BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

KHOA KHOA HỌC CƠ BẢN

BÁO CÁO ĐỀ TÀI KHAI PHÁ DỮ LIỆU

Đề Tài: Phân cụm dữ liệu K-Medoid

NHÓM 5

Giảng viên: **Nguyễn Quốc Tuấn**

Thành viên: **Vũ Thị Minh Ngọc - 213012716**

Đỗ Thị Huệ - 223030629

Đặng Hoàng Phúc - 223000645

Trần Minh Tuấn - 213046718

Tạ Phương Duy - 213010502

Lớp: **Toán ứng dụng** Khoá: **K63**

Hà Nội, 2024

LỜI CẢM ƠN

Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc đến các thầy cô giáo của trường Đại học Giao Thông Vận Tải, đặc biệt là các thầy cô thuộc khoa Khoa học Cơ bản. Chúng em rất biết ơn sự tận tâm trong công tác quản lý, tổ chức môn học và sự quan tâm, tạo điều kiện thuận lợi để chúng em có đủ thời gian và môi trường học tập để hoàn thiện môn học một cách tốt nhất.

Chúng em xin gửi lời cảm ơn đặc biệt tới thầy Nguyễn Quốc Tuấn, người đã tận tình hướng dẫn và chỉ bảo chúng em trong suốt quá trình học tập và thực hiện báo cáo môn Khai phá Dữ liệu. Những kiến thức quý báu và những lời khuyên của thầy đã giúp chúng em rất nhiều trong việc hoàn thành môn học này. Chúng em rất trân trọng sự hỗ trợ và sự tận tâm của thầy trong suốt thời gian qua.

Trong suốt quá trình thực hiện bài báo cáo, dù chúng em đã cố gắng hết sức nhưng không tránh khỏi những thiếu sót và sai sót. Chúng em rất mong thầy thông cảm và bỏ qua những lỗi nhỏ này. Với kinh nghiệm và kiến thức còn hạn chế, chúng em hy vọng sẽ nhận được những ý kiến, đóng góp từ thầy để bài báo cáo có thể hoàn thiện hơn nữa, từ đó giúp chúng em học hỏi và phát triển thêm trong các môn học sau.

Một lần nữa, chúng em xin chân thành cảm ơn thầy và các thầy cô đã đồng hành, giúp đỡ chúng em trong suốt quá trình học tập. Những đóng góp và sự quan tâm của thầy cô sẽ là nguồn động lực lớn lao để chúng em không ngừng nỗ lực và phấn đấu.

Chúng em xin chân thành cảm ơn!

LỜI MỞ ĐẦU

Trong thời đại bùng nổ của dữ liệu lớn và sự phát triển không ngừng của các kỹ thuật học máy, phân cụm dữ liệu đã trở thành một công cụ phân tích vô cùng quan trọng. Phân cụm giúp khám phá cấu trúc ẩn trong dữ liệu và phân loại các đối tượng thành những nhóm có đặc điểm tương đồng. Một trong những phương pháp phổ biến trong phân cụm là **K-medoids**, một thuật toán mạnh mẽ và linh hoạt, phù hợp cho các bài toán yêu cầu độ chính xác cao và khả năng chống chịu với các giá trị ngoại lệ. So với phương pháp K-means, K-medoids không bị ảnh hưởng nhiều bởi các giá trị bất thường do chọn các điểm đại diện nằm trong dữ liệu thay vì tính toán từ trung bình của cụm, điều này làm cho K-medoids đặc biệt hữu ích khi xử lý các tập dữ liệu sinh học phức tạp, trong đó dữ liệu thường có tính biến đổi cao và chứa nhiều nhiễu.

Chim cánh cụt là một đối tượng nghiên cứu lý thú và quan trọng trong sinh thái học, bảo tồn, và biến đổi khí hậu. Các loài chim cánh cụt sống chủ yếu ở vùng Nam Cực và các vùng lân cận, môi trường sống khắc nghiệt và thay đổi lớn giữa các mùa tạo ra nhiều áp lực lên các đặc điểm sinh học của chúng. Do đó, chúng thể hiện sự đa dạng đáng kể về kích thước, cân nặng, cấu trúc cơ thể, và các đặc điểm khác để thích nghi với môi trường khắc nghiệt này. Việc phân tích và phân cụm các đặc điểm sinh học của chim cánh cụt không chỉ giúp hiểu rõ hơn về sự đa dạng sinh học của chúng mà còn có thể cung cấp dữ liệu hữu ích cho việc nghiên cứu tác động của biến đổi khí hậu đối với các loài sinh vật ở Nam Cực.

Đề tài này áp dụng K-medoids để phân tích các đặc điểm sinh trắc học của chim cánh cụt như chiều dài mỏ, độ sâu mỏ, chiều dài vây và khối lượng cơ thể. Mục tiêu là tìm ra các cụm đặc trưng cho từng loài hoặc nhóm chim cánh cụt, giúp phát hiện sự tương đồng trong các nhóm và cung cấp thông tin về sự thích nghi sinh học.

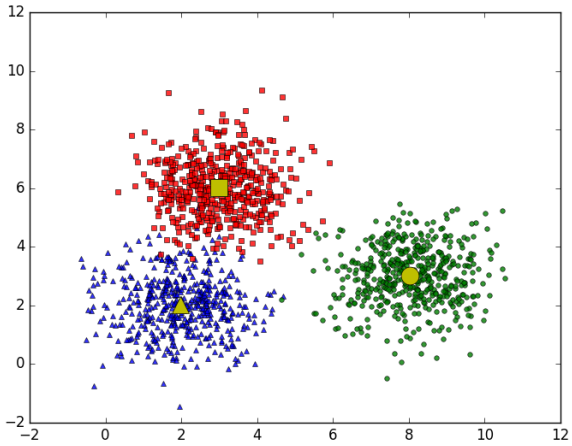
Kết quả của nghiên cứu này không chỉ có ý nghĩa trong việc hiểu rõ hơn về cấu trúc quần thể và sự đa dạng của các loài chim cánh cụt mà còn mở ra tiềm năng ứng dụng trong các nghiên cứu liên quan đến bảo tồn sinh thái. Cụ thể, việc phân cụm các loài chim cánh cụt dựa trên đặc điểm sinh học có thể giúp các nhà nghiên cứu đánh giá mức độ đa dạng di truyền và hiểu sâu hơn về cách mà các loài này thích ứng với môi trường thay đổi nhanh chóng. Bên cạnh đó, việc áp dụng K-medoids vào bài toán phân cụm dữ liệu sinh học có thể mở rộng sang các nghiên cứu sinh học khác, đóng góp cho lĩnh vực khoa học dữ liệu trong sinh học.

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN VỀ PHÂN CỤM DỮ LIỆU

1. Phân cụm dữ liệu là gì?

Phân cụm là quá trình phân vùng một tập hợp các đối tượng dữ liệu (hoặc quan sát) thành các tập hợp con. Mỗi tập hợp con là một cụm, sao cho các đối tượng trong một cụm tương tự nhau, nhưng không giống với các đối tượng trong các cụm khác. Tập hợp các cụm kết quả từ phân tích cụm có thể được gọi là phân cụm.



Hình 1.1: Phân cụm dữ liệu.

Trong bối cảnh này, các phương pháp phân cụm khác nhau có thể tạo ra các cụm khác nhau trên cùng một tập dữ liệu. Phân cụm không được thực hiện bởi con người, mà bởi thuật toán phân cụm.

2. Ứng dụng của phân tích cụm.

Phân tích cụm là một công cụ mạnh mẽ được áp dụng rộng rãi trong nhiều lĩnh vực khác nhau, góp phần vào việc khai thác và phân tích dữ liệu hiệu quả. Một số ứng dụng tiêu biểu bao gồm:

- **Nghiên Cứu Thị Trường**: Phân tích cụm giúp các nhà tiếp thị xác định và khám phá các nhóm khách hàng khác biệt trong cơ sở dữ liệu của họ. Qua đó, họ có thể mô tả các đặc điểm và hành vi mua sắm của từng nhóm, từ đó xây dựng các chiến lược tiếp cận phù hợp.
- **Sinh Học**: Trong lĩnh vực sinh học, phân tích cụm được sử dụng để phân loại các đơn vị sinh vật, chẳng hạn như thực vật và động vật. Nó cũng hỗ trợ trong việc phân

loại các gen có chức năng tương tự, giúp hiểu rõ hơn về cấu trúc di truyền của quần thể.

- **Phân Tích Địa Lý**: Phân cụm có thể được ứng dụng để xác định các khu vực sử dụng đất tương tự trong cơ sở dữ liệu quan sát trái đất. Bằng cách phân nhóm các khu vực, nhà nghiên cứu có thể dễ dàng phân tích và đưa ra quyết định về quy hoạch đô thị và phát triển bền vững.
- **Xử Lý Hình Ảnh**: Trong lĩnh vực xử lý hình ảnh, phân tích cụm giúp phân loại và nhận diện các đối tượng trong hình ảnh, từ đó nâng cao khả năng phân tích và truyền tải thông tin.
- **Khám Phá Thông Tin Web**: Phân tích cụm cũng được áp dụng để phân loại tài liệu trên web, giúp khám phá và tổ chức thông tin một cách hiệu quả, từ đó cải thiện trải nghiệm tìm kiếm và truy cập thông tin của người dùng.

Những ứng dụng này cho thấy tính đa dạng và hiệu quả của phân tích cụm trong việc hỗ trợ các quyết định chiến lược và cải thiện quy trình làm việc trong nhiều lĩnh vực khác nhau.

3. Các phương pháp phân cụm cơ bản.

Phân cụm là một trong những phương pháp học máy không giám sát quan trọng, giúp tìm ra sự tương đồng và các mẫu mối quan hệ giữa các mẫu dữ liệu. Các phương pháp này phân nhóm các mẫu dựa trên các đặc trưng chung của chúng. Có thể phân chia các phương pháp phân cụm thành bốn nhóm chính:

- **Phương Pháp Phân Hoạch** (Partitioning Methods): Trong nhóm phương pháp này, các cụm được hình thành bằng cách chia các đối tượng thành k cụm, trong đó số lượng cụm tương ứng với số lượng phân vùng. Các ví dụ tiêu biểu bao gồm thuật toán K-means và CLARANS, một thuật toán phân nhóm ứng dụng dựa trên tìm kiếm ngẫu nhiên.
- **Phương Pháp Phân Cấp** (Hierarchical Methods): Các phương pháp này tạo ra cấu trúc cụm theo dạng cây, dựa trên hệ thống phân cấp. Chúng được chia thành hai loại chính:
    - Agglomerative (Cách tiếp cận từ dưới lên): Bắt đầu từ các đối tượng riêng lẻ và dần dần hợp nhất thành các cụm lớn hơn.
    - Divisive (Cách tiếp cận từ trên xuống): Bắt đầu từ một cụm duy nhất và phân chia thành các cụm nhỏ hơn. Ví dụ về các phương pháp này bao gồm CURE và BIRCH, các thuật toán phân cụm dựa trên cấu trúc phân cấp.

- **Phương Pháp Dựa Trên Mật Độ** (Density-based Methods): Các phương pháp này hình thành các cụm dựa trên vùng mật độ cao. Chúng có ưu điểm nổi bật về độ chính xác và khả năng kết hợp các cụm. Các ví dụ điển hình là DBSCAN, một thuật toán phân cụm không gian với khả năng xử lý tiếng ồn, và OPTICS, một phương pháp xác định cấu trúc phân cụm.
- **Phương Pháp Dựa Trên Lưới** (Grid-based Methods): Trong các phương pháp này, các cụm được hình thành dưới dạng cấu trúc lưới. Ưu điểm của chúng là cho phép thực hiện các hoạt động phân cụm nhanh chóng và không phụ thuộc vào số lượng đối tượng dữ liệu. Ví dụ tiêu biểu bao gồm STING và CLIQUE, các thuật toán phân cụm dựa trên cấu trúc lưới.

Những phương pháp này không chỉ giúp cải thiện hiệu quả trong việc phân tích dữ liệu mà còn mở ra nhiều cơ hội ứng dụng trong thực tiễn.

4. Một số thuật toán phân cụm phổ biến.

Trong phần này, chúng ta sẽ điểm qua một số thuật toán phân cụm phổ biến như:

- **K-means**: Đây là thuật toán phân cụm dựa trên centroid, hoạt động bằng cách chia dữ liệu thành kkk cụm với mục tiêu tối thiểu hóa tổng bình phương khoảng cách giữa các điểm dữ liệu và centroid tương ứng của chúng. K-means rất phổ biến nhờ tính đơn giản và hiệu quả trong việc phân nhóm các dữ liệu lớn.
- **Hierarchical Clustering**: Thuật toán phân cụm phân cấp xây dựng một cây phân cấp các cụm, cho phép người dùng dễ dàng quan sát mối quan hệ giữa các cụm. Phương pháp này có thể được chia thành hai hướng tiếp cận: từ dưới lên (Agglomerative) và từ trên xuống (Divisive).
- **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise): Thuật toán phân cụm dựa trên mật độ này có khả năng phát hiện các cụm với hình dạng bất kỳ và xử lý hiệu quả các điểm dữ liệu nhiễu. DBSCAN rất hữu ích trong các ứng dụng thực tế, nơi mà dữ liệu có thể không phân bố đồng đều.
- **K-medoids**: K-medoids là một kỹ thuật phân cụm dựa trên đối tượng tiêu chuẩn, tương tự như K-means, nhưng thay vì sử dụng centroid, nó chọn một điểm thực tế trong dữ liệu (medoid) làm đại diện cho mỗi cụm. Điều này giúp thuật toán trở nên mạnh mẽ hơn trong việc xử lý dữ liệu chứa nhiễu hoặc ngoại lệ.

Những thuật toán này cung cấp các phương pháp linh hoạt và hiệu quả trong việc phân tích và nhóm dữ liệu, phù hợp với nhiều tình huống và yêu cầu khác nhau trong nghiên cứu và ứng dụng thực tiễn.

CHƯƠNG 2: THUẬT TOÁN K-MEDOIDS

1. Thuật toán phân cụm K-medoids.

- Thuật toán K-medoids - Một kỹ thuật dựa trên đối tượng tiêu chuẩn là một thuật toán phân cụm phân chia tập hợp các điểm dữ liệu xung quanh một trung gian (điểm khác biệt nhỏ nhất) và liên tục cố gắng giảm sự khác biệt giữa các điểm trong cùng một cụm. Điểm mấu chốt ở đây là medoid về cơ bản là một điểm dữ liệu từ tập hợp đầu vào, không giống như k có nghĩa là giá trị trung bình đơn thuần.
- Thuật toán **PAM (Partitioning Around Medoids)** là một phương pháp phân cụm phổ biến, đặc biệt trong bài toán phân cụm theo mô hình **K-medoids**. PAM được sử dụng chủ yếu trong các trường hợp mà thuật toán **K-means** không hoạt động tốt, do đó nó được áp dụng trong một số trường hợp cụ thể nhưng không phổ biến rộng rãi như K-means.

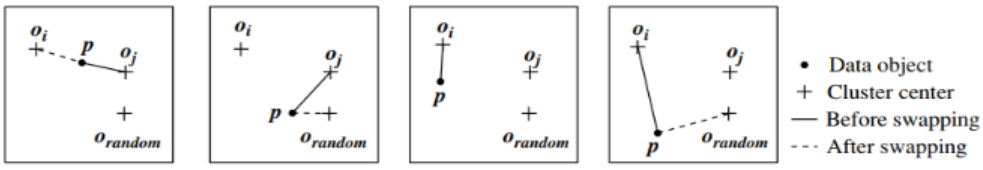
1.1. Ý tưởng của thuật toán.

- Để tìm ra k cụm cho n đối tượng thì K-medoids chọn ngẫu nhiên n đối tượng vào k cụm
  - Coi mỗi đối tượng trong tập n đối tượng là trọng tâm của cụm
  - Phân bố các đối tượng còn lại vào cụm mà sự khác nhau của nó với đối tượng trọng tâm của cụm là ít nhất (hay còn gọi là gần nhất)
  - Lặp lại quá trình : Thay đổi đối tượng trọng tâm của cụm được đánh giá bởi 1 hàm đo sự khác nhau giữa 1 đối tượng và đối tượng của nó. Quá trình này lặp cho đến khi không còn sự thay đổi nào về lực lượng cũng như hình dạng của các cụm.

- Cụ thể :
  - Giả sử {O1,...,Ok}. k là tập hợp các đối tượng đại diện hiện tại (medoids).

- Để xác định liệu một đối tượng không phải đại diện, được ký hiệu là O_{random}. có thể thay thế tốt cho một medoid hiện tại Oj (với $1 \leq j \leq k$) .Chúng ta tính khoảng cách từ mỗi đối tượng **p** đến đối tượng gần nhất trong tập hợp $\{O_1, O_{j-1}, \dots, O_{random}, O_k\}$ và sử dụng khoảng cách này để cập nhật hàm chi phí.

- Xét 4 trường hợp xảy ra:



(a) : Gán lại cho điểm Oi
(b) : Gán lại cho điểm O_{random}
(c) : Không thay đổi
(d) : Gán lại cho điểm O_{random}

Data object : Điểm dữ liệu
Cluster center : Cụm trung tâm
Before swapping : Trước khi đổi
After swapping : Sau khi đổi

Hình 2.1: Bốn trường hợp của hàm chi phí cho cụm k-medoids.

- TH1 : **p** đang thuộc vào cụm có trọng tâm $\mathbf{O_i}$. Nếu O_j được 'thay thế' bởi O_{random} và **p** "gần nhất" với O_i ($i \neq j$) thì p *gán lại* vào O_i (hình 2.1a).
- TH2 : đang thuộc vào $\mathbf{O_i}$. Nếu O_i ' thay thế' bởi O_{random} và p "gần nhất" với O_{random} thì p được *gán lại* vào O_{random} (hình 2.1b).
- TH3: **p** đang thuộc vào **Oj**. Nếu Oj thay thế bởi Orandom và p vẫn "gần nhất" với Oj → p không thay đổi (tức p *vẫn thuộc* Oj) (hình 2.1c).
- TH4: **p** đang thuộc vào **Oj .** . Nếu Oj thay thế bởi Orandom và p "gần nhất" với Orandom thì p được gán lại vào Orandom (hình 2.1d).

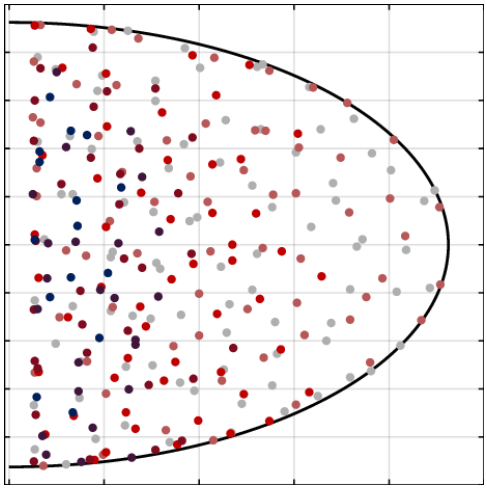
1.2. Chi tiết thuật toán.

Trong thuật toán K-medoids bao gồm:

- **Input** (đầu vào).
  - k: Số cụm mà bạn muốn chia dữ liệu thành.

Hình 2.2: Hình ảnh biểu diễn k - cụm.

➢ D: Tập dữ liệu chứa n đối tượng.

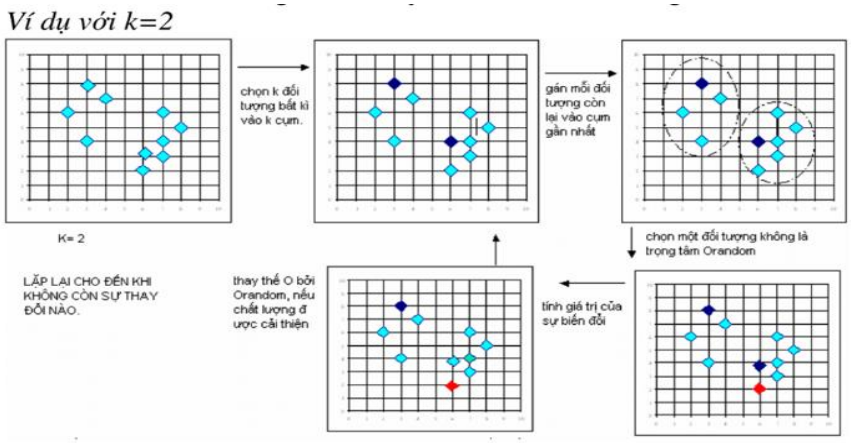


Hình 2.3: Hình ảnh biểu diễn n – đối tượng.

- **Output** (đầu ra).
  - Oj: Một đối tượng đại diện hiện tại (medoid) trong số n đối tượng đại diện ban đầu được chọn ngẫu nhiên.
  - O_{random}: Một đối tượng không phải là đại diện hiện tại (non-representative object) được chọn ngẫu nhiên từ tập dữ liệu.
  - S: Tổng chi phí của việc hoán đổi giữa một đối tượng đại diện hiện tại Oj và một đối tượng không đại diện O_{random}. Chi phí này dựa trên việc tính tổng khoảng cách giữa các đối tượng và đối tượng đại diện tương ứng trong cụm.

1.3. Quy trình hoạt động của thuật toán.

- Bước 1: Ban đầu chọn ngẫu nhiên k đối tượng trong n đối tượng từ tập dữ liệu D làm các đối tượng đại diện ban đầu (medoids).
- Bước 2:Bước lặp:
  - Bước 3: Gán mỗi đối tượng còn lại vào cụm chứa đối tượng đại diện gần nhất.
  - Bước 4: Chọn ngẫu nhiên một đối tượng không phải là đại diện $O_{random.}$
  - Bước 5: Tính tổng chi phí S của việc hoán đổi đối tượng đại diện hiện tại Oj với O_{random}.
  - Bước 6: Nếu chi phí S < 0 (nghĩa là việc hoán đổi cải thiện tổng chi phí), thực hiện hoán đổi Oj với O_{random} để hình thành tập mới gồm k đối tượng đại diện.
- Bước 7: Dừng lại khi không còn sự thay đổi nào.



Hình 2.4: Tổng quát về các bước thực hiện.

2. Ví dụ về thuật toán K-Medoids.

2.1.Ví dụ 1.

Cho tập dữ liệu sau, hãy phân dữ liệu sau thành 2 cụm:

$A_1(2,10)$ $A_2(2,5)$ $A_3(8,4)$ $B_1(5,8)$ $B_2(7,5)$ $B_3(6,4)$ $C_1(1,2)$ $C_2(4,9)$

Bài làm

- Với số cụm cho trước là k = 2 (cụm). Ta chọn 2 điểm ngẫu nhiên trong tập dữ liệu trên là: A_1, B_1 để tạo thành các medoid (đại diện của cụm)
- Ta có :

Medoid 1: $A_1(2,10)$ Medoid 2: $B_1(5,8)$

- Phân cụm dữ liệu bằng cách tính toán khoảng cách giữa các điểm với các Medoid đã chọn ban đầu. Sử dụng công thức khoảng cách Manhattan và gán điểm đó vào cụm có medoid gần nhất.
- Công thức khoảng cách Manhattan: Manhattan Distance = $|X1 - X2| + |Y1 - Y2|$
- Bảng tính toán khoảng cách giữa các điểm tới Medoid của cụm:

Điểm	Medoid 1 $A_1(2,10)$	Medoid 2 $B_1(5,8)$	Gán
$A_2(2,5)$	5	6	Cụm 1
$A_3(8,4)$	12	7	Cụm 2
$B_2(7,5)$	10	5	Cụm 2
$B_3(6,4)$	10	5	Cụm 2
$C_1(1,2)$	9	10	Cụm 1
$C_2(4,9)$	3	2	Cụm 2

⇨ Kết quả phân cụm ban đầu:

Cụm 1 (Medoid A_1): A_1, A_2, C_1 Cụm 2 (Medoid B_1): B_1,C_2, A_3, B_2, B_3

⇨ Tính tổng chi phí của các điểm thuộc các cụm trên:

Cụm 1 : Cost1 = 5 + 9 = 14 Cụm 2 : Cost2 = 7 + 5 + 5 + 2 = 19

- Tổng chi phí ban đầu (TotalCost) = Cost1 + Cost2 = 33.
- Thay thế Medoids :
  - Ta xét các điểm ngẫu nhiên để tính khoảng cách từ các điểm khác đến 2 điểm medoids ngẫu nhiên :
  - Xét 2 điểm ngẫu nhiên A_1(2,10) và A_2 (2,5) lần lượt làm medoids của cụm 1 và cụm 2.
  - Ta có bảng sau :

Điểm	Medoids A_1(2,10)	Medoids A_2 (2,5)	Gán
A_1(2,10)	0	5	Cụm 1
A_2 (2,5)	5	0	Cụm 2
A_3(8,4)	12	7	Cụm 2
B_1(5,8)	5	6	Cụm 1
B_2(7,5)	10	5	Cụm 2
B_3(6,4)	10	5	Cụm 2
C_1(1,2)	9	4	Cụm 2
C_2(4,9)	3	6	Cụm 1

⇨ Kết quả phân cụm:

Cụm 1 (Medoid A_1): A_1, B_1, C_2 Cụm 2 (Medoid A_2): A_2, A_3, B_2, B_3, C_1

⇨ Tính tổng chi phí của các điểm thuộc các cụm trên:

- Cụm 1 : Cost1_new = 0 + 5 + 3 = 8
- Cụm 2 : Cost2_new = 0 + 7 + 5 + 5 + 4 = 21

⇨ Tổng chi phí mới (TotalCost_new) = Cost1_new + Cost2_new = 29

- Ta thấy : S = TotalCost_new – TotalCost = 29 – 33 = -4 < 0

⇨ Thay điểm A_2 (2,5) làm Medoids mới của cụm 2.

- Xét 2 điểm ngẫu nhiên A_1(2,10) và B_3(6,4) lần lượt làm medoids của cụm 1 và cụm 2.
- Ta có bảng sau :

Điểm	Medoids $A_1(2,10)$	Medoids $B_3(6,4)$	Gán
$A_1(2,10)$	0	10	Cụm 1
A_2 (2,5)	5	5	Cụm 2
$A_3(8,4)$	12	2	Cụm 2
$B_1(5,8)$	5	5	Cụm 1
$B_2(7,5)$	10	2	Cụm 2
$B_3(6,4)$	10	0	Cụm 2
$C_1(1,2)$	9	7	Cụm 2
$C_2(4,9)$	3	7	Cụm 1

⇨ Kết quả phân cụm:

Cụm 1 (Medoid A_1): A_1, B_1, C_2 Cụm 2 (Medoid B_3): A_2, A_3, B_2, B_3, C_1

⇨ Tính tổng chi phí của các điểm thuộc các cụm trên

- Cụm 1 : Cost1_new = 0 + 5 + 3 = 8
- Cụm 2 : Cost2_new = 5 + 2 + 2 + 0 + 7 = 16

⇨ Tổng chi phí mới (TotalCost_new) = Cost1_new + Cost2_new = 24.

- Ta thấy : S = TotalCost_new – TotalCost = 24 – 29 = -5 < 0

⇨ Thay điểm $B_3(6,4)$làm Medoids mới của cụm 2.

*** Tương tự, ta xét các điểm còn lại và tính khoảng cách (TotalCost) để tìm S và kiểm tra.

⇨ Kết quả cho ta vẫn là dữ liệu không thay đổi:

Cụm 1 (Medoid A_1): A_1, B_1, C_2 Cụm 2 (Medoid B_3): A_2, A_3, B_2, B_3, C_1

2.2. Mở rộng ví dụ.

Trong phần này, câu hỏi chúng ta đặt ra như sau: Giả sử nếu dữ liệu chưa cho trước số cụm thì làm thế nào ?

- Một phương pháp đơn giản là đặt số cụm thành khoảng n2 cho một tập dữ liệu gồm n điểm. Theo kỳ vọng, mỗi cụm có 2n điểm (Tùy trong mọi bài toán)
- **Phương pháp khuỷu tay (**Elbow Method) cũng thường được sử dụng, dựa trên việc giảm phương sai trong cụm và chọn điểm ngoặt của đường cong phương sai để tìm số cụm hợp lý.
- Các phương pháp tiên tiến hơn sử dụng tiêu chí thông tin hoặc lý thuyết thông tin để Axác định số cụm. Một cách khác là **xác thực chéo**, chia tập dữ liệu thành nhiều

phần và sử dụng mỗi phần để kiểm tra chất lượng của mô hình phân cụm. Sau đó, so sánh chất lượng của các giá trị k khác nhau để tìm số cụm phù hợp nhất.

2.3. Ví dụ 2.

Thực hiện phương pháp Elbow Method cho 1 tập dữ liệu gồm 5 điểm. Hãy tìm số cụm k sao cho tối ưu nhất?

A(1,2), B(2,3), C(3,3), D(6,7), E(7,8).

Bài làm

- **Bước 1: Xác định số cụm.**
  Chúng ta sẽ thực hiện phân cụm với k=1, k=2, và k=3, sau đó tính tổng chi phí cho từng giá trị của k.
- **Bước 2: Phân cụm với k = 1.**
  - Với k = 1 tất cả các điểm đều nằm trong 1 cụm duy nhất và chỉ có 1 medoid. Để tìm medoid, ta tính tổng khoảng cách từ từng điểm đến các điểm còn lại và chọn điểm có tổng khoảng cách nhỏ nhất làm medoid.
  - Tính tổng khoảng cách từ mỗi điểm đến các điểm còn lại (sử dụng khoảng cách Euclide) :

    d(A) = d(A,B) + d(A,C) + d(A,D) + d(A,E)
    d(A,B) = 2-12+3-22 ≈1,41
    d(A,C) = 3-12+3-22 ≈2,24
    d(A,D) = 6-12+(7-2)2 ≈7,07
    d(A,E) = 7-12+8-22 ≈8,49

⇨ Tổng d(A) = 1,41 + 2,24 + 7,07 + 8,49 = 19,21

- Tương tự tính cho các điểm B, C, D, E:
  d(B) ≈17,38 d(C) ≈16,95 d(D) ≈12,67 d(E) ≈ 14,08
- Chọn Medoid :
  - Do d(D) có tổng khoảng cách nhỏ nhất nên D(6,7) là medoid của cụm duy nhất
  - Tổng chi phí (cost) với K=1 - Chi phí tổng là tổng các khoảng cách từ các điểm còn lại đến medoid D.

    d(A,D) ≈ 7.07 d(B,D) ≈ 6.40 d(C,D) ≈ 5.66 d(E,D) ≈ 1.41

⇨ Tổng chi phí = 7.07 + 6.40 + 5.66 + 1.41 = 20.54

- **Bước 3 : Phân cụm với k = 2.**
  - Với k = 2, chúng ta sẽ chia các điểm thành 2 cụm gồm 2 medoid.
  - Chọn 2 medoid: Giả sử chọn ban đầu D(6,7) và A(1,2) làm medoid.
  - Gán các điểm vào cụm gần nhất:

    d(B,A) ≈ 1.41 d(B,D) ≈ 6.40 => B nằm trong cụm của A.
    d(C,A) ≈ 2.24 d(C,D) ≈ 5.66 => C nằm trong cụm của A.
    d(E,D) ≈ 1.41 d(E,A) ≈ 1.41 => E nằm trong cụm của D.

⇨ Tổng chi phí (với k = 2) = d(B,A) + d(C,A) + d(E,D) = 1.41 + 2.24 + 1.41 = 5.06

- **Bước 4 : Phân cụm với k = 3.**
  - Với k=3, chọn 3 medoid giả định là A(1,2), D(6,7), C(3,3)
  - Gán các điểm vào các cụm gần nhất :

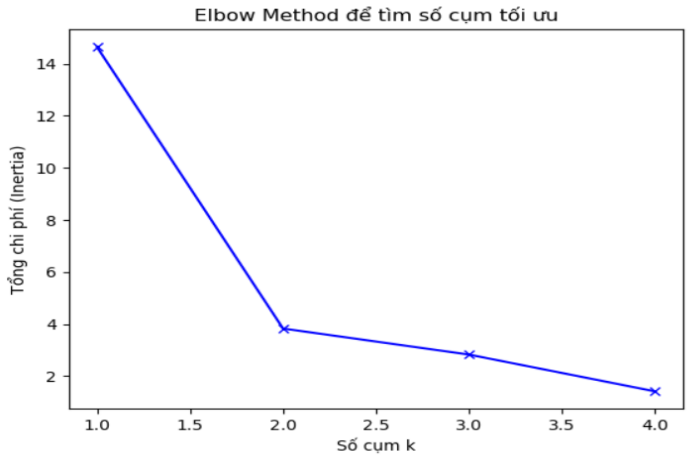
    A(1,2): Gồm A,B

    D(6,7): Gồm D,E

    C(3,3): Chỉ có C

⇨ Tổng chi phí (với k = 3) = d(B,A) + d(E,D) = 1.41 + 1.41 = 2.82

- **Bước 5 : Áp dụng Elbow Method.**
  - Tổng chi phí đã tính được cho các giá trị k :
    - k=1: Tổng chi phí = 20.54
    - k=2: Tổng chi phí = 5.06
    - k=3: Tổng chi phí = 2.82
  - Ta có biểu đồ Elbow method:



Hình 2.5: Biểu đồ elbow method.

⇨ Từ biểu đồ ta thấy được k = 2 cụm là số cụm phù hợp nhất, vì đó là điểm ngoặt đầu tiên khi giá trị của tổng chi phí giảm mạnh, sau đó mức giảm trở nên ít hơn.

3. Ưu điểm – nhược điểm của thuật toán K-Medoids.

- Phương pháp k-medoids mạnh hơn k-means khi có nhiễu và giá trị ngoại lai vì **Medoid ít bị ảnh hưởng bởi giá trị ngoại lai hoặc các giá trị cực đoan khác hơn giá trị trung bình.**
- Tuy nhiên, độ phức tạp của mỗi lần lặp trong thuật toán k-medoids là $O(k(n-k)^2)$. Đối với các giá trị lớn của n và k, việc tính toán như vậy trở nên rất tốn kém và tốn kém hơn nhiều so với phương pháp k-means. Cả hai phương pháp đều yêu cầu người dùng chỉ định k, số lượng cụm.

- Một thuật toán phân cụm k-medoids điển hình như PAM hoạt động hiệu quả đối với các tập dữ liệu nhỏ, nhưng không mở rộng tốt cho các tập dữ liệu lớn. Để xử lý các tập dữ liệu lớn hơn, có thể sử dụng một phương pháp dựa trên mẫu ngẫu nhiên gọi là CLARA (Clustering Large Applications) hoặc CLARANS (Phân cụm các ứng dụng lớn dựa trên Tìm kiếm ngẫu nhiên).

4. Cải tiến thuật toán.

4.1. PAM (Partitioning Around Medoids).

- Đặc điểm: Đây là thuật toán cơ bản của K-Medoid, hoạt động bằng cách tìm các điểm trung tâm (medoids) đại diện cho mỗi cụm trong dữ liệu. PAM có độ chính xác cao vì nó duyệt qua toàn bộ tập dữ liệu để tìm giải pháp tốt nhất cho từng cụm.
- Ứng dụng: Tốt nhất cho tập dữ liệu nhỏ hoặc vừa, do chi phí tính toán cao. Khi số lượng dữ liệu tăng lên, thuật toán sẽ tiêu tốn nhiều tài nguyên, dẫn đến giảm hiệu suất.

4.2. CLARA (Clustering Large Applications).

- Đặc điểm: Đây là phiên bản cải tiến của PAM, sử dụng một mẫu nhỏ từ dữ liệu lớn để giảm chi phí tính toán. CLARA chọn một mẫu dữ liệu và thực hiện thuật toán PAM trên mẫu này để giảm độ phức tạp.
- Ứng dụng: Phù hợp cho tập dữ liệu lớn vì giảm số lượng điểm cần xử lý. Tuy nhiên, kết quả có thể không đạt mức tối ưu nếu mẫu không đại diện đầy đủ cho toàn bộ tập dữ liệu.

4.3. CLARANS (Clustering Large Applications based on Randomized Search).

- Đặc điểm: Kết hợp phương pháp lấy mẫu ngẫu nhiên và tìm kiếm cục bộ, cho phép tìm kiếm nhanh hơn và linh hoạt hơn so với PAM. Thay vì duyệt toàn bộ tập dữ liệu hoặc chọn mẫu cố định như CLARA, CLARANS chọn ngẫu nhiên và thực hiện tìm kiếm trên các điểm lân cận.
- Ứng dụng: Tốt cho tập dữ liệu lớn, nhưng việc cài đặt tham số (như số lượng bước tìm kiếm ngẫu nhiên) có thể phức tạp, và lựa chọn tham số không phù hợp có thể ảnh hưởng đến kết quả.

4.4. FastPAM.

- Đặc điểm: Đây là một biến thể của PAM được tối ưu hóa về tốc độ, giảm thiểu số lần tính toán cần thiết khi lựa chọn medoids, giúp giảm chi phí tính toán mà vẫn duy trì độ chính xác tương đối.
- Ứng dụng: Phù hợp với tập dữ liệu từ vừa đến lớn, khi cần một phương pháp nhanh chóng hơn PAM. Tuy nhiên, FastPAM có thể không đạt độ tối ưu như PAM do có ít phép tính và ít duyệt qua toàn bộ dữ liệu.

CHƯƠNG 3: ỨNG DỤNG CỦA THUẬT TOÁN K-MEDOIDS

1. Tên ứng dụng.

Trong bài báo này, chúng tôi thực hiện ứng dụng *"phân cụm loài chim cánh cụt (penguins)"*. Mục tiêu của bài toán là sử dụng phương pháp phân cụm K-Medoids để nhóm các cá thể chim cánh cụt vào các cụm dựa trên đặc điểm hình thái học chung, giúp hiểu rõ hơn về sự phân bố và phân loại các loài chim cánh cụt.

Bộ dữ liệu bao gồm các đặc trưng như chiều dài mỏ, chiều sâu mỏ, chiều dài cánh và khối lượng cơ thể. Chúng tôi áp dụng phương pháp phân cụm K-Medoids để phân loại chim cánh cụt thành các nhóm, sau đó phân tích kết quả phân cụm dựa trên các đặc điểm của từng nhóm.

2. Nguồn dữ liệu.

Bộ dữ liệu "data.csv" được sử dụng trong nghiên cứu này, được lấy từ trang Kaggle: https://www.kaggle.com/code/totathewarrior/clustering-penguins-species/notebook, chứa thông tin về các loài chim cánh cụt, bao gồm các đặc trưng hình thái học của chúng. Có 344 đối tượng được xét và gồm 5 biến số trong bộ dữ liệu bao gồm:

- **culmen_length_mm**: Chiều dài mỏ (tính bằng mm)
- **culmen_depth_mm**: Chiều sâu mỏ (tính bằng mm)
- **flipper_length_mm**: Chiều dài vây (tính bằng mm)
- **body_mass_g**: Khối lượng cơ thể (tính bằng gram)
- **sex:** giới tính chim cánh cụt (FEMALE và MALE)

Chúng tôi sử dụng bộ dữ liệu đã được chuẩn hóa (sau khi thực hiện PCA để giảm số chiều) để phân tích và phân cụm. Chúng ta sẽ sử dụng bốn đặc trưng hình thái học để thực hiện phân cụm, vì **giới tính** không phải là đặc trưng sinh học chủ yếu giúp phân biệt giữa các loài chim cánh cụt trong bài toán phân cụm này. Do đó, biến "sex" sẽ không được đưa vào phân tích phân cụm.

3. Cách bước tiến hành.

3.1. Tiền xử lý dữ liệu.

3.1.1. Làm sạch dữ liệu.

- **Bước 1**: Tải và khám phá dữ liệu.
  - Đọc dữ liệu:

```
# Đọc dữ liệu
data = pd.read_csv("data.csv")
data.head()
```

	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex
0	39.1	18.7	181.0	3750.0	MALE
1	39.5	17.4	186.0	3800.0	FEMALE
2	40.3	18.0	195.0	3250.0	FEMALE
3	NaN	NaN	NaN	NaN	NaN
4	36.7	19.3	193.0	3450.0	FEMALE

- Phân tích dữ liệu:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 344 entries, 0 to 343
Data columns (total 5 columns):
culmen_length_mm     342 non-null float64
culmen_depth_mm      342 non-null float64
flipper_length_mm    342 non-null float64
body_mass_g          342 non-null float64
sex                  335 non-null object
dtypes: float64(4), object(1)
memory usage: 13.5+ KB
```

⇨ Bộ dữ liệu gồm 344 đối tượng và 5 thuộc tính.

- Tính dữ liệu:
  - count: Tổng số giá trị không rỗng
  - Mean: Gía trị trung bình các giá trị cột
  - Std: Độ lệch chuẩn của các giá trị cột
  - 25%: phần trăm thứ 25
  - min /max: giá trị nhỏ nhất/ lớn nhất từ cột

	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g
count	342.000000	342.000000	342.000000	342.000000
mean	43.921930	17.151170	214.014620	4201.754386
std	5.459584	1.974793	260.558057	801.954536
min	32.100000	13.100000	-132.000000	2700.000000
25%	39.225000	15.600000	190.000000	3550.000000
50%	44.450000	17.300000	197.000000	4050.000000
75%	48.500000	18.700000	213.750000	4750.000000
max	59.600000	21.500000	5000.000000	6300.000000

- **Bước 2:** Tìm các giá trị khuyết thiếu.
  - Đây là bước rất quan trọng vì các thuật toán học máy không thể xử lý các giá trị thiếu.

```
# Gía trị khuyết thiếu
data.isnull().sum()
```

```
culmen_length_mm     2
culmen_depth_mm      2
flipper_length_mm    2
body_mass_g          2
sex                  9
dtype: int64
```

  ⇨ Sau khi phân tích, đã phát hiện rằng có **2 giá trị thiếu** trong các cột "culmen_length_mm, culmen_depth_mm, flipper_length_mm, body_mass_g" và **9 giá trị thiếu** trong cột "sex".

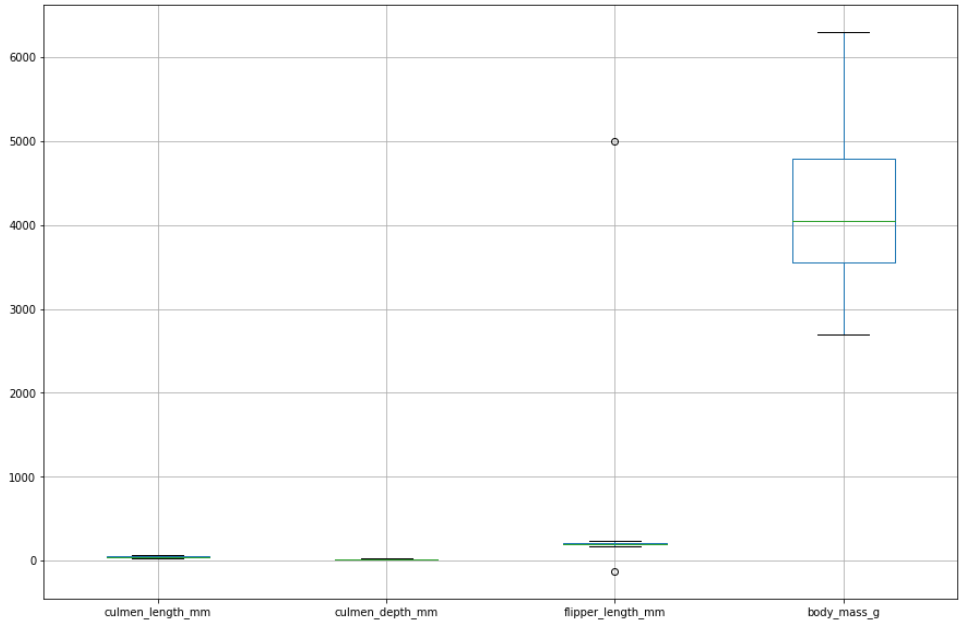
- **Bước 3:** Xử lý các giá trị khuyết thiếu.
  - Loại bỏ các dòng có giá trị thiếu: Nếu số lượng các dòng thiếu là không đáng kể.
  - Thay thế các giá trị thiếu: Dùng giá trị trung bình (mean), trung vị (median), hoặc mode (mode) cho các giá trị thiếu trong các đặc trưng số học.

```
# Xử lý các giá trị bị thiếu
data.dropna(inplace=True)

data.isnull().sum()
```

```
culmen_length_mm     0
culmen_depth_mm      0
flipper_length_mm    0
body_mass_g          0
sex                  0
dtype: int64
```

- **Bước 4:** Xác định giá trị ngoại lai.
  - Các giá trị dị biệt có thể ảnh hưởng đến chất lượng của phân tích phân cụm, vì vậy cần kiểm tra và xử lý chúng nếu cần thiết. Sử dụng biểu đồ hộp cho các cột số để kiểm tra các giá trị ngoại lai.

```
# biểu đồ hộp cho các cột số để kiểm tra các giá trị ngoại lai

plt.figure(figsize=(15, 10))
data.boxplot()
plt.show()
```



- Bước 5: Xử lý ngoại lai.
  - Loại bỏ các giá trị ngoại lai:

```
# Xóa các hàng có giá trị sai
data = data[data['sex'] != '.']
data = data[data['flipper_length_mm'] >= 0]
data = data[data['flipper_length_mm'] < 4000]
# reset index
data.reset_index(drop=True, inplace=True)

data.describe()
```

  - Sau khi loại bỏ các giá trị ngoại lai, ta thu được kết quả:

	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g
count	332.000000	332.000000	332.000000	332.000000
mean	44.021084	17.153012	200.975904	4206.475904
std	5.452462	1.960275	14.035971	806.361278
min	32.100000	13.100000	172.000000	2700.000000
25%	39.500000	15.600000	190.000000	3550.000000
50%	44.700000	17.300000	197.000000	4025.000000
75%	48.625000	18.700000	213.000000	4781.250000
max	59.600000	21.500000	231.000000	6300.000000

3.1.2. Tích hợp dữ liệu.

- **Bước 1:** Xử lý giá trị trùng lặp.
  - Tìm và xử lý các giá trị trùng lặp:

```
# Tìm các hàng trùng lặp
duplicate_rows = data[data.duplicated(keep=False)]

# Hiển thị số hàng trùng lặp
print(f"Bộ dữ liệu chứa {data.duplicated().sum()} hàng trùng lặp cần được loại bỏ.")

# Loại bỏ các hàng trùng lặp
data.drop_duplicates(inplace=True)

# Kiểm tra số hàng còn lại trong dataframe
data.shape[0]
```

```
Bộ dữ liệu chứa 0 hàng trùng lặp cần được loại bỏ.

332
```

⇨ Từ dữ liệu ban đầu 344 đối tượng, sau khi loại bỏ sự trùng lặp dư thừa thì còn 332 đối tượng.

- **Bước 2:** Mã hóa cột " sex_MALE ".
  - Vì giới tính không phải là đặc trưng sinh học chủ yếu giúp phân biệt giữa các loài chim cánh cụt trong bài toán phân cụm này, vì vậy ta phải mã hóa 'MALE' thành '1' và ' FEMALE' thành '0'. Ta có bảng kết quả:

	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex_MALE
0	39.1	18.7	181.0	3750.0	1
1	39.5	17.4	186.0	3800.0	0
2	40.3	18.0	195.0	3250.0	0
3	36.7	19.3	193.0	3450.0	0
4	39.3	20.6	190.0	3650.0	1

- **Bước 3:** Chuẩn hóa dữ liệu.
  - Tính toán giá trị trung bình và độ lệch chuẩn của từng thuộc tính.Sau khi chuẩn hóa, sẽ giúp mô hình phân cụm hoạt động hiệu quả hơn.

```
# Chuẩn hóa dữ liệu (tính toán giá trị trung bình và độ lệch chuẩn của từng thuộc tính)
Scaler = StandardScaler()
scaled = Scaler.fit_transform(data)

data_scaled = pd.DataFrame(scaled, columns=data.columns)
data_scaled.head()
```

  - Ta có bảng kết quả chuẩn hóa:

	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex_MALE
0	-0.903906	0.790360	-1.425342	-0.566948	0.993994
1	-0.830434	0.126187	-1.068577	-0.504847	-1.006042
2	-0.683490	0.432728	-0.426399	-1.187953	-1.006042
3	-1.344738	1.096901	-0.569105	-0.939551	-1.006042
4	-0.867170	1.761074	-0.783164	-0.691149	0.993994

3.1.3. Giảm chiều dữ liệu.

Khi làm việc với nhiều đặc trưng, bạn có thể sử dụng **PCA (Principal Component Analysis)** để giảm số chiều mà vẫn giữ được thông tin quan trọng. PCA giúp chuyển đổi các đặc trưng ban đầu thành các thành phần chính (principal components), giúp giảm độ phức tạp và giúp trực quan hóa dữ liệu tốt hơn.

- **Bước 1:** Áp dụng PCA để giảm số chiều xuống còn 2.

```
pca = PCA(n_components=None)  # phân tích dữ liệu chính để giảm chiều
pca_data = pca.fit(data_scaled)

pca_data.explained_variance_ratio_
```

```
array([0.56820593, 0.28153159, 0.09633697, 0.03399271, 0.01993278])
```

- **Bước 2:** Chuyển PCA thành data và thêm vào bộ dữ liệu.

```
pca = PCA(n_components=2)
pca_data = pca.fit_transform(data_scaled)
```

3.2. Thuật toán K-Medoids vào ứng dụng.

3.2.1. Tính số cụm (k).

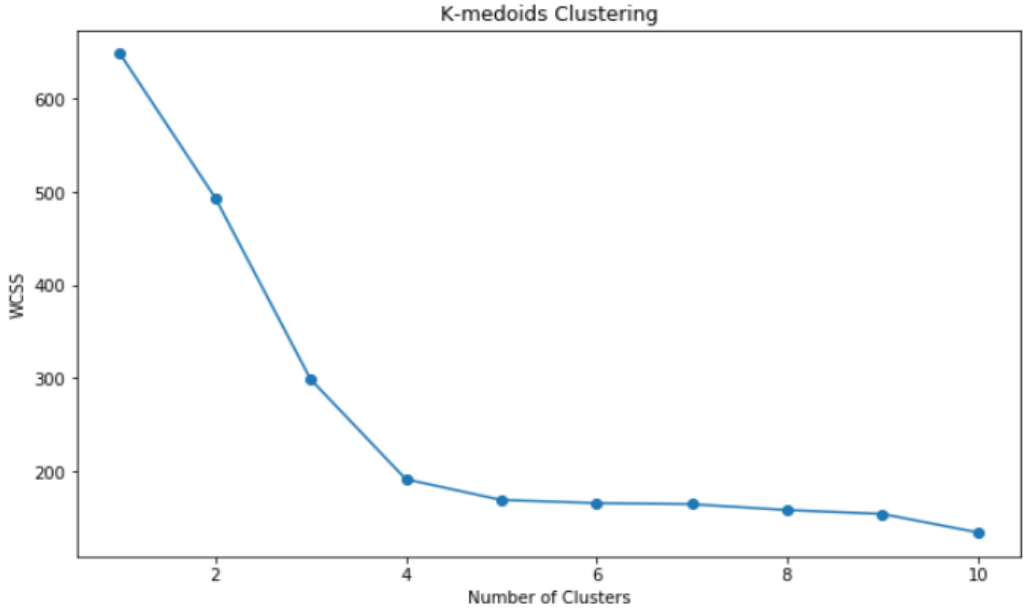
Để xác định được số cụm-k tối ưu, ta sử dụng phương pháp Elbow:

```
# elbow method
wcss = []   # Trong cụm tổng bình phương
for i in range(1, 11):
    kmedoids = KMedoids(n_clusters=i, random_state=42)
    kmedoids.fit(pca_data)
    wcss.append(kmedoids.inertia_)
```

Sau đó, ta vẽ biểu đồ Elbow Method:

```
# Biểu đồ của Elbow method
plt.figure(figsize=(10, 6))
plt.plot(range(1, 11), wcss, marker='o', linestyle='-')
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.title('K-medoids Clustering')
plt.show()
```

Sau khi áp dụng phương pháp Elbow, ta xác định được k = 4 là số cụm tối ưu. Điểm gãy xuất hiện ở giá trị k = 4, cho thấy rằng bộ dữ liệu này có thể được phân thành 4 cụm chính.



3.2.2. Phân cụm.

Sử dụng thuật toán K-Medoids với k = 4, ta thu được 4 cụm chính tương ứng với bốn loại chim cánh cụt với các số lượng khác nhau:

```python
# 4. Sử dụng K-Medoids với số cụm k xác định từ Elbow Method
k = 4  # Giả sử bạn đã xác định k từ Elbow Method

kmedoids = KMedoids(n_clusters=k, random_state=42)
kmedoids.fit(pca_data)

# Lấy nhãn phân cụm và các cụm
labels = kmedoids.labels_

# Đếm số lượng phần tử trong mỗi cụm
cluster_sizes = [np.sum(labels == i) for i in range(k)]

# In ra số lượng phần tử trong mỗi cụm
print(f"Số lượng phần tử trong từng cụm:")
for i in range(k):
    print(f"Cụm {i + 1}: {cluster_sizes[i]} con")
```

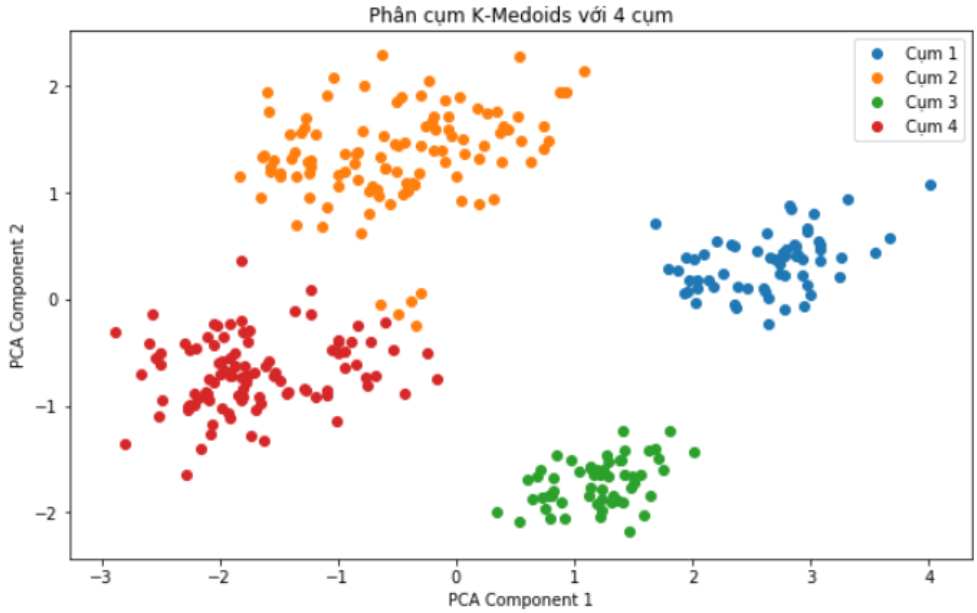
```
Số lượng phần tử trong từng cụm:
Cụm 1: 61 con
Cụm 2: 111 con
Cụm 3: 58 con
Cụm 4: 102 con
```

⇨ Như vậy, ta thấy được số lượng chim cánh cụt ở từng cụm: Cụm 1 có 61 con, cụm 2, Cụm 2 có 111 con, Cụm 3 có 58 con, và Cụm 4 có 102 con.

Từ đó, ta vẽ biểu đồ K-Medoids với 4 cụm:

```python
# Vẽ đồ thị phân cụm
plt.figure(figsize=(10, 6))

# Vẽ điểm cho từng cụm
for cluster_idx in range(k):
    cluster_points = pca_data[labels == cluster_idx]  # Lấy các điểm trong cụm
    plt.scatter(cluster_points[:, 0], cluster_points[:, 1], label=f"Cụm {cluster_idx + 1}")

# Thêm tiêu đề và nhãn cho đồ thị
plt.title(f"Phân cụm K-Medoids với {k} cụm")
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.legend()
plt.show()
```

và có biểu đồ như sau:



Ta hiển thị bảng tính giá trị trung bình của từng cụm:

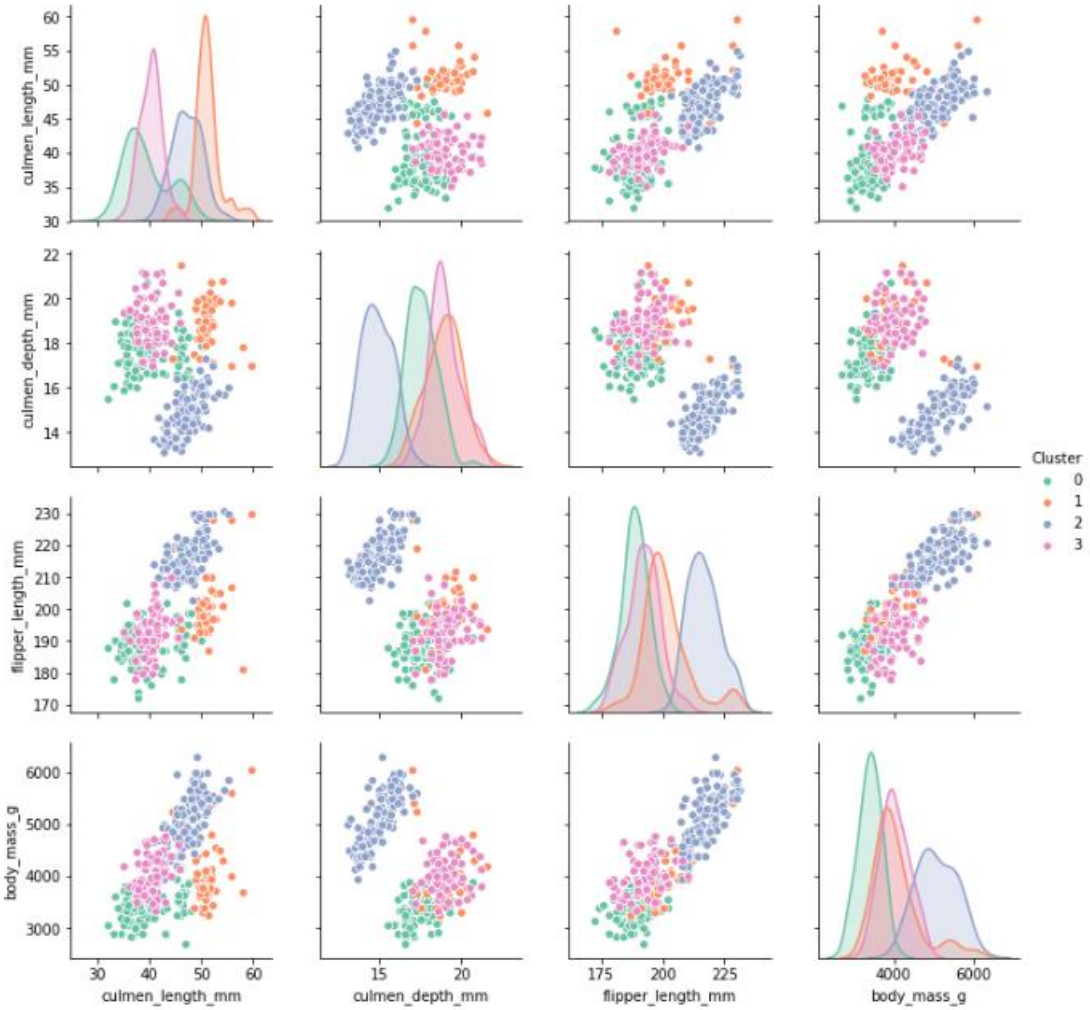
```
cluster_means = data.groupby('Cluster')[['culmen_length_mm', 'culmen_depth_mm', 'flipper_length_mm', 'body_mass_g']].mean()
cluster_means_rounded = cluster_means.round(3)
# Hiển thị kết quả đẹp hơn với tabulate
print(tabulate(cluster_means_rounded, headers='keys', tablefmt='pretty', showindex=True))
```

```
+---------+-------------------+-----------------+--------------------+-------------+
| Cluster | culmen_length_mm | culmen_depth_mm | flipper_length_mm | body_mass_g |
+---------+-------------------+-----------------+--------------------+-------------+
|    0    |       39.646      |      17.591     |       188.775      |   3408.088  |
|    1    |       51.243      |      18.968     |       201.568      |   4060.227  |
|    2    |       47.378      |      14.923     |       216.922      |   5075.652  |
|    3    |       40.393      |       19.01     |       192.31       |   4036.268  |
+---------+-------------------+-----------------+--------------------+-------------+
```

Ngoài ra, ta còn xác định được biểu đồ hiển thị sự phân cụm của các dữ liệu theo 4 thuộc tính (được đo lường bằng các đơn vị khác nhau). Mỗi thuộc tính được so sánh với nhau dưới dạng phân tán và phân bố theo từng cụm.

```
kmedoids = KMedoids(n_clusters=4, random_state=42)
data['Cluster'] = kmedoids.fit_predict(scaled)
```

```
sns.pairplot(data, hue='Cluster', palette='Set2', vars=['culmen_length_mm', 'culmen_depth_mm', 'flipper_length_mm', 'body_mass_g
plt.show()
```



4. Kết quả.

4.1. Phân tích từng đặc tính.

4.1.1. Culmen_length_mm.

- Độ dài culmen_length có sự phân biệt rõ rệt giữa các cụm, từ đó giúp nhận dạng và phân chia các cụm dựa trên kích thước.
    - Cụm 1 (màu xanh dương): Đây là cụm có độ dài culmen_length ngắn nhất, chủ yếu nằm trong khoảng 35-45 mm. Điều này cho thấy các cá thể trong cụm này có mỏ ngắn hơn, có thể liên quan đến các loài hoặc nhóm kích thước nhỏ hơn.
    - Cụm 0 (màu xanh lá): Cụm này có giá trị culmen_length trung bình, dao động từ 40-50 mm. Những cá thể này có độ dài mỏ không quá dài cũng không quá ngắn, có thể là nhóm có kích thước mỏ trung bình.
    - Cụm 2 (màu cam) và Cụm 3 (màu hồng): Hai cụm này có độ dài culmen_length lớn hơn, chủ yếu nằm trên 45 mm. Đặc biệt, Cụm 3 là cụm có độ dài

culmen_length lớn nhất, với nhiều giá trị vượt 55 mm. Điều này cho thấy những cá thể trong các cụm này có thể thuộc về nhóm loài hoặc giai đoạn phát triển có mỏ dài.

4.1.2. Culmen_depth_mm.

- Độ sâu culmen_depth cũng có sự khác biệt rõ ràng giữa các cụm, cho thấy đây là đặc tính hữu ích để phân biệt.

    - Cụm 1: Là cụm có độ sâu culmen_depth cao nhất, với phần lớn các giá trị nằm trên 18 mm. Điều này có thể cho thấy những cá thể này có mỏ dày hơn, phù hợp với những yêu cầu sinh học hoặc môi trường đặc biệt.
    - Cụm 0 và Cụm 2: Hai cụm này có độ sâu trung bình, dao động trong khoảng 14-17 mm. Những cá thể này có độ dày mỏ trung bình, không quá dày cũng không quá mỏng.
    - Cụm 3: Đây là cụm có độ sâu culmen_depth thấp nhất, với giá trị chủ yếu nằm dưới 16 mm. Điều này cho thấy các cá thể trong cụm này có mỏ mỏng hơn, có thể phù hợp với các môi trường hoặc loài cần mỏ mỏng để kiếm ăn hoặc sinh tồn.

4.1.3. Flipper_length_mm.

- Độ dài flipper_length giữa các cụm cho thấy sự khác biệt đáng kể, có thể liên quan đến khả năng bơi lội hoặc nhu cầu sinh học của từng loài.

    - Cụm 1: Có độ dài flipper_length ngắn nhất, chủ yếu nằm dưới 200 mm. Điều này có thể cho thấy các cá thể trong cụm này không cần vây quá dài, hoặc chúng là loài có vây ngắn.
    - Cụm 0 và Cụm 2: Độ dài vây của các cụm này dao động từ 200-220 mm, ở mức trung bình, cho thấy các cá thể có vây dài hơn cụm 1 nhưng ngắn hơn cụm 3.
    - Cụm 3: Đây là cụm có độ dài flipper_length cao nhất, chủ yếu nằm trên 210 mm, thậm chí có một số cá thể vượt quá 225 mm. Điều này có thể liên quan đến những loài cần vây dài hơn để di chuyển hoặc bơi lội hiệu quả.

4.1.4. Body_mass_g.

- Khối lượng cơ thể là đặc tính nổi bật để phân biệt kích thước tổng thể của các cá thể giữa các cụm.

    - Cụm 1: Đây là cụm có khối lượng cơ thể thấp nhất, phần lớn dưới 4000 g. Có thể đây là các cá thể nhỏ hơn, có thể là những loài có trọng lượng cơ thể thấp hoặc ở giai đoạn chưa trưởng thành.

- Cụm 0 và Cụm 2: Các cụm này có khối lượng cơ thể trung bình, chủ yếu dao động trong khoảng 3500-4500 g. Điều này cho thấy các cá thể thuộc cụm này có trọng lượng cơ thể lớn hơn cụm 1 nhưng không quá nặng.
- Cụm 3: Đây là cụm có khối lượng cơ thể cao nhất, với phần lớn các cá thể nặng trên 4500 g, thậm chí có một số cá thể vượt quá 6000 g. Những cá thể trong cụm này có thể thuộc loài lớn hoặc là những cá thể trưởng thành có kích thước và trọng lượng lớn.

⇨ Tóm lại: Mỗi đặc tính cung cấp thông tin quan trọng về sự khác biệt giữa các cụm. Cụm 1 đại diện cho nhóm cá thể có mỏ ngắn, dày và cơ thể nhỏ. Cụm 0 và Cụm 2 là những nhóm có đặc điểm trung bình về mỏ, vây, và trọng lượng. Cụm 3 là nhóm lớn nhất với mỏ dài, vây dài, và trọng lượng cơ thể lớn nhất. Điều này cho thấy các cụm có sự khác biệt rõ rệt về mặt sinh học, có thể phản ánh sự đa dạng loài hoặc giai đoạn phát triển của từng cá thể.

4.2. Phân tích đặc điểm của từng cụm.

Cụm 0:

- Loài chim trong cụm này có chiều dài mỏ trung bình khá thấp so với các cụm khác.
- Độ sâu mỏ tương đối vừa phải.
- Chiều dài vây thuộc mức trung bình, không quá lớn cũng không quá nhỏ.
- Khối lượng cơ thể trong khoảng trung bình, thấp hơn so với các cụm khác.

⇨ Cụm này có thể đại diện cho các loài chim có kích thước mỏ nhỏ đến vừa (dao động trong khoảng 40-50 mm), vây dài vừa phải (chủ yếu dao động từ 200-220 mm), và khối lượng cơ thể không quá lớn (chủ yếu nằm trong khoảng 3500-4500 g).

Cụm 1:

- Loài chim trong cụm này có chiều dài mỏ dài nhất, vượt trội so với các cụm còn lại.
- Độ sâu mỏ cao, tương đối lớn.
- Chiều dài vây cũng khá lớn, chỉ đứng sau cụm 2.
- Khối lượng cơ thể khá lớn, lớn hơn so với cụm 0 và cụm 3.

⇨ Cụm này có thể đại diện cho các loài chim có mỏ dài và sâu, vây lớn (chủ yếu trong khoảng 210-225 mm), và khối lượng cơ thể khá cao (trong khoảng 4000-5000 g), thường là những loài chim lớn và mạnh mẽ.

Cụm 2:

- Loài chim trong cụm này có chiều dài mỏ khá dài, tương tự cụm 1, nhưng độ sâu mỏ lại nhỏ hơn nhiều.
- Chiều dài vây dài nhất trong các cụm, có thể vượt quá 225 mm.

- Khối lượng cơ thể cao nhất trong các cụm, đạt trung bình khoảng 5075.65 g.

⇨ Cụm này có thể đại diện cho các loài chim có mỏ dài nhưng mỏng, vây dài nhất (thường trên 225 mm), và khối lượng cơ thể lớn nhất (thường trên 5000 g). Đặc điểm này cho thấy đây là những loài chim có kích thước cơ thể lớn và sức bền tốt, có thể thích hợp cho những hành trình di cư hoặc môi trường sống yêu cầu di chuyển xa.

Cụm 3:

- Chiều dài mỏ của loài chim trong cụm này tương đối vừa phải, không quá dài.
- Độ sâu mỏ cao nhất trong tất cả các cụm, vượt trội so với các cụm khác.
- Chiều dài vây và khối lượng cơ thể trung bình, tương đối lớn hơn so với cụm 0 nhưng không bằng cụm 2.

⇨ Cụm này có thể đại diện cho các loài chim có mỏ vừa và sâu, vây và cơ thể trung bình (dao động từ 210-225 mm cho vây và 4000-5000 g cho khối lượng). Những loài này có thể có khả năng thích ứng tốt với các môi trường đòi hỏi sức mạnh của mỏ, nhưng không cần di chuyển xa như các loài trong cụm 2.

BẢNG PHÂN CÔNG NHIỆM VỤ

Nhiệm vụ		Thành viên	Note
Nội dung	Chương 1: Tổng quan về phân cụm dữ liệu	Đỗ Thị Huệ	Hoàn thành tốt
		Tạ Phương Duy	
	Chương 2: Thuật toán K-Medoids	Đặng Hoàng Phúc	Hoàn thành tốt
		Trần Minh Tuấn	
	Chương 3: Ứng dụng phân cụm chim cánh cụt	Vũ Thị Minh Ngọc	Hoàn thành tốt
Báo cáo	Chương 1: Tổng quan về phân cụm dữ liệu	Đỗ Thị Huệ	Hoàn thành còn thiếu xót
	Chương 2: Thuật toán K-Medoids	Đặng Hoàng Phúc	
	Chương 3: Ứng dụng phân cụm chim cánh cụt	Vũ Thị Minh Ngọc	Hoàn thiện báo cáo