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

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN 3**

**Báo cáo thuật toán K-Means**

**GVHD: TS. Lê Văn Vinh**

**SVTH: Trần Ngọc Hùng**

**MSSV: 17110157**

**SVTH: Châu Huỳnh Phước Toàn**

**MSSV: 17110240**

**TP. Hồ Chí Minh, tháng 01 năm 2021**

**NHẬN XÉT CỦA GIẢNG VIÊN**

|  |  |
| --- | --- |
| **ĐIỂM** |  |

Giảng viên hướng dẫn

Lê Văn Vinh

# **Mục Lục**

[**Mục Lục** 2](#_Toc61720012)

[**LỜI CẢM ƠN** 4](#_Toc61720013)

[**DANH MỤC HÌNH VẼ** 5](#_Toc61720014)

[**CHƯƠNG 1: MỞ ĐẦU** 7](#_Toc61720015)

[**1.1 Lý do chọn đề tài** 7](#_Toc61720016)

[**1.2 Mục tiêu đề tài** 8](#_Toc61720017)

[**CHƯƠNG 2: TỔNG QUAN VỀ THUẬT TOÁN GOM CỤM (CLUSTERING)** 9](#_Toc61720020)

[**2.1 Giới thiệu chung** 9](#_Toc61720021)

[**2.2 Thuật toán gom cụm là gì ?** 9](#_Toc61720022)

[**2.3 Ví dụ về gom cụm** 10](#_Toc61720023)

[**2.4 Quy trình gom cụm** 12](#_Toc61720024)

[**2.5 Ứng dụng thực tế** 13](#_Toc61720025)

[**2.6 Một số các thuật toán gom cụm** 16](#_Toc61720026)

[**2.6.1 Phân loại các thuật toán gom cụm** 16](#_Toc61720027)

[**2.6.2 Một số các thuật toán gom cụm** 17](#_Toc61720028)

[**CHƯƠNG 3: TỔNG QUAN VỀ THUẬT TOÁN GOM CỤM K-MEANS (K-MEANS CLUSTERING)** 19](#_Toc61720029)

[**3.1 Giới thiệu chung** 19](#_Toc61720030)

[**3.2 Thuật toán gom cụm K-Means là gì ?** 20](#_Toc61720031)

[**3.3 Các bước thực hiện thuật toán K-means** 22](#_Toc61720032)

[**3.3.1 Phương pháp khởi tạo tâm cụm (Centroid)** 23](#_Toc61720033)

[**3.3.2 Accelerated K-Means và Mini-batch K-Means** 25](#_Toc61720034)

[**3.3.3 Tìm số lượng nhóm tối ưu** 26](#_Toc61720035)

[**3.4 Tổng kết** 30](#_Toc61720036)

[**3.4.1 Tổng quát** 30](#_Toc61720037)

[**3.4.2 Ưu điểm của thuật toán** 30](#_Toc61720038)

[**3.4.3 Nhược điểm của thuật toán** 31](#_Toc61720039)

[**3.4.4 Các giải pháp thay thế** 32](#_Toc61720040)

[**CHƯƠNG 4: CHƯƠNG TRÌNH MINH HỌA** 33](#_Toc61720041)

[**4.1 Cấu hình máy và phần mềm sử dụng** 33](#_Toc61720042)

[**4.2 Đầu vào và kết quả thuật toán K-Mean** 33](#_Toc61720043)

[**CHƯƠNG 5: KẾT LUẬN** 38](#_Toc61720044)

[**5.1 Kiến thức tìm hiểu được** 38](#_Toc61720045)

[**5.2 Khó khăn** 38](#_Toc61720046)

[**5.3 Hướng phát triển** 38](#_Toc61720047)

[**CHƯƠNG 6: TÀI LIỆU THAM KHẢO** 39](#_Toc61720048)

# **LỜI CẢM ƠN**

Qua đề tài này chúng em đã có thêm những kiến thức về học máy. Chúng em xin chân thành gửi lời cảm ơn đến những người đã hỗ trợ, giúp đỡ chúng em rất nhiều trong thời gian qua thực hiện đề tài.

Đầu tiên chúng em xin gửi lời cảm ơn tới các Thầy Cô trong trường Đại Học Sư Phạm Kỹ Thuật TP.Hồ Chí Minh, đặc biệt là các Thầy Cô trong Khoa Công Nghệ Thông Tin và Khoa Đào Tạo Chất Lượng Cao, đã nhiệt tình giảng dạy, trao dồi, vun đắp kiến thức cho chúng em trong nhiều năm ngồi trên ghế nhà trường.

Đặc biệt chúng em chân thành cảm ơn Thầy Lê Văn Vinh trong thời gian qua đã cung cấp cho chúng em những kiến thức về cũng như tài liệu rất hữu ích và hướng dẫn chi tiết trong quá trình chúng em làm để tài. Thầy luôn giải đáp tận tình những vướng mắc khi chúng em gặp phải. Và chúng em cũng xin gửi lời cảm ơn đến bạn bè trong và ngoài lớp - những người bạn đã cùng chia sẻ, trao đổi kiến thức với chúng em.

Trong phạm vi khả năng cho phép, chúng em đã rất cố gắng để hoàn thành đề tài một cách tốt nhất. Tuy nhiên không thể tránh khỏi nhiều thiếu sót, chúng em kính mong nhận được sự cảm thông và những ý kiến đóng góp của quý thầy cô và các bạn.

Sinh viên thực hiện

Trần Ngọc Hùng

Châu Huỳnh Phước Toàn

# **DANH MỤC HÌNH VẼ**

[Hình 2. 1 Quy trình gom cụm dữ liệu 11](#_Toc61720721)

[Hình 3. 1.Thuật toán K-Means 20](#_Toc61720726)

[Hình 3. 2. Các giải pháp dưới mức tối ưu do khởi tạo tâm cụm không may mắn (giá trị tâm cụm lệch so với mức tối ưu) 20](#_Toc61720727)

[Hình 3. 3. Các bước thực hiện thuật toán K-Means 22](#_Toc61720728)

[Hình 3. 4. Ví dụ kết quả phân cụm với các giá trị tâm cụm k 22](#_Toc61720729)

[Hình 3. 5. K-Means loạt nhỏ so với K-Means: quán trính kém hơn khi k tăng (trái) nhứng nhanh hơn nhiều(phải) 25](#_Toc61720730)

[Hình 3. 6. Kết quả phân cụm không tốt khi lựa chọn giá trị tâm cụm không chính xác 26](#_Toc61720731)

[Hình 3. 7. Chọn số lượng cụm k bằng cách sử dụng “ellbow rule” 26](#_Toc61720732)

[Hình 3. 8. Chọn số lượng cụm k bằng cách sử dụng điểm hình bóng 27](#_Toc61720733)

[Hình 3. 9. Phân tích Silhouette: so sánh các sơ đồ hình bóng cho các giá trị khác nhau của k. 28](#_Toc61720734)

[Hình 4. 1. Implement các thư viện môi trường. 32](#_Toc61720735)

[Hình 4. 2. : Khởi tạo tập dữ liệu 32](#_Toc61720736)

[Hình 4. 3. Kết quả dataset đã khởi tạo dữ liệu 33](#_Toc61720737)

[Hình 4. 4. : Import thư viện K-Means và khởi tạo các tâm cụm. 33](#_Toc61720738)

[Hình 4. 5. Kết quả sau khi khởi tạo K-Means 34](#_Toc61720739)

[Hình 4. 6. Thể hiện tập điểm dữ liệu của tập dữ liệu lên biểu đồ 34](#_Toc61720740)

[Hình 4. 7. Sử dụng thư viện KMean để tìm tọa độ tâm cụm 35](#_Toc61720741)

[Hình 4. 8. In ra các giá trị tâm cụm. 35](#_Toc61720742)

[Hình 4. 9. Sử dụng thư viện matplotlib để thể hiện các cụm và tâm cụm. 35](#_Toc61720743)

[Hình 4. 10. Hiển thị dữ liệu trên biểu đồ 36](#_Toc61720744)

# **CHƯƠNG 1: MỞ ĐẦU**

## **1.1 Lý do chọn đề tài**

Với mong muốn tiếp cận với bộ môn học máy và cụ thể hơn là các kiến thức chung,cách sử dụng và cài đặt các thuật toán liên quan đến việc phân cụm dữ liệu dựa trên các thuộc tính từ các tập dữ liệu thu thập được, nhóm chúng em đã chọn đề tài về thuật toán K-Means thuộc nhóm gom cụm (clustering) để tìm hiểu và thực hiện triển khai theo mong muốn học hỏi đã đề ra của nhóm. Ngày nay, các công việc liên quan tới các tác vụ phân tích và gọm nhóm gom cụm dữ liệu ngày càng nhiều, nhất là trong lĩnh vực phân tích hành vi người dùng nhằm nâng cao trải nghiệm sử dụng của khách hàng, ngoài ra các lĩnh vực kinh doanh cũng sử dụng các dữ liệu phân tích nhằm xác định các xu hướng hiện thời, phân tích thị trường hiện tại và còn rất nhiều các nghành nghề khác cũng đang áp dụng các thuật toán học máy cụ thể hơn là các thuật toán liên quan tới việc phân cụm dữ liệu. Phân tích cụm nhằm mục đích xác định các nhóm đối tượng tương tự và do đó giúp khám phá sự phân biệt của các mẫu và mối tương quan trong các tập dữ liệu lớn. Nó đã là chủ đề nghiên cứu rộng rãi vì nó nổi lên trong nhiều lĩnh vực ứng dụng trong kỹ thuật, kinh doanh và khoa học xã hội. Đặc biệt, trong những năm qua, với sự sẵn có của các bộ dữ liệu giao dịch và số lượng bộ dữ liệu thử nghiệm khổng lồ và các yêu cầu phát sinh để khai thác dữ liệu đã tạo ra nhu cầu cho các thuật toán phân cụm mở rộng quy mô và có thể được áp dụng trong các lĩnh vực khác nhau. Hơn nữa, nó giải quyết một vấn đề quan trọng của quá trình phân cụm liên quan đến việc đánh giá chất lượng các kết quả phân nhóm. Điều này cũng liên quan đến các tính năng vốn có của tập dữ liệu đang được quan tâm. Hiện nay, các thuật toán gom nhóm phân cụm dữ liệu cho việc học máy có số lượng rất nhiều cũng như rất đa dạng về cách cài đặt, phân tích, gom cụm dữ liệu, cách thức vận hành… trong phạm vi tìm hiểu các kiến thức cơ bản đã được học và giải thích trên lớp trên lớp nhóm chúng em xin trình bày những hiểu biết chung về thuật toán K-Means mà nhóm đã đúc kết trong quá trình học tập. Với tính chất dễ tiếp cận, nhóm em mong muốn thông qua đề tài có thể hiểu hơn về học máy để có thể áp dụng vào thực tế và các đồ án sau này.

Các từ khóa trọng tâm (keywords) giúp chúng ta có thể tìm hiểu 1 cách sâu và mở rộng hơn các kiến thức trong báo cáo này: thuật toán phân cụm (clustering algorithms), học không giám sát (unsupervised learning), tính hợp lệ của cụm (cluster validity), chỉ số hợp lệ (validity indice), thuật toán K-Means (K-Means algorithm).

## **1.2 Mục tiêu đề tài**

## Tìm hiểu được các cài đặt cơ bản và cách thức vận hành của thuật toán K-Means.

Nắm được các trường hợp sử dụng và áp dụng cài đặt vào bài toán cụ thể.

## Hoàn thành tập tài liệu nghiên cứu sử dụng thuật toán K-Means.

# **CHƯƠNG 2: TỔNG QUAN VỀ THUẬT TOÁN GOM CỤM (CLUSTERING)**

## **2.1 Giới thiệu chung**

Phân cụm là một trong những tác vụ hữu ích nhất trong quá trình khai thác dữ liệu để khám phá các nhóm và xác định các phân phối (distributions) và mẫu (patterns) trong bộ dữ liệu cơ bản. Bài toán phân cụm là việc phân vùng một tập dữ liệu nhất định thành các nhóm (cụm) sao cho các điểm dữ liệu trong cụm giống nhau hơn so với các điểm trong các cụm khác nhau (Guha et al., 1998). Ví dụ: hãy xem xét một bản ghi cơ sở dữ liệu bán lẻ chứa các mặt hàng mà khách hàng đã mua. Một quy trình phân nhóm có thể nhóm các khách hàng theo cách mà những khách hàng có thói quen mua hàng giống nhau ở trong cùng một nhóm. Do đó, mối quan tâm chính trong quy trình phân nhóm là tiết lộ việc tổ chức các mẫu thành các nhóm “hợp lý”, cho phép chúng ta phát hiện ra các điểm giống và khác nhau, cũng như rút ra các kết luận hữu ích về chúng. Ý tưởng này có thể áp dụng trong nhiều lĩnh vực, chẳng hạn như khoa học đời sống, khoa học y tế và kỹ thuật. Phân cụm có thể được tìm thấy dưới các tên khác nhau trong các ngữ cảnh khác nhau, chẳng hạn như học tập không giám sát (trong nhận dạng mẫu), phân loại số (trong sinh học, sinh thái học), phân loại học (trong khoa học xã hội) và phân vùng (trong lý thuyết đồ thị) (Theodoridis và Koutroubas, 1999). Trong quá trình phân cụm, không có các lớp được định nghĩa trước và không có ví dụ nào cho thấy loại quan hệ mong muốn nào sẽ hợp lệ giữa dữ liệu, đó là lý do tại sao nó được coi là một quá trình không được giám sát (Berry và Linoff, 1996). Mặt khác, phân loại là bắt buộc gán một mục dữ liệu cho một tập hợp các danh mục định nghĩa trước (Fayyad và cộng sự, 1996). Việc gộp tạo ra các danh mục ban đầu trong đó các giá trị của tập dữ liệu được phân loại trong quá trình phân loại lớp.

## **2.2 Thuật toán gom cụm là gì ?**

Phân cụm là :

* Kỹ thuật rất quan trọng trong khai phá dữ liệu [1]
* Là một kỹ thuật học tập không giám sát (unsupervised learning), chúng ta sẽ cần phải cố gắng nhóm các điểm dữ liệu dựa trên các đặc điểm cụ thể [1]
* Có rất nhiều định nghĩa khác nhau về kỹ thuật này [1]
* Về bản chất có thể hiểu phân cụm là các quy trình tìm cách nhóm các đối tượng đã cho vào các cụm (clusters), sao cho các đối tượng trong cùng một cụm tương tự nhau và các đối tượng khác cụm thì không tương tự nhau [1]
* Mục đích của phân cụm là tìm ra bản chất bên trong các nhóm của dữ liệu [1]
* Tuy nhiên, không có tiêu chí nào được xem là tốt nhất để đánh giá hiệu quả của phân tích phân cụm, điều này phụ thuộc vào mục đích của phân cụm như: giảm dữ liệu (data reduction), “cụm tự nhiên” (“natural clusters”), cụm “có ích” (“useful” clusters), phát hiện phần tử ngoại lai (outlier detection). [1]
* Trong thực tế sẽ có rất nhiều các tác vụ liên quan đến việc phân cụm nhóm dữ liệu như: phân cụm tài liệu (Document Clustering), công cụ khuyến nghị (Recommendation Engine), phân đoạn hình ảnh (Image Segmentation), phân khúc thị trường (Market Segmentation), nhóm kết quả tìm kiếm (Search Result Grouping), phát hiện bất thường (Anomaly Detection) và còn rất nhiều tác vụ khác. [1]

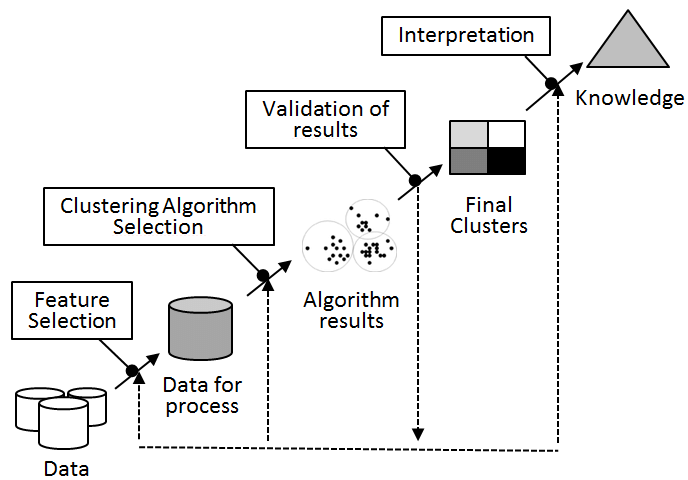
## **2.3 Ví dụ về gom cụm**

Giả sử có một công ty định ra mắt một mẫu sản phẩm mới T-shirt vào thị trường. Tất nhiên họ sẽ phải sản xuất rất nhiều size để phù hợp với sự đa dạng của thị trường người dùng. Với định hướng đo, công ty đã tiến hành khảo sát dữ liệu chiều cao và cân nặng của người dùng. Công ty này ko thể đủ nguồn lực để có thể sản xuất áo với tất cả mọi size. Thực tế này trong kinh doanh bạn cũng dễ dàng hiểu được. Thay vì đó, họ sẽ chia số lượng người dùng thành các size như là Small, Medium, Large và sản xuất chỉ 3 mẫu như thế. 3 mẫu này là đủ khớp với tất cả mọi người và thị trường. Ở đây việc phân chia các người dùng vào 3 nhóm trên sẽ được xử lý bằng kỹ thuật gom cụm Thuật toán này sẽ cho ta 3 size áo tối ưu nhất - thỏa mãn tất cả mọi người. Tất nhiên, nếu như ko thể tìm được 3 size áo vừa vặn thoả mãn mọi người trong nhóm, công ty sẽ chia nhỏ nhóm thêm thành nhiều nhóm khác, có thể là 5, có thể là nhiều hơn nữa [1].

Một ví dụ khác, khi bạn đi bộ trên núi, bạn tình cờ gặp một loại cây mà chưa từng thấy trước đây. Bạn nhìn xung quanh và nhận thấy một vài điều nữa rằng chúng không hoàn toàn giống nhau, nhưng chúng có đủ các đặc tính tương đồng để chúng ta biết rằng rất có thể các loại cây ấy thuộc cùng một loài (hoặc ít nhất là cùng một chi). Bạn có thể cần một nhà thực vật học cho bạn biết đó là loài gì, nhưng bạn chắc chắn không cần một chuyên gia để xác định các nhóm vật thể có hình dáng giống nhau. Đây được gọi là phân cụm: xác định các cá thể tương tự và gán chúng cho các cụm, tức là các nhóm các cá thể tương tự. Cũng giống như trong phân loại, mỗi cá thể được gán cho một nhóm. Tuy nhiên, đây là một nhiệm vụ không được giám sát (unsupervised task), trong đó loài của mỗi cá thể (tức là lớp của nó) được thể hiện bằng một điểm đánh dấu khác nhau. Nó là một tập dữ liệu được gắn nhãn mà các thuật toán phân loại như Hồi quy logistic (Logistic Regression), SVMs hoặc bộ phân loại Rừng ngẫu nhiên (Random Forest) rất phù hợp. Với việc cùng một tập dữ liệu, nhưng không có nhãn, chúng ta không thể sử dụng thuật toán phân loại nữa. Đây là lúc các thuật toán phân cụm được áp dụng: nhiều trong số chúng có thể dễ dàng phát hiện ra cụm trên cùng bên trái. Chúng ta cũng có thể dễ dàng nhìn thấy bằng mắt thường nhưng không quá rõ ràng rằng cụm thực sự bao gồm hai cụm con riêng biệt. Điều đó nói rằng, tập dữ liệu thực sự có hai tính năng bổ sung (chiều dài và chiều rộng riêng), các thuật toán phân nhóm có thể sử dụng tốt tất cả các đặc điểm của dữ liệu, vì vậy thực tế là chúng xác định ba cụm khá tốt (ví dụ: sử dụng mô hình hỗn hợp Gaussian, chỉ có 5 trường hợp trong số 150 trường hợp được gán cho cụm sai).

Trong thực tế, chúng ta không hề có bất kỳ định nghĩa chung nào về cụm (cluster) là gì, nó luôn phụ thuộc vào ngữ cảnh sử dụng và các thuật toán khác nhau sẽ phù hợp cũng như tối ưu cho việc xử lý từng loại cụm khác nhau. Ví dụ, một số thuật toán tìm kiếm các điểm dữ liệu tập trung xung quanh một điểm cụ thể được gọi là tâm cụm (centroid). Những thuật toán khác tìm kiếm các vùng liên tục (continuous regions) của các cụm dữ liệu có mật độ phân bố dày đặc, những cụm này có thể có hình dạng bất kỳ. Một số thuật toán có thứ bậc (hierarchical) tìm kiếm các cụm của từng cụm (clusters of clusters). Trong báo cáo này, chúng em sẽ tìm hiểu về thuật toán phân cụm phổ biến hiện nay K-Means và tìm hiểu các ứng dụng thực tế dựa trên cách vận hành cũng như đặc tính của nó, chẳng hạn như giảm kích thước phi tuyến tính (non-linear dimensionality reduction), học bán giám sát (semi-supervised learning) và phát hiện dị thường (anomaly detection).

## **2.4 Quy trình gom cụm**



Hình 2. 1 Quy trình gom cụm dữ liệu

Quá trình phân cụm có thể dẫn đến việc phân vùng khác nhau của một tập dữ liệu, tùy thuộc vào tiêu chí cụ thể được sử dụng để phân nhóm. Do đó, cần phải tiền xử lý trước khi chúng ta phân tích một tác vụ phân cụm trong một tập dữ liệu. Các bước cơ bản để phát triển quy trình phân cụm có thể được tóm tắt như sau (Fayyad et al., 1996):

Lựa chọn tính năng (**Feature selection**): mục tiêu của quá trình này là chọn đúng các đặc tính mà phân cụm sẽ được thực hiện để mã hóa càng nhiều thông tin càng tốt liên quan đến nhiệm vụ mà bạn quan tâm. Do đó, việc xử lý trước dữ liệu có thể cần thiết trước khi sử dụng chúng.

Thuật toán phân cụm (**Clustering algorithm**): bước này đề cập đến việc lựa chọn một thuật toán dẫn đến việc xác định một lược đồ phân cụm tốt cho một tập dữ liệu. Thước đo độ gần (proximity measure) và tiêu chuẩn phân cụm (clustering criterion) chủ yếu đặc trưng cho thuật toán phân cụm cũng như hiệu quả của nó để xác định sơ đồ phân nhóm tương ứng với tập dữ liệu.

*Độ đo lân cận (Proximity measure):* là thước đo định lượng mức độ “giống nhau” của hai điểm dữ liệu (tức là vectơ đặc tính). Trong hầu hết các trường hợp, chúng ta phải đảm bảo rằng tất cả các tính năng đã chọn đều phân bổ như nhau cho việc tính toán độ gần nhau và không có đặc tính nào lấn át các đặc tính khác.

*Tiêu chí phân nhóm (Clustering criterion):* trong bước này, chúng ta phải xác định tiêu chí phân cụm, tiêu chí này có thể được biểu thị thông qua hàm chi phí (cost function) hoặc một số loại quy tắc khác. Chúng ta nên bắt đầu từ việc phải tính đến loại cụm dự kiến ​​sẽ xuất hiện trong tập dữ liệu. Do đó, chúng ta có thể xác định tiêu chí phân nhóm "tốt", dẫn đến việc phân vùng có lợi cho tập dữ liệu

Xác nhận kết quả (**Validation of the results**): tính đúng đắn của các kết quả thuật toán phân cụm được xác minh bằng cách sử dụng các tiêu chí và kỹ thuật thích hợp. Vì các thuật toán phân cụm bao gồm các cụm không được biết trước, bất kể phương pháp phân nhóm nào thì phân vùng cuối cùng của dữ liệu yêu cầu một số loại đánh giá trong hầu hết các ứng dụng (Rezaee et al., 1998).

Giải thích kết quả (**Interpretation of the results**): trong nhiều trường hợp, các chuyên gia trong lĩnh vực ứng dụng phải tích hợp các kết quả phân nhóm với các bằng chứng và phân tích thực nghiệm khác để rút ra kết luận đúng.

## **2.5 Ứng dụng thực tế**

Phân cụm được sử dụng trong nhiều ứng dụng, bao gồm:

* Phân khúc khách hàng (customer segmentation): chúng ta có thể phân nhóm khách hàng dựa trên lịch sử và thói quen mua hàng của họ, các hoạt động trên ứng dụng, trang web hoặc nền tảng của bạn, xác định tính cách (personas) dựa trên sở thích (interests), tạo hồ sơ dựa trên giám sát hoạt động v.v. Điều này rất hữu ích để hiểu khách hàng của bạn là ai và họ cần gì, từ đó có thể điều chỉnh sản phẩm và chiến dịch tiếp thị của mình cho từng phân khúc. Ví dụ: điều này có thể hữu ích trong các hệ thống đề xuất (recommender systems) để đề xuất nội dung mà những người dùng khác trong cùng một cụm yêu thích.
* Phân tích dữ liệu (data analysis): khi phân tích một tập dữ liệu mới, sẽ hữu ích hơn rất nhiều nếu trước tiên chúng ta có thể phát hiện ra các cụm của các trường hợp tương tự, vì việc phân tích các cụm riêng biệt thường dễ dàng hơn.
* Là một kỹ thuật giảm kích thước dữ liệu (data reduction):đối với 1 tập dữ liệu đã được phân nhóm chúng ta thường có thể đo lường mức độ phù hợp của từng điểm dữ liệu với từng cụm (sức hấp dẫn là bất kỳ thước đo nào về mức độ phù hợp của một đối tượng với một cụm). Sau đó, vectơ đặc trưng x của mỗi đối tượng có thể được thay thế bằng vectơ của các mối quan hệ cụm của nó. Nếu có k cụm, thì vectơ này là k chiều. Chiều này thường thấp hơn nhiều so với vector đặc trưng ban đầu, nhưng nó có thể bảo toàn đủ thông tin để xử lý thêm.
* Phát hiện dị thường [anomaly detection] (còn gọi là phát hiện ngoại lệ [outlier detection]): bất kỳ trường hợp nào có ái lực (affinity) thấp với tất cả các cụm dữ liệu đều có khả năng là dị thường. Ví dụ: nếu bạn đã nhóm những người dùng trang web của mình dựa trên hành vi của họ, bạn có thể phát hiện những người dùng có hành vi bất thường, chẳng hạn như số lượng yêu cầu bất thường mỗi giây, v.v. Phát hiện bất thường đặc biệt hữu ích trong việc phát hiện các khiếm khuyết trong quá trình sản xuất hoặc phát hiện gian lận.
* Học bán giám sát (semi-supervised learning): nếu bạn chỉ có một vài nhãn, bạn có thể thực hiện phân cụm và truyền các nhãn cho tất cả các phiên bản trong cùng một cụm. Điều này có thể làm tăng đáng kể số lượng nhãn có sẵn cho thuật toán học tập có giám sát ở lần tiếp theo giúp cải thiện hiệu suất của nó.
* Công cụ tìm kiếm (search engines): ví dụ, một số công cụ tìm kiếm cho phép bạn tìm kiếm những hình ảnh tương tự như một hình ảnh tham chiếu. Để xây dựng một hệ thống như vậy, trước tiên bạn sẽ áp dụng thuật toán phân cụm cho tất cả các hình ảnh trong cơ sở dữ liệu của mình: các hình ảnh tương tự sẽ kết thúc trong cùng một cụm. Sau đó, khi người dùng cung cấp hình ảnh tham chiếu, tất cả những gì bạn cần làm là tìm cụm của hình ảnh này bằng cách sử dụng nhóm được đào tạo và sau đó bạn có thể chỉ cần trả lại tất cả các hình ảnh từ cụm này.
* Để phân đoạn hình ảnh (segment an image): bằng cách nhóm các pixel theo màu của chúng, sau đó thay thế màu của mỗi pixel bằng màu trung bình của cụm của nó, có thể giảm đáng kể số lượng màu khác nhau trong hình ảnh. Kỹ thuật này được sử dụng trong nhiều hệ thống phát hiện và theo dõi đối tượng do nó có khả năng giúp phát hiện đường viền của từng đối tượng dễ dàng hơn.
* Phân loại các phép đo cảm biến (Sorting sensor measurements): phát hiện các loại hoạt động trong cảm biến chuyển động (motion sensors), nhóm loại hình ảnh (group images),phân tácg âm thanh (separate audio),xác định các nhóm trong giám sát sức khỏe (health monitoring).
* Phân loại hàng tồn kho (Inventory categorization):nhóm hàng tồn kho theo hoạt động bán hàng (sales activity),nhóm hàng tồn kho theo chỉ số sản xuất (manufacturing metrics).
* Ngoài ra còn 1 số trường hợp áp dụng vào các lĩnh vực cụ thể như kinh doanh (Business). Trong kinh doanh, phân nhóm có thể giúp các nhà tiếp thị phát hiện ra các nhóm đáng kể trong cơ sở dữ liệu của khách hàng của họ và mô tả đặc điểm của chúng dựa trên các mô hình mua hàng. Sinh học (Biology): trong sinh học, phân cụm có thể được sử dụng để xác định các đơn vị phân loại, phân loại các gen có chức năng tương tự và hiểu sâu hơn về các cấu trúc vốn có trong quần thể. Phân tích dữ liệu không gian (Spatial data analysis): do lượng dữ liệu không gian khổng lồ có thể thu được từ hình ảnh vệ tinh, thiết bị y tế, Hệ thống thông tin địa lý (GIS), thăm dò cơ sở dữ liệu ảnh, v.v., việc kiểm tra chi tiết dữ liệu không gian là rất tốn kém và khó khăn cho người dùng. Phân cụm có thể giúp tự động hóa quá trình phân tích và hiểu dữ liệu không gian. Nó được sử dụng để xác định và trích xuất các đặc điểm và mẫu thú vị có thể tồn tại trong cơ sở dữ liệu không gian lớn. Các thuật toán phân cụm còn được áp dụng rất nhiều vào các lĩnh vực khác trong đời sống của chúng ta.
* Khai thác web (web mining): trong trường hợp này, phân cụm được sử dụng để khám phá các nhóm tài liệu rất lớn trong kho lưu trữ tài liệu bán cấu trúc khổng lồ của Web. Phân loại của tài liệu Web này hỗ trợ việc khám phá thông tin

## **2.6 Một số các thuật toán gom cụm**

## **2.6.1 Phân loại các thuật toán gom cụm**

Các thuật toán phân cụm có thể phân loại theo:

* Loại dữ liệu đầu vào cho thuật toán.
* Tiêu chí phân nhóm xác định sự giống nhau giữa các điểm dữ liệu.
* Lý thuyết và các khái niệm cơ bản dựa trên kỹ thuật phân tích phân cụm (ví dụ: lý thuyết mờ, thống kê).

Do đó, theo phương pháp được áp dụng cho các cụm phân loại, các thuật toán có thể được phân loại rộng rãi thành các loại sau (Jain et al., 1999):

* Phân cụm từng phần (Partitional clustering): chúng ta sẽ cố gắng phân rã trực tiếp tập dữ liệu thành một tập hợp các nhóm rời rạc. Cụ thể hơn, chúng ta cố gắng xác định một số nguyên phân vùng tối ưu hóa một hàm tiêu chí nhất định. Hàm tiêu chí (criterion function) có thể nhấn mạnh cấu trúc tổ chức cục bộ của dữ liệu và việc tối ưu hóa nó là một quy trình lặp đi lặp lại.
* Phân cụm phân cấp (Hierarchical clustering): tiến hành liên tiếp bằng cách hợp nhất các cụm nhỏ hơn vào trong các cụm lớn hơn hoặc bằng cách tách các cụm lớn hơn. Kết quả của thuật toán là một cây gồm các cụm, được gọi là dendrogram, cho biết các cụm liên quan với nhau như thế nào. Bằng cách cắt dendrogramat ở mức mong muốn, sẽ thu được một nhóm các mục dữ liệu thành các nhóm riêng biệt.
* Phân cụm dựa trên mật độ (Density-based clustering): ý tưởng chính của loại phân cụm này là nhóm các đối tượng lân cận của một tập dữ liệu thành các cụm dựa trên điều kiện mật độ.
* Phân cụm dựa trên lưới (Grid-based clustering): loại thuật toán này chủ yếu được đề xuất để khai thác dữ liệu không gian. Đặc điểm chính của chúng là chúng định lượng không gian thành một số lượng nhỏ các ô và sau đó chúng thực hiện tất cả các hoạt động trên không gian lượng tử (quantised space).

## **2.6.2 Một số các thuật toán gom cụm**

* Bên cạnh các thuật toán phân cụm thông dụng và đơn giản như K-Means, DBSCAN, mô hình kết hợp Gaussian (Gaussian mixture models), Scikit-Learn triển khai một số thuật toán phân cụm khác mà chúng ta nên tìm hiểu:
* Phân cụm tích hợp (Agglomerative clustering): một hệ thống phân cấp các cụm được xây dựng từ dưới lên, bao gồm nhiều bong bóng nhỏ nổi trên mặt nước và dần dần gắn vào nhau cho đến khi chỉ còn một nhóm bong bóng lớn. Tương tự, tại mỗi lần lặp lại, phân cụm tích tụ kết nối cặp cụm gần nhất (bắt đầu với các cá thể riêng lẻ). Nếu bạn vẽ một cây với một nhánh cho mỗi cặp cụm đã hợp nhất, bạn sẽ có được một cây nhị phân gồm các cụm, trong đó các lá là các điểm dữ liệu riêng lẻ. Cách tiếp cận này chia tỷ lệ rất tốt với số lượng lớn các điểm dữ liệu hoặc cụm, nó có thể nắm bắt các cụm có nhiều hình dạng (shape) khác nhau, tạo ra một cây cụm linh hoạt và nhiều thông tin thay vì buộc bạn phải chọn một tỷ lệ cụm cụ thể và nó có thể được sử dụng với bất kỳ khoảng cách theo cặp nào. Thuật toán này có thể mở rộng số lượng lớn các điểm dữ liệu nếu bạn cung cấp ma trận kết nối. Đây là một ma trận m x m chỉ ra các cặp dữ liệu nào là hàng xóm của nhau (ví dụ: trả về bởi sklearn.neighbors.kneighbors\_graph ()). Nếu không có ma trận kết nối, thuật toán không mở rộng quy mô tốt đến các tập dữ liệu lớn.
* Birch: thuật toán này được thiết kế đặc biệt cho các tập dữ liệu rất lớn và nó có thể nhanh hơn K-Means batch với cùng kết quả tương tự, miễn là số lượng đặc tính không quá lớn (<20). Thuật toán này xây dựng một cấu trúc cây trong quá trình đào tạo có chứa đủ thông tin để nhanh chóng gán từng phiên bản mới cho một cụm, mà không cần phải lưu trữ tất cả các cá thể trong cây: điều này cho phép nó sử dụng bộ nhớ hạn chế, đồng thời xử lý các tập dữ liệu khổng lồ.
* MeanShift: thuật toán này bắt đầu bằng cách đặt một vòng tròn ở giữa mỗi trường hợp, sau đó đối với mỗi vòng tròn chúng ta sẽ tính giá trị trung bình của tất cả các điểm dữ liệu nằm bên trong nó và thuật toán sẽ dịch chuyển vòng tròn để nó được căn giữa vào giá trị trung bình. Tiếp theo, nó lặp lại bước dịch chuyển trung bình này cho đến khi tất cả các vòng tròn ngừng di chuyển (nghĩa là cho đến khi mỗi vòng tròn được căn giữa vào trung bình của các điểm dữ liệu mà nó chứa). Thuật toán này dịch chuyển các vòng tròn theo hướng có mật độ cao hơn, cho đến khi mỗi vòng tròn tìm thấy mật độ cục bộ tối đa. Cuối cùng, tất cả các điểm dữ liệu có các vòng kết nối đã ổn định ở cùng một vị trí (hoặc đủ gần) được gán cho cùng một cụm. Điều này có một số điểm tương đồng như DBSCAN, đặc biệt nó có thể tìm thấy bất kỳ số lượng cụm có hình dạng nào, nó chỉ có một hyperparameter là bán kính của các vòng tròn, được gọi là băng thông (bandwidth) và nó dựa vào ước tính mật độ cục bộ. Tuy nhiên, thuật toán này có xu hướng chia các cụm thành nhiều mảnh khi chúng có sự thay đổi mật độ bên trong. Thật không may, độ phức tạp tính toán của nó là O (m2), vì vậy nó không phù hợp với các bộ dữ liệu lớn.
* Truyền bá Anity (Anity propagation): thuật toán này sử dụng hệ thống bỏ phiếu (voting system), trong đó các điểm dữ liệu bỏ phiếu cho các cá thể tương tự là đại diện của chúng và khi thuật toán hội tụ, mỗi đại diện và những đại diện bỏ phiếu của nó sẽ tạo thành một cụm. Thuật toán này có thể phát hiện bất kỳ số lượng cụm có kích thước khác nhau. Thật không may,độ phức tạp tính toán của thuật toán này của O (m2), vì vậy nó không phù hợp với các bộ dữ liệu lớn.
* Phân cụm quang phổ (Spectral clustering): thuật toán này lấy một ma trận tương tự giữa các điểm dữ liệu và tạo ra một nhúng chiều thấp từ nó (tức là nó làm giảm số chiều của nó), sau đó nó sử dụng một thuật toán phân cụm khác trong không gian chiều thấp này (triển khai của Scikit-Learn sử dụng K-Means). Phân cụm quang phổ có thể xử lý các cấu trúc cụm phức tạp và nó cũng có thể được sử dụng để cắt biểu đồ (ví dụ: để xác định các nhóm bạn bè trên mạng xã hội), tuy nhiên, nó không mở rộng quy mô phù hợp với số lượng lớn các trường hợp và nó không hoạt động tốt khi các cụm có kích thước rất khác nhau.

# **CHƯƠNG 3: TỔNG QUAN VỀ THUẬT TOÁN GOM CỤM K-MEANS (K-MEANS CLUSTERING)**

## **3.1 Giới thiệu chung**

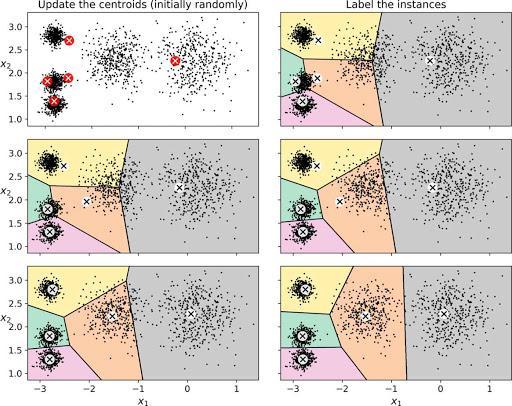
Thuật toán K-Means là một thuật toán đơn giản có khả năng phân cụm loại tập dữ liệu rất nhanh chóng và hiệu quả chỉ với một vài lần lặp. Nó được đề xuất bởi Stuart Lloyd tại Phòng thí nghiệm Bell vào năm 1957 như một kỹ thuật điều chế mã xung, nhưng nó chỉ được xuất bản bên ngoài công ty vào năm 1982, trong một bài báo có tiêu đề “Lượng tử hóa bình phương ít nhất trong PCM”. Sau đó, vào năm 1965. , Edward W. Forgy đã xuất bản gần như cùng một thuật toán, vì vậy K-Means đôi khi được gọi là LloydForgy.

Thuật toán phân cụm K-means là một loại học tập không giám sát (unsupervised learning), được sử dụng khi chúng ta có dữ liệu chưa được gắn nhãn (tức là dữ liệu không có danh mục hoặc nhóm được xác định). Mục tiêu của thuật toán này là tìm các nhóm trong dữ liệu, với số lượng nhóm được đại diện bởi biến K. Thuật toán hoạt động lặp đi lặp lại để gán mỗi điểm dữ liệu cho một trong K nhóm dựa trên các đặc tính của dữ liệu được cung cấp. Các điểm dữ liệu được phân nhóm dựa trên sự giống nhau về mặt tính chất và thuộc tính của nó. Kết quả của thuật toán phân cụm K-Means là:

* Các tâm cụm của K cụm có thể được sử dụng để gắn nhãn (label) dữ liệu mới
* Nhãn cho dữ liệu đào tạo [training data] (mỗi điểm dữ liệu được gán cho một cụm duy nhất)
* Thay vì xác định các nhóm trước khi xem dữ liệu, phân cụm cho phép chúng ta tìm và phân tích các nhóm đã hình thành một cách có tổ chức.
* Mỗi trung tâm của một cụm là một tập hợp các giá trị đặc trưng xác định các nhóm kết quả. Việc kiểm tra các trọng số của tính năng tâm cụm có thể được sử dụng để giải thích một cách định tính loại nhóm nào mà mỗi cụm đại diện.

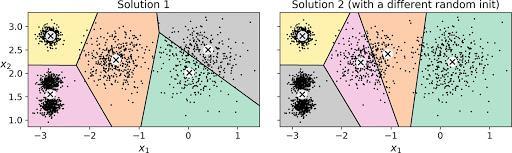
## **3.2 Thuật toán gom cụm K-Means là gì ?**

Vậy thuật toán K-means hoạt động như thế nào? Giả sử chúng ta đã được cung cấp các tâm cụm (centroid): chúng ta có thể dễ dàng gắn nhãn tất cả các điểm dữ liệu (instance) trong cụm dữ liệu (cluster) bằng cách gán các điểm dữ liệu cho cụm có tâm cụm gần nhất. Ngược lại, nếu chúng ta được cung cấp tất cả các nhãn của điểm dữ liệu thì có thể dễ dàng xác định vị trí tất cả các tâm cụm (centroid) bằng cách tính giá trị trung bình của các điểm dữ liệu cho mỗi cụm. Nhưng nếu chúng ta không được cung cấp nhãn lẫn tâm cụm, vậy làm thế nào có thể tiếp tục phân cụm? Vâng, chúng ta chỉ cần bắt đầu bằng cách đặt ngẫu nhiên số lượng các tâm cụm (chọn ngẫu nhiên k trường hợp và sử dụng vị trí của chúng làm trung tâm). Sau đó gắn nhãn các điểm dữ liệu, cập nhật các trung tâm và lặp lại quá trình này cho đến khi các tâm cụm ngừng di chuyển (giá trị không đổi). Thuật toán được đảm bảo hội tụ trong một số bước hữu hạn (thường là khá nhỏ), nó sẽ không dao động mãi mãi,điều này có thể được chứng minh bằng cách chỉ ra rằng khoảng cách bình phương trung bình giữa các cá thể và tâm gần nhất của chúng chỉ có thể giảm xuống ở mỗi bước. Các tâm cụm được khởi tạo ngẫu nhiên (trên cùng bên trái), sau đó các thể hiện được gắn nhãn (trên cùng bên phải), sau đó các tâm cụm được cập nhật (giữa bên trái), các thể hiện được gắn nhãn lại (giữa bên phải), v.v. Cuối cùng, chỉ trong 3 lần lặp lại, thuật toán đã đạt đến một cụm có vẻ gần với mức tối ưu. [2]



Hình 3. 1.Thuật toán K-Means

Trong thực tế, mặc dù thuật toán được đảm bảo hội tụ, nhưng nó có thể không hội tụ đến giải pháp phù hợp (tức là nó có thể hội tụ đến mức tối ưu cục bộ): điều này phụ thuộc vào khởi tạo tâm cụm. Ví dụ, hình dưới đây sẽ cho thấy hai giải pháp tối ưu phụ mà thuật toán có thể hội tụ nếu chúng không may mắn với bước khởi tạo tâm cụm ngẫu nhiên:



Hình 3. 2. Các giải pháp dưới mức tối ưu do khởi tạo tâm cụm không may mắn (giá trị tâm cụm lệch so với mức tối ưu)

K-means là thuật toán rất quan trọng và được sử dụng phổ biến trong kỹ thuật phân cụm dữ liệu.Ý tưởng chính của thuật toán K-means là tìm cách phân nhóm các đối tượng (objects) đã cho vào k cụm (k là số các cụm được xác định trước, k là số nguyên dương) sao cho tổng bình phương khoảng cách giữa các đối tượng đến tâm nhóm (centroid) là nhỏ nhất. [2]

Thuật toán K-means áp dụng cho các đối tượng được biểu diễn bởi các điểm trong không gian vectơ d chiều U = {xi | i = 1, … , N}, với xi ∈ ℜ biểu thị đối tượng (hay điểm dữ liệu) thứ i [2]

Thuật toán k-means gom cụm toàn bộ các điểm dữ liệu trong U thành k cụm C = {C1, C2,…, Ck} sao cho mỗi điểm dữ liệu xi nằm trong một cụm duy nhất. Để biết điểm dữ liệu thuộc cụm nào người ta gán cho nó một mã cụm. Các điểm có cùng mã cụm ở cùng cụm, trong khi các điểm khác mã cụm ở trong các cụm khác nhau. Một cụm có thể biểu thị bằng vectơ liên thuộc cụm v có độ dài N, với vi là mã cụm của xi. Giá trị k là đầu vào của thuật toán. Giá trị k dựa trên tiêu chuẩn tri thức trước đó. Trong các thuật toán gom cụm, các điểm được nhóm theo khái niệm “độ gần” hay “độ tương tự”. Với K-means, phép đo mặc định cho “độ tương tự” là khoảng cách Euclide [2]

## **3.3 Các bước thực hiện thuật toán K-means**

Thuật toán k-means thực hiện qua các bước chính sau:

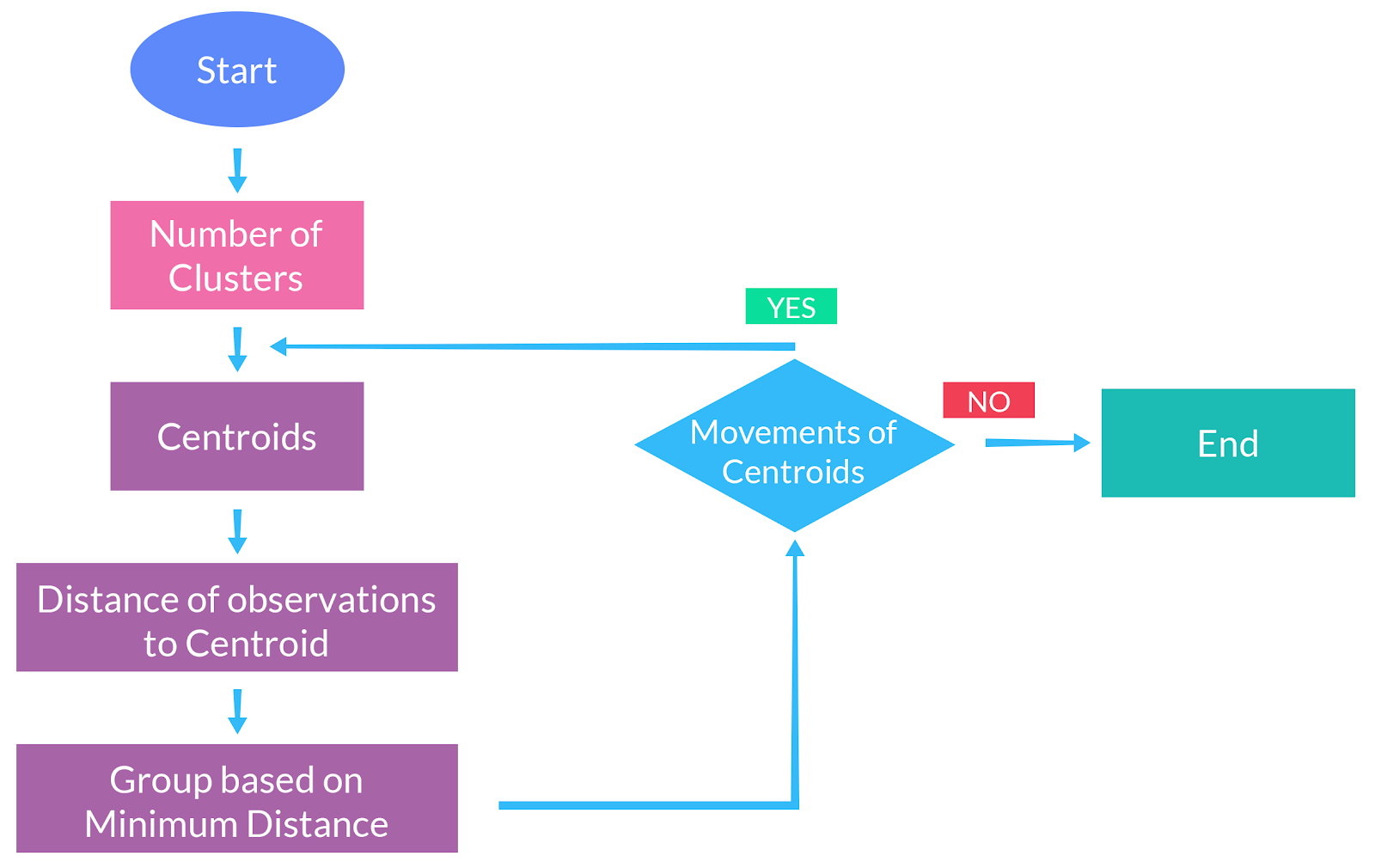
1. Chọn ngẫu nhiên k tâm cụm (centroid) cho k cụm (cluster). Mỗi cụm được đại diện bằng các tâm của cụm. [2]

2. Tính khoảng cách giữa các đối tượng (objects) đến k tâm (thường dùng khoảng cách Euclide). [2]

3. Nhóm các đối tượng vào nhóm gần nhất. [2]

4. Xác định lại tâm mới cho các nhóm. [2]

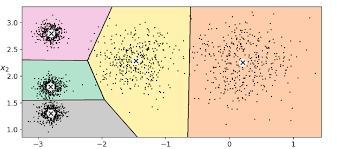
5. Thực hiện lại bước 2 cho đến khi không có sự thay đổi nhóm nào của các đối tượng. [2]



Hình 3. 3. Các bước thực hiện thuật toán K-Means

Một cách cụ thể hơn chúng ta sẽ đi vào chi tiết các bước cài đặt chính sau:

### **3.3.1 Phương pháp khởi tạo tâm cụm (Centroid)**



Hình 3. 4. Ví dụ kết quả phân cụm với các giá trị tâm cụm k

Nếu chúng ta có thể xác định gần đúng vị trí của các tâm cụm (ví dụ: nếu bạn đã chạy một thuật toán phân cụm khác trước đó), thì bạn có thể đặt siêu tham số init thành một mảng NumPy chứa danh sách các trung tâm và đặt n\_init thành 1:

|  |
| --- |
| good\_init = np.array([[-3, 3], [-3, 2], [-3, 1], [-1, 2], [0, 2]])  kmeans = KMeans(n\_clusters=5, init=good\_init, n\_init=1) |

Một giải pháp khác là chạy thuật toán nhiều lần với các lần khởi tạo ngẫu nhiên khác nhau và giữ nguyên giải pháp tốt nhất. Điều này được kiểm soát bởi hyperparameter n\_init: theo mặc định, nó bằng 10, có nghĩa là toàn bộ thuật toán được mô tả trước đó thực sự chạy 10 lần khi chúng ta gọi lệnh fit () và Scikit-Learn giữ giải pháp tốt nhất. Nhưng làm thế nào giải thuật có thể biết chính giải pháp nào là tốt nhất? Đó là nhờ sử dụng thước đo hiệu suất (performance metric). Nó được gọi là quán tính của mô hình: đây là khoảng cách bình phương trung bình giữa mỗi trường hợp và tâm gần nhất của nó. Nó gần bằng 223,3 cho mô hình bên trái Hình 2; 237,5 cho mô hình bên phải Hình 2, và 211.6 cho mô hình trong Hình 3. Lớp K-Means chạy thuật toán n\_init lần và giữ cho mô hình có quán tính thấp nhất: trong ví dụ này, mô hình trong Hình 3 sẽ được chọn (trừ khi chúng ta rất không may mắn với n\_init lần khởi tạo ngẫu nhiên liên tiếp). Nếu bạn tò mò, quán tính của một mô hình có thể truy cập được thông qua biến thể hiện quán tính inertia\_: [2]

|  |
| --- |
| >>> kmeans.inertia\_  211.59853725816856 |

Phương thức score() trả về quán tính âm. Tại sao lại là giá trị âm? Đó là bởi vì phương thức score() của người dự đoán phải luôn tôn trọng quy tắc "great is better". [2]

|  |
| --- |
| >>> kmeans.score(X)  -211.59853725816856 |

Một cải tiến quan trọng đối với thuật toán K-Means, được gọi là K-Means + \ +, được đề xuất trong một bài báo năm 2006 bởi David Arthur và Sergei Vassilvitskii: 2 họ đã giới thiệu một bước khởi tạo thông minh hơn có xu hướng chọn các trung tâm ở xa nhau và điều này làm cho thuật toán K-Means ít có khả năng hội tụ đến một giải pháp tối ưu hơn nhiều. Họ đã chỉ ra rằng tính toán bổ sung cần thiết cho bước khởi tạo thông minh hơn là rất đáng giá vì nó có thể giúp giảm đáng kể số lần thuật toán cần phải chạy để tìm ra giải pháp tối ưu. Đây là thuật toán khởi tạo K-Means ++: [2]

* Chọn 1 tâm cụm ,được chọn ngẫu nhiên đồng nhất từ ​​tập dữ liệu. [2]
* Chọn 1 tâm cụm mới và 1 điểm dữ liệu với xác suất  với là khoảng cách giữa điểm dữ liệu với tâm cụm đã được chọn gần nhất.Phân phối xác suất này đảm bảo rằng các trường hợp xa hơn các tâm cụm đã được chọn có nhiều khả năng được chọn làm trung tâm. [2]
* Lặp lại bước trước đó cho đến khi tất cả k tâm cụm đã được chọn. [2]

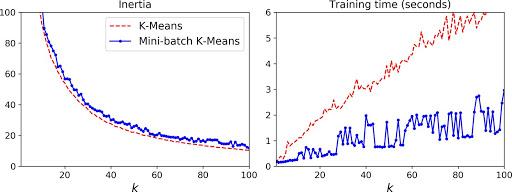
Trong thực tế K-Means mặc định sử dụng phương thức khởi tạo này. Nếu chúng muốn buộc thuật sử dụng phương pháp gốc (tức là chọn ngẫu nhiên k trường hợp để xác định tâm cụm ban đầu), thì bạn có thể đặt hyperparameter init thành "random", chúng ta thực sự sẽ hiếm khi cần phải làm điều này. [2]

### **3.3.2 Accelerated K-Means và Mini-batch K-Means**

Một cải tiến quan trọng khác đối với thuật toán K-Means đã được đề xuất trong một bài báo năm 2003 của . Nó tăng tốc đáng kể thuật toán bằng cách giảm thiểu nhiều các tính toán khoảng cách không cần thiết: điều này đạt được bằng cách khai thác bất đẳng thức tam giác (tức là đường thẳng luôn ngắn ) và bằng cách theo dõi các giới hạn dưới và trên để biết khoảng cách giữa các điểm dữ liệu và tâm cụm. Đây là thuật toán được sử dụng theo mặc định bởi lớp K-Means (chúng ta có thể buộc nó sử dụng thuật toán ban đầu bằng cách đặt hyperparameter của thuật toán thành "full" mặc dù có thể chúng ta sẽ không bao giờ cần đến). Tuy nhiên, một biến thể quan trọng khác của thuật toán K-Means đã được đề xuất trong một bài báo năm 2010 bởi .Thay vì sử dụng tập dữ liệu đầy đủ ở mỗi lần lặp, thuật toán có khả năng sử dụng các lô nhỏ (mini-batches), chỉ di chuyển các tâm cụm một chút ở mỗi lần lặp. Điều này giúp tăng tốc thuật toán theo hệ số 3 hoặc 4 và có thể phân cụm các bộ dữ liệu khổng lồ có kích thước quá lớn so với khả năng xử lý bộ nhớ. Scikit-Learn triển khai thuật toán này trong lớp **MiniBatchKMeans**. Chúng ta chỉ có thể sử dụng lớp này như lớp KMeans: [2]

|  |
| --- |
| from **sklearn.cluster** import MiniBatchKMeans  minibatch\_kmeans = MiniBatchKMeans(n\_clusters=5)  minibatch\_kmeans.fit(X) |

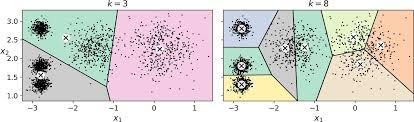
Nếu tập dữ liệu không vừa trong bộ nhớ, tùy chọn đơn giản nhất là sử dụng lớp **memmap**. Ngoài ra, chúng ta có thể chuyển một lô nhỏ tại một thời điểm cho phương thức part fit (), nhưng điều này sẽ đòi hỏi nhiều tác vụ hơn vì chúng ta sẽ cần thực hiện nhiều lần khởi tạo và tự mình chọn kết quả tốt nhất. Mặc dù thuật toán K-Means theo lô nhỏ nhanh hơn nhiều so với thuật toán K-Means thông thường, nhưng quán tính (inertia) của nó thường kém hơn một chút, đặc biệt là khi số lượng cụm tăng lên. Bạn có thể thấy điều này trong Hình a: biểu đồ bên trái so sánh độ trơ của các mô hình K-Means theo lô nhỏ và các mô hình K-Means thông thường được đào tạo trên tập dữ liệu trước đó bằng cách sử dụng nhiều cụm k khác nhau. Sự khác biệt giữa hai đường cong vẫn không đổi, nhưng sự khác biệt này ngày càng trở nên có ý nghĩa hơn khi k tăng, vì quán tính ngày càng nhỏ. Tuy nhiên, trong biểu đồ bên phải, bạn có thể thấy rằng K-Means theo lô nhỏ nhanh hơn nhiều so với K-Means thông thường, và sự khác biệt này tăng lên theo k. [2]



Hình 3. 5. K-Means loạt nhỏ so với K-Means: quán trính kém hơn khi k tăng (trái) nhứng nhanh hơn nhiều(phải)

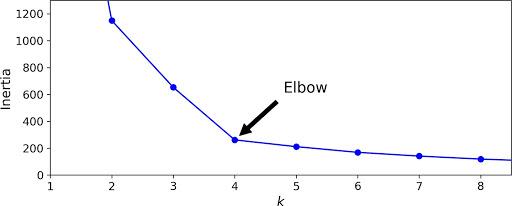
**3.3.3 Tìm số lượng nhóm tối ưu**

Cho đến nay, chúng ta đã đặt số lượng cụm k thành 5 bởi vì bằng cách xem xét dữ liệu rõ ràng rằng đây là số cụm chính xác. Nói chung, sẽ không dễ dàng biết cách cài đặt giá trị k và kết quả có thể không tối ưu nếu chúng ta dự đoán và đặt sai giá trị. Ví dụ, như bạn có thể thấy trong Hình b, việc đặt k thành 3 hoặc 8 dẫn đến kết quả các mô hình phân cụm khá tệ: [2]



Hình 3. 6. Kết quả phân cụm không tốt khi lựa chọn giá trị tâm cụm không chính xác

Chúng ta có thể nghĩ rằng chỉ cần đơn giản chọn mô hình có quán tính (inertia) thấp nhất là tối ưu nhất phải không? Thật không may, điều này không đơn giản như thế. Quán tính của k = 3 là 653,2; cao hơn nhiều so với k = 5 là 211,6; nhưng với k = 8, quán tính chỉ là 119,1. Quán tính không phải là một thước đo hiệu suất tốt khi cố gắng chọn k vì nó tiếp tục thấp hơn khi tăng giá trị k.. Thật vậy, càng có nhiều cụm, mỗi điểm dữ liệu sẽ càng gần tâm cụm gần nhất của nó, và do đó quán tính sẽ càng thấp. Hãy vẽ đồ thị quán tính dưới dạng hàm của k (xem Hình c): [2]

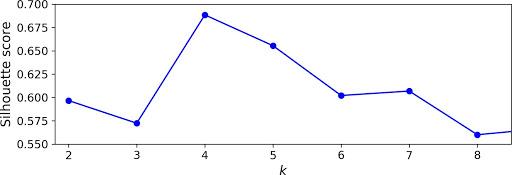


Hình 3. 7. Chọn số lượng cụm k bằng cách sử dụng “ellbow rule”

Dựa vào kết quả trên chúng ta có thể thấy, quán tính giảm rất nhanh khi chúng ta tăng k lên 4, nhưng sau đó giảm chậm hơn nhiều khi chúng ta tiếp tục tăng k. Đường cong này có hình dạng gần giống như một cánh tay và có một "khuỷu tay" ở k = 4, vì vậy nếu chúng ta không biết rõ hơn, nó sẽ là một lựa chọn tốt: bất kỳ giá trị nào thấp hơn kết quả sẽ rất tệ, trong khi bất kỳ giá trị nào cao hơn sẽ không giúp ích được gì nhiều, và chúng ta có thể chỉ tách các cụm hoàn toàn tốt thành một nửa mà vì không lý do chính đáng. Kỹ thuật này chọn giá trị tốt nhất cho số lượng cụm thô. Một cách tiếp cận chính xác hơn (nhưng cũng tốn kém hơn về mặt tính toán) là sử dụng điểm hình bóng (silhouette score), là hệ số hình bóng trung bình trong tất cả các điểm dữ liệu. Hệ số hình bóng của một đối tượng bằng (b - a) / max (a, b) trong đó a là khoảng cách trung bình đến các đối tượng khác trong cùng một cụm (nó là khoảng cách trung bình trong cụm) và b là giá trị khoảng cách đến với tâm cụm gần nhất, đó là khoảng cách trung bình đến các điểm dữ liệu của cụm gần nhất tiếp theo (được định nghĩa là giảm thiểu b, không bao gồm cụm dữ liệu mà nó thuộc về). Hệ số hình bóng có thể thay đổi giữa -1 và +1: hệ số gần +1 có nghĩa là đối tượng nằm bên trong cụm của chính nó và xa các cụm khác, trong khi hệ số gần 0 có nghĩa là nó gần với ranh giới cụm, và cuối cùng một hệ số gần -1 có nghĩa là điểm dữ có thể đã được gán cho sai cụm. Để tính toán điểm hình bóng, bạn có thể sử dụng hàm silhouette score () của Scikit-Learn, cung cấp cho hàm tập dữ liệu và các nhãn mà chúng được gán: [2]

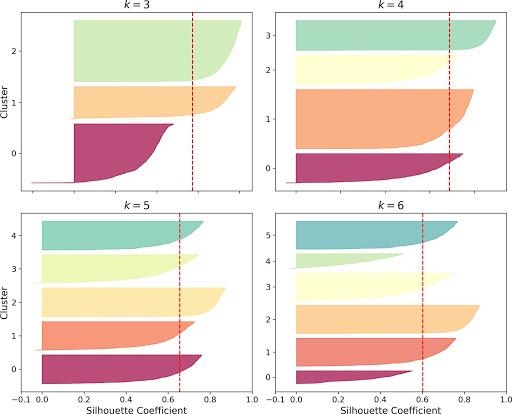
|  |
| --- |
| >>> from **sklearn.metrics** import silhouette\_score  >>> silhouette\_score(X, kmeans.labels\_)  0.655517642572828 |

Hãy so sánh điểm hình bóng cho số lượng các cụm khác nhau (xem Hình d)



Hình 3. 8. Chọn số lượng cụm k bằng cách sử dụng điểm hình bóng

Như chúng ta có thể thấy, các thể hiện bằng cách ảo hóa dữ liệu dưới dạng biểu đồ này sẽ giúp chúng ta dễ dàng xem xét kết quả, dựa vào biểu đồ trên có thể nhận biết 1 điều rằng, mặc dù nó xác nhận rằng k = 4 là một lựa chọn rất tốt, nó cũng nhấn mạnh thực tế rằng k = 5 cũng khá tốt và tốt hơn nhiều so với k = 6 hoặc 7. Điều này không thể nhìn thấy khi so sánh quán tính. Thậm chí, chúng ta còn có thể hình dung nhiều thông tin hơn khi vẽ biểu đồ hệ số hình bóng của mọi đối tượng, được sắp xếp theo cụm chúng được gán và theo giá trị của hệ số. Đây được gọi là sơ đồ hình bóng (xem Hình e): [2]



Hình 3. 9. Phân tích Silhouette: so sánh các sơ đồ hình bóng cho các giá trị khác nhau của k.

Các đường đứt nét dọc thể hiện điểm số hình bóng (silhouette score) cho mỗi số cụm. Khi hầu hết các điểm dữ trong một cụm có hệ số thấp hơn điểm số hình bóng(nếu nhiều trường hợp dừng lại gần đường đứt nét và kết thúc ở bên trái của nó), thì cụm này sẽ phân cụm không tốt vì điều này có nghĩa là các điểm dữ liệu của nó quá gần với các cụm khác. Chúng ta có thể thấy rằng khi k = 3 và khi k = 6, chúng ta nhận được các cụm không tốt như vậy. Nhưng khi k = 4 hoặc k = 5, các cụm được nhóm 1 cách khá tốt - hầu hết các trường hợp mở rộng ra ngoài đường đứt nét, sang bên phải và gần hơn với 1,0. Khi k = 4, cụm ở chỉ số 1 (thứ ba từ trên xuống), khá lớn, trong khi khi k = 5, tất cả các cụm có kích thước tương tự nhau, vì vậy mặc dù điểm hình bóng tổng thể từ k = 4 lớn hơn một chút so với k = 5 nhưng sử dụng k = 5 để nhận các cụm có kích thước tương tự cho việc phân cụm là khá tốt. [2]

## **3.4 Tổng kết**

### **3.4.1 Tổng quát**

Thuật toán phân cụm K-Means tạo ra một số lượng cụ thể các cụm rời rạc, phẳng (không phân cấp - non-hierarchical), rất thích hợp để tạo ra các cụm hình cầu. Phương pháp K-Means là phương pháp số (non-hierarchical), không giám sát (unsupervised), không xác định (unsupervised) và lặp lại (iterative).

**Thuộc tính:**

* Luôn có K cụm.
* Luôn có ít nhất một điểm dữ liệu trong mỗi cụm.
* Các cụm không phân cấp và chúng không chồng chéo lên nhau.
* Mọi thành viên của một cụm đều gần với cụm của nó hơn bất kỳ cụm nào khác bởi vì sự gần gũi không phải lúc nào cũng liên quan đến tâm cụm của các cụm.

**Quy trình thuật toán K-Means:**

Tập dữ liệu được phân chia thành K cụm và các điểm dữ liệu được gán ngẫu nhiên cho các cụm dẫn đến các cụm có số điểm dữ liệu gần như giống nhau.

Đối với mỗi điểm dữ liệu:

* Tính toán khoảng cách từ điểm dữ liệu đến từng cụm.
* Nếu điểm dữ liệu gần nhất với cụm của chính nó, hãy để nó ở vị trí cũ. Nếu điểm dữ liệu không gần với cụm của chính nó nhất, hãy di chuyển nó vào cụm gần nhất.
* Lặp lại bước trên cho đến khi hoàn toàn duyệt qua tất cả các điểm dữ liệu dẫn đến không có điểm dữ liệu nào di chuyển từ cụm này sang cụm khác. Tại thời điểm này các cụm ổn định và quá trình phân cụm kết thúc.

Việc lựa chọn phân vùng ban đầu có thể ảnh hưởng lớn đến các cụm cuối cùng, về khoảng cách và sự gắn kết giữa các cụm và giữa các cụm.

### **3.4.2 Ưu điểm của thuật toán**

* Việc cài đặt và vận hành thuật toán tương đối đơn giản.
* Cân bằng và phân chia tỷ lệ tốt với tập dữ liệu lớn đảm bảo khả năng áp dụng vào thuật toán giúp việc thực hiện trở nên trơn tru.
* Đảm bảo hội tụ kết quả (guarantees convergence) trong đa số trường hợp.
* Có thể khởi tạo vị trí của các tâm cụm.
* Dễ dàng thích ứng với các tập dữ liệu mới.
* Tổng quát hóa thành các cụm có hình dạng và kích thước khác nhau, chẳng hạn như các cụm hình elip.
* Với một số lượng lớn các biến dữ liệu, K-Means có thể nhanh hơn về mặt tính toán so với phân cụm phân cấp (hierarchical clustering) nếu giá trị K nhỏ.
* K-Means có thể tạo ra các cụm chặt chẽ hơn so với phân cụm phân cấp (hierarchical clustering), đặc biệt nếu các cụm là hình cầu.

### **3.4.3 Nhược điểm của thuật toán**

* Khó khăn trong việc so sánh chất lượng của các cụm được tạo ra (ví dụ: đối với các phân vùng ban đầu khác nhau hoặc giá trị của K ảnh hưởng đến kết quả).
* Trong đa số trường hợp, người dùng thường phải dự đoán và chọn thủ công giá trị k để cài đặt cho thuật toán. Số lượng cụm cố định có thể gây khó khăn cho việc dự đoán giá trị phù hợp cho k.
* Hiệu suất hoạt động của nó thường không cạnh tranh bằng các kỹ thuật phân cụm phức tạp khác vì các biến thể nhỏ trong dữ liệu có thể dẫn đến phương sai cao
* Không tối ưu cho các cụm dữ liệu có hình dạng (shape) không phải hình cầu, các cụm (trong dữ liệu gốc) có kích thước khác nhau và mật độ khác nhau, để phân cụm dữ liệu này, chúng ta cần phải tổng quát hóa K-Means. Ngoài ra, đối vợ trường hợp nhóm các ngoại lệ (outliers), các tâm cụm có thể bị lôi kéo bởi dữ liệu nhiễu, các phần tử ngoại lai hoặc các phần tử ngoại lai có thể nhận được cụm riêng của chúng thay vì bị bỏ qua. Cân nhắc loại bỏ hoặc cắt bớt các điểm ngoại lệ trước khi phân cụm.
* Phụ thuộc vào giá trị khởi tạo ban đầu. Các phân vùng ban đầu khác nhau có thể tạo ra các cụm cuối cùng khác nhau. Sẽ rất hữu ích nếu bạn chạy lại chương trình bằng cách sử dụng các giá trị K giống nhau cũng như khác nhau, để so sánh kết quả đạt được. Đối với các cụm dữ liệu mật độ thấp, chúng ta có thể giảm thiểu sự phụ thuộc này bằng cách chạy K-Means nhiều lần với các giá trị ban đầu khác nhau và chọn kết quả tốt nhất. Khi mật độ dữ liệu tăng lên, chúng ta cần các phiên bản nâng cao của K-Means để chọn các giá trị tốt hơn của các tâm cụm ban đầu (được gọi là K-Mean seeding, để tìm hiểu đầy đủ về phương pháp này,chúng ta có thể tham khảo Nghiên cứu so sánh về các phương pháp khởi tạo hiệu quả cho thuật toán phân cụm K-Means của M. Emre Celebi, Hassan A. Kingravi, Patricio A. Vela [A Comparative Study of Efficient Initialization Methods for the K-Means Clustering Algorithm by M. Emre Celebi, Hassan A. Kingravi, Patricio A. Vela]).
* Chia tỷ lệ với số chiều (dimension): khi số chiều tăng lên, thước đo độ tương tự dựa trên khoảng cách hội tụ về một giá trị không đổi giữa bất kỳ ví dụ nhất định nào. Giảm kích thước bằng cách sử dụng PCA trên đặc tính dữ liệu hoặc bằng cách sử dụng "phân cụm quang phổ" (spectral clustering) để sửa đổi thuật toán phân cụm.

### **3.4.4 Các giải pháp thay thế**

Hiện nay có rất nhiều thuật toán phân cụm thay thế cho K-Mean có thể kể đến như DBSCAN, phân cụm quang phổ (spectral clustering) và mô hình hóa kết hợp Gaussian (Gaussian mixtures). Kỹ thuật giảm kích thước, chẳng hạn như phân tích thành phần chính, có thể được sử dụng để tách các nhóm mẫu trong dữ liệu.

Một kết quả có thể xảy ra là không có cụm có tổ chức (organic clusters) nào trong dữ liệu; thay vào đó, tất cả dữ liệu nằm dọc theo phạm vi đặc tính liên tục trong một nhóm duy nhất. Trong trường hợp này, chúng ta có thể cần phải xem lại các đặc điểm dữ liệu để xem liệu các phép đo khác nhau có cần được đưa vào hay không hoặc một phép chuyển đổi tính năng sẽ thể hiện tốt hơn sự thay đổi trong dữ liệu. Ngoài ra, bạn có thể muốn áp đặt các danh mục hoặc nhãn dựa trên kiến ​​thức về miền (domain) và sửa đổi phương pháp phân tích của mình.

# **CHƯƠNG 4: CHƯƠNG TRÌNH MINH HỌA**

## **4.1 Cấu hình máy và phần mềm sử dụng**

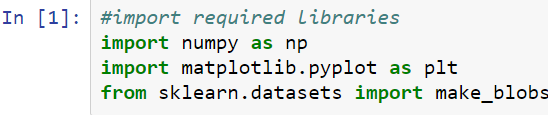
Cấu hình máy:

* Intel core i7 7700
* 16GB RAM
* Win 10 - 32bit

Phần mềm sử dụng: Anaconda3 Jupyter.

## **4.2 Đầu vào và kết quả thuật toán K-Mean**

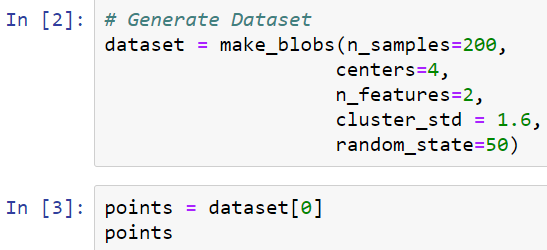
Mã lệnh:



Hình 4. 1. Implement các thư viện môi trường.

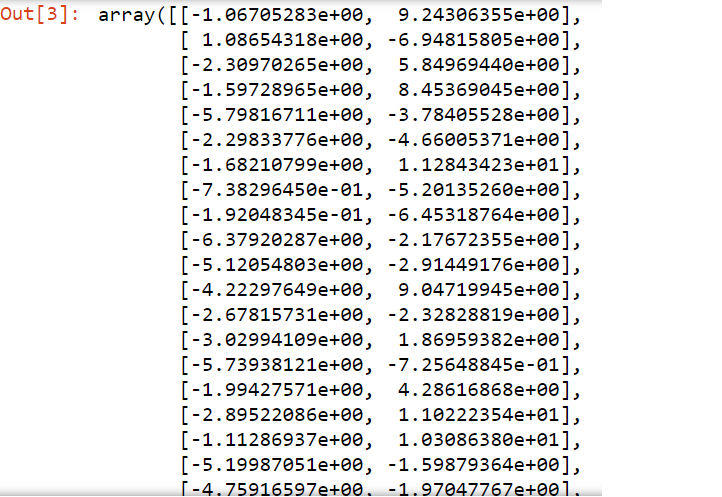
Chức năng dòng lệnh: import các thư viện xử lý cần thiết như numpy để xử lý mảng, matplotlib.pyplot thực hiện các thao tác liên quan đến biểu đồ, sklearn.datasets để load và trả về tập dữ liệu.

Mã lệnh:



Hình 4. 2. : Khởi tạo tập dữ liệu

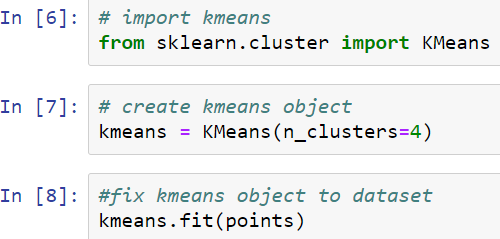
Kết quả trả về:



Hình 4. 3. Kết quả dataset đã khởi tạo dữ liệu

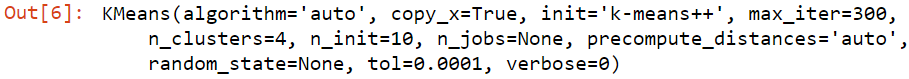
Chức năng dòng lệnh: khởi tạo tập dataset ngẫu nhiên làm dữ liệu đầu vào ( ở đây tập giá trị sẽ gồm 200 điểm với 2 thuộc tính x và y, 4 tâm cụm và giá trị sẽ được random).

Mã lệnh:



Hình 4. 4. : Import thư viện K-Means và khởi tạo các tâm cụm.

Kết quả trả về:



Hình 4. 5. Kết quả sau khi khởi tạo K-Means

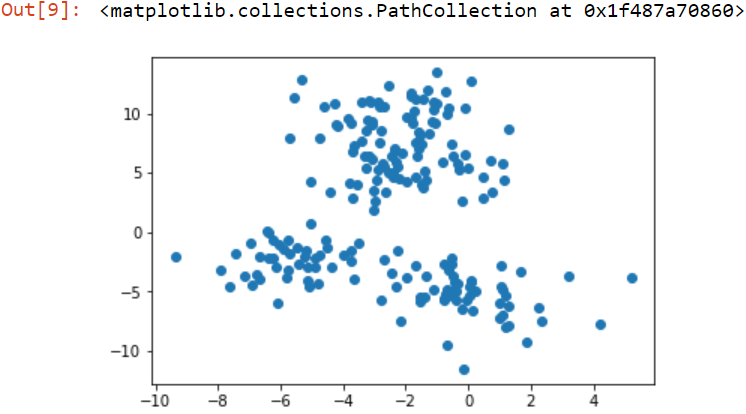
Chức năng dòng lệnh: import thư viện xử lý K-means sẵn có của Python, khởi tạo giá trị số cụm cần gom nhóm (chương trình demo là 4 cụm) và lấy dataset đã khởi tạo ở trên làm dữ liệu thực thi.

Mã lệnh:



Hình 5.5: Sử dụng thư viện matplotlib để in các giá trị trong tập dữ liệu (dataset).

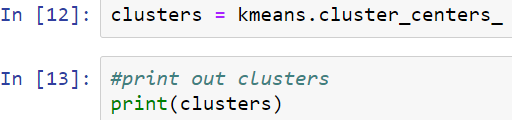
Kết quả trả về:



Hình 4. 6. Thể hiện tập điểm dữ liệu của tập dữ liệu lên biểu đồ

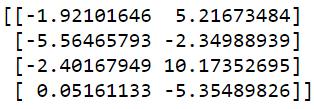
Chức năng dòng lệnh: sử dụng thư viện matplotlib để thể hiện các điểm trong tập dữ liệu dataset lên biểu đồ.

Mã lệnh:



Hình 4. 7. Sử dụng thư viện KMean để tìm tọa độ tâm cụm

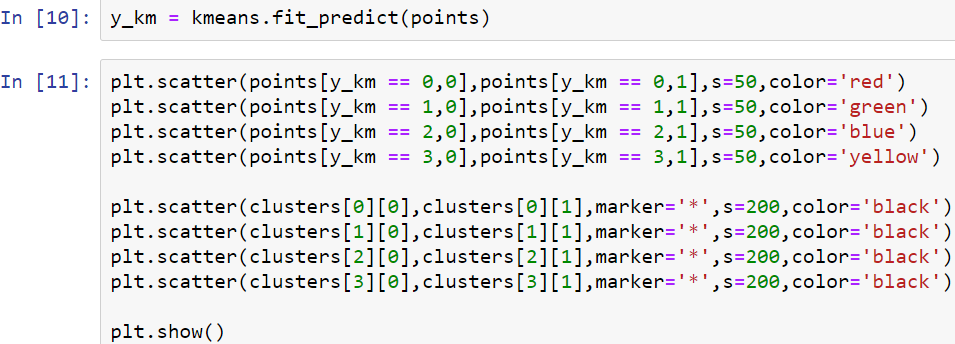
Kết quả trả về:



Hình 4. 8. In ra các giá trị tâm cụm.

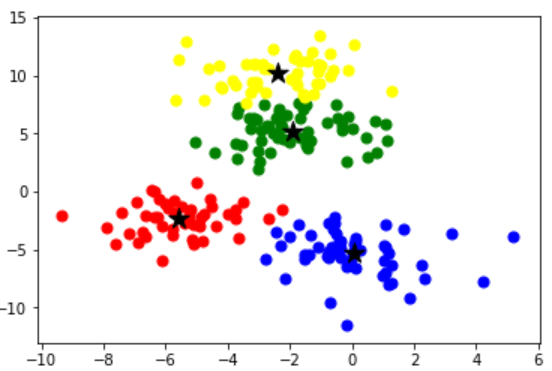
Chức năng dòng lệnh: sử dụng thư viện K-means để tính toán các giá trị tâm cụm và in ra màn hình.

Mã lệnh:



Hình 4. 9. Sử dụng thư viện matplotlib để thể hiện các cụm và tâm cụm.

Kết quả trả về:



Hình 4. 10. Hiển thị dữ liệu trên biểu đồ

Chức năng dòng lệnh: sử dụng thư viện matplotlib thể hiện các giá trị sau khi gom cụm và các tâm cụm lệnh biểu đồ.

# **CHƯƠNG 5: KẾT LUẬN**

## **5.1 Kiến thức tìm hiểu được**

Nắm bắt được thuật toán gom cụm K-means, có thể sử dụng được tuy nhiên do thời gian và khả năng của các thành viên trong nhóm có hạn nên 1 số phần còn chưa hiểu rõ.

## **5.2 Khó khăn**

|  |  |  |
| --- | --- | --- |
| **STT** | **Khó khăn** | **Khắc phục** |
| 1 | Ngôn ngữ mới hoàn toàn chưa có kinh nghiệm sử dụng | Học thêm từ các nguồn bên ngoài |
| 2 | Chưa hiểu rõ hết được thư viện | Đọc từ các nguồn bên ngoài |
| 3 | Thiếu dữ liệu để train cho thuật toán, quá lớn dữ liệu quá nhiều đặc điểm, quá ít thì không thể train | Đọc document |

## **5.3 Hướng phát triển**

Nắm bắt chắc chắn hơn thuật toán gom cụm K-Means, DBSCAN để có thể áp dụng trong tương lai

# **CHƯƠNG 6: TÀI LIỆU THAM KHẢO**

<https://drive.google.com/file/d/1NEWlDjfZZXz9QgihAMBzkOfcarlja5hh/view?usp=sharing> [1]

<https://drive.google.com/file/d/1yYrGHJn_ZJ4ePqDqY11O7EzVVGZYta8e/view?usp=sharing> [2]