 9 và thuộc tính 2 = 9, ta thực hiện các bước sau:

Tính khoảng cách giữa điểm dữ liệu mới và 10 điểm dữ liệu trong tập huấn luyện.

Chọn ra 5 điểm dữ liệu gần nhất với điểm dữ liệu mới.

Trong 5 điểm dữ liệu gần nhất, có 4 điểm dữ liệu thuộc lớp A và 1 điểm dữ liệu thuộc lớp B.

Do đó, điểm dữ liệu mới sẽ được gán nhãn lớp A.

Ví dụ:

Giả sử chúng ta có tập dữ liệu đào tạo sau:

| Điểm dữ liệu | Nhãn |

|---|---|

| [1, 2] | A |

| [3, 4] | B |

| [5, 6] | C |

| [7, 8] | D |

Tập dữ liệu này có 4 điểm dữ liệu, mỗi điểm dữ liệu có hai thuộc tính và một nhãn. Chúng ta có thể chọn tham số K là 3.

Để phân loại điểm dữ liệu mới [9, 10], chúng ta sẽ tìm ba điểm dữ liệu gần nhất với điểm dữ liệu này. Các điểm dữ liệu gần nhất với điểm dữ liệu [9, 10] là [7, 8], [5, 6] và [3, 4]. Nhãn của ba điểm dữ liệu này là D, C và B. Do đó, chúng ta sẽ dự đoán nhãn của điểm dữ liệu [9, 10] là B.

Một số ứng dụng thực tế của thuật toán KNN cho bài toán hồi quy:

Dự đoán giá nhà

Dự đoán doanh số bán hàng

Dự đoán chi phí sản xuất

Dự đoán điểm thi

Dự đoán thời tiết

Ví dụ:

Giả sử chúng ta có tập dữ liệu sau về giá trị của một căn nhà và số phòng ngủ:

Số phòng ngủ Giá trị (USD)

2 100.000

3 200.000

4 300.000

5 400.000

Chúng ta muốn sử dụng KNN để dự đoán giá trị của một căn nhà có 6 phòng ngủ.

Để làm được điều này, chúng ta cần chọn một giá trị K. Giá trị K càng lớn thì kết quả dự đoán càng ít bị ảnh hưởng bởi các điểm dữ liệu ngoại lệ. Trong ví dụ này, chúng ta sẽ chọn K = 3.

Tiếp theo, chúng ta sẽ tìm ra 3 điểm dữ liệu gần nhất với điểm dữ liệu có 6 phòng ngủ trong tập dữ liệu huấn luyện.

Số phòng ngủ Giá trị (USD)

5 400.000

4 300.000

3 200.000

Giá trị trung bình của 3 điểm dữ liệu này là 300.000 USD. Do đó, chúng ta dự đoán giá trị của một căn nhà có 6 phòng ngủ là 300.000 USD.

* **Ứng dụng KNN cho bài toán hồi quy**

Trong bài toán hồi quy, thuật toán KNN được sử dụng để dự đoán giá trị đầu ra y của một điểm dữ liệu mới x dựa trên các điểm dữ liệu gần nhất với x trong tập huấn luyện. Thuật toán này hoạt động như sau:

1. Tính khoảng cách giữa điểm dữ liệu mới x và các điểm dữ liệu trong tập huấn luyện.
2. Chọn K điểm dữ liệu gần nhất với x.
3. Tính giá trị trung bình của các điểm dữ liệu K này để dự đoán giá trị đầu ra y của điểm dữ liệu mới x.

Ví dụ

Giả sử chúng ta có tập dữ liệu sau:

|  |  |
| --- | --- |
| x | y |
| 1 | 2 |
| 2 | 3 |
| 3 | 4 |

drive\_spreadsheetXuất sang Trang tính

Chúng ta muốn sử dụng thuật toán KNN để dự đoán giá trị đầu ra y của điểm dữ liệu mới x = 4.

1. Tính khoảng cách giữa điểm dữ liệu mới x = 4 và các điểm dữ liệu trong tập huấn luyện.

distance(1, 4) = 3

distance(2, 4) = 2

distance(3, 4) = 1

distance là khoảng cách giữa hai điểm dữ liệu

1. Chọn K = 2 điểm dữ liệu gần nhất với x = 4.

K = 2

sorted\_distances = [distance(1, 4), distance(2, 4)]

sorted\_distances.sort()

sorted\_distances = [1, 2]

closest\_points = [1, 2]

sorted\_distances là một danh sách các khoảng cách giữa điểm dữ liệu mới x và các điểm dữ liệu trong tập huấn luyện, được sắp xếp theo thứ tự tăng dần.

1. Tính giá trị trung bình của các điểm dữ liệu K này để dự đoán giá trị đầu ra y của điểm dữ liệu mới x.

y\_hat = (2 + 3) / 2

y\_hat = 2.5

Trong bài toán hồi quy, y\_hat là giá trị trung bình của các điểm dữ liệu K gần nhất với x.

Vậy, thuật toán KNN dự đoán giá trị đầu ra y của điểm dữ liệu mới x = 4 là 2.5.

Ưu điểm của KNN trong bài toán hồi quy

* Đơn giản, dễ hiểu và dễ triển khai.
* Không cần giả định gì về phân phối của dữ liệu.
* Có thể được sử dụng để hồi quy các dữ liệu có số lượng đặc trưng lớn.
* Có thể được sử dụng để hồi quy các dữ liệu không tuyến tính.

Nhược điểm của KNN trong bài toán hồi quy

* Độ chính xác phụ thuộc vào số lượng láng giềng được sử dụng (K).
* Độ phức tạp phụ thuộc vào số chiều của không gian đặc trưng.
* Không thể học các đặc trưng phức tạp của dữ liệu.

Các phương pháp cải thiện hiệu quả của KNN trong bài toán hồi quy

* Thử nghiệm các giá trị K khác nhau: Chọn giá trị K phù hợp với tập dữ liệu.
* Sử dụng các thước đo khoảng cách khác nhau: Chọn thước đo khoảng cách phù hợp với tập dữ liệu.
* Sử dụng kỹ thuật đánh trọng số: Gán các trọng số khác nhau cho các láng giềng để cải thiện độ chính xác trong trường hợp dữ liệu bị nhiễu hoặc phân lớp không đồng đều.

Kết luận

Thuật toán KNN là một thuật toán học máy đơn giản và hiệu quả. Thuật toán này có thể được sử dụng để hồi quy các dữ liệu có số lượng đặc trưng lớn và không tuyến tính. Để cải thiện hiệu quả của KNN trong bài toán hồi quy, có thể áp dụng các phương pháp như thử nghiệm các giá trị K khác nhau, sử dụng các thước đo khoảng cách khác nhau và sử dụng kỹ thuật đánh trọng số.

**TUẦN 8: THỬ NGHIỆM MỘT SỐ PHƯƠNG PHÁP CẢI THIỆN HIỆU QUẢ CỦA THUẬT TOÁN KNN**

Thuật toán KNN là một thuật toán phân loại dựa trên học tập gần nhất, trong đó một điểm dữ liệu mới được phân loại dựa trên các điểm dữ liệu gần nhất với nó trong tập huấn luyện. Thuật toán này có ưu điểm là đơn giản, dễ hiểu và dễ triển khai, tuy nhiên cũng có một số nhược điểm như:

* Độ chính xác phụ thuộc vào số lượng láng giềng được sử dụng (K). Nếu K quá nhỏ, thuật toán có thể bị quá khớp (overfitting), dẫn đến kết quả kém chính xác trên tập dữ liệu mới. Nếu K quá lớn, thuật toán có thể bị quá đơn giản hóa (underfitting), dẫn đến kết quả kém chính xác trên tập dữ liệu huấn luyện.
* Độ phức tạp phụ thuộc vào số chiều của không gian đặc trưng. Nếu không gian đặc trưng có nhiều chiều, thuật toán sẽ mất nhiều thời gian để tính toán khoảng cách giữa các điểm dữ liệu.

Để cải thiện hiệu quả của thuật toán KNN, có thể thử nghiệm một số phương pháp sau:

* Sử dụng kỹ thuật đánh trọng số. Trong phương pháp này, các láng giềng được gán các trọng số khác nhau, tùy thuộc vào mức độ gần gũi của chúng với điểm dữ liệu mới. Kỹ thuật này có thể giúp cải thiện độ chính xác của thuật toán, đặc biệt là trong trường hợp dữ liệu bị nhiễu hoặc phân lớp không đồng đều.
* Sử dụng kỹ thuật phân tích cấu trúc. Trong phương pháp này, dữ liệu được phân tích để tìm ra các cấu trúc hoặc mô hình có thể được sử dụng để cải thiện hiệu quả của thuật toán KNN. Ví dụ, thuật toán K-means có thể được sử dụng để nhóm các điểm dữ liệu thành các cụm, sau đó thuật toán KNN có thể được sử dụng để phân loại các điểm dữ liệu mới dựa trên cụm mà chúng thuộc về.
* Sử dụng kỹ thuật học không giám sát. Trong phương pháp này, thuật toán KNN được sử dụng để học các đặc trưng của dữ liệu mà không cần biết nhãn của các điểm dữ liệu. Các đặc trưng này sau đó có thể được sử dụng để cải thiện hiệu quả của thuật toán KNN trong quá trình phân loại.

Dưới đây là một số ví dụ cụ thể về việc áp dụng các phương pháp này để cải thiện hiệu quả của thuật toán KNN:

* Sử dụng kỹ thuật đánh trọng số. Trong bài toán phân loại ảnh hoa, có thể sử dụng kỹ thuật đánh trọng số để ưu tiên các láng giềng thuộc cùng lớp với điểm dữ liệu mới. Điều này sẽ giúp cải thiện độ chính xác của thuật toán trong trường hợp dữ liệu bị nhiễu hoặc phân lớp không đồng đều.
* Sử dụng kỹ thuật phân tích cấu trúc. Trong bài toán phân loại văn bản, có thể sử dụng thuật toán K-means để nhóm các điểm dữ liệu thành các cụm dựa trên các đặc trưng ngữ nghĩa của chúng. Sau đó, thuật toán KNN có thể được sử dụng để phân loại các điểm dữ liệu mới dựa trên cụm mà chúng thuộc về.
* Sử dụng kỹ thuật học không giám sát. Trong bài toán phân loại dữ liệu khách hàng, có thể sử dụng thuật toán KNN để học các đặc trưng của dữ liệu khách hàng mà không cần biết nhãn của các điểm dữ liệu. Các đặc trưng này sau đó có thể được sử dụng để cải thiện hiệu quả của thuật toán KNN trong quá trình phân loại khách hàng theo nhóm.

Việc lựa chọn phương pháp cải thiện hiệu quả của thuật toán KNN cần được xem xét dựa trên các đặc điểm của tập dữ liệu và yêu cầu của bài toán.

Dưới đây là một số hướng dẫn chung để lựa chọn phương pháp cải thiện hiệu quả của thuật toán KNN:

* Sử dụng kỹ thuật đánh trọng số nếu dữ liệu bị nhiễu hoặc phân lớp không đồng đều.
* Sử dụng kỹ thuật phân tích cấu trúc nếu dữ liệu có nhiều đặc trưng hoặc các đặc trưng có mối quan hệ với nhau.
* Sử dụng kỹ thuật học không giám sát nếu dữ liệu có số lượng điểm dữ liệu lớn hoặc dữ liệu không có nhãn.

Trên thực tế, có thể kết hợp nhiều phương pháp để cải thiện hiệu quả của thuật toán KNN. Ví dụ, có thể sử dụng kỹ thuật đánh trọng số kết hợp với kỹ thuật phân tích cấu trúc để cải thiện độ chính xác của thuật toán trong trường hợp dữ liệu bị nhiễu hoặc phân lớp không đồng đều.

**Tuần 9+10: PHÂN TÍCH KẾT QUẢ PHÂN LOẠI CỦA THUẬT TOÁN KNN VÀ PHÂN TÍCH HIỆU QUẢ CỦA THUẬT TOÁN KNN**

* **Để phân tích kết quả phân loại của thuật toán KNN, có thể sử dụng các chỉ số đánh giá sau:**
* Độ chính xác (accuracy): Số điểm dữ liệu được phân loại chính xác trên tập dữ liệu thử nghiệm chia cho tổng số điểm dữ liệu trên tập dữ liệu thử nghiệm.
* Độ nhạy (sensitivity): Số điểm dữ liệu thuộc lớp dương được phân loại chính xác chia cho tổng số điểm dữ liệu thuộc lớp dương.
* Độ đặc hiệu (specificity): Số điểm dữ liệu thuộc lớp âm được phân loại chính xác chia cho tổng số điểm dữ liệu thuộc lớp âm.
* Mức độ lỗi (error rate): Số điểm dữ liệu được phân loại sai chia cho tổng số điểm dữ liệu trên tập dữ liệu thử nghiệm.
* Độ chính xác trung bình (mean accuracy): Độ chính xác của thuật toán trên nhiều lần chạy với các giá trị K khác nhau.

Ngoài ra, có thể sử dụng các phương pháp trực quan hóa để phân tích kết quả phân loại của thuật toán KNN, chẳng hạn như:

* Ma trận phân loại (confusion matrix): Bảng hiển thị số điểm dữ liệu được phân loại chính xác và sai cho từng lớp.
* Biểu đồ ROC (receiver operating characteristic): Biểu đồ hiển thị độ nhạy và độ đặc hiệu của thuật toán.
* Biểu đồ AUC (area under the curve): Diện tích dưới đường cong ROC.

Dưới đây là một số ví dụ cụ thể về phân tích kết quả phân loại của thuật toán KNN:

* Trong trường hợp độ chính xác thấp, cần xem xét nguyên nhân gây ra lỗi. Có thể là do dữ liệu bị nhiễu, phân lớp không đồng đều, hoặc giá trị K không phù hợp.
* Trong trường hợp độ nhạy thấp, cần xem xét liệu thuật toán có thể phân loại chính xác các điểm dữ liệu thuộc lớp dương hay không. Có thể là do dữ liệu bị nhiễu, hoặc thuật toán không phân biệt được các điểm dữ liệu thuộc lớp dương và lớp âm.
* Trong trường hợp độ đặc hiệu thấp, cần xem xét liệu thuật toán có thể phân loại chính xác các điểm dữ liệu thuộc lớp âm hay không. Có thể là do dữ liệu bị nhiễu, hoặc thuật toán phân loại quá nhiều điểm dữ liệu thuộc lớp dương thành lớp âm.

Việc phân tích kết quả phân loại của thuật toán KNN giúp đánh giá hiệu quả của thuật toán và đưa ra các biện pháp cải thiện nếu cần thiết.

* **Phân tích hiệu quả của thuật toán KNN**

Hiệu quả của thuật toán KNN phụ thuộc vào nhiều yếu tố, bao gồm:

* Số lượng láng giềng (K): K càng lớn thì thuật toán càng có khả năng học được các đặc trưng phức tạp của dữ liệu. Tuy nhiên, K quá lớn cũng có thể dẫn đến quá đơn giản hóa.
* Thước đo khoảng cách: Thước đo khoảng cách được sử dụng để tính toán khoảng cách giữa các điểm dữ liệu có thể ảnh hưởng đến kết quả phân loại.
* Dữ liệu huấn luyện: Dữ liệu huấn luyện cần phải đại diện cho các dữ liệu thực tế. Nếu dữ liệu huấn luyện không đại diện, thuật toán có thể không hiệu quả.

Để đánh giá hiệu quả của thuật toán KNN, có thể sử dụng các chỉ số đánh giá sau:

* Độ chính xác (accuracy): Số điểm dữ liệu được phân loại chính xác trên tập dữ liệu thử nghiệm chia cho tổng số điểm dữ liệu trên tập dữ liệu thử nghiệm.
* Độ nhạy (sensitivity): Số điểm dữ liệu thuộc lớp dương được phân loại chính xác chia cho tổng số điểm dữ liệu thuộc lớp dương.
* Độ đặc hiệu (specificity): Số điểm dữ liệu thuộc lớp âm được phân loại chính xác chia cho tổng số điểm dữ liệu thuộc lớp âm.
* Mức độ lỗi (error rate): Số điểm dữ liệu được phân loại sai chia cho tổng số điểm dữ liệu trên tập dữ liệu thử nghiệm.

Các phương pháp cải thiện hiệu quả của thuật toán KNN

Để cải thiện hiệu quả của thuật toán KNN, có thể áp dụng các phương pháp sau:

* Thử nghiệm các giá trị K khác nhau: Chọn giá trị K phù hợp với tập dữ liệu.
* Sử dụng các thước đo khoảng cách khác nhau: Chọn thước đo khoảng cách phù hợp với tập dữ liệu.
* Sử dụng kỹ thuật đánh trọng số: Gán các trọng số khác nhau cho các láng giềng để cải thiện độ chính xác trong trường hợp dữ liệu bị nhiễu hoặc phân lớp không đồng đều.

Ví dụ về ứng dụng của thuật toán KNN

Thuật toán KNN có thể được ứng dụng trong nhiều bài toán phân loại, chẳng hạn như:

* Phân loại hình ảnh
* Phân loại văn bản
* Phân loại âm thanh
* Phân loại dữ liệu khách hàng

Kết luận

Thuật toán KNN là một thuật toán học máy đơn giản và hiệu quả. Thuật toán này có thể được sử dụng để phân loại các dữ liệu có số lượng đặc trưng lớn và không tuyến tính. Để cải thiện hiệu quả của thuật toán KNN, có thể áp dụng các phương pháp như thử nghiệm các giá trị K khác nhau, sử dụng các thước đo khoảng cách khác nhau và sử dụng kỹ thuật đánh trọng số.