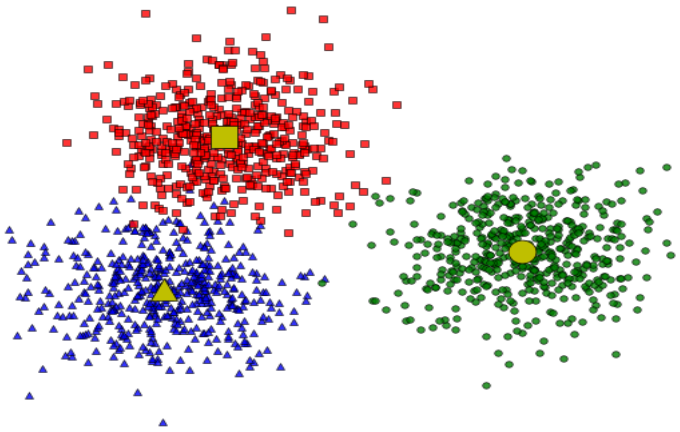
ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
UIT-HCM

ĐỒ ÁN CUỐI KÌ

HIERARCHICAL CLUSTERING



Lập trình Python cho Máy học

CS116.M12.KHCL

Giảng viên: Nguyễn Vinh Tiệp

Thực hiện:

Trịnh Tuấn Nam (19521874)

Nguyễn Thành Trung (19522432)

Nguyễn Dương Hải (19521464)

MỤC LỤC:

A/ GIỚI THIỆU MÔ HÌNH:

Bài toán phân cụm là một nhánh ứng dụng chính của lĩnh vực Học không giám sát (Unsupervised Learning), trong đó dữ liệu huấn luyện trong bài toán chưa được dán nhãn. Trong trường hợp này, thuật toán sẽ tìm cách phân cụm - chia dữ liệu thành từng nhóm có đặc điểm tương tự nhau, nhưng đồng thời đặc tính giữa các nhóm đó lại phải càng khác biệt càng tốt.

Có nhiều phương pháp để thực hiện gom nhóm dữ liệu, mỗi cách sẽ có mục đích cũng như hiệu quả khác nhau, tùy vào mục đích và yêu cầu của bài toán. Trong bài viết này, nhóm sẽ đề cập đến một trong những phương pháp gom cụm khá phổ biến, đó là:

Hierarchical Clustering (phân cụm phân cấp)

B/ THUẬT TOÁN:

Khi sử dụng thuật toán **Hierarchical Clustering**, chúng ta không cần phải khai báo trước số lượng cụm, thay vào đó, thuật toán chỉ yêu cầu phải xác định thước đo về sự khác biệt giữa hai cụm (không giao nhau) với nhau.

Phương pháp phân cụm phân cấp hoạt động thông qua việc nhóm dữ liệu thành một cây các cụm. Theo phương pháp này, chúng ta biểu diễn được những sự phân cấp, trong đó mỗi cụm được hình thành bằng cách hợp nhất tất cả các cụm ở cấp thấp hơn. Sự phân cấp này được thể hiện qua **đồ thị *dendrogram***.

Trước khi thực hiện gom cụm, ta cần phải xác định *tiêu chí gom cụm* và *điều kiện dừng* của bài toán. Khi đã xác định được, bài toán được mô tả như sau:

WHILE (điều kiện dừng = = False) DO

- Chọn hai cụm gần nhau nhất theo tiêu chí đã xác định ban đầu.
- Sát nhập hai cụm gần nhau thành cụm lớn hơn.

END WHILE;

1/ Những tiêu chí hợp nhất hai cụm

+ Centroid-linkage: gom hai cụm có khoảng cách giữa hai tâm của hai cụm này là nhỏ nhất.

+ Single-linkage: khoảng cách giữa hai điểm gần nhau nhất thuộc hai cụm. Gom hai cụm có khoảng cách này nhỏ nhất.

+ Average-linkage: trung bình các khoảng cách giữa hai cặp điểm bất kì thuộc hai cụm. Gom hai cụm có khoảng cách này nhỏ nhất.

+ Complete-linkage: khoảng cách giữa hai điểm xa nhau nhất của hai cụm, gom hai cụm có khoảng cách này là nhỏ nhất.

+ Radius: bán kính của một cụm là khoảng cách từ tâm tới điểm xa nhất của cụm, gom hai cụm nếu hai cụm tạo ra một cụm có bán kính nhỏ nhất.

+ Diameter: đường kính của một cụm là khoảng cách của hai điểm xa nhau nhất trong cụm, gom hai cụm nếu chúng tạo nên một cụm có đường kính nhỏ nhất.

2/ Điều kiệu dừng

- Phỏng đoán được số tự nhiên k – số cụm trong tập sữ liệu. Sau đó chúng ta sẽ dừng thuật toán phân chia nếu như số lượng các cụm đạt được là chạm ngưỡng bằng k.
- Khi việc sát nhập hai cụm tạo ra một cụm kém chất lượng (có độ gắn kết các điểm dữ liệu thấp).

*Tham khảo thêm phương pháp khảo sát độ gắn kết của một cụm theo phương pháp ***tiếp cận mật độ (density-based).***

3/ Các chiến lược phân cụm phân cấp

Giả sử có 6 điểm dữ liệu: A, B, C, D, E và F (như hình vẽ)

3.a/ Chiến lược hợp nhất (agglomerative):

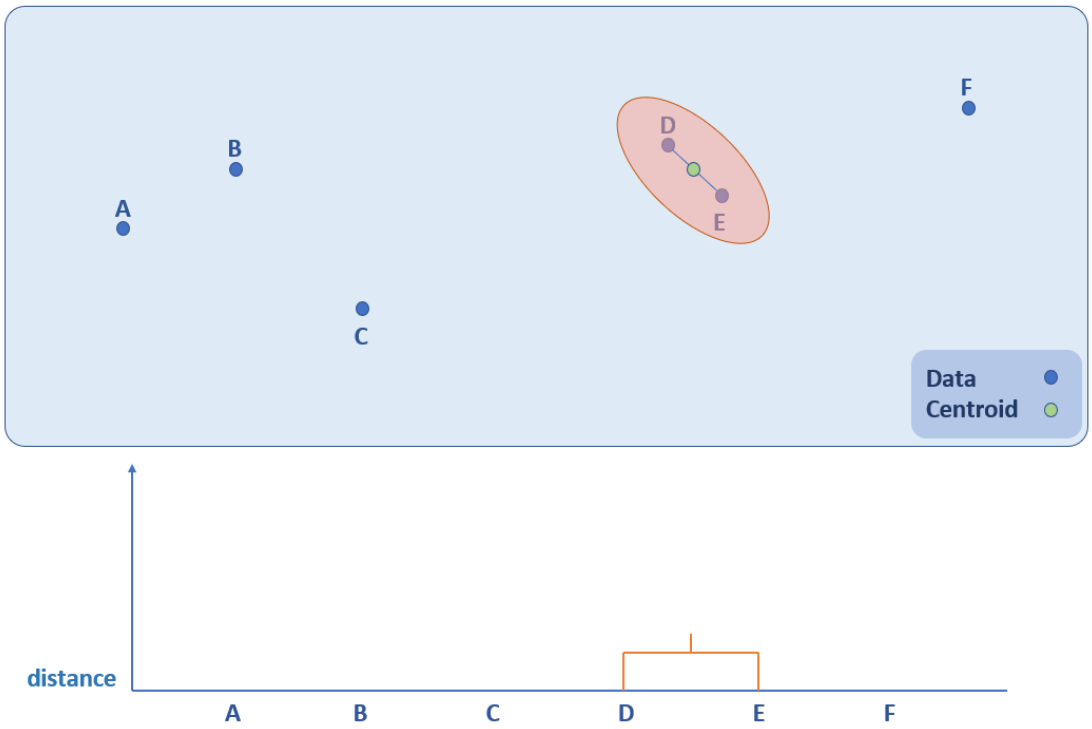
Chiến lược hợp nhất khi bắt đầu sẽ xem như mỗi điểm dữ liệu là một cụm đơn lẻ. Giả sử chúng ta có N quan sát, thuật toán cần thực hiện N−1 bước để hợp nhất hai nhóm có khoảng cách gần nhất lại với nhau và đồng thời giảm số lượng cụm trước khi chúng đạt được tới node gốc chứa toàn bộ các điểm dữ liệu. Với 6 điểm dữ liệu như trên, cụ thể như sau:

Trong phần này, nhóm sẽ thống nhất sử dụng tiêu chí **Centroid-linkage** (dựa trên khoảng cách Euclide) để cụ thể hóa thuật toán.

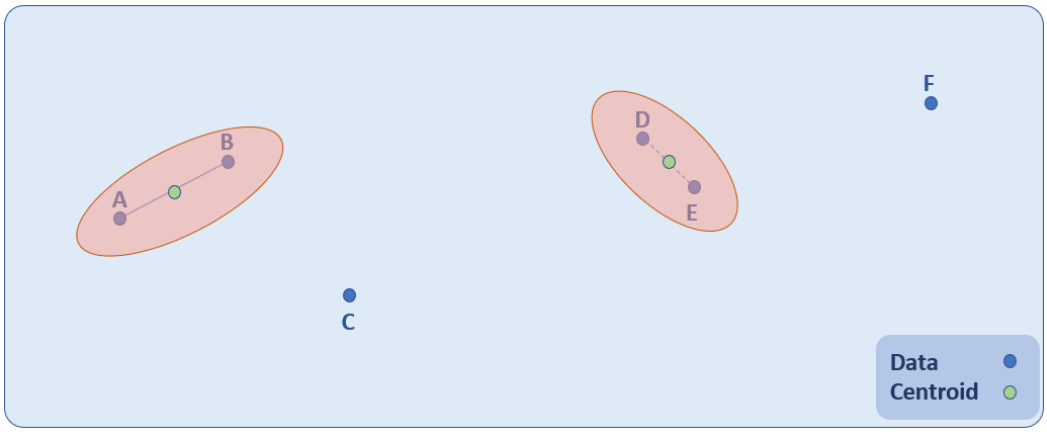
***Bước 1*: Phân cụm phân cấp theo chiến lược hợp nhất bắt đầu bằng cách coi mỗi điểm dữ liệu là một cụm riêng biệt.**

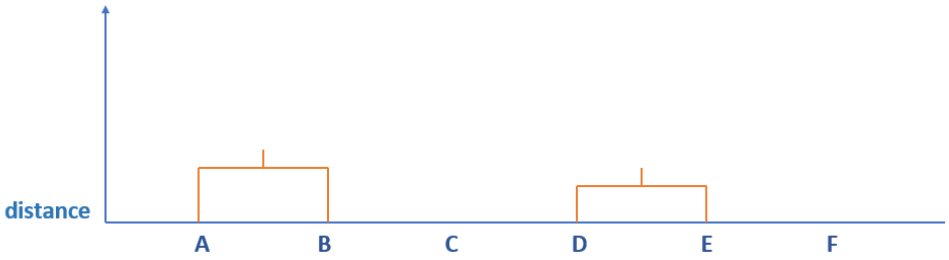
Khi đó, đồ thị **dendrogram** sẽ như sau:

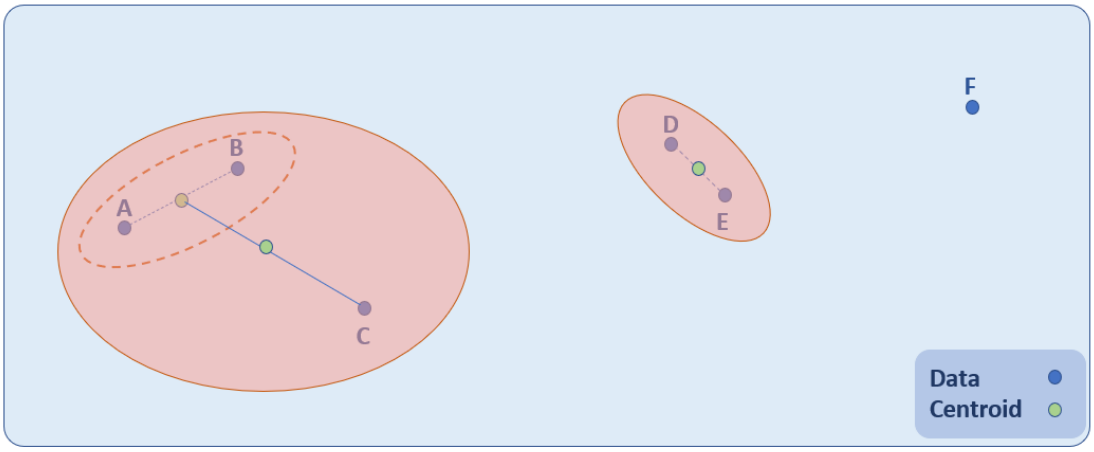
***Bước 2*: Xác định khoảng cách giữa các cụm với nhau. Gom 2 cụm có khoảng cách gần nhất thành 1. Đồng thời xác định tâm mới của cụm vừa gom.**

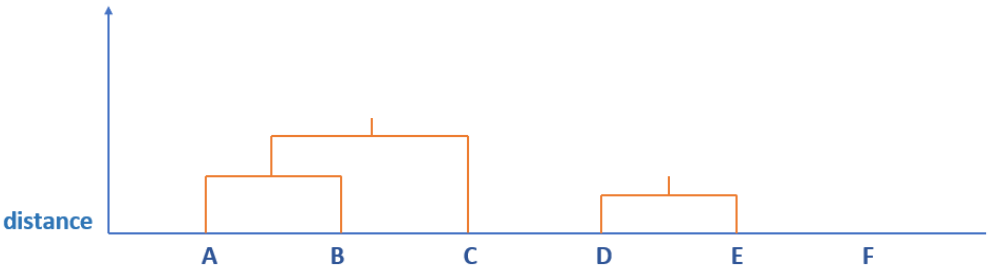


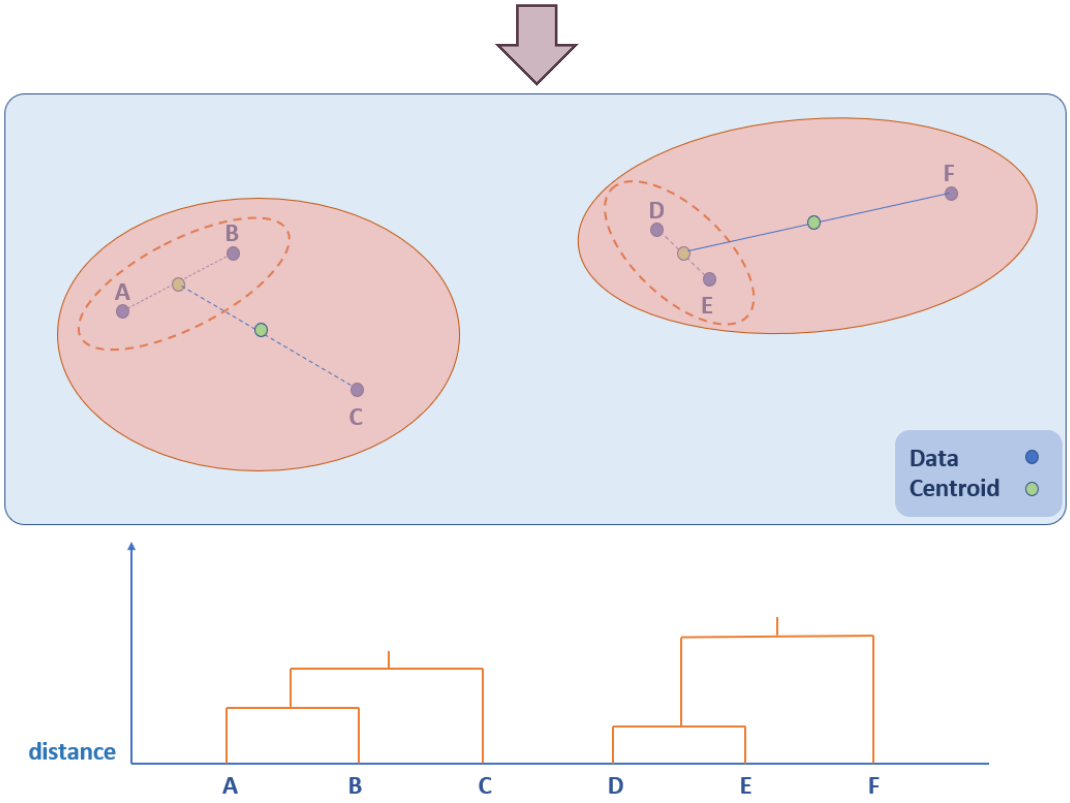
<u>Bước 3:</u> Tiếp tục lặp lại bước 2 cho các cụm còn lại cho đến khi chỉ còn 1 cụm duy nhất.

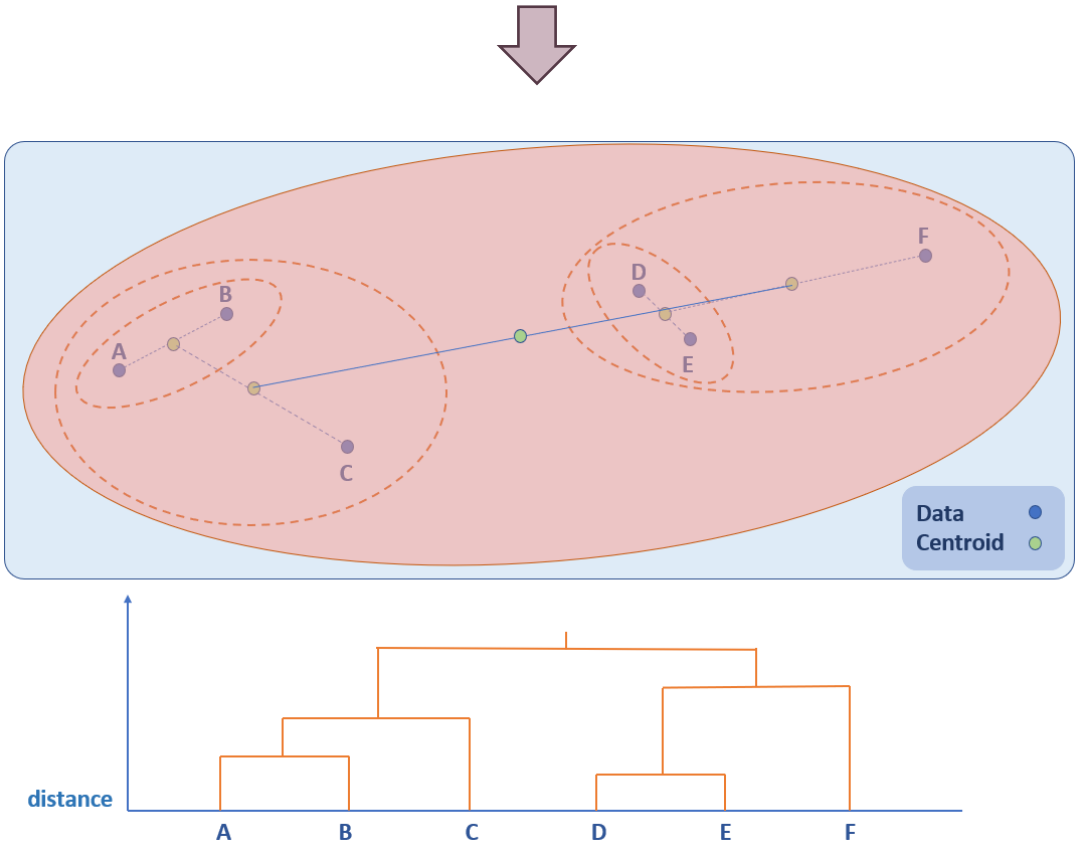




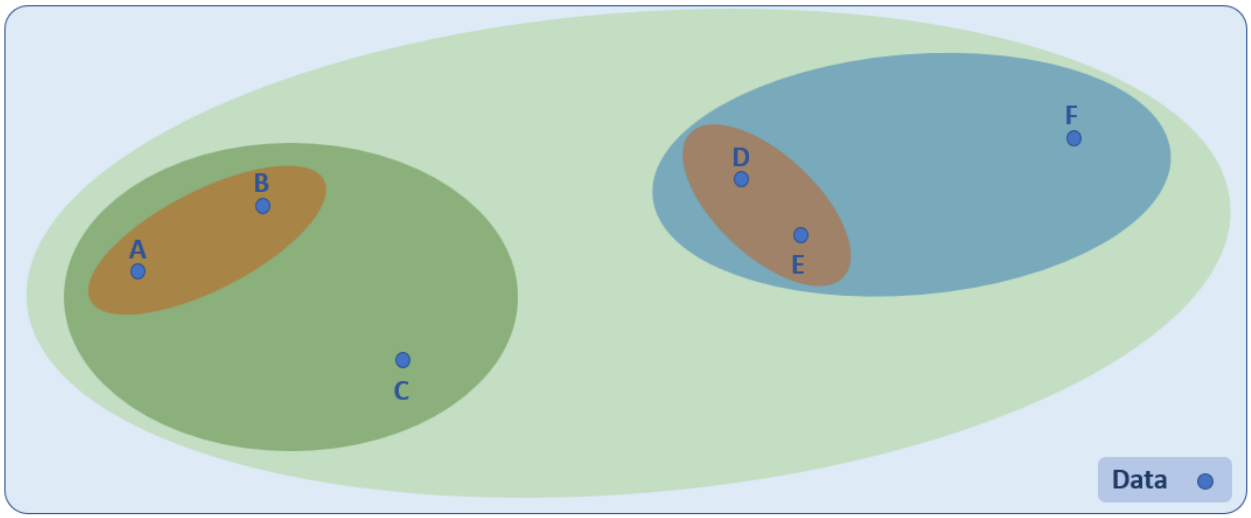





A
B
C
D
E
F
Data
Centroid
distance
A
B
C
D
E
F


A
B
C
D
E
F
Data
Centroid
distance
A
B
C
D
E
F

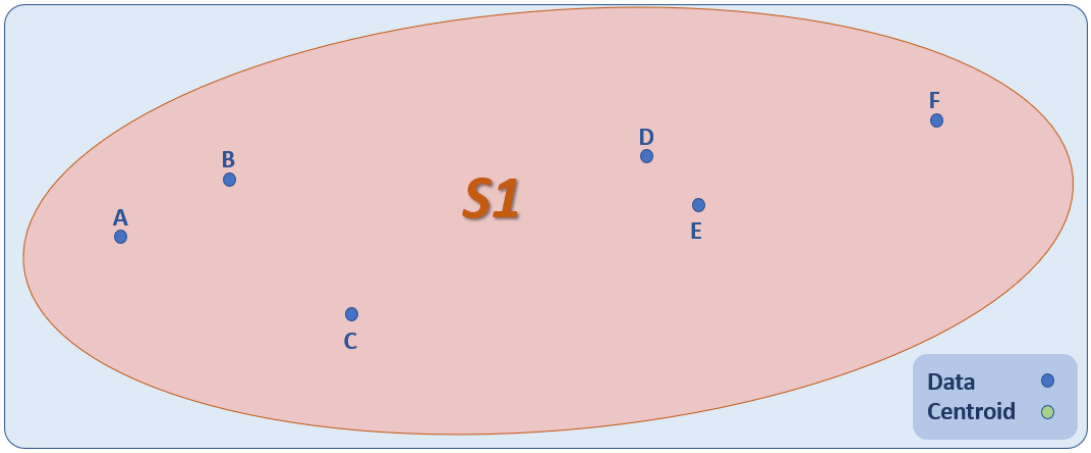
Hoàn thành:



3.b/ Chiến lược phân chia (devise):

Vẫn ví dụ như trên, ***Chiến lược phân chia*** sẽ bắt đầu từ một cụm ban đầu (gọi là S1) gồm toàn bộ các điểm dữ liệu bên trong cụm và sau đó phân chia đệ qui những cụm đang tồn tại thành hai cụm con tại mỗi bước theo hướng *top-down*.

Bước 1: Phân cụm phân cấp theo chiến lược phân chia bắt đầu bằng cách coi mọi điểm dữ liệu đều nằm trong một cụm duy nhất (S1).



Việc chuyển cụm một điểm ***xi*** từ S1 sang cụm khác cần tuân theo 2 điều kiện:

Trung bình khoảng cách từ ***xi*** *tới toàn bộ các điểm còn lại trong S1 phải là* ***lớn nhất*** *(có nghĩa là* ***xi*** *là điểm tách biệt nhất so với phần còn lại của S1).*

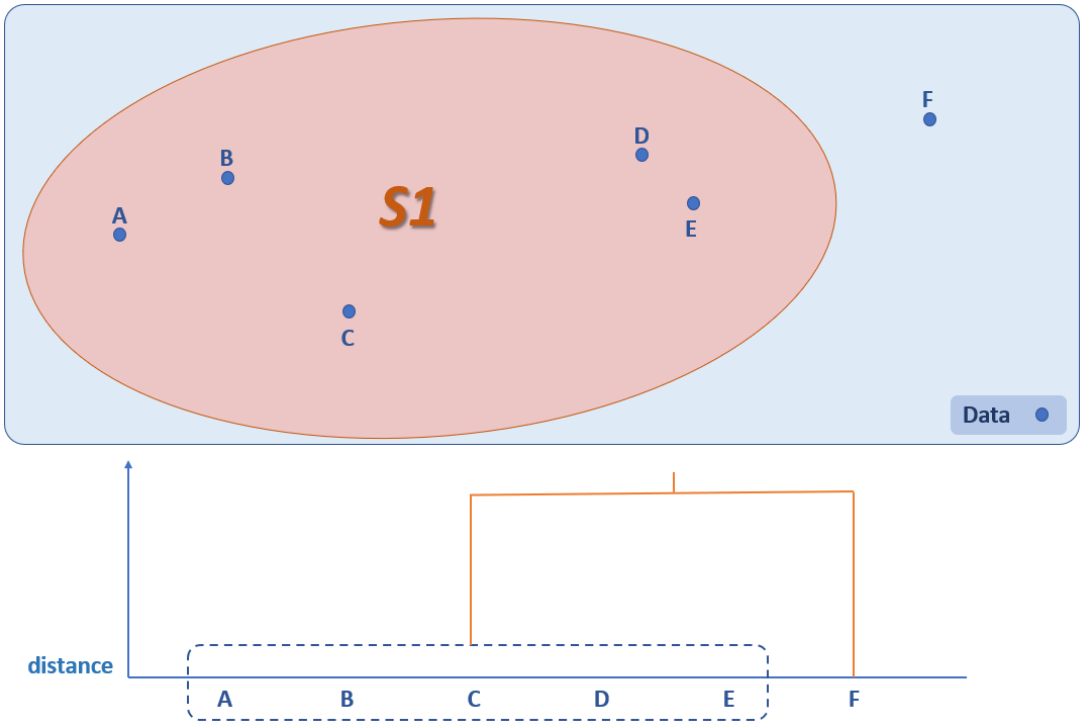
$$\mathbf{x}_i = \arg\max_{\mathbf{x}_i} \frac{1}{|\mathcal{S}_1| - 1} \sum_{j=1, j\neq i}^{|\mathcal{S}_1|} d(\mathbf{x}_i, \mathbf{x}_j)$$

Khoảng cách tối thiểu từ ***xi*** *tới các điểm trong tập tách ra (gọi là S2) phải* ***bé hơn*** *khoảng cách tối thiểu từ* ***xi*** *tới các điểm trong S1 (điều này nhằm mục đích khiến cho* ***xi*** *phải gần với cụm S2 hơn cụm S1).*

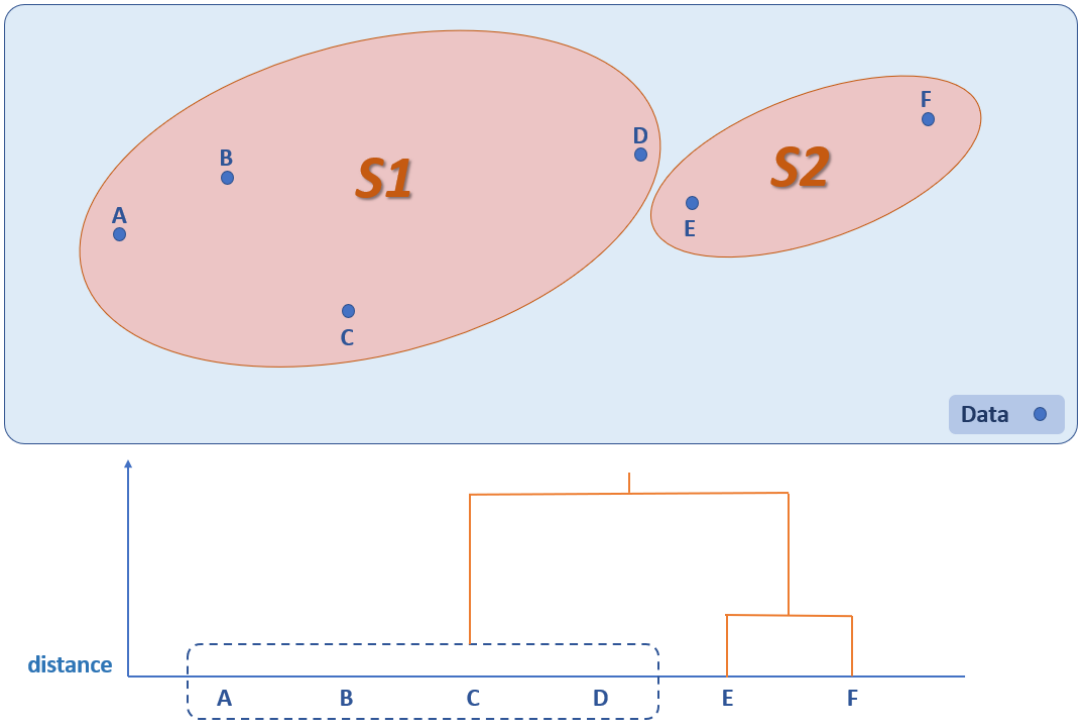
$$d(\mathbf{x}_i, \mathcal{S}_1) \geq d(\mathbf{x}_i, \mathcal{S}_2)$$

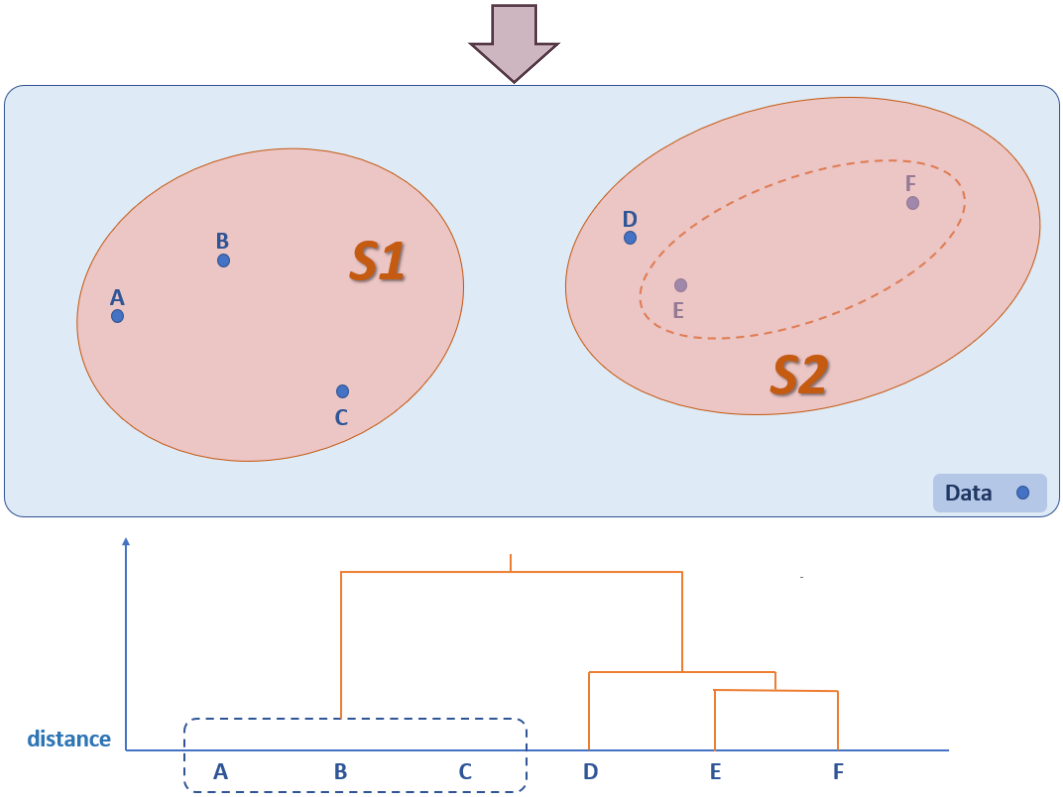
Quá trình chuyển cụm sẽ **kết thúc** khi không còn điểm nào thoả mãn **cả hai** điều kiện trên. Khi đó chúng ta lại thực hiện đệ qui lại quá trình trên trên từng tập S1 và S2.

***Bước 2**: Chúng ta sẽ lựa chọn ra điểm F là điểm đầu tiên thuộc cụm mới vì khoảng cách so với các điểm còn lại là xa nhất.*



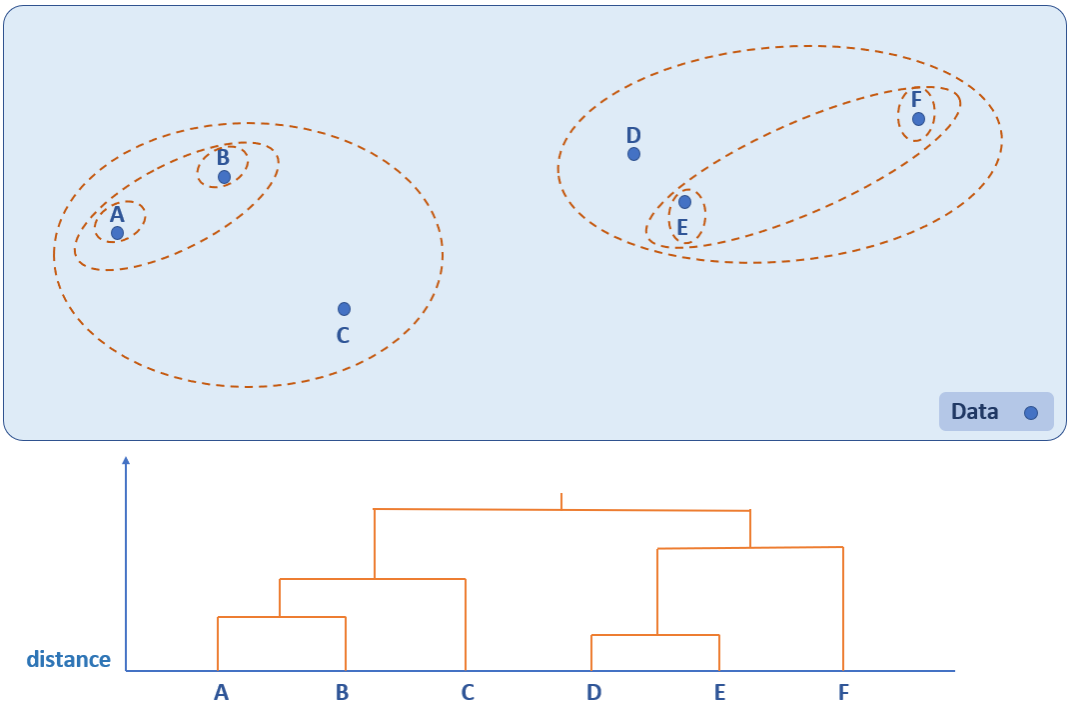
***Bước 3:** Tiếp tục lặp lại bước 2 cho các điểm dữ liệu trong S1.*



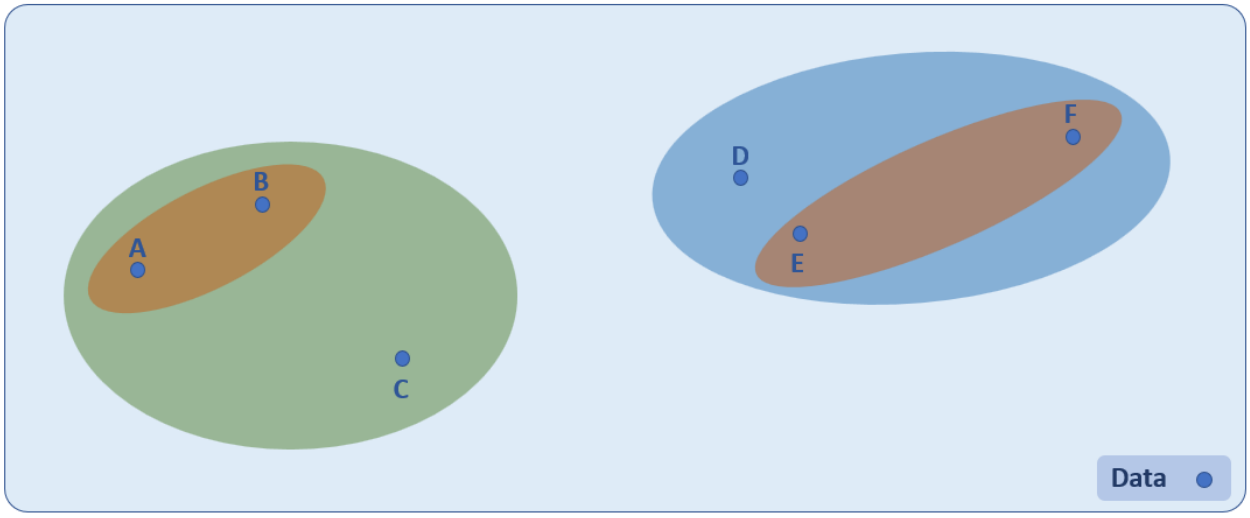


Đến đây, trong S1, mặc dù trung bình khoảng cách từ điểm C tới các điểm dữ liệu còn lại là lớn nhất nhưng vì điểm C có khoảng cách gần với cụm S1 hơn S2 nên không thể chuyển điểm C ra khỏi cụm ban đầu được, tương tự đối với điểm A và B. Do đó, quy trình **chuyển cụm** sẽ kết thúc, thuật toán đã đạt sự hội tụ về 2 cụm.

<u>Bước 4:</u> Đệ quy trên từng cụm con đến khi mỗi điểm dữ liệu là 1 cụm riêng biệt.



Hoàn thành:



4/ So sánh Agglomerative clustering và Divisive clustering:

- **Phân cụm phân chia** phức tạp hơn so với **phân cụm hợp nhất**, vì nó cần một "chương trình con" để tách từng cụm cho đến khi chúng ta có mỗi dữ liệu có một cụm singleton của riêng nó.

- **Phân cụm phân chia** sẽ hiệu quả hơn nếu dữ liệu bố trí với mật độ phức tạp.

- **Phân cụm hợp nhất** đưa ra quyết định bằng cách xem xét các mẫu cục bộ hoặc các điểm lân cận mà ban đầu không tính đến sự phân phối toàn cầu của dữ liệu. Những quyết định ban đầu này không thể được hoàn tác, trong khi phân cụm phân chia có tính đến sự phân phối toàn cầu của dữ liệu khi đưa ra quyết định phân vùng cấp cao nhất.

C/ AGGLOMERATIVE CLUSTERING VỚI SCIKIT-LEARN:

```
(class) AgglomerativeClustering(n_clusters=2, affinity="euclidean",
memory=None, connectivity=None, compute_full_tree="auto",
linkage="ward", distance_threshold=None, compute_distances=False)
```

*** n_clusters:** Số lượng các cụm cần tìm. *(default = 2)*

*** affinity:**

euclide *(default)*	$d(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$	được sử dụng khi khoảng cách giữa các điểm là quan trọng.
manhattan	$d(x, y) = \sum_{n=1} \lvert x_i - y_i \rvert$	
cosine	$\cos(\theta) = AB^{\cdot} \lvert\lvert A \rvert\rvert * \lvert\lvert B \rvert\rvert \cos(\theta)$ $= AB^{\cdot} \lvert\lvert A \rvert\rvert * \lvert\lvert B \rvert\rvert$	được sử dụng khi khoảng cách giữa các điểm không quan trọng nhưng hướng thì có.
precomputed	*ma trận khoảng cách cho trước*	được sử dụng nếu bạn đã có ma trận khoảng cách được tính toán trước cho các điểm dữ liệu của mình.

*** linkage:**

Tùy thuộc vào sự phân bố của dữ liệu của bạn.

single	single-linkage dựa trên khoảng cách nhỏ nhất của tất cả các điểm dữ liệu của hai tập hợp được xem xét để hợp nhất.	hoạt động nhanh nhất, thích hợp để sử dụng với những bộ dữ liệu lớn. Có thể hoạt động tốt trên dữ liệu phân bố không đều, nhưng nó hoạt động kém khi có nhiễu.
complete	complete-linkage dựa trên khoảng cách lớn nhất của tất cả các điểm dữ liệu của hai tập hợp được xem xét để hợp nhất.	cả hai đều hoạt động tốt trên các cụm hình cầu được phân tách rõ ràng, nhưng có các kết quả khác nhau.
average	average-linkage dựa trên trung bình khoảng cách của tất cả các điểm dữ liệu của hai tập hợp được xem xét để hợp nhất.	
ward *(default)* *chỉ có thể sử dụng đồng thời với affinity: 'euclide'*	ward giúp giảm thiểu phương sai khoảng cách giữa các cụm được hợp nhất.	hoạt động tốt với dữ liệu được phân chia hoàn toàn thành các tập con rời rạc hoặc hiệu quả nhất cho dữ liệu nhiễu. Nó là một lựa chọn tốt khi sử dụng với centroid.

D/ SO SÁNH VỚI K-MEANS CLUSTERING:

- Hierarchical Clustering: phân cụm phân cấp.
- K-Means Clustering: phân cụm phân hoạch.

Hierarchical Clustering	K-Means Clustering
là phương pháp phân chia hoặc hợp nhất, **không phải** khai báo trước số lượng cụm.	sử dụng số lượng cụm được chỉ định trước để phân cụm, **phải** khai báo trước số lượng cụm.
không có yếu tố lựa chọn ngẫu nhiên, kết quả luôn luôn chỉ có 1.	mỗi lần bắt đầu với sự lựa chọn ngẫu nhiên của các cụm, kết quả có thể khác nhau trong những lần chạy khác nhau.
phân cụm phân cấp không hoạt động tốt như k-mean khi hình dạng của các cụm là siêu hình cầu.	hoạt động tốt khi cấu trúc của các cụm là siêu hình cầu (như hình tròn trong 2D, hình cầu trong 3D).
Độ phức tạp: O(N^2.logN) – tối ưu nhất *N* là số lượng điểm dữ liệu.	**Độ phức tạp:** O(N.K.i) *N* là số lượng điểm dữ liệu. *K* là số lượng cụm. *i* là số lần lặp cần thiết cho đến khi hội tụ.
Ưu điểm: 1. Dễ dàng xử lý với dữ liệu có hình thức tương tự hoặc khoảng cách tương đồng. 2. Áp dụng cho mọi loại thuộc tính.	**Ưu điểm:** 1. Sự hội tụ được đảm bảo. 2. Chuyên dùng cho các cụm có kích thước và hình dạng khác nhau.
Nhược điểm: 1. Phân cụm phân cấp yêu cầu tính toán và lưu trữ ma trận khoảng cách n × n. Đối với bộ dữ liệu rất lớn, gây tốn kém bộ nhớ và thời gian chạy lớn.	**Nhược điểm:** 1. Số lượng cụm khó dự đoán 2. Không hoạt động tốt với cụm phân bố đều.

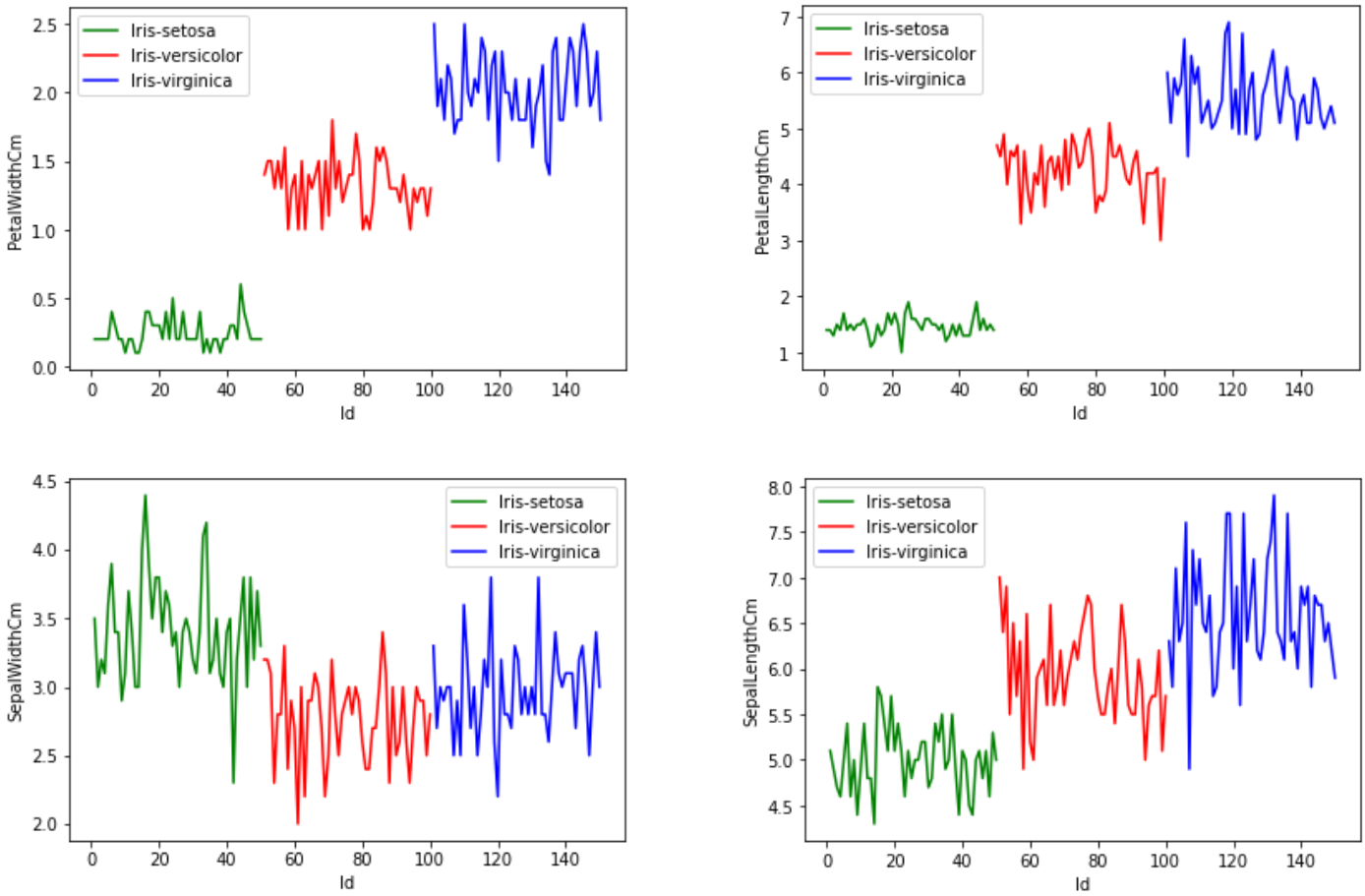
E/ THỰC NGHIỆM MÔ HÌNH:

1/ Mô tả bộ dữ liệu:

Nhóm sử dụng bộ dữ liệu IRIS để tiến hành bài toán Hierachical Clustering. Bộ IRIS bao gồm **150 samples** của **3 loại hoa** khác nhau được mô tả như bên dưới:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns



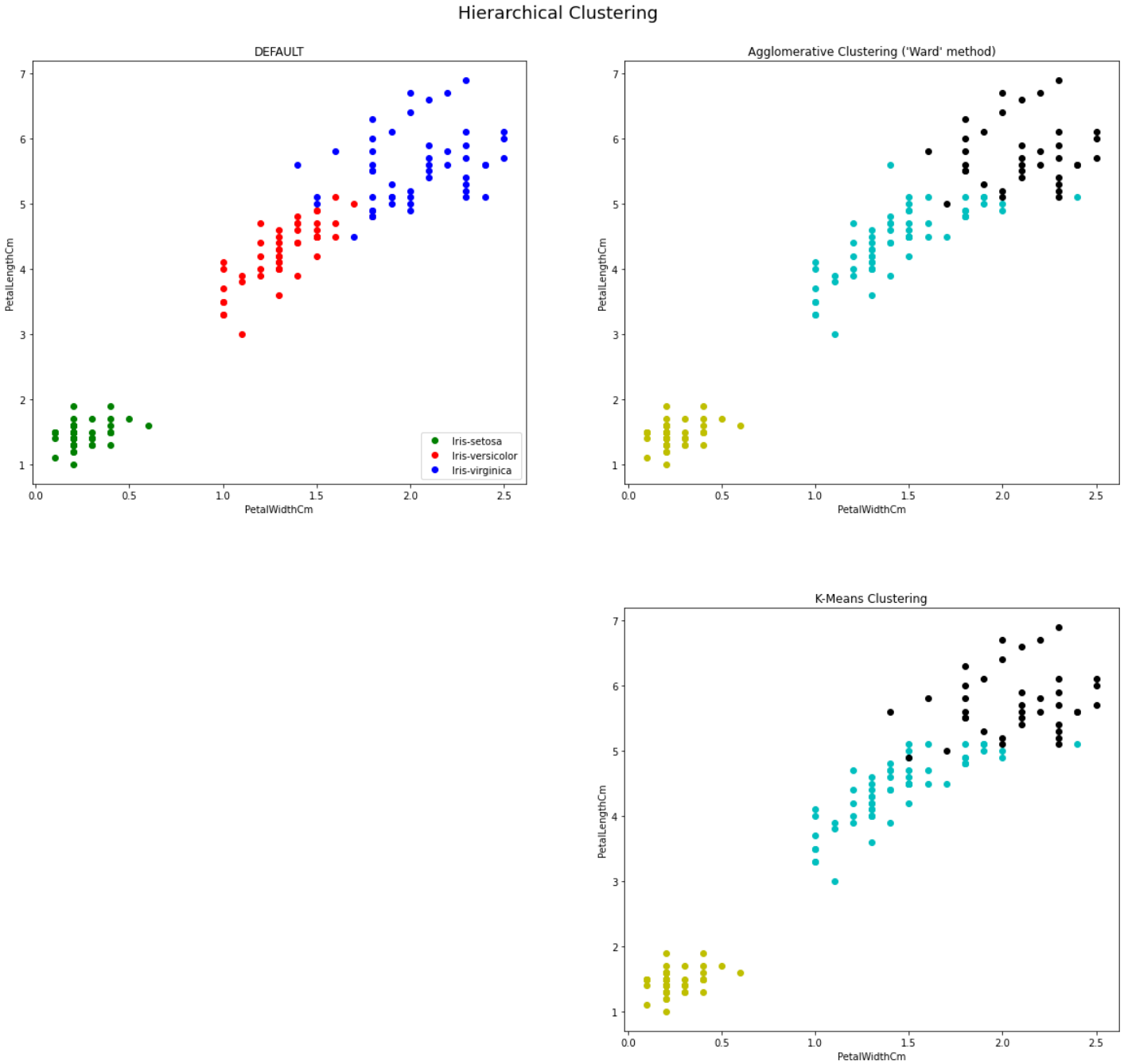
Miền dữ liệu của các thuộc tính trong bộ IRIS

Lí do chọn bộ IRIS để hiện thực hóa bài toán Hierachical Clustering:

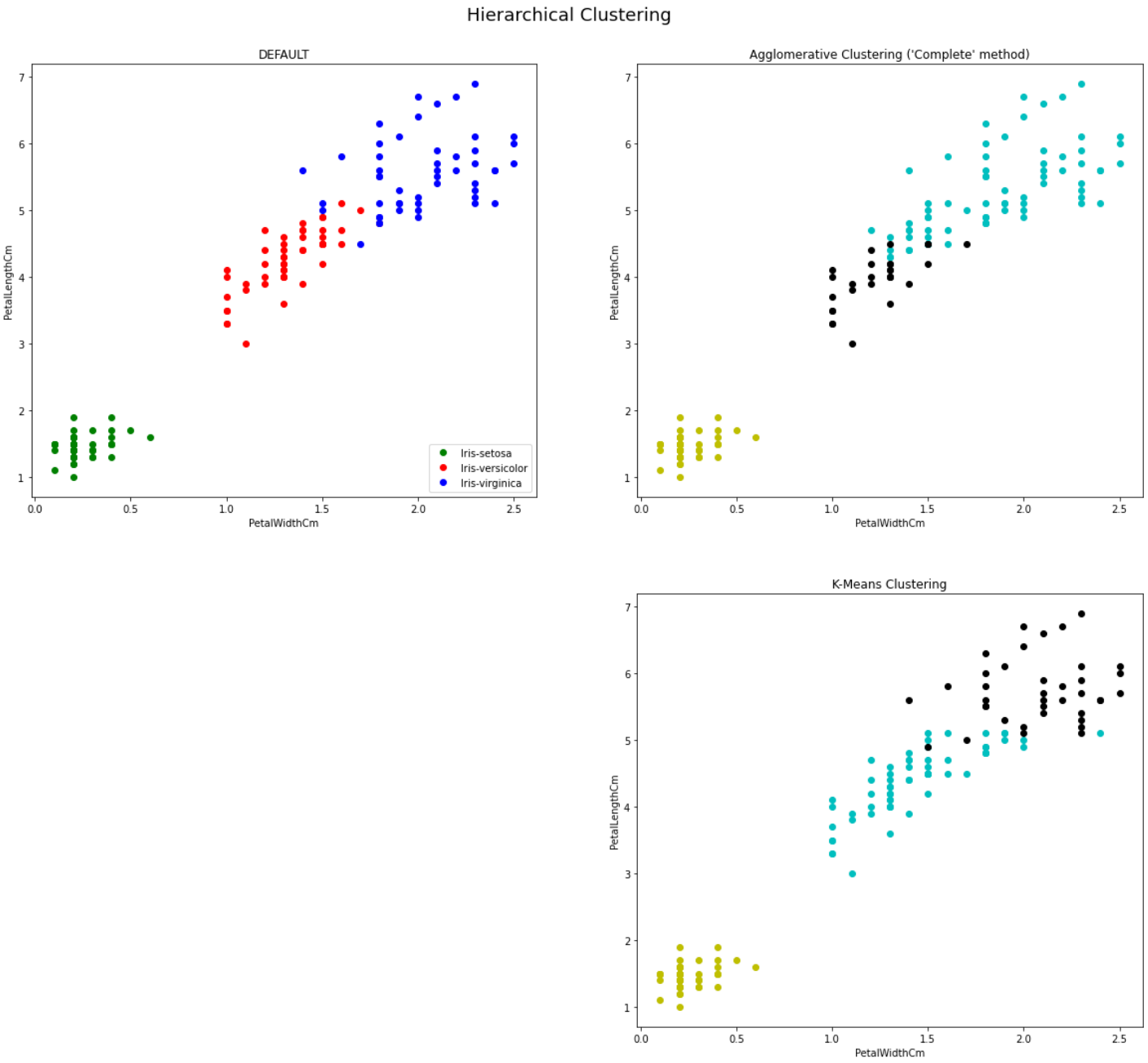
Đây là một bài toán Học không giám sát nhưng nhóm vẫn sử dụng một bộ dataset có gắn nhãn sẵn để tiện cho việc so sánh và đánh giá kết quả sau khi thực hiện thuật toán với kết quả ban đầu.

2/ Kết quả thực nghiệm: (AgglomerativeClustering trong thư viện sklearn)

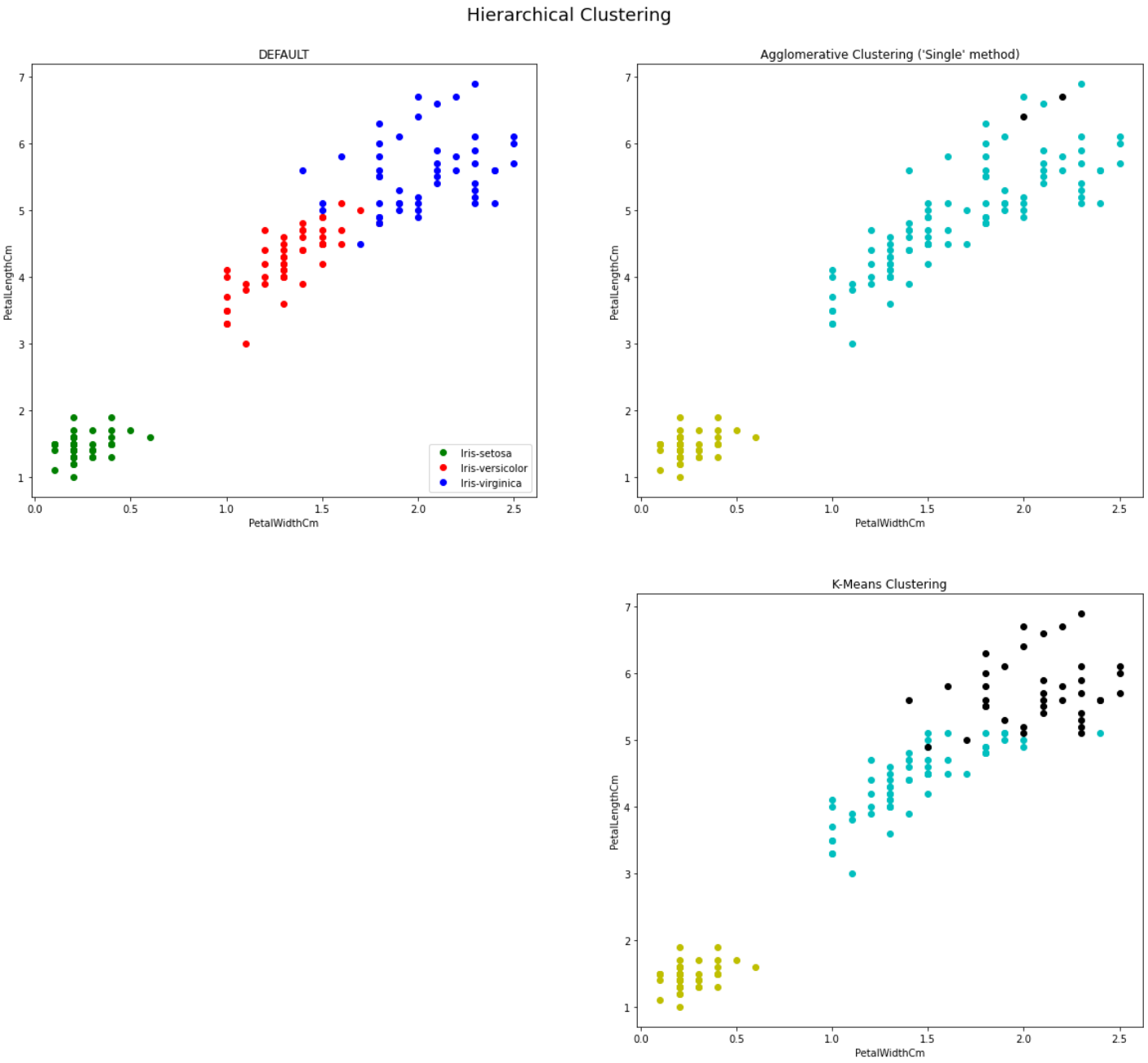
a) linkage = 'ward', affinity = 'euclide', n_clusters = 3



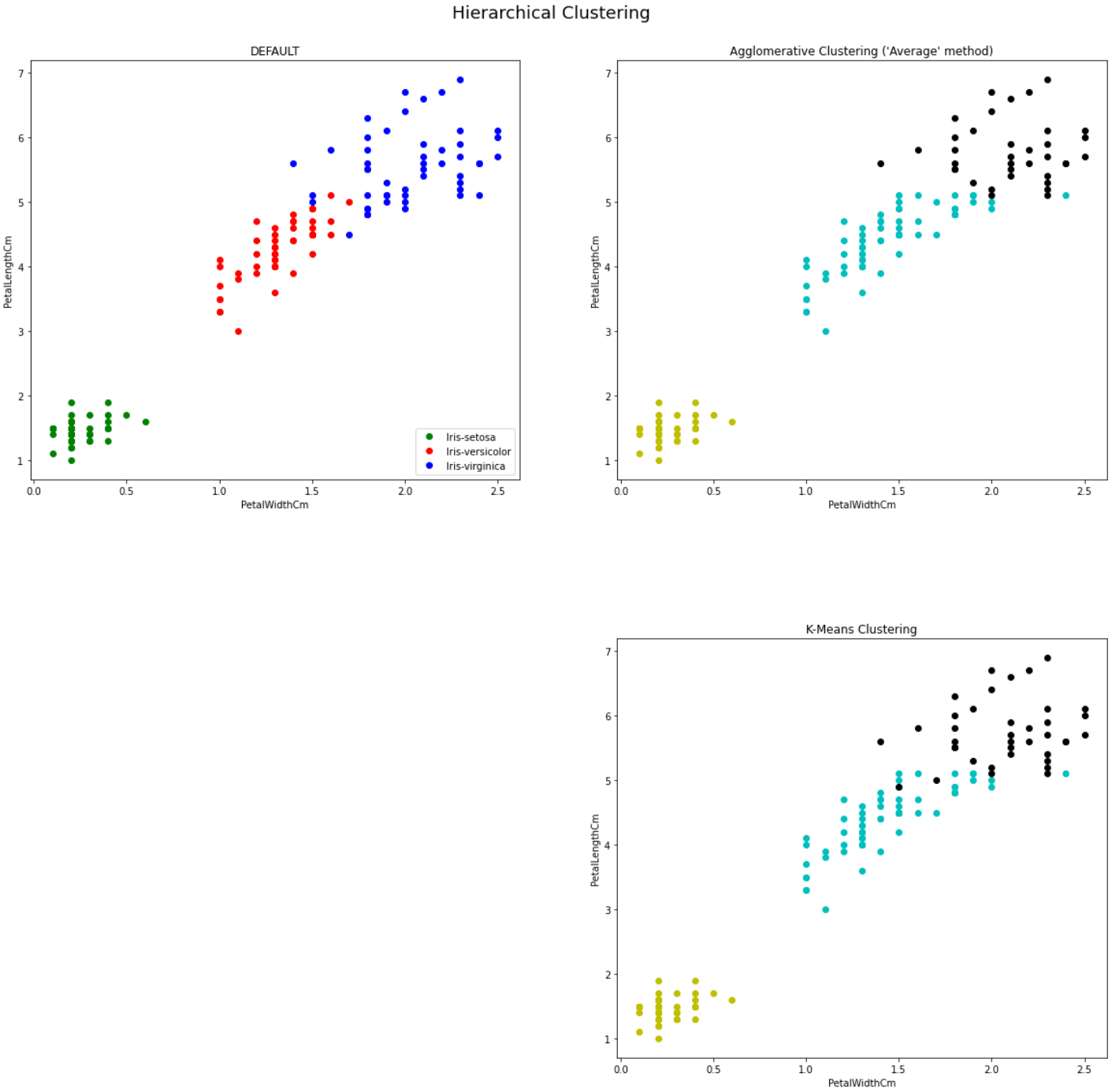
b) linkage = ‘complete’, affinity = ‘euclide’, n_clusters = 3:



c) linkage = ‘single’, affinity = ‘euclide’, n_clusters = 3:



d) linkage = ‘average’, affinity = ‘euclide’, n_clusters = 3:



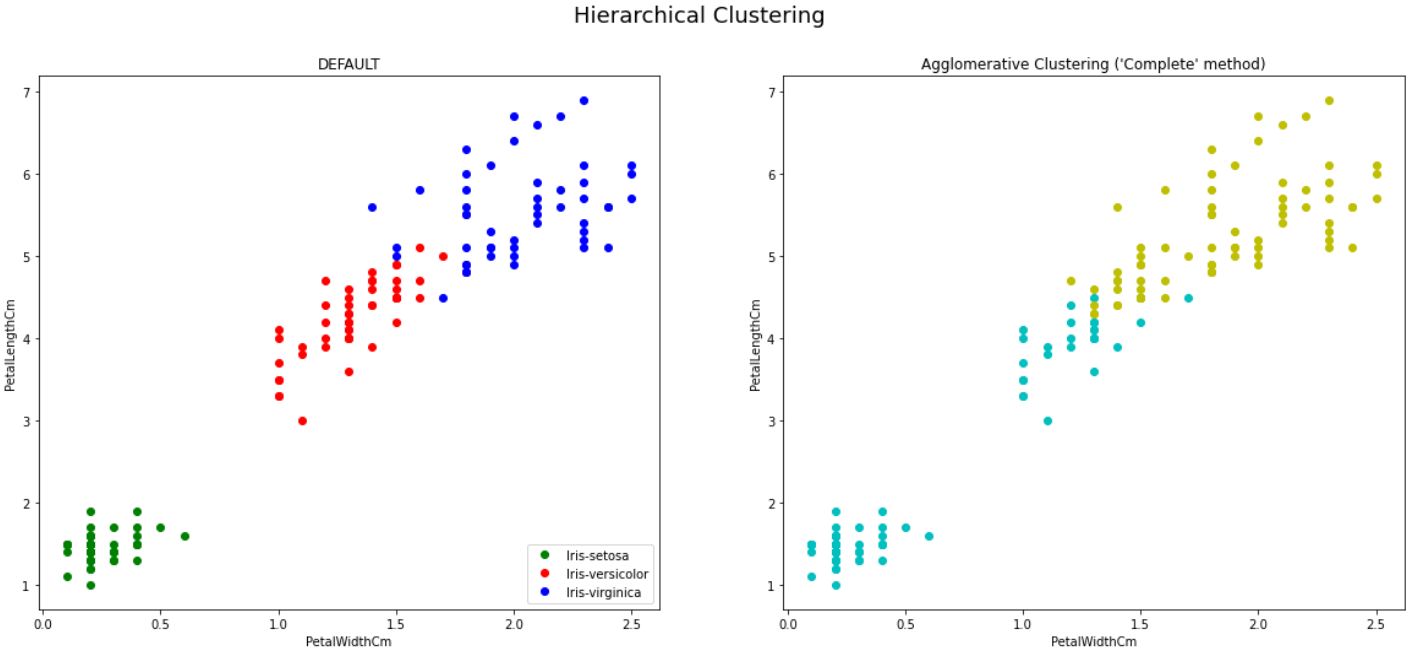
- **<u>Nhận xét:</u>**

Từ kết quả thực nghiệm đối với bộ IRIS, có thể thấy thuật toán gom cụm cho kết quả tốt nhất với method **linkage=‘complete’**.

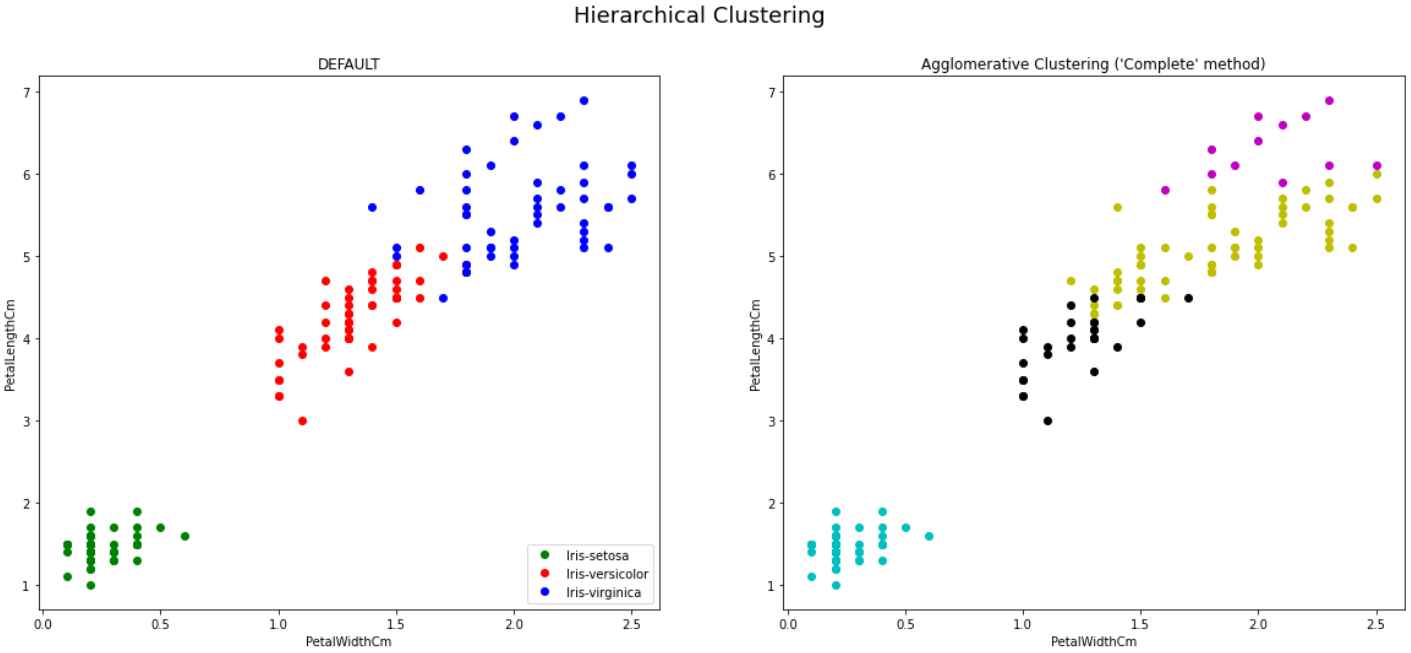
Còn đối với method linkage=‘single’ lại cho kết quả gom cụm khá tệ, với 1 cụm chỉ có 2 điểm dữ liệu.

Thử nghiệm với số cụm khác:

- **n_clusters = 2:**



- **n_clusters = 4:**



Github: **https://github.com/namt9/CS116-HierarchicalClustering**

TÀI LIỆU THAM KHẢO:

https://viblo.asia/p/hierarchical-clustering-phan-cum-du-lieu-maGK7q2elj2

https://phamdinhkhanh.github.io/deepai-book/ch_ml/HierarchicalClustering.html#do-phuc-tap-cua-thuat-toan-phan-cum-phan-cap

http://scholar.vimaru.edu.vn/sites/default/files/thinhnv/files/dm_-_chapter_5_-_clustering.pdf

https://www.analyticsvidhya.com/blog/2019/05/beginners-guide-hierarchical-clustering/

https://www.geeksforgeeks.org/hierarchical-clustering-in-data-mining/

https://www.javatpoint.com/hierarchical-clustering-in-machine-learning

https://towardsdatascience.com/introduction-to-hierarchical-clustering-part-1-theory-linkage-and-affinity-e3b6a4817702