**Shape, square

Description automatically generated**

**TRƯỜNG ĐẠI HỌC QUY NHƠN**

KHOA CÔNG NGHỆ THÔNG TIN

-----🙞🙜🕮🙞🙜-----

A blue and white logo

Description automatically generated ‘;

**BÁO CÁO CUỐI KÌ**

**THỰC HÀNH LÀM VIỆC NHÓM**

**CHỦ ĐỀ: THUẬT TOÁN K Nearest Neighbor**

**Đề tài: Chẩn đoán tim bằng K Nearest Neighbor**

***Giảng viên hướng dẫn:*** Ths 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

***Khoa*:** Công nghệ thông tin

Bình Định, tháng 11 năm 2023

**MỤC LỤC**

[**I. Tìm hiểu chung:**](#_Toc9690)

1. Tìm hiểu tổng quan về học máy.

2. Tìm hiểu về thuật toán K-Nearest Neigbor(KNN).

3. Tìm hiểu tổng quan về Github và Git

[**II. Phát biểu về đề tài chẩn đoán bệnh tim:**](#_Toc9049)

1. Ý tưởng.

2. Cách thức thực hiện.

3. Dữ liệu sử dụng

**III. Demo chẩn đoán bệnh tim bằng thuật toán KNN:**

1. Ngôn ngữ được sử dụng.

2. Quản lý code

3. Code demo và Chi tiết các hàm được sử dụng.

**IV. Đánh giá và kết quả đạt được:**

1. [**Tìm hiểu chung:**](#_Toc9690)
2. **Tìm hiểu tổng quan về học máy:**

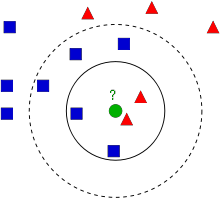
Học máy ( machine learning) là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể. Các thuật toán học máy xây dựng một mô hình dựa trên dữ liệu mẫu, được gọi là dữ liệu huấn luyện, để đưa ra dự đoán hoặc quyết định mà không cần được lập trình chi tiết về việc đưa ra dự đoán hoặc quyết định này.

* 1. **Phân loại:**
* 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.
  1. **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.
  1. **Ứng dụng của học máy:**
* 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
  1. **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.

1. **Tìm hiểu về thuật toán K-Nearest Neigbor(KNN):**



2.1. **Khái niệm về thuật toán 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.

* 1. **Nguyên lý hoạt động của thuật toán 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:

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.

* 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.

* 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.

* 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 và nhược điểm của thuật toán KNN:**

**a. 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.

**b. 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.
  1. **Các loại khoảng cách 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)

* 1. **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.
  1. **Các phương pháp xử lý dữ liệu trước khi sử dụng thuật toán KNN:**

**a. 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.

**b. 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.

**c. 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.

**d. 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.
  1. **Áp dụng KNN cho bài toán phân loại và bài toán hồi quy:**
     1. **Áp dụng 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.

1. **Cách hoạt động của thuật toán KNN trong bài toán phân loại:**

* 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.

1. **Ưu nhược điểm của thuật toán KNN cho bài toán phân loại:**

* 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

1. **Các ví dụ về áp dụng thuật toán KNN cho bài toán phân loại**

Ví dụ 1:

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 2:

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.

* + 1. **Áp 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.

1. **Cách hoạt động của thuật toán KNN trong bài toán hồi quy:**

* Tính khoảng cách giữa điểm dữ liệu mới x và các điểm dữ liệu trong tập huấn luyện.
* Chọn K điểm dữ liệu gần nhất với x.
* 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.

1. **Ưu nhược điểm của thuật toán KNN cho bài toán hồi quy:**

* Ưu điểm
* 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
* 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.

1. **Các ví dụ về áp dụng thuật toán KNN cho bài toán hồi quy:**

Giả sử chúng ta có tập dữ liệu sau:

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

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

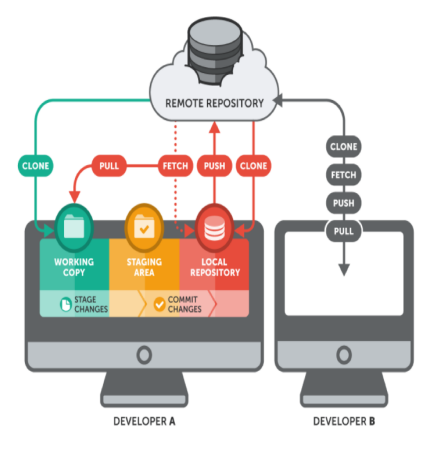
y\_hat = 2.5

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.

1. **Tìm hiểu tổng quan về Github và Git :**
   1. **Tìm hiểu tổng quan về 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.



* **Hệ thống kiểm soát phiên bản Git**: GitHub sử dụng Git, một hệ thống quản lý phiên bản phân tán, để theo dõi lịch sử thay đổi trong mã nguồn. Git cho phép bạn quản lý lịch sử thay đổi, tạo branches để làm việc đồng thời trên nhiều tính năng và hợp nhất các thay đổi.
* **Tài khoản cá nhân và tổ chức**: GitHub cung cấp cả tài khoản cá nhân và tổ chức. Tài khoản cá nhân được sử dụng cho các dự án cá nhân và tổ chức dành cho các tổ chức hoặc dự án có nhiều thành viên.
* **Hợp tác và kiểm tra mã:** GitHub cung cấp nhiều tính năng cho hợp tác trong phát triển phần mềm. Bạn có thể tạo issues để theo dõi công việc, thực hiện kiểm tra mã (code review), và sử dụng tính năng ghép nối (merge) để hợp nhất các thay đổi.
* **Dự án mã nguồn mở**: GitHub là một trung tâm cho dự án mã nguồn mở, nơi cộng đồng nguồn mở có thể đóng góp vào các dự án, báo cáo lỗi và xây dựng phần mềm cùng nhau.

* **Dịch vụ tích hợp và quản lý dự án**: GitHub cung cấp dịch vụ tích hợp với nhiều công cụ và dịch vụ khác nhau, bao gồm CI/CD, kiểm tra mã tự động và quản lý dự án. Điều này giúp tối ưu hóa quá trình phát triển.
* **Cộng đồng lớn**: GitHub có một cộng đồng lớn và đa dạng của các nhà phát triển và tổ chức trên khắp thế giới. Điều này tạo ra một môi trường tương tác và học hỏi.
* **Phí và miễn phí**: GitHub cung cấp tài khoản trả phí cho mã nguồn riêng tư và tài khoản miễn phí cho các dự án mã nguồn mở, đồng thời có giá trị cho các tổ chức và doanh nghiệp lớn.
  1. **Tìm hiểu tổng quan về Git:**

Git là một hệ thống kiểm soát phiên bản phân tán code nguồn mở. Nó được thiết kế để xử lý các dự án từ nhỏ đến lớn với tốc độ và hiệu quả cao. Nó được phát triển để điều phối công việc giữa các developer. Kiểm soát phiên bản cho phép chúng ta theo dõi và làm việc cùng với các thành viên trong nhóm của chúng ta tại cùng một không gian làm việc.

* + 1. **Tại sao phải sử dụng Git:**
* Git dễ sử dụng, an toàn và nhanh chóng.
* Có thể giúp quy trình làm việc code theo nhóm hoặc cá nhân đơn giản hơn rất nhiều bằng việc kết hợp các phân nhánh (branch).
* Bạn có thể làm việc ở bất cứ đâu vì chỉ cần clone mã nguồn từ kho chứa hoặc clone một phiên bản thay đổi nào đó từ kho chứa, hoặc một nhánh nào đó từ kho chứa.
* Dễ dàng trong việc deployment sản phẩm.
  + 1. **Các lệnh git cơ bản:**
* 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

1. [**Phát biểu về đề tài chẩn đoán bệnh tim:**](#_Toc9049)
2. **Ý tưởng:**

Đề tài chẩn đoán bệnh tim bằng thuật toán K Nearest Neighbor. Đây là một thử nghiệm về thuật toán KNN áp dụng trong lĩnh vực ý tế cụ thể là dự đoán một người mắc bệnh về tim. Ý tưởng của đề tài này là sử dụng các dữ liệu về các triệu chứng của một người mắc vấn đề về tim mạch để xác định một người có mắc bệnh về tim hay không. Ý tưởng trên nhằm giúp người dùng có thể thuận lợi hơn trong việc theo dõi tình hình sức khỏe của bản thân.

1. **Cách thức thực hiện:**

Bước 1: Chuẩn bị Dữ liệu

* Tìm hoặc thu thập dữ liệu về các trường hợp đã được chẩn đoán bệnh tim. Dữ liệu này cần bao gồm các đặc trưng (features) như huyết áp, lượng cholesterol, nhịp tim, và các thông tin y tế khác.
* Phân chia dữ liệu thành hai phần: tập dữ liệu huấn luyện (training set) và tập dữ liệu kiểm thử (test set). Thông thường, bạn sẽ sử dụng khoảng 70-80% dữ liệu cho việc huấn luyện và 20-30% để kiểm thử.

Bước 2: Tiền Xử lý Dữ liệu

* Kiểm tra dữ liệu để đảm bảo không có giá trị thiếu hoặc ngoại lệ.
* Tiêu chuẩn hóa dữ liệu nếu cần thiết để đảm bảo rằng mọi đặc trưng có cùng phạm vi giá trị.

Bước 3: Thực Hiện Thuật Toán KNN

* Lựa chọn số lân cận k (k-neighbors) cho thuật toán KNN. Điều này có thể được thực hiện thông qua quá trình đánh giá hoặc sử dụng một giá trị mặc định.
* Sử dụng thư viện machine learning, ví dụ như scikit-learn trong Python, để triển khai thuật toán KNN. Tạo một mô hình KNN và huấn luyện nó trên tập dữ liệu huấn luyện.

Bước 4: Đánh Giá Mô Hình

* Sử dụng tập dữ liệu kiểm thử để đánh giá hiệu suất của mô hình. Đánh giá có thể bao gồm việc tính toán độ chính xác, độ nhạy (recall), độ chính xác dương tính (precision), và F1 score.
* Hiểu rõ về các yếu tố đánh giá mô hình giúp bạn hiểu rõ hơn về khả năng chẩn đoán của thuật toán KNN trên bệnh tim.

Bước 5: Thực Hiện Chẩn Đoán

* Sử dụng mô hình đã huấn luyện để chẩn đoán bệnh tim trên các dữ liệu mới.
* Nhập dữ liệu mới, chạy mô hình, và xem kết quả chẩn đoán.

Bước 6: Tối Ưu Hóa Mô Hình (Tùy Chọn)

* Nếu kết quả không đạt được như mong đợi, bạn có thể tối ưu hóa mô hình bằng cách thử nghiệm với các giá trị khác nhau cho k hoặc điều chỉnh các tham số khác của thuật toán.

Bước 7: Hiển Thị Kết Quả

* Hiển thị kết quả của chẩn đoán, có thể là một thông báo đơn giản hoặc một biểu đồ minh họa.

1. **Dữ liệu sử dụng:**

* Dữ liệu sử dụng trong đề tài trên được lấy từ thông số của những người mắc bệnh tim dể có thể chẩn đoán chính xác nhất.
* Dữ liệu bao gồm những thông như: tuổi, cholesterol, nhịp tim …

1. **Demo chẩn đoán bệnh tim bằng thuật toán KNN:**
   1. **Ngôn ngữ được sử dụng:**

Ngôn ngữ được sử dụng trong đề tài trên là: C++

* 1. **Quản lý code:**

GitHub là nền tảng quản lý mã nguồn mạnh mẽ, đóng vai trò quan trọng trong sự hợp tác của nhóm em. Với quản lý nhánh linh hoạt và tính năng theo dõi vấn đề, các thành viên trong nhóm dễ dàng theo kịp sự thay đổi của dự án, giúp mỗi thành viên đóng góp một cách hiệu quả.

Khám phá đề tài chẩn đoán bệnh tim bằng thuật toán KNN trên:

<https://github.com/LATxBONA/demo_KNN_Diagnostic_CNTT44D>

**3. Code demo và Chi tiết các hàm được sử dụng:**

**A screen shot of a computer program

Description automatically generated**

* Khai báo thư viện
* Tạo một struct Patient để lưu các thông tin của một bệnh nhân bao gồm tuổi, cholesterol, Nhịp tim và kết quả chẩn đoán

**A computer screen shot of text

Description automatically generated**

**Hàm KhoangCachEuclidean:**

* **Chức năng:** Tính toán khoảng cách Euclidean giữa hai bệnh nhân.
* **Đối số:**
  + **p1** và **p2**: Hai đối tượng **Patient** đại diện cho thông tin về hai bệnh nhân.
* **Trả về:** Khoảng cách Euclidean giữa **p1** và **p2**.

**Hàm DuDoanKNN:**

* **Chức năng:** Dự đoán kết quả chẩn đoán cho một bệnh nhân mới sử dụng thuật toán KNN.
* **Đối số:**
  + **patients**: Một vector chứa thông tin về các bệnh nhân từ tập dữ liệu huấn luyện.
  + **newPatient**: Đối tượng **Patient** đại diện cho thông tin của bệnh nhân mới cần chẩn đoán.
  + **k**: Số lân cận gần nhất được sử dụng trong thuật toán KNN.
* **Trả về:** Kết quả chẩn đoán cho bệnh nhân mới (0 hoặc 1, tùy thuộc vào kết quả của thuật toán KNN).

Hàm main trong demo chẩn đoán bệnh tim bằng thuật toán KNN.

**A screen shot of a computer program

Description automatically generated**

**Đọc Dữ Liệu:** Trong phần **main**, có một đoạn mã sử dụng **ifstream** để mở và đọc dữ liệu từ một tệp CSV ("heart\_disease\_data.csv" trong trường hợp này). Dữ liệu này chứa thông tin về các bệnh nhân, bao gồm tuổi, mức cholesterol, tốc độ tim tối đa và kết quả chẩn đoán.

****

**Chuẩn bị Dữ Liệu Mới:** Sau đó, có một đoạn mã tạo một đối tượng **Patient** mới để biểu diễn thông tin của một bệnh nhân mới cần được chẩn đoán. Dữ liệu của bệnh nhân mới này sẽ được chèn vào trong đoạn mã.



**Gọi Hàm Dự Đoán KNN:** Tiếp theo, hàm **predictKNN** được gọi để dự đoán kết quả chẩn đoán cho bệnh nhân mới dựa trên thuật toán KNN.



**Xuất Kết Quả:** Cuối cùng, kết quả chẩn đoán được in ra màn hình dựa trên giá trị trả về từ hàm **predictKNN**

**A black background with white text

Description automatically generated**

1. **Đánh giá và kết quả đạt được:**

* Trong bài báo cáo này, chúng ta đã có cơ hội khám phá và trải nghiệm sự tiềm năng của việc sử dụng thuật toán K-Nearest Neighbors (KNN) trong việc demo chẩn đoán bệnh tim. Dù chỉ là một demo và vẫn còn nhiều sai sót, nhưng hành trình này đã mang lại những nhận thức quý báu và đánh thức sự sáng tạo trong lĩnh vực y học.
* Demo chẩn đoán bệnh tim bằng KNN không chỉ là việc đưa ra các dự đoán, mà còn là một hành trình giáo dục cho chúng ta về sức mạnh và hạn chế của thuật toán. Những sai sót mà chúng ta gặp phải không chỉ là thách thức, mà còn là cơ hội để học hỏi và cải thiện.
* Cuối cùng, qua demo này, chúng ta đã thấy rõ rằng sự kết hợp giữa công nghệ và y học không chỉ giúp chúng ta hiểu biết sâu rộng hơn về bệnh lý, mà còn mở ra những cánh cửa mới cho việc đổi mới trong chẩn đoán và điều trị. Dù chỉ là một bước nhỏ, demo của chúng ta có thể đánh dấu một khởi đầu mới trong sự phát triển của lĩnh vực này.
* Trước khi chúng ta kết thúc, hãy nhìn xa hơn nữa và nhận ra rằng mỗi sai sót là một bài học, và mỗi demo là một cơ hội để phát triển. Chúng ta đang đứng trước một thế giới nơi những ý tưởng đột phá có thể nảy sinh từ những nỗ lực nhỏ. Với sự học hỏi từ demo chẩn đoán bệnh tim bằng KNN này, chúng ta có thể tiếp tục xây dựng những cầu nối giữa y học và công nghệ để cùng nhau đưa ngành y học lên một tầm cao mới, mang lại lợi ích to lớn cho sức khỏe cộng đồng.