AI VIET NAM
@aivietnam.edu.vn

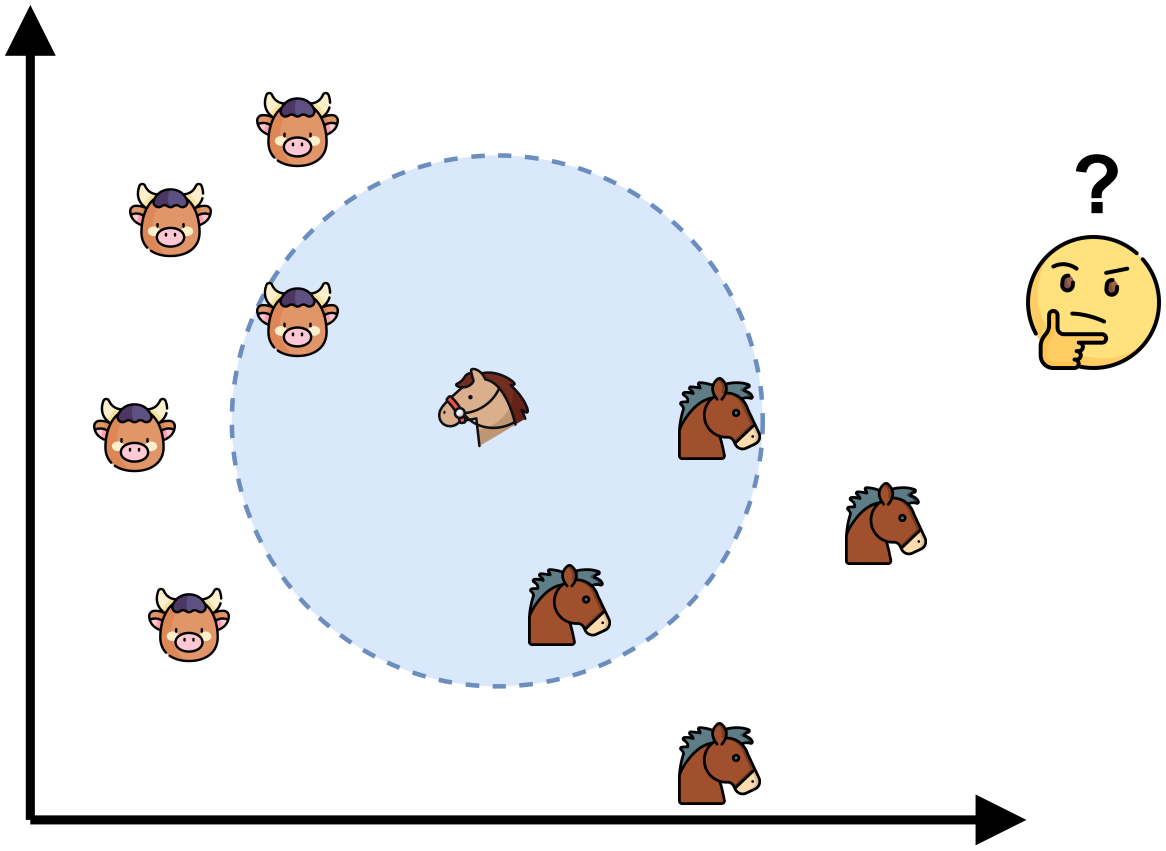
AI VIET NAM – AI COURSE 2025

Step-by-Step: k-Nearest Neighbors

Dương Trường Bình, Dương Đình Thắng, Đinh Quang Vinh

I. Dẫn nhập

Ngạn ngữ có câu *"Ngưu tầm ngưu, mã tầm mã (Birds of a feather flock together)"*, hay một câu nói quen thuộc khác: *"Hãy cho tôi biết bạn của bạn là ai, tôi sẽ cho bạn biết bạn là người như thế nào"*. Thật vậy, trong cuộc sống, chúng ta thường có xu hướng đưa ra đánh giá hoặc nhận định dựa trên một nguyên tắc rất tự nhiên: những gì giống nhau thường có xu hướng ở gần nhau. Chính cách suy luận quen thuộc này, tưởng chừng chỉ thuộc về đời sống, lại vô tình phản ánh triết lý cốt lõi của một trong những thuật toán học máy trực quan và dễ hiểu nhất: **k-Nearest Neighbors (KNN)**.



Hình 1: Minh họa cho câu ngạn ngữ.

Theo đó, ý tưởng của KNN vô cùng đơn giản nhưng lại rất trực quan: để xác định bản chất của một đối tượng mới, chúng ta chỉ cần quan sát những "người hàng xóm" gần nhất của nó trong một không gian đã biết và đưa ra dự đoán dựa trên thông tin từ họ. Chẳng hạn, nếu đa số hàng xóm của một email mới là "Spam", khả năng cao email đó cũng sẽ là "Spam". Tương tự, nếu những ngôi nhà có cùng diện tích, vị trí và số phòng ngủ đều có mức giá nhất định, thì một ngôi nhà mới với các đặc điểm tương tự cũng sẽ có giá xấp xỉ như vậy. Cách suy luận này rất gần gũi với cách con người chúng ta đưa ra phán đoán trong đời sống hằng ngày: quan sát môi trường xung quanh và dựa trên các mẫu quen thuộc để đi đến kết luận.

Được giới thiệu lần đầu trong một báo cáo kỹ thuật vào năm 1951 của Trường Y Hàng không Không quân Hoa Kỳ, thuật toán KNN do Evelyn Fix và Joseph Hodges phát triển không cố gắng "học" một mô hình phức tạp từ dữ liệu như các thuật toán học máy khác. Thay vào đó, nó chỉ đơn giản ghi nhớ toàn bộ tập dữ liệu huấn luyện và trì hoãn mọi tính toán quan trọng cho đến khi có yêu cầu dự đoán. Chính vì sự "lười biếng" trong việc xây dựng mô hình trước này mà KNN được xếp vào nhóm các thuật toán *lazy learning*. Nhờ đặc điểm này, KNN vừa dễ triển khai vừa có thể thích ứng tốt khi dữ liệu huấn luyện thay đổi hoặc được mở rộng.

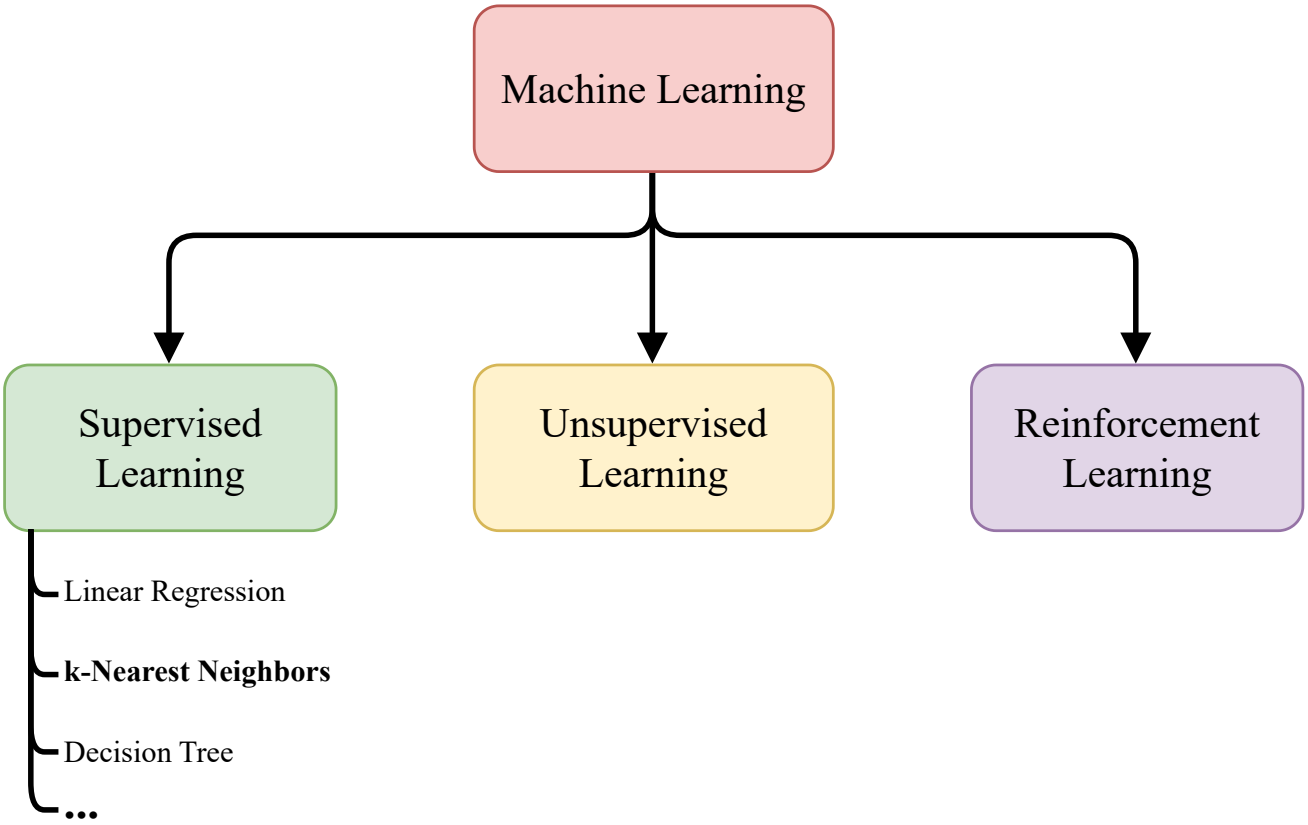
Trong bài viết này, chúng ta sẽ cùng "giải phẫu" thuật toán KNN một cách chi tiết: tìm hiểu cách nó định nghĩa khái niệm "gần gũi", cơ chế mà nó "tham khảo ý kiến" các láng giềng, cũng như cách nó đưa ra quyết định cuối cùng cho cả hai dạng bài toán phổ biến trong học máy: phân loại và hồi quy.

Mục lục

II. Thuật toán k-Nearest Neighbors

II.1. Tổng quan

Theo lý thuyết, k-Nearest Neighbors (KNN) là một trong những thuật toán cơ bản trong học máy. Như được minh họa trong Hình 2, KNN thuộc nhóm thuật toán **supervised learning** và còn được đặc trưng bởi hai tính chất quan trọng khác là **non-parametric** và **instance-based**. Các tính chất này có thể được hiểu cụ thể như sau:



Hình 2: Sơ đồ phân loại các thuật toán học máy và vị trí của KNN.

- **Supervised Learning (tạm dịch: học có giám sát):** Đây là đặc điểm cơ bản nhất. KNN yêu cầu một tập dữ liệu huấn luyện mà trong đó mỗi điểm dữ liệu (đầu vào) đã được gán sẵn một "nhãn" hoặc "giá trị" (đầu ra) chính xác. Mục tiêu của thuật toán là học cách dự đoán đầu ra cho các điểm dữ liệu mới chưa từng thấy dựa trên mối liên hệ đã học được từ dữ liệu huấn luyện.
- **Non-parametric (tạm dịch: Phi tham số):** Thuật ngữ này không có nghĩa là mô hình không có tham số nào, mà là nó không đưa ra bất kỳ giả định nào về dạng phân phối của dữ liệu (ví dụ: không giả định dữ liệu tuân theo phân phối chuẩn hay có mối quan hệ tuyến tính). Mô hình không bị giới hạn bởi một số lượng tham số cố định. Thay vào đó, độ phức tạp của mô hình (chính là toàn bộ tập dữ liệu) sẽ tăng lên khi chúng ta cung cấp thêm dữ

liệu. Điều này giúp KNN rất linh hoạt và có thể học được các ranh giới quyết định phức tạp.

- **Instance-based / Lazy Learning (tạm dịch: Dựa trên thực thể / Học lười):** Đây là đặc điểm độc đáo nhất của KNN. Không giống như các thuật toán khác (như hồi quy tuyến tính, mạng nơ-ron) cố gắng xây dựng một hàm tổng quát trong giai đoạn huấn luyện, KNN không có giai đoạn huấn luyện thực sự. Nó chỉ đơn giản là **lưu trữ toàn bộ tập dữ liệu huấn luyện**. Quá trình tính toán chính, bao gồm việc tìm kiếm láng giềng và dự đoán, bị trì hoãn cho đến khi có một yêu cầu dự đoán mới.

Nhờ vào sự linh hoạt này, KNN có thể được áp dụng hiệu quả cho cả hai loại bài toán phổ biến trong học máy:

- **Classification (tạm dịch: Phân loại):** Dự đoán một nhãn rời rạc cho một đối tượng (ví dụ: "Spam" hay "Not Spam", "Mèo" hay "Chó").
- **Regression (tạm dịch: Hồi quy):** Dự đoán một giá trị liên tục (ví dụ: giá nhà, nhiệt độ, doanh thu).

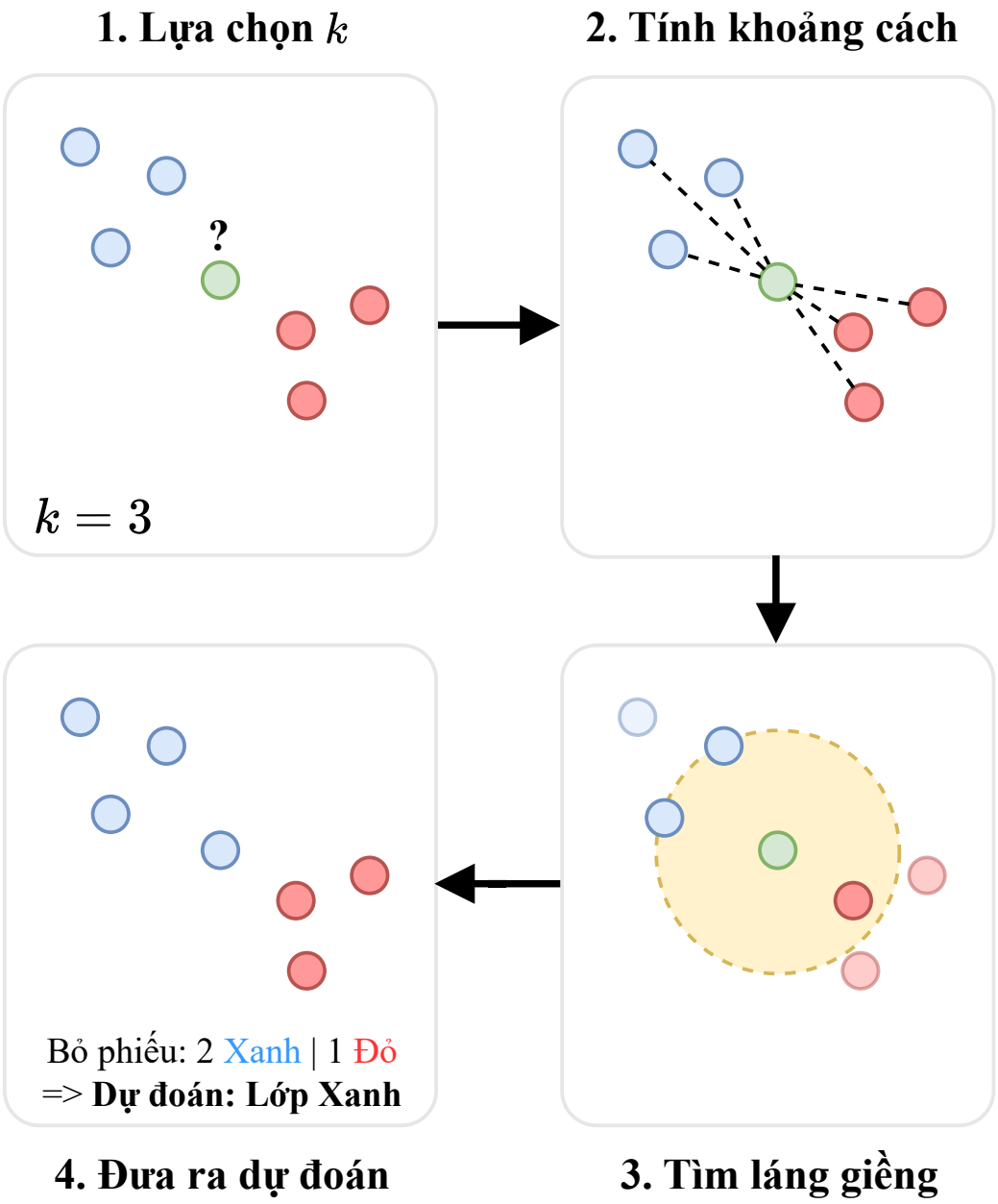
Để đảm bảo tính nhất quán và rõ ràng, bảng dưới đây tổng hợp các ký hiệu toán học chính sẽ được sử dụng xuyên suốt bài viết.

Bảng Ký hiệu Toán học

Ký hiệu	Ý nghĩa
k	Siêu tham số, là số lượng láng giềng gần nhất được xem xét để đưa ra dự đoán.
N	Tổng số điểm dữ liệu trong tập huấn luyện.
n	Số chiều (tức là số đặc trưng) của dữ liệu.
$\mathbf{p}, \mathbf{q}$	Các điểm dữ liệu (vector) trong không gian đặc trưng. Ví dụ: $\mathbf{p} = (p_1, p_2, \ldots, p_n)$.
p_i, q_i	Giá trị của đặc trưng thứ i của điểm dữ liệu tương ứng $\mathbf{p}$ và $\mathbf{q}$.
$\mathbf{p}_{new}$	Điểm dữ liệu mới cần được phân loại hoặc dự đoán giá trị.
$d(\mathbf{p}, \mathbf{q})$	Ký hiệu cho hàm khoảng cách giữa hai điểm dữ liệu $\mathbf{p}$ và $\mathbf{q}$.
y_i	Nhãn (bài toán phân loại) hoặc giá trị mục tiêu (bài toán hồi quy) của điểm dữ liệu huấn luyện thứ i.
$\hat{y}_{new}$	Nhãn hoặc giá trị được dự đoán cho điểm dữ liệu mới $\mathbf{p}_{new}$.
w_i	Trọng số của láng giềng thứ i trong thuật toán KNN có trọng số, thường được tính là $w_i = 1/d_i$.

II.2. Cách hoạt động

Để dự đoán cho một điểm dữ liệu mới, thuật toán không cần trải qua một quá trình "huấn luyện" phức tạp để học một hàm số. Thay vào đó, nó chỉ đơn giản là ghi nhớ toàn bộ tập dữ liệu huấn luyện và thực hiện các bước tính toán khi có yêu cầu dự đoán. Quá trình này có thể được mô tả qua 4 bước chính như trong Hình 1.



Hình 3: Tổng quan các bước của thuật toán KNN.

Hãy cùng đi sâu vào từng bước để hiểu rõ cơ chế hoạt động của thuật toán.

Bước 1: Lựa chọn siêu tham số k

Bước đầu tiên và mang tính định hướng cho toàn bộ thuật toán là việc lựa chọn siêu tham số **k**. Theo định nghĩa ở bảng ký hiệu toán học, k chính là số lượng "hàng xóm" mà chúng ta sẽ tham khảo ý kiến để quyết định tính chất của điểm dữ liệu mới. Việc chọn k có ảnh hưởng trực tiếp đến độ phức tạp và khả năng tổng quát hóa của mô hình:

- **Khi k quá nhỏ (ví dụ, $k = 1$):** Mô hình trở nên cực kỳ linh hoạt. Ranh giới quyết định sẽ rất phức tạp, cố gắng phân loại chính xác từng điểm một. Điều này khiến mô hình rất nhạy cảm với nhiễu (noise) và các điểm dữ liệu ngoại lai (outliers). Hệ quả là mô hình có thể hoạt động hoàn hảo trên dữ liệu huấn luyện nhưng lại dự đoán kém trên dữ liệu mới, một hiện tượng gọi là **overfitting (tạm dịch: quá khớp)**.
- **Khi k quá lớn (ví dụ, $k = N$):** Mô hình trở nên quá "bảo thủ". Nó sẽ xem xét một vùng lân cận rất rộng, có thể chứa các điểm từ nhiều lớp khác nhau. Điều này làm cho ranh giới quyết định trở nên quá mượt mà, bỏ qua các chi tiết quan trọng và cấu trúc cục bộ của dữ liệu. Mô hình có thể dự đoán tất cả các điểm mới đều thuộc về lớp phổ biến nhất trong toàn bộ tập dữ liệu, dẫn đến hiện tượng **underfitting (tạm dịch: dưới khớp)**.

Vậy làm thế nào để chọn k tối ưu? Không có một công thức toán học nào cho giá trị k hoàn hảo. Tuy nhiên, có những phương pháp và quy tắc kinh nghiệm hiệu quả:

- **Cross-Validation:** Đây là phương pháp tiêu chuẩn và đáng tin cậy nhất. Ta chia tập huấn luyện ra thành nhiều phần (folds), sau đó thử nghiệm với nhiều giá trị k khác nhau. Với mỗi giá trị k, ta huấn luyện mô hình trên một số phần và đánh giá hiệu suất trên phần còn lại. Giá trị k nào cho lỗi trung bình thấp nhất trên các tập kiểm tra sẽ được chọn.
- **Rule of Thumb:** Một gợi ý thường được sử dụng là chọn $k = \sqrt{N}$, trong đó N là tổng số điểm trong tập dữ liệu huấn luyện.
- **Chọn k là số lẻ cho bài toán phân loại:** Đối với các bài toán phân loại nhị phân (hoặc có số lớp chẵn), việc chọn k là một số lẻ (3, 5, 7,...) giúp tránh trường hợp "hòa phiếu". Ví dụ, nếu $k = 4$ và có 2 hàng xóm thuộc lớp A và 2 hàng xóm thuộc lớp B, thuật toán sẽ không thể đưa ra quyết định rõ ràng.

Bước 2: Tính khoảng cách

Sau khi đã quyết định sẽ "hỏi ý kiến" bao nhiêu hàng xóm (k), chúng ta cần một phương pháp để xác định ai là "hàng xóm" gần nhất. "Sự gần gũi" trong KNN được định lượng hóa bằng các thước đo khoảng cách.

Lưu ý quan trọng

Trước khi tính khoảng cách, có một bước tiền xử lý bắt buộc là **chuẩn hóa dữ liệu**. Thuật toán KNN cực kỳ nhạy cảm với thang đo của các đặc trưng.

Hãy tưởng tượng ta dự đoán giá nhà dựa trên 2 đặc trưng: `diện tích` (đơn vị m^2, giá trị từ 30 đến 200) và `số phòng ngủ` (giá trị từ 1 đến 5). Khi tính khoảng cách, sự chênh lệch về giá trị của `diện tích` (ví dụ: $150 - 80 = 70$) sẽ hoàn toàn lấn át sự chênh lệch của `số phòng ngủ` (ví dụ: $3 - 2 = 1$). Điều này làm cho đặc trưng `số phòng ngủ` gần như vô dụng. Chuẩn hóa dữ liệu sẽ đưa tất cả các đặc trưng về cùng một thang đo, đảm bảo mỗi đặc trưng đều có cơ hội đóng góp một cách công bằng vào việc tính khoảng cách.

Hai phương pháp chuẩn hóa phổ biến nhất là:

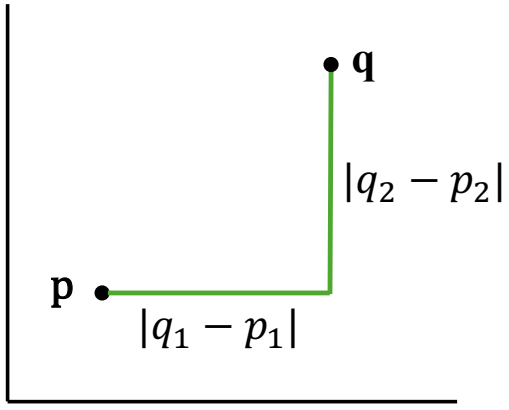
- **Min-Max Scaling (Normalization):** Đưa tất cả giá trị của một đặc trưng về một khoảng $[0, 1]$. Công thức:
$$X_{\text{scaled}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}$$
Phương pháp này hiệu quả nhưng nhạy cảm với các giá trị ngoại lai (outliers).

- **Standardization (Z-score Normalization):** Biến đổi dữ liệu sao cho đặc trưng có giá trị trung bình (μ) là 0 và độ lệch chuẩn (σ) là 1. Công thức:
$$X_{\text{scaled}} = \frac{X - \mu}{\sigma}$$
Phương pháp này ít bị ảnh hưởng bởi outliers và thường được ưu tiên sử dụng trong hầu hết các trường hợp.

Các thước đo khoảng cách phổ biến Sau khi dữ liệu đã được chuẩn hóa, ta có thể áp dụng các công thức đo khoảng cách. Cho hai điểm dữ liệu $\mathbf{p} = (p_1, p_2, ..., p_n)$ và $\mathbf{q} = (q_1, q_2, ..., q_n)$ trong không gian đặc trưng n chiều:

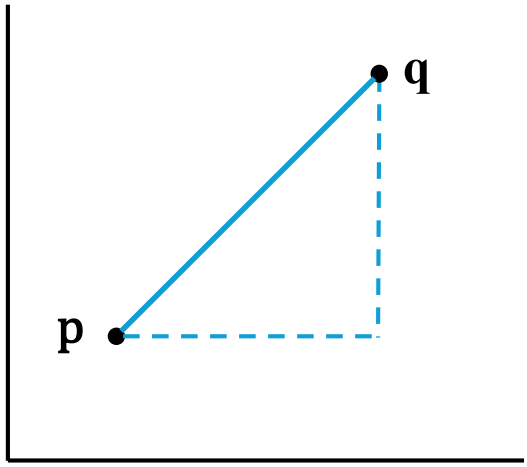
- **Khoảng cách Manhattan (L_1):** Là tổng khoảng cách di chuyển theo từng trục tọa độ, giống như đi qua các tòa nhà trong một thành phố.

$$d(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^{n} |q_i - p_i|$$



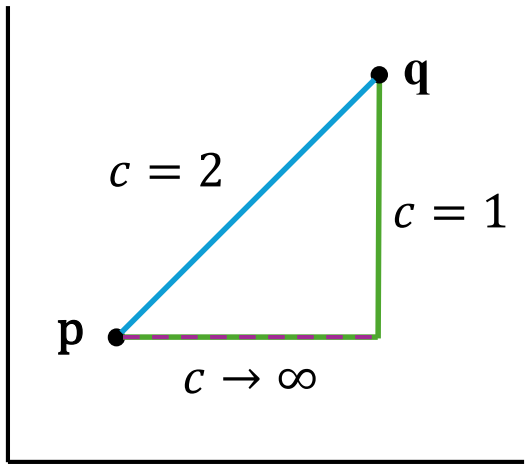
- **Khoảng cách Euclidean (L_2):** Là khoảng cách "đường chim bay" thẳng nhất giữa hai điểm. Đây là lựa chọn mặc định và phổ biến nhất.

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$



- **Khoảng cách Minkowski (L_c):** Là dạng tổng quát của hai loại trên, với tham số c (hay p). Khi $c = 1$, nó trở thành khoảng cách Manhattan. Khi $c = 2$, nó trở thành khoảng cách Euclidean.

$$d(\mathbf{p}, \mathbf{q}) = \left(\sum_{i=1}^{n}|q_i - p_i|^c\right)^{1/c}$$



Sau khi chuẩn hóa dữ liệu và lựa chọn được thước đo phù hợp, thuật toán sẽ tính khoảng cách từ điểm dữ liệu mới đến mọi điểm trong tập huấn luyện. Kết quả thu được là một danh sách đầy đủ các khoảng cách.

Bước 3: Tìm k láng giềng gần nhất

Ở bước này, chúng ta sẽ lần lượt:

1. Sắp xếp các khoảng cách đã tính ở trên theo thứ tự từ nhỏ đến lớn.
2. Chọn ra **k** điểm dữ liệu ứng với **k** khoảng cách nhỏ nhất. Những điểm này chính là "hội đồng" láng giềng sẽ đưa ra quyết định cuối cùng.

Bước 4: Ra quyết định - Dự đoán kết quả

Cách "hội đồng" k láng giềng đưa ra quyết định phụ thuộc vào bản chất của bài toán.

Đối với bài toán Phân loại - Lấy ý kiến số đông (Majority Vote):

- **Nguyên tắc:** Điểm dữ liệu mới sẽ được gán cho lớp nào chiếm **đa số** trong số k láng giềng gần nhất.
- **Ví dụ:** Ta đang phân loại một email là "Spam" hay "Not Spam" với $k = 5$. Sau khi tìm kiếm, 5 email gần nhất có nhãn là: [Spam, Not Spam, Spam, Spam, Not Spam]. Ta đếm: có 3 phiếu cho "Spam" và 2 phiếu cho "Not Spam". Vì "Spam" chiếm đa số, email mới sẽ được dự đoán là "Spam".
- **Cải tiến:** Một cách tinh vi hơn là không coi phiếu bầu của mọi láng giềng có giá trị như nhau. Láng giềng càng gần thì ý kiến càng có trọng lượng. Một phương pháp phổ biến là lấy trọng số bằng nghịch đảo của khoảng cách ($w = 1/d$). Láng giềng ở rất gần (d nhỏ) sẽ có trọng số w lớn, do đó có ảnh hưởng mạnh mẽ hơn đến kết quả cuối cùng.

Đối với bài toán Hồi quy - Lấy giá trị trung bình (Averaging):

- **Nguyên tắc:** Giá trị dự đoán cho điểm dữ liệu mới là **giá trị trung bình** của các giá trị mục tiêu của k láng giềng gần nhất.
- **Ví dụ:** Ta đang dự đoán giá một ngôi nhà với $k = 3$. Ba ngôi nhà tương đồng nhất trong tập dữ liệu có giá lần lượt là 2 tỷ, 2.2 tỷ và 2.4 tỷ. Giá dự đoán cho ngôi nhà mới sẽ là trung bình cộng: $(2 + 2.2 + 2.4)/3 = 2.2$ tỷ.
- **Cải tiến:** Tương tự như phân loại, ta có thể tính trung bình có trọng số. Những ngôi nhà ở gần hơn (tương đồng hơn) sẽ có ảnh hưởng lớn hơn đến giá dự đoán. Công thức có thể là:

$$\hat{y}_{new} = \frac{\sum_{i=1}^{k} w_i y_i}{\sum_{i=1}^{k} w_i} \quad \text{với} \quad w_i = \frac{1}{d(\mathbf{p}_{new}, \mathbf{p}_i)}$$

  Trong đó, y_i là giá trị mục tiêu của láng giềng thứ i, w_i là trọng số, $d(\mathbf{p}_{new}, \mathbf{p}_i)$ là khoảng cách từ điểm mới đến láng giềng thứ i, và $\hat{y}_{new}$ là giá trị được dự đoán.

III. Thực hành

Trong phần này, để hiểu rõ hơn về cơ chế hoạt động của thuật toán, chúng ta ứng dụng KNN thực hiện tính toán từng bước và minh họa trên một ví dụ cụ thể nhằm dự đoán kết quả học tập của một sinh viên mới.

Bối cảnh bài toán

Giả sử chúng ta có một tập dữ liệu nhỏ về kết quả học tập của 5 sinh viên dựa trên số giờ học và số giờ ngủ mỗi ngày.

Bảng 1: Dữ liệu huấn luyện về kết quả học tập của sinh viên.

Sinh viên	**Giờ học** (X_1)	**Giờ ngủ** (X_2)	**Kết quả** (y_{cls})	**Điểm thi** (y_{reg})
$\mathbf{p}_1$	2	8	Trượt	4.5
$\mathbf{p}_2$	3	7	Trượt	5.5
$\mathbf{p}_3$	5	6	Đậu	7.5
$\mathbf{p}_4$	6	5	Đậu	8.5
$\mathbf{p}_5$	4	6	Đậu	7.0

Bài toán: Có một sinh viên mới, ký hiệu là $\mathbf{p}_{new}$, có số liệu như sau:

- **Giờ học = 4**
- **Giờ ngủ = 7**

Dựa vào các sinh viên đã biết, chúng ta hãy cùng dự đoán:

1. Sinh viên $\mathbf{p}_{new}$ này sẽ "Đậu" hay "Trượt" (bài toán phân loại)?
2. Điểm thi của sinh viên $\mathbf{p}_{new}$ sẽ là bao nhiêu (bài toán hồi quy)?

Chúng ta sẽ thực hiện quy trình KNN từng bước để trả lời cả hai câu hỏi này. Ba bước đầu tiên (chọn K, tính khoảng cách, tìm láng giềng) sẽ được dùng chung cho cả hai bài toán.

Bước 1: Lựa chọn siêu tham số k

Đây là bước đầu tiên. Với một tập dữ liệu nhỏ gồm 5 điểm, việc chọn $k = 1$ có thể quá nhạy cảm, trong khi $k = 5$ sẽ lấy tất cả các điểm. Một lựa chọn hợp lý là $\mathbf{k = 3}$. Đây cũng là một số lẻ, giúp chúng ta tránh được trường hợp hòa phiếu giữa hai lớp "Đậu" và "Trượt" trong bài toán phân loại.

Lựa chọn: $k = 3$

Bước 2: Tính khoảng cách

Tiếp theo, chúng ta cần tính khoảng cách từ điểm dữ liệu mới $\mathbf{p}_{new}(4, 7)$ đến **tất cả** các điểm trong tập huấn luyện.

Lựa chọn thước đo khoảng cách: Chúng ta sẽ sử dụng thước đo phổ biến nhất là **Khoảng cách Euclidean (L_2)**. Với hai điểm $\mathbf{p} = (p_1, p_2)$ và $\mathbf{q} = (q_1, q_2)$:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}$$

Lưu ý về chuẩn hóa dữ liệu: Như đã đề cập ở phần lý thuyết, chuẩn hóa dữ liệu là bước cực kỳ quan trọng. Tuy nhiên, trong ví dụ đơn giản này, đơn vị của "Giờ học" và "Giờ ngủ" đều là giờ, nên chúng ta sẽ tạm thời bỏ qua bước chuẩn hóa.

Bắt đầu tính toán:

- **Khoảng cách từ $\mathbf{p}_{new}(4, 7)$ đến $\mathbf{p}_1(2, 8)$:**

$$d(\mathbf{p}_{new}, \mathbf{p}_1) = \sqrt{(4-2)^2 + (7-8)^2} = \sqrt{2^2 + (-1)^2} = \sqrt{4+1} = \sqrt{5} \approx 2.24$$

- **Khoảng cách từ $\mathbf{p}_{new}(4, 7)$ đến $\mathbf{p}_2(3, 7)$:**

$$d(\mathbf{p}_{new}, \mathbf{p}_2) = \sqrt{(4-3)^2 + (7-7)^2} = \sqrt{1^2 + 0^2} = \sqrt{1} = 1.00$$

- **Khoảng cách từ $\mathbf{p}_{new}(4, 7)$ đến $\mathbf{p}_3(5, 6)$:**

$$d(\mathbf{p}_{new}, \mathbf{p}_3) = \sqrt{(4-5)^2 + (7-6)^2} = \sqrt{(-1)^2 + 1^2} = \sqrt{1+1} = \sqrt{2} \approx 1.41$$

- **Khoảng cách từ $\mathbf{p}_{new}(4, 7)$ đến $\mathbf{p}_4(6, 5)$:**

$$d(\mathbf{p}_{new}, \mathbf{p}_4) = \sqrt{(4-6)^2 + (7-5)^2} = \sqrt{(-2)^2 + 2^2} = \sqrt{4+4} = \sqrt{8} \approx 2.83$$

- **Khoảng cách từ $\mathbf{p}_{new}(4, 7)$ đến $\mathbf{p}_5(4, 6)$:**

$$d(\mathbf{p}_{new}, \mathbf{p}_5) = \sqrt{(4-4)^2 + (7-6)^2} = \sqrt{0^2 + 1^2} = \sqrt{1} = 1.00$$

Bước 3: Tìm k láng giềng gần nhất

Bây giờ, chúng ta sẽ tập hợp các khoảng cách vừa tính và sắp xếp chúng theo thứ tự từ nhỏ đến lớn để tìm ra 3 "hàng xóm" gần nhất.

Bảng 2: Sắp xếp khoảng cách từ $\mathbf{p}_{new}$ đến các điểm huấn luyện.

Điểm dữ liệu	Khoảng cách Euclidean	Kết quả	Điểm thi
$\mathbf{p}_2$	1.00	**Trượt**	5.5
$\mathbf{p}_5$	1.00	**Đậu**	7.0
$\mathbf{p}_3$	1.41	**Đậu**	7.5
$\mathbf{p}_1$	2.24	Trượt	4.5
$\mathbf{p}_4$	2.83	Đậu	8.5

Dựa trên bảng đã sắp xếp, 3 láng giềng gần nhất của $\mathbf{p}_{new}$ là: $\mathbf{p_2}, \mathbf{p_5}, \mathbf{p_3}$.

Bước 4: Ra quyết định - Dự đoán kết quả

Với 3 láng giềng đã xác định, chúng ta sẽ đưa ra dự đoán riêng cho từng bài toán.

Dự đoán cho bài toán Phân loại (Classification)

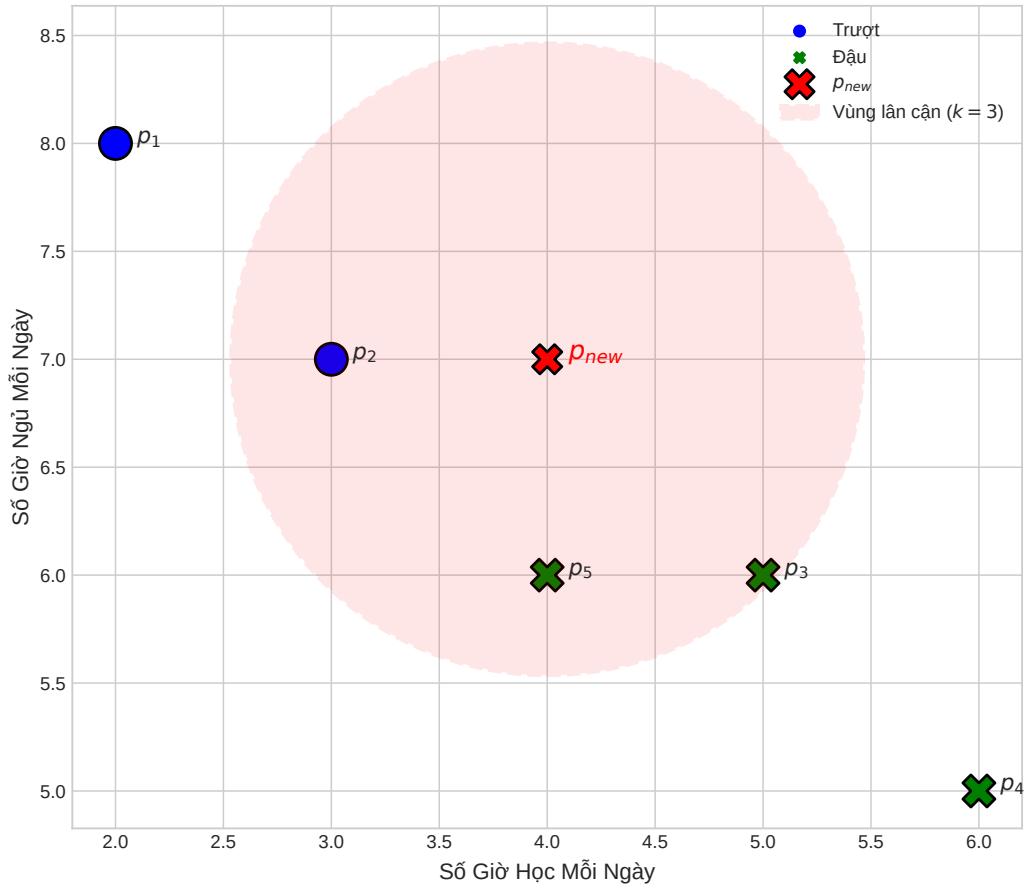
Chúng ta sẽ "tham khảo ý kiến" của 3 láng giềng gần nhất để quyết định nhãn cho $\mathbf{p}_{new}$. Vì đây là bài toán phân loại, chúng ta sẽ sử dụng phương pháp **biểu quyết theo đa số (majority vote)**.

- Láng giềng $\mathbf{p}_2$ có nhãn là: **Trượt**
- Láng giềng $\mathbf{p}_5$ có nhãn là: **Đậu**
- Láng giềng $\mathbf{p}_3$ có nhãn là: **Đậu**

Đếm số phiếu:

- Số phiếu cho "Đậu": 2
- Số phiếu cho "Trượt": 1

Kết luận: Lớp "Đậu" chiếm đa số phiếu. Do đó, thuật toán KNN dự đoán rằng sinh viên mới $\mathbf{p}_{new}$ sẽ **Đậu**.



Hình 4: Minh họa kết quả dự đoán phân loại cho sinh viên $\mathbf{p}_{new}$ bằng thuật toán KNN.

Dự đoán cho bài toán Hồi quy (Regression)

Sau khi dự đoán sinh viên mới sẽ "Đậu", chúng ta tiếp tục sử dụng các láng giềng đã tìm được để dự đoán điểm thi cụ thể ($\hat{y}_{reg}$) cho sinh viên này.

Phương pháp 1: Lấy trung bình (Averaging) Nguyên tắc là lấy trung bình cộng điểm thi của 3 láng giềng gần nhất.

- Điểm thi của $\mathbf{p}_2$: 5.5
- Điểm thi của $\mathbf{p}_5$: 7.0
- Điểm thi của $\mathbf{p}_3$: 7.5

Điểm dự đoán được tính như sau:

$$\hat{y}_{new} = \frac{5.5 + 7.0 + 7.5}{3} = \frac{20.0}{3} \approx 6.67$$

Phương pháp 2: Lấy trung bình có trọng số (Weighted Averaging) Đây là một cải tiến, trong đó láng giềng càng gần thì ý kiến (điểm số) càng có trọng lượng. Ta sử dụng trọng số $w_i = 1/d_i$.

- $\mathbf{p}_2$: $d_2 = 1.00 \implies w_2 = 1/1.00 = 1.0$
- $\mathbf{p}_5$: $d_5 = 1.00 \implies w_5 = 1/1.00 = 1.0$
- $\mathbf{p}_3$: $d_3 \approx 1.41 \implies w_3 = 1/1.41 \approx 0.709$

Điểm dự đoán được tính theo công thức trung bình có trọng số:

$$\hat{y}_{new} = \frac{\sum_{i=1}^{K} w_i y_i}{\sum_{i=1}^{K} w_i} = \frac{(1.0 \times 5.5) + (1.0 \times 7.0) + (0.709 \times 7.5)}{1.0 + 1.0 + 0.709}$$

$$\hat{y}_{new} = \frac{5.5 + 7.0 + 5.3175}{2.709} = \frac{17.8175}{2.709} \approx 6.58$$

Kết luận: Sử dụng phương pháp trung bình đơn giản, điểm thi dự đoán là **6.67**. Nếu dùng trung bình có trọng số để ưu tiên các láng giềng gần hơn, điểm thi dự đoán là **6.58**.
Ngoài ví dụ tính toán thủ công trên, để có thể trải nghiệm tương tác trực quan và sâu hơn, bạn có thể truy cập và thử nghiệm với nhiều bộ dữ liệu và tham số khác nhau tại **chương trình demo thuật toán KNN**.

IV. Câu hỏi trắc nghiệm

1. Thuật toán k-Nearest Neighbors (KNN) thuộc loại thuật toán học máy nào?

   (a) Học không giám sát (Unsupervised Learning).

   (b) Học có giám sát (Supervised Learning).

   (c) Học tăng cường (Reinforcement Learning).

   (d) Học bán giám sát (Semi-supervised Learning).

2. Nguyên tắc cơ bản nhất của thuật toán KNN là gì?

   (a) Tìm một hàm số để ánh xạ đầu vào tới đầu ra.

   (b) Phân chia không gian dữ liệu bằng các siêu phẳng.

   (c) Xây dựng một cây bao gồm các quy tắc quyết định.

   (d) Phân loại một điểm dữ liệu mới dựa trên các điểm lân cận gần nhất của nó.

3. Trong thuật toán KNN, siêu tham số "k" đại diện cho điều gì?

   (a) Số lượng lớp trong bài toán.

   (b) Số lượng thuộc tính của dữ liệu.

   (c) Số lượng láng giềng gần nhất được xem xét để đưa ra dự đoán.

   (d) Số lần lặp tối đa của thuật toán.

4. Đối với bài toán phân loại (classification), KNN đưa ra dự đoán cho một điểm dữ liệu mới bằng cách nào?

   (a) Tính giá trị trung bình của các nhãn của k láng giềng.

   (b) Gán nhãn của láng giềng xa nhất.

   (c) Gán nhãn phổ biến nhất (majority vote) trong số k láng giềng.

   (d) Gán một nhãn ngẫu nhiên từ các láng giềng.

5. Đối với bài toán hồi quy (regression), giá trị dự đoán của KNN cho một điểm dữ liệu mới được tính như thế nào?

   (a) Bằng giá trị trung bình (average) của các giá trị mục tiêu của k láng giềng.

   (b) Bằng giá trị trung vị (median) của các giá trị mục tiêu của k láng giềng.

   (c) Bằng giá trị phổ biến nhất (mode) của các giá trị mục tiêu của k láng giềng.

   (d) Bằng tổng các giá trị mục tiêu của k láng giềng.

6. Tại sao việc chuẩn hóa dữ liệu (feature scaling) thường là một bước bắt buộc trước khi áp dụng KNN?

   (a) Để thuật toán hội tụ nhanh hơn.

(b) Để loại bỏ các điểm dữ liệu ngoại lai.

(c) Vì các thuộc tính có thang đo lớn có thể lấn át các thuộc tính có thang đo nhỏ khi tính khoảng cách.

(d) Để chuyển đổi các thuộc tính dạng chuỗi thành dạng số.

7. KNN được gọi là một thuật toán "lazy learning" (học lười biếng). Điều này có nghĩa là gì?

(a) Giai đoạn huấn luyện của nó rất phức tạp và tốn thời gian.

(b) Nó không thực sự xây dựng một mô hình tổng quát từ dữ liệu huấn luyện mà chỉ lưu trữ lại toàn bộ dữ liệu.

(c) Nó chỉ hoạt động hiệu quả với các tập dữ liệu nhỏ.

(d) Nó không cần bất kỳ siêu tham số nào.

8. Trong phiên bản "Weighted KNN" (KNN có trọng số), các láng giềng ảnh hưởng đến kết quả dự đoán như thế nào?

(a) Tất cả các láng giềng có phiếu bầu/ảnh hưởng ngang nhau.

(b) Láng giềng ở xa hơn có trọng số (ảnh hưởng) lớn hơn.

(c) Láng giềng ở gần hơn có trọng số (ảnh hưởng) lớn hơn.

(d) Chỉ láng giềng gần nhất mới có trọng số, các láng giềng khác bằng 0.

9. Dựa vào bộ dữ liệu trong phần thực hành, hãy tính khoảng cách Euclidean (L_2) giữa sinh viên $P_1(2, 8)$ và sinh viên $P_4(6, 5)$.

(a) 25.0.

(b) 7.0.

(c) 5.0.

(d) 4.12.

10. Vẫn sử dụng bộ dữ liệu trong phần thực hành, giả sử có một sinh viên mới $Q(5, 8)$. Nếu sử dụng thuật toán KNN với $k = 1$ và khoảng cách Euclidean, kết quả của sinh viên Q sẽ được dự đoán là gì?

(a) Đậu.

(b) Trượt.

(c) Không thể xác định.

(d) Cả Đậu và Trượt.

Phụ lục

1. **Datasets:** Các file dataset được đề cập trong bài có thể được tải tại đây.
2. **Hint:** Các file code gợi ý có thể được tải tại đây.
3. **Demo:** Chương trình demo cho nội dung trong bài có thể được truy cập tại đây.
4. **Solution:** Các file code cài đặt hoàn chỉnh và phần trả lời nội dung trắc nghiệm có thể được tải tại đây.
5. **Rubric:**

Mục	Kiến Thức	Đánh Giá
I.	- Lý thuyết nền tảng của thuật toán KNN (học có giám sát, phi tham số, lazy learning). - Các bước hoạt động cốt lõi của KNN. - Vai trò của siêu tham số k và sự đánh đổi (overfitting/underfitting). - Tầm quan trọng của việc chuẩn hóa dữ liệu (feature scaling) và các thước đo khoảng cách phổ biến. - Cơ chế dự đoán cho bài toán phân loại (majority vote) và hồi quy (averaging), bao gồm cả phiên bản có trọng số (weighted). - Kỹ năng triển khai thuật toán KNN bằng code.	- Trình bày và giải thích được nguyên lý, các đặc điểm của KNN. - Phân tích được ảnh hưởng của việc lựa chọn các giá trị k khác nhau đến kết quả mô hình. - Giải thích được tại sao phải chuẩn hóa dữ liệu và vận dụng được các thước đo khoảng cách. - Có khả năng vận dụng KNN để giải quyết bài toán phân loại và hồi quy trên một bộ dữ liệu thực tế.