#### **ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ**

#### **MINH TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA TOÁN - ỨNG DỤNG**

#### 



**TIỂU LUẬN**

**KHAI PHÁ DỮ LIỆU**

**ĐỀ TÀI:** Thực hành Phân Cụm Dữ Liệu với K-means và Phân Cụm Đa Cấp

**Giáo viên hướng dẫn:** Đỗ Như Tài

**Nhóm thực hiện:** Nhóm 12

TP. Hồ Chí Minh, Tháng 12, năm 2025

**DANH SÁCH PHÂN CÔNG THÀNH VIÊN**

|  |  |  |
| --- | --- | --- |
| **Tên thành viên** | **MSSV** | **Công việc** |
| **Lê Phương Vy** | 3123580064 | Nhóm trưởng  Phân công công việc  Phần 1 code  Báo cáo |
| **Nguyễn Văn Hậu** | 3123580012 | Phần 2 code  Báo cáo |
| **Nguyễn Thị Minh Thùy** | 3123580049 | Phần 2 code  Báo cáo  Kết luận |
| **Đỗ Ngọc Như Ý** | 3123580067 | Phần 1 code  Báo cáo |

**LỜI MỞ ĐẦU**

Trong lĩnh vực học máy, các kỹ thuật phân cụm (clustering) đóng vai trò quan trọng trong việc khám phá cấu trúc tiềm ẩn của dữ liệu khi không có nhãn (unsupervised learning). Phân cụm giúp nhóm các đối tượng có đặc điểm tương đồng lại với nhau, hỗ trợ quá trình phân tích, rút trích thông tin và hiểu rõ hơn bản chất phân bố của dữ liệu.

Bài báo cáo thực hành này tập trung vào hai phương pháp phân cụm phổ biến là thuật toán K-means và phân cụm đa cấp (Hierarchical Clustering). Đây là những kỹ thuật được ứng dụng rộng rãi trong khai phá dữ liệu nhờ khả năng mô tả cấu trúc nhóm, trực quan hóa quá trình phân cụm và cung cấp góc nhìn tổng quát về mối quan hệ giữa các điểm dữ liệu.

Trong quá trình thực hành, người học sẽ được hướng dẫn cách triển khai K-means và phân cụm đa cấp bằng Python, đồng thời thực hiện trực quan hóa kết quả để đánh giá mức độ phân nhóm. Ngoài ra, báo cáo cũng giúp hệ thống hóa và ôn tập lại các kiến thức lý thuyết cốt lõi liên quan đến hai thuật toán này nhằm củng cố nền tảng học thuật.

Báo cáo được thực hiện nhằm nâng cao kỹ năng xử lý dữ liệu, hiểu rõ nguyên lý vận hành của các phương pháp phân cụm, và ứng dụng chúng một cách đúng kỹ thuật trong các bài toán thực tế thuộc lĩnh vực phân tích và khai phá dữ liệu.

**LỜI CẢM ƠN**

Nhóm chúng em xin gửi lời cảm ơn chân thành đến thầy Đỗ Như tài – giảng viên phụ trách môn Khai phá dữ liệu đã tận tình giảng dạy và hướng dẫn chúng em trong suốt quá trình học.  
Thông qua các buổi học và thực hành trên JupyterLab, thầy đã giúp chúng em hiểu rõ hơn về quy trình khai phá dữ liệu, từ bước tiền xử lý, phân tích khám phá dữ liệu (EDA) cho đến trực quan hóa và rút ra nhận xét từ dữ liệu thực tế.

Nhờ sự giảng dạy nhiệt tình, dễ hiểu và những ví dụ minh họa cụ thể của thầy, nhóm không chỉ nắm vững kiến thức lý thuyết mà còn biết cách áp dụng vào bài thực hành một cách hiệu quả.  
Môn học này đã giúp chúng em rèn luyện thêm kỹ năng lập trình với Python, tư duy phân tích dữ liệu và cách trình bày kết quả khoa học.

Bài báo cáo tuy đã được nhóm cố gắng hoàn thiện, nhưng chắc chắn vẫn còn những thiếu sót. Nhóm rất mong nhận được những góp ý quý báu từ thầy để rút kinh nghiệm và cải thiện hơn trong các bài làm sau.

Nhóm chúng em xin chân thành cảm ơn thầy!

**MỤC LỤC**

[A. GIẢI THUẬT K-Means 1](#_Toc216047977)

[I. Ôn tập lý thuyết 1](#_Toc216047978)

[1. Giải thuật K-Means hoạt động như thế nào? Hãy giải thích các bước chính trong quy trình phân cụm 1](#_Toc216047979)

[2. Tại sao cần chọn số lượng cụm (K) trước khi chạy K-Means? Làm thế nào để xác định giá trị K tối ưu? 1](#_Toc216047980)

[3. Hàm mục tiêu (objective function) của K-Means là gì? Nó đo lường điều gì trong quá trình phân cụm? 1](#_Toc216047981)

[4. Những hạn chế của K-Means là gì? Trong trường hợp nào K-Means có thể cho kết quả không tốt? 2](#_Toc216047982)

[5. Viết đoạn code mẫu bằng Python (sử dụng Scikit-learn) để triển khai K-Means Clustering không? Hãy mô tả các bước thực hiện 2](#_Toc216047983)

[6. Sử dụng phương pháp nào trong Python để chọn số cụm K tối ưu (ví dụ: Elbow Method, Silhouette Score)? Hãy chia sẻ một đoạn code mẫu 3](#_Toc216047984)

[7. K-Means nhạy cảm với giá trị khởi tạo (initial centroids), bạn sẽ làm gì để đảm bảo kết quả ổn định (ví dụ: K-Means++)? 3](#_Toc216047985)

[8. Làm thế nào để đánh giá chất lượng của các cụm được tạo bởi K-Means? Bạn sử dụng chỉ số nào (ví dụ: Silhouette Score, Within-Cluster Sum of Squares)? 3](#_Toc216047986)

[II. Bài tập ứng dụng 4](#_Toc216047987)

[1. Xây dựng mô hình dựa vào thuật toán phân cụm K-Means trên tập dữ liệu Mall-Customer. 4](#_Toc216047988)

[*1.1.* *Thực hiện giải thuật K-Means để phân cụm dữ liệu khách.* 4](#_Toc216047989)

[*1.2.* *Phân cụm dữ liệu trên tập dữ liệu iris-data* 11](#_Toc216047990)

[2. Xây dựng mô hình phân cụm K-means trên tập dữ liệu chim cánh cụt. 19](#_Toc216047991)

[*2.1. Khai báo và đọc dữ liệu* 19](#_Toc216047992)

[*2.2. Tiền xử lý dữ liệu* 20](#_Toc216047993)

[*2.3. Phân bố dữ liệu giữa hai nhóm giới tính* 21](#_Toc216047994)

[*2.4. Phân tích biến culmen\_length\_mm* 22](#_Toc216047995)

[*2.5. Phân tích biến culmen\_depth\_mm* 24](#_Toc216047996)

[*2.6. Phân tích biến flipper\_length\_mm* 25](#_Toc216047997)

[*2.7. Xử lý ngoại lai* 27](#_Toc216047998)

[*2.8. Phân tích biến body\_mass\_g* 27](#_Toc216047999)

[*2.9. Ma trận tương quan* 29](#_Toc216048000)

[*2.10. Phân tích 2 biến body\_mass\_g và flipper\_length\_mm* 31](#_Toc216048001)

[*2.11. Phân tích 2 biến flipper\_length\_mm và culmen\_length\_mm* 31](#_Toc216048002)

[*2.12. Phân cụm dữ liệu bằng thuật toán K-Means* 32](#_Toc216048003)

[*2.13. Trực quan hóa kết quả phân cụm bằng PCA* 34](#_Toc216048004)

[3. Xây dựng mô hình phân cụm K-means trên tập dữ liệu mua sắm tại siêu thị. 36](#_Toc216048005)

[*3.1. Khai báo và đọc dữu liệu* 36](#_Toc216048006)

[*3.2. Khám phá dữ liệu* 37](#_Toc216048007)

[*3.3. Kiểm tra giá trị thiếu* 38](#_Toc216048008)

[*3.4. Xử lý giá trị thiếu* 38](#_Toc216048009)

[*3.5. Chuẩn hóa và trích xuất thông tin thời gian* 39](#_Toc216048010)

[*3.6. Tạo biến mới và Loại bỏ giao dịch không hợp lệ* 39](#_Toc216048011)

[*3.7. Khám phá dữ liệu EDA* 40](#_Toc216048012)

[*3.8. Chuẩn bị dữ liệu cho phân cụm* 46](#_Toc216048013)

[*3.9. Phân cụm dữ liệu bằng thuật toán K-Means* 49](#_Toc216048014)

[*3.10. Trực quan hóa kết quả phân cụm bằng PCA* 51](#_Toc216048015)

[B. GIẢI THUẬT PHÂN CỤM ĐA CẤP 55](#_Toc216048016)

[I. Ôn tập lý thuyết 55](#_Toc216048017)

[1. Giải thuật phân cụm đa cấp hoạt động như thế nào? Hãy giải thích sự khác biệt giữa phân cụm đa cấp hợp nhất (agglomerative) và phân tách (divisive) 55](#_Toc216048018)

[2. Các phương pháp liên kết (linkage) như single linkage, complete linkage, average linkage, và Ward’s method khác nhau ra sao? Khi nào nên sử dụng từng loại? 55](#_Toc216048019)

[3. Dendrogram trong phân cụm đa cấp là gì? Làm thế nào để sử dụng nó để chọn số lượng cụm? 56](#_Toc216048020)

[4. Phân cụm đa cấp có thể áp dụng cho dữ liệu phi số (non-numeric data) như thế nào? Hãy giải thích 56](#_Toc216048021)

[5. Viết đoạn code mẫu bằng Python (sử dụng Scikit-learn) để triển khai phân cụm đa cấp hợp nhất (agglomerative clustering) không? Hãy mô tả các bước thực hiện 57](#_Toc216048022)

[6. Làm thế nào để vẽ dendrogram trong Python sử dụng thư viện như scipy hoặc matplotlib? Hãy chia sẻ một đoạn code mẫu 58](#_Toc216048023)

[7. Các lớp trong gói Scipy hỗ trợ phân cụm đa cấp? và so sánh giữa cách tiếp cận scikit-learn và cách tiếp cận sử dụng Scipy 58](#_Toc216048024)

[II. Bài tập ứng dụng 59](#_Toc216048025)

[1. Tìm hiểu một số khái niệm cơ bản của phân cụm đa cấp 59](#_Toc216048026)

[*1.1.* *Khai báo và tạo dữ liệu giả* 59](#_Toc216048027)

[*1.2.* *Hiển thị ma trận khoảng cách giữa các điểm dữ liệu với điểm trung tâm* 60](#_Toc216048028)

[*1.3. Vẽ hình dendrogram từ ma trận khoảng cách* 61](#_Toc216048029)

[*1.4. Vẽ các cụm dữ liệu dựa ma trận khoảng cách và thông tin từ dendrogram với hàm fcluster* 62](#_Toc216048030)

[*1.5. Vẽ các cụm dữ liệu với các phương pháp linkage khác nhau* 63](#_Toc216048031)

[*1.6. Xây dựng mô hình phân cụm đa cấp theo tiếp cận của thư viện scikit-learn* 67](#_Toc216048032)

[2. Phân cụm đa cấp trên tập dữ liệu rượu vang 67](#_Toc216048033)

[*2.1. Khai báo dữ liệu và thư viện* 67](#_Toc216048034)

[*2.2. Xây dựng mô hình phân cụm phân cấp (Agglomerative Clustering)* 68](#_Toc216048035)

[*2.3. Phân cụm K-Means* 69](#_Toc216048036)

[*2.4. Đánh giá chất lượng phân cụm bằng Silhouette Score* 70](#_Toc216048037)

[3. Xây dựng mô hình phân cụm đa cấp trên tập dữ liệu chim cánh cụt 71](#_Toc216048038)

[*3.1. Khai báo dữ liệu và tiền xử lý* 71](#_Toc216048039)

[*3.2. X* *ây dựng mô hình phân cụm phân cấp (Agglomerative Clustering)* 72](#_Toc216048040)

[*3.3. Xây dựng mô hình phân cụm phân cấp* 74](#_Toc216048041)

[*3.4. Biểu đồ phân tán* 75](#_Toc216048042)

[*3.5. Kết quả mô hình* 76](#_Toc216048043)

[*3.6. Đánh giá chất lượng phân cụm* 77](#_Toc216048044)

[4. Xây dựng mô hình phân cụm đa cấp trên tập dữ liệu mua sắm tại siêu thị 77](#_Toc216048045)

[*4.1. Khai báo thư viện và đọc dữ liệu* 77](#_Toc216048046)

[*4.2. Xem thông tin dữ liệu* 78](#_Toc216048047)

[*4.3. Tiền xử lý dữ liệu* 79](#_Toc216048048)

[*4.4. Xây dựng mô hình phân cụm đa cấp* 81](#_Toc216048049)

[*4.5. Biểu đồ Dendrogram* 82](#_Toc216048050)

[*4.6. Gán nhãn cụm và xác định phân khúc khách hàng* 83](#_Toc216048051)

[*4.7. Kết quả thống kê trung bình theo từng phân khúc* 83](#_Toc216048052)

[C. Tóm tắt thực hành 85](#_Toc216048053)

# **A. GIẢI THUẬT K-Means**

## **I. Ôn tập lý thuyết**

### **1. Giải thuật K-Means hoạt động như thế nào? Hãy giải thích các bước chính trong quy trình phân cụm**

Thuật toán K-Means là phương pháp phân cụm dựa trên khoảng cách, nhằm chia dữ liệu thành *K cụm* sao cho các điểm trong cùng một cụm giống nhau nhất có thể. Quy trình hoạt động gồm các bước chính:

* Khởi tạo K tâm cụm (centroid): Có thể chọn ngẫu nhiên hoặc dùng phương pháp K-Means++.
* Gán cụm: Mỗi điểm dữ liệu được gán vào cụm có tâm gần nhất dựa trên khoảng cách (thường là Euclidean).
* Cập nhật tâm cụm: Tâm cụm mới được tính bằng trung bình của tất cả các điểm thuộc cụm đó.
* Lặp lại hai bước gán cụm và cập nhật tâm cho đến khi:
  + Tâm cụm hội tụ (không thay đổi đáng kể), hoặc
  + Đạt số vòng lặp tối đa.

Kết quả cuối cùng là K cụm ổn định.

### **2. Tại sao cần chọn số lượng cụm (K) trước khi chạy K-Means? Làm thế nào để xác định giá trị K tối ưu?**

K-Means yêu cầu số cụm K trước khi chạy vì thuật toán cần biết có bao nhiêu tâm cụm cần được khởi tạo. Nếu K không hợp lý, kết quả phân cụm có thể bị sai lệch (quá chi tiết hoặc quá chung chung).

Các phương pháp xác định K tối ưu:

* Elbow Method: Tính Within-Cluster Sum of Squares (WCSS) cho các giá trị K khác nhau, chọn K tại “điểm khuỷu tay” – nơi WCSS giảm chậm lại.
* Silhouette Score: Chọn K có điểm Silhouette cao nhất (điểm gần 1 cho thấy phân cụm tốt).
* Gap Statistic: So sánh phân cụm thực tế với dữ liệu ngẫu nhiên (ít dùng hơn trong thực hành cơ bản).

### **3. Hàm mục tiêu (objective function) của K-Means là gì? Nó đo lường điều gì trong quá trình phân cụm?**

Hàm mục tiêu của K-Means là tối thiểu hóa tổng bình phương khoảng cách từ các điểm đến tâm cụm tương ứng, còn gọi là Within-Cluster Sum of Squares (WCSS):

A black and white math symbols

AI-generated content may be incorrect.

Trong đó:

* : cụm thứ i
* : tâm của cụm i

Hàm mục tiêu đo lường độ chặt của các cụm – càng nhỏ thì phân cụm càng tốt.

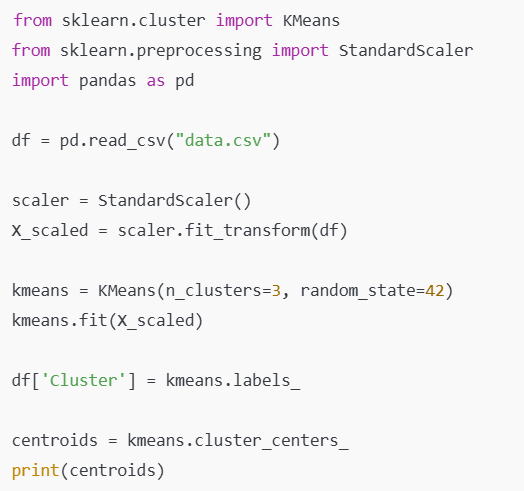
### **4. Những hạn chế của K-Means là gì? Trong trường hợp nào K-Means có thể cho kết quả không tốt?**

Hạn chế chính:

* Yêu cầu chọn trước số cụm K, đôi khi không rõ ràng.
* Nhạy cảm với giá trị ban đầu → có thể hội tụ vào nghiệm cục bộ.
* Không phù hợp với cụm có hình dạng phức tạp, ví dụ dạng phi tuyến hoặc lồi lõm.
* Không tốt với dữ liệu chứa nhiều nhiễu hoặc outlier.
* Phụ thuộc vào thang đo dữ liệu → cần chuẩn hóa.

K-Means cho kết quả không tốt khi:

* Dữ liệu phân bố không tròn hoặc không chia tách bằng khoảng cách Euclidean.
* Số lượng cụm thực tế chênh lệch lớn về kích thước.
* Dữ liệu có nhiều outlier hoặc không được chuẩn hóa.
  1. **Viết đoạn code mẫu bằng Python (sử dụng Scikit-learn) để triển khai K-Means Clustering không? Hãy mô tả các bước thực hiện**

****

Các bước thực hiện bao gồm: chuẩn hóa dữ liệu, chọn số cụm K, huấn luyện mô hình và gán nhãn cụm.

* 1. **Sử dụng phương pháp nào trong Python để chọn số cụm K tối ưu (ví dụ: Elbow Method, Silhouette Score)? Hãy chia sẻ một đoạn code mẫu**

a) Elbow Method

A screenshot of a computer program

AI-generated content may be incorrect.

b) Silhouette Score

A screen shot of a computer code

AI-generated content may be incorrect.

* 1. **K-Means nhạy cảm với giá trị khởi tạo (initial centroids), bạn sẽ làm gì để đảm bảo kết quả ổn định (ví dụ: K-Means++)?**

Để tăng độ ổn định:

* Sử dụng K-Means++ (phương pháp khởi tạo thông minh, giảm nguy cơ rơi vào nghiệm xấu).
* Tăng số lần chạy (n\_init) để thử nhiều lần khởi tạo:

****

* Chuẩn hóa dữ liệu để giảm biến dạng do thang đo.
  1. **Làm thế nào để đánh giá chất lượng của các cụm được tạo bởi K-Means? Bạn sử dụng chỉ số nào (ví dụ: Silhouette Score, Within-Cluster Sum of Squares)?**

Một số chỉ số đánh giá phổ biến:

* Silhouette Score: đo độ tách biệt giữa các cụm (giá trị gần 1 là tốt).
* WCSS (Within-Cluster Sum of Squares): tổng khoảng cách trong cụm, càng nhỏ càng tốt.
* Between-Cluster Sum of Squares (BCSS): khoảng cách giữa các cụm, càng lớn càng tốt.
* Davies–Bouldin Index: càng nhỏ càng tốt.

**II. Bài tập ứng dụng**

* 1. **Xây dựng mô hình dựa vào thuật toán phân cụm K-Means trên tập dữ liệu Mall-Customer.**
  2. ***Thực hiện giải thuật K-Means để phân cụm dữ liệu khách.***

*1.1.1. Khai báo dữ liệu*

**A screen shot of a computer code

AI-generated content may be incorrect.**

Đầu tiên, chương trình nạp các thư viện cần thiết để xử lý dữ liệu, trực quan hóa và thực hiện mô hình K-Means. Các thư viện bao gồm NumPy, Pandas, Matplotlib, Seaborn và các hàm phân cụm của Scikit-learn. Việc tắt cảnh báo giúp quá trình chạy gọn gàng hơn.

*1.1.2. Đọc dữ liệu và tiền xử lý*

[Input]

**A computer screen shot of a code

AI-generated content may be incorrect.**

[Output]

**A screenshot of a computer code

AI-generated content may be incorrect.**

Sau khi đọc tập dữ liệu *Mall\_Customers.csv* và thực hiện mã hóa biến giới tính (Gender) từ dạng ký tự sang dạng số (Male = 1, Female = 0), thông tin tổng quan về DataFrame cho thấy:

* Tập dữ liệu gồm 200 quan sát và 5 thuộc tính, bao gồm:  
   CustomerID, Gender, Age, Annual Income (k$) và Spending Score (1–100).
* Tất cả các cột đều có 200 giá trị hợp lệ (non-null), nghĩa là không tồn tại dữ liệu bị thiếu, giúp việc phân cụm diễn ra thuận lợi mà không cần xử lý thiếu hụt.
* Kiểu dữ liệu của toàn bộ các cột đều là int64, phù hợp với yêu cầu của thuật toán K-Means (vì K-Means chỉ xử lý được dữ liệu số).
* Việc mã hóa cột Gender thành giá trị nhị phân (0 và 1) giúp mô hình tính toán khoảng cách giữa các điểm dễ dàng hơn.

Tiếp theo, để chuẩn bị phân cụm, dữ liệu sử dụng cho mô hình được chọn dùng 4 biến đầu vào cho quá trình phân cụm:

* Gender
* Age
* Annual Income (k$)
* Spending Score (1–100)

Các biến này đều mang tính định lượng hoặc đã được chuẩn hóa về dạng số, đảm bảo phù hợp với thuật toán K-Means. Đây cũng là những đặc trưng quan trọng giúp mô hình phân nhóm khách hàng dựa trên thu nhập, độ tuổi và hành vi chi tiêu.

*1.1.3. Xây dựng mô hình K-Means*

[Input]

**A white background with colorful text

AI-generated content may be incorrect.**

[Output]

**A screenshot of a computer

AI-generated content may be incorrect.**

* Xét theo Inertia
  + Inertia giảm dần khi tăng số cụm — đây là hành vi bình thường của K-Means.
  + Tuy nhiên, giảm mạnh nhất trong khoảng:
    - từ k=3 → 4
    - từ k=4 → 5
    - từ k=5 → 6
  + Sau k=6, mức giảm không còn đáng kể ⇒ biểu hiện điểm “khuỷu tay” (Elbow).

k = 5 hoặc k = 6 có thể là lựa chọn hợp lý theo tiêu chí Inertia.

* Xét theo Silhouette Score
  + Silhouette tăng từ k=3 → 6.
  + Mức cao nhất đạt tại k = 6 (0.4507).
  + Sau đó giảm dần ở k = 7 và k = 8

k = 6 là số cụm tối ưu theo tiêu chí Silhouette.

**Kết luận:**

Dựa trên cả hai chỉ số:

* Inertia giảm mạnh đến khoảng k = 5–6
* Silhouette Score đạt mức cao nhất khi k = 6

Số cụm phù hợp nhất là k = 6.

*1.1.4. Tìm số cụm tối ưu cho mô hình K-Means bằng phương pháp Elbow*

[Input]

**A close-up of a computer code

AI-generated content may be incorrect.**

[Output]

**A graph with a line

AI-generated content may be incorrect.**

* Inertia giảm rất mạnh khi tăng k từ 3 → 4 → 5.
* Giảm tiếp nhưng chậm hơn ở k = 6.
* Từ k = 7 trở đi, độ giảm gần như không còn đáng kể.

Đây chính là đặc điểm thường thấy khi tìm “điểm khuỷu tay” (Elbow).  
 Điểm gãy của đường cong xuất hiện ở khoảng k = 5 hoặc k = 6, khi độ dốc bắt đầu giảm rõ rệt.

**Kết luận:**

Dựa trên độ cong của đồ thị:

* Trước k = 6: Inertia giảm nhanh → mô hình phân cụm được cải thiện đáng kể.
* Sau k = 6: Đường cong “phẳng” hơn → tăng số cụm không mang lại nhiều lợi ích.

k = 6 được xem là số cụm phù hợp nhất theo phương pháp Elbow.

Kết quả này đồng nhất với phân tích Silhouette Score, cũng đạt cao nhất tại k = 6, cho thấy mô hình ở mức này có sự tách cụm rõ ràng, cấu trúc chặt chẽ và cân bằng tốt giữa số lượng cụm và mức độ phân tách dữ liệu.

*1.1.5. Tìm số cụm tối ưu bằng phương pháp Silhoutte*

[Input]

**A white background with black and red text

AI-generated content may be incorrect.**

[Output]

A graph with a line

AI-generated content may be incorrect.

* Khi k = 3, điểm khá thấp (~0.31), cho thấy cấu trúc phân cụm chưa rõ ràng.
* Score tăng mạnh tại k = 4 và k = 5, lần lượt đạt khoảng 0.39 và 0.44.
* Giá trị cao nhất xuất hiện tại k = 6, đạt 0.45+, thể hiện đây là số cụm cho chất lượng phân tách tốt nhất.
* Từ k = 7, điểm giảm xuống và tiếp tục giảm mạnh khi đến k = 8, chứng tỏ phân cụm không còn rõ ràng.
* k = 9 có cải thiện nhẹ nhưng vẫn thấp hơn đáng kể so với k = 6.

**Kết luận**

Biểu đồ cho thấy k = 6 mang lại:

* Mức Silhouette Score cao nhất
* Sự cân bằng tốt nhất giữa độ tách rời giữa các cụm và độ liên kết nội cụm
* Chất lượng phân cụm vượt trội so với các giá trị k khác

Do đó, k = 6 là lựa chọn tối ưu dựa trên Silhouette Score.

Kết luận này hoàn toàn phù hợp với kết quả từ phương pháp Elbow, càng củng cố việc chọn 6 cụm là hợp lý nhất cho bộ dữ liệu Mall Customers.

*1.1.6. Gán nhãn cho các mẫu dữ liệu dựa vào mô hình K-Means với số cụm là 6*

[Input]

**A screenshot of a computer code

AI-generated content may be incorrect.**

[Output]

A screenshot of a computer screen

AI-generated content may be incorrect.A screenshot of a computer screen

AI-generated content may be incorrect.

Dựa trên các bảng thống kê thu được, đặc điểm của từng cụm có thể được mô tả như sau:

* **Cluster 0:** Nhóm khách hàng có thu nhập cao nhưng mức chi tiêu ở mức trung bình. Đây thường là nhóm trung niên, có hành vi chi tiêu khá ổn định và ít biến động.
* **Cluster 1:** Nhóm khách hàng trẻ, thu nhập cao và Spending Score cũng cao. Đây là nhóm “khách hàng giá trị cao”, có xu hướng tiêu dùng mạnh và sẵn sàng chi trả cho các dịch vụ cao cấp.
* **Cluster 2:** Nhóm có thu nhập thấp và mức chi tiêu thấp. Đây là nhóm ít tiềm năng về mặt doanh thu và thường phù hợp cho các chương trình kích cầu hoặc khuyến mãi đơn giản.
* **Cluster 3:** Nhóm khách hàng trung niên với thu nhập mức vừa nhưng Spending Score thấp. Nhóm này có xu hướng chi tiêu tiết kiệm và ít tham gia vào các hoạt động mua sắm lớn.
* **Cluster 4:** Nhóm khách hàng trẻ, thu nhập không cao nhưng Spending Score lại rất cao. Điều này thể hiện hành vi chi tiêu cảm tính và dễ bị thu hút bởi khuyến mãi hoặc các sản phẩm thời thượng.
* **Cluster 5:** Nhóm có thu nhập cao nhưng Spending Score thấp. Đây là nhóm khách hàng tiềm năng nhưng hiện chưa chi tiêu nhiều; có thể cần các chiến lược marketing phù hợp để khai thác thêm.

Nhìn chung, kết quả phân cụm với K = 6 đã phân chia khách hàng thành các nhóm có đặc điểm khác biệt rõ rệt về độ tuổi, thu nhập và hành vi tiêu dùng. Các thông tin này có giá trị quan trọng trong việc xây dựng chiến lược marketing, phân khúc khách hàng, và thiết kế chương trình chăm sóc khách hàng phù hợp.

* 1. ***Phân cụm dữ liệu trên tập dữ liệu iris-data***

*1.2.1. Khai báo dữ liệu*

A screenshot of a computer code

AI-generated content may be incorrect.

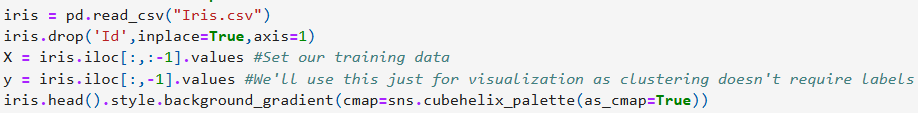
Trong bước đầu tiên, các thư viện cần thiết cho việc xử lý, trực quan hóa và phân cụm dữ liệu được import, bao gồm:

* NumPy, Pandas: xử lý dữ liệu dạng bảng và số học.
* Seaborn, Plotly Express, Plotly Graph Objects: trực quan hóa dữ liệu dưới dạng boxplot, histogram, scatter plot.
* Scikit-learn: triển khai thuật toán K-Means.
* MinMaxScaler: chuẩn hóa dữ liệu về cùng một thang đo.

Việc khai báo đầy đủ thư viện giúp quá trình phân tích diễn ra thuận lợi và thống nhất cho toàn bộ pipeline xử lý dữ liệu.

*1.2.2. Đọc dữ liệu và xử lý ban đầu*

[Input]



[Output]

A screenshot of a computer screen

AI-generated content may be incorrect.

Tập dữ liệu *Iris.csv* được đọc bằng Pandas và cột Id được loại bỏ vì không mang ý nghĩa cho việc phân cụm.  
Dữ liệu còn lại gồm 4 thuộc tính:

* SepalLengthCm
* SepalWidthCm
* PetalLengthCm
* PetalWidthCm

và một thuộc tính nhãn Species, chỉ dùng cho mục đích trực quan hóa, không đưa vào mô hình phân cụm.

Việc xem trước 5 dòng đầu giúp xác định nhanh cấu trúc dữ liệu và kiểm tra chất lượng ban đầu. Màu sắc được gán theo gradient giúp quan sát mức độ biến thiên của từng thuộc tính.

*1.2.3. Phân bố nhãn (Species)*

[Input]



[Output]

A circle with a pie chart

AI-generated content may be incorrect.

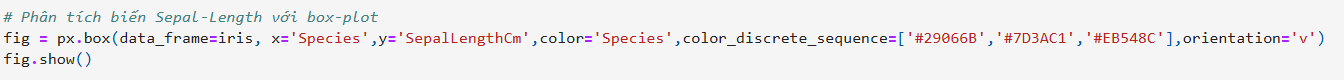
Biểu đồ tròn cho thấy ba loài *Iris-setosa*, *Iris-versicolor* và *Iris-virginica* đều chiếm **33.3%** tổng số mẫu.

**Nhận xét**

* Tập dữ liệu hoàn toàn cân bằng, mỗi lớp có số lượng bằng nhau.
* Điều này rất thuận lợi cho các thuật toán học máy và phân cụm vì không tồn tại tình trạng lệch lớp (imbalanced data).
* Không cần các kỹ thuật bổ sung như oversampling hoặc undersampling.

*1.2.4. Phân tích biến Sepal-Length với box-plot*

[Input]



[Output]

A graph with a purple rectangle

AI-generated content may be incorrect.

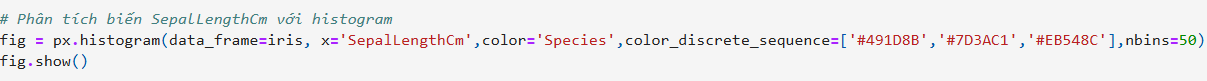
Boxplot cho thấy:

* **Iris-setosa** có Sepal Length nhỏ nhất và tách biệt rõ ràng khỏi hai loài còn lại.
* **Iris-versicolor** và **Iris-virginica** có phân bố tương đối gần nhau, trong đó Virginica có giá trị trung bình cao hơn một chút.
* Không xuất hiện giá trị ngoại lai đáng kể.

**Nhận xét:**  
Sepal Length chỉ phân biệt tốt *setosa*; Versicolor và Virginica rất khó tách biệt dựa trên thuộc tính này do đặc trưng sinh học tương đối giống nhau.

*1.2.5. Phân tích biến SepalLengthCm với histogram*

[Input]



[Output]

A graph of different colored squares

AI-generated content may be incorrect.

Histogram của Sepal Length thể hiện:

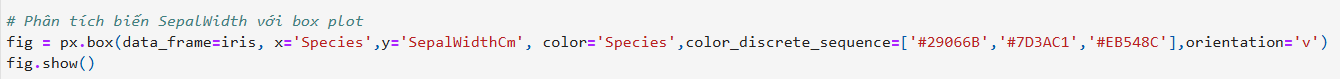
* Mỗi loài đều có dạng phân bố **unimodal** (một đỉnh).
* Setosa tập trung hoàn toàn ở vùng giá trị thấp → tách biệt tốt.
* Versicolor và Virginica tiếp tục chồng lấn nhẹ quanh vùng giá trị trung bình.

**Nhận xét:**  
Phân bố của Sepal Length xác nhận:

* Setosa dễ nhận dạng nhờ chiều dài đài hoa nhỏ.
* Versicolor và Virginica khó phân tách hoàn toàn, chỉ khác biệt nhẹ về trung bình.

*1.2.6. Phân tích biến SepalWidth với box plot*

[Input]



[Output]

A graph with a purple rectangle

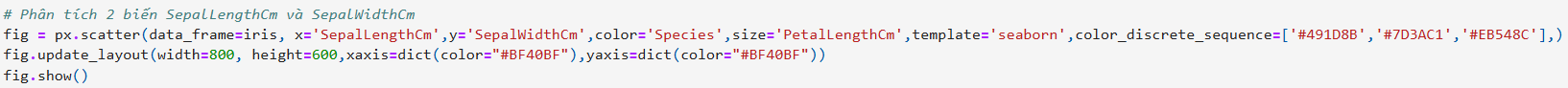
AI-generated content may be incorrect.

* Setosa có Sepal Width lớn hơn đáng kể so với hai loài còn lại.
* Virginica có độ dao động nhỏ nhất, thể hiện phân bố ổn định hơn.
* Versicolor nằm giữa hai nhóm còn lại.

**Nhận xét:**  
Mặc dù tồn tại sự khác biệt, Sepal Width không tạo ra ranh giới phân tách hoàn toàn giữa ba loài. Sepla Width chỉ hỗ trợ nhận dạng Setosa, nhưng hiệu quả phân loại giữa Versicolor và Virginica không cao.

*1.2.7. Phân tích 2 biến SepalLengthCm và SepalWidthCm*

[Input]



[Output]

A graph of dots with different colors

AI-generated content may be incorrect.

* Setosa tạo thành một nhóm riêng biệt và dễ nhận dạng.
* Versicolor và Virginica phân bố gần nhau, có sự chồng chéo rõ ràng ở vùng trung gian.
* Không có giá trị bất thường (outlier lớn).

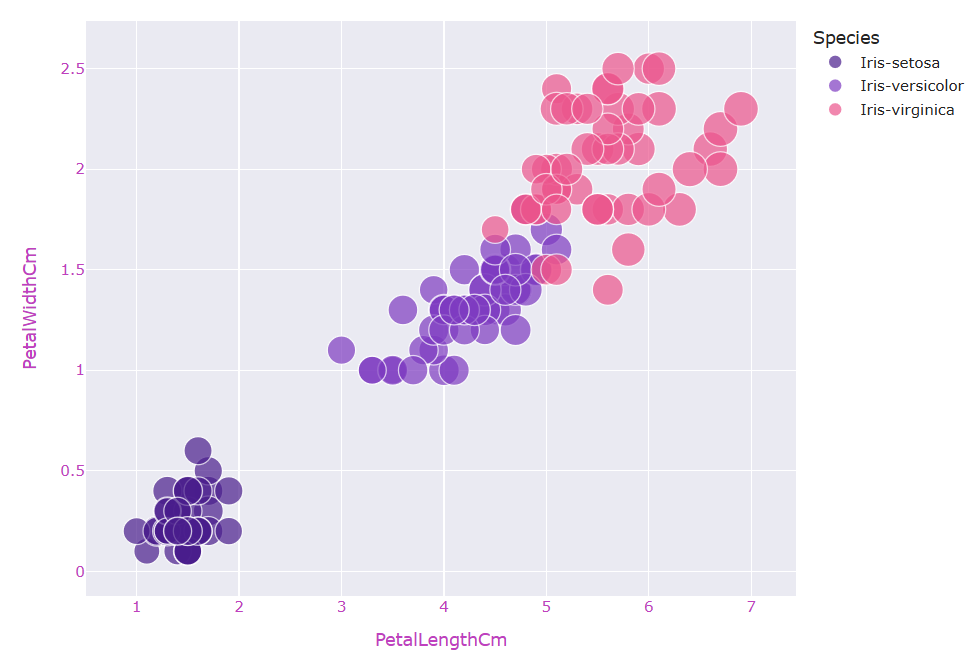
**Nhận xét:**  
Tổ hợp hai biến sepal cho cấu trúc dữ liệu tương đối rõ, nhưng không đủ mạnh để phân biệt hoàn toàn ba loài. Điều này phù hợp với đặc điểm sinh học: các thông số sepal giữa Versicolor và Virginica khá tương đồng.

*1.2.8. Phân tích 2 biến PetalLengthCm và PetalWidthCm*

[Input]



[Output]



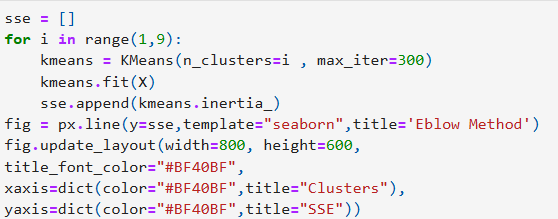
* Setosa tách biệt hoàn toàn với giá trị petal rất nhỏ → không có điểm giao nhau.
* Versicolor và Virginica tạo thành hai cụm rõ hơn:
  + Versicolor có giá trị petal ở mức trung bình
  + Virginica có giá trị petal lớn hơn đáng kể
* Chỉ xuất hiện một vùng giao nhau nhỏ giữa hai cụm.

**Nhận xét:**  
Hai thuộc tính Petal Length và Petal Width thể hiện khả năng phân tách dữ liệu mạnh nhất:

* Tạo ranh giới tự nhiên cho cả ba loài
* Cho thấy đây là hai biến quan trọng nhất trong việc phân nhóm dữ liệu Iris

*1.2.9. Xác định số cụm tối ưu bằng Elbow Method*

[Input]



[Output]

A graph with a line

AI-generated content may be incorrect.

Số cluster tối ưu = 3

**Nhận xét:**

* Điểm “khuỷu tay” xuất hiện rõ tại k = 3, nghĩa là mô hình đạt sự cân bằng tốt nhất giữa độ chính xác và số cụm.
* Đây cũng là con số trùng khớp với số loài thực tế trong tập Iris (Setosa, Versicolor, Virginica).
* Vì vậy, k = 3 được chọn cho bước phân cụm tiếp theo.

*1.2.10. Kết quả phân cụm KMeans với k = 3*

[Input]

A screenshot of a computer

AI-generated content may be incorrect.

[Output]

A screen shot of a computer screen

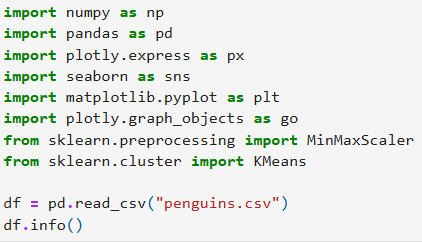
AI-generated content may be incorrect.

**Nhận xét:**

* Iris-setosa: tạo thành cụm tách biệt hoàn toàn, phân bố rất rõ ràng.
* Iris-versicolor: nằm ở vùng trung gian, có một số điểm gần ranh giới với Virginica.
* Iris-virginica: có giá trị lớn nhất về các kích thước petal, cụm nằm xa so với hai loài kia.
* Kết quả này phù hợp với phân tích biến petal (mục trước): Petal Length và Petal Width là hai thuộc tính phân tách tốt nhất.
* Mô hình KMeans đã phân cụm hợp lý và phản ánh đúng cấu trúc tự nhiên của dữ liệu
  1. **Xây dựng mô hình phân cụm K-means trên tập dữ liệu chim cánh cụt.**

***2.1. Khai báo và đọc dữ liệu***

[Input]

****

[Output]

A screenshot of a computer code

AI-generated content may be incorrect.

Đầu tiên, chương trình thực hiện import các thư viện cần thiết như numpy, pandas, cùng với các công cụ trực quan hoá như matplotlib, seaborn và plotly. Đây đều là những thư viện quan trọng giúp xử lý dữ liệu dạng bảng, chuẩn hóa giá trị số, vẽ biểu đồ và chuẩn bị dữ liệu cho các mô hình phân cụm sau này.

Tiếp theo, chương trình đọc bộ dữ liệu Penguins từ file *penguins.csv* bằng hàm pd.read\_csv(). Bộ dữ liệu này chứa thông tin sinh học của ba loài chim cánh cụt sống ở quần đảo Palmer, bao gồm chiều dài mỏ, độ sâu mỏ, chiều dài cánh, khối lượng cơ thể, giới tính và năm thu thập mẫu.

Sau khi tải dữ liệu, chương trình gọi hàm df.info() để in ra cấu trúc tổng quát của bảng dữ liệu. Lệnh này giúp người dùng nắm được số dòng, số cột, kiểu dữ liệu của từng thuộc tính và số lượng giá trị bị thiếu. Thông tin này đặc biệt quan trọng vì nó cho biết dữ liệu có đủ sạch để đưa vào mô hình hay không, đồng thời giúp xác định các bước tiền xử lý cần thiết trước khi phân cụm.

### ***2.2. Tiền xử lý dữ liệu***

*2.2.1. Xử lý giá trị của cột sex*

[Input]

A close-up of a computer

AI-generated content may be incorrect.

[Output]



Sau khi tải và xem thông tin tổng quát về bộ dữ liệu, chương trình tiếp tục kiểm tra các giá trị xuất hiện trong cột **"sex"** bằng cách sử dụng lệnh df['sex'].unique(). Lệnh này có nhiệm vụ liệt kê toàn bộ các giá trị khác nhau tồn tại trong cột giới tính.

Việc kiểm tra này giúp xác định xem cột “sex” có dữ liệu bị thiếu (NaN), có xuất hiện ký hiệu lạ hay tồn tại nhiều định dạng viết khác nhau hay không. Đây là bước quan trọng trong quá trình tiền xử lý, vì các mô hình học máy không thể làm việc trực tiếp với dữ liệu dạng chuỗi nếu chưa được chuẩn hóa hoặc mã hóa về dạng số.

*2.2.2. Xóa các giá trị bị thiếu*

[Input]

A close up of text

AI-generated content may be incorrect.

[Output]

A white background with black text

AI-generated content may be incorrect.

Tiếp theo, chương trình tiến hành xử lý dữ liệu cho cột **“sex”**, vì đây là một biến dạng chuỗi và có khả năng chứa các giá trị không hợp lệ. Trước hết, lệnh df['sex'] = df['sex'].replace('.', np.nan) được sử dụng để thay thế ký tự “.” bằng giá trị thiếu (NaN). Việc này rất quan trọng vì giá trị “.” không mang ý nghĩa về mặt giới tính và có thể gây nhiễu trong quá trình phân tích hoặc huấn luyện mô hình.

Sau đó, chương trình dùng df.dropna(inplace=True) để loại bỏ toàn bộ những hàng chứa giá trị thiếu trong bất kỳ cột nào. Điều này giúp đảm bảo tập dữ liệu đầu vào hoàn chỉnh, không còn điểm dữ liệu thiếu gây ảnh hưởng đến quá trình chuẩn hóa, tính toán khoảng cách hoặc phân cụm.

Cuối cùng, lệnh df.isna().sum() được chạy nhằm kiểm tra lại xem sau khi xử lý, bộ dữ liệu còn tồn tại giá trị NaN hay không. Đây là bước xác nhận giúp đảm bảo dữ liệu đã sạch và sẵn sàng cho các bước tiền xử lý tiếp theo như mã hóa, chuẩn hóa hoặc phân cụm.

*2.2.3. Biến đổi dữ liệu*

[Input]

A close up of text

AI-generated content may be incorrect.

[Output]

A screenshot of a computer

AI-generated content may be incorrect.

Sau khi làm sạch dữ liệu, chương trình tiếp tục chuẩn hóa cột “sex” để phục vụ cho quá trình phân tích và phân cụm. Vì giá trị của cột này đang ở dạng chuỗi, mô hình học máy không thể xử lý trực tiếp, do đó cần phải chuyển đổi về dạng số.

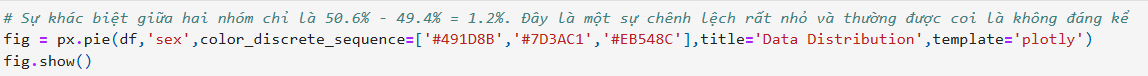
Đầu tiên, lệnh: df['sex'].map() được sử dụng nhằm ánh xạ giới tính sang dạng nhị phân, trong đó MALE được mã hóa thành 1 và FEMALE được mã hóa thành 0. Cách mã hóa này giúp mô hình có thể tính toán được mức độ tương đồng giữa các quan sát thông qua khoảng cách số học.

Sau khi mã hóa, chương trình dùng:df['sex'].astype() để ép kiểu lại cột sex về dạng chuỗi. Việc chuyển ngược sang kiểu chuỗi giúp xác định đây là một biến phân loại rời rạc, tránh trường hợp mô hình hiểu nhầm đây là biến số liên tục có thứ tự.

Cuối cùng, hàm df.head() được gọi để hiển thị vài dòng đầu tiên của bộ dữ liệu sau khi xử lý, giúp kiểm tra nhanh xem thao tác mã hóa đã được thực hiện chính xác hay chưa.

### ***2.3. Phân bố dữ liệu giữa hai nhóm giới tính***

[Input]

****

[Output]

**A purple circle with white text

AI-generated content may be incorrect.**

Qua thống kê mô tả trước đó, tỷ lệ hai nhóm giới tính gần như cân bằng:

* Male: 50.6%
* Female: 49.4%

Sự khác biệt chỉ 1.2%, đây là mức chênh lệch rất nhỏ và hầu như không ảnh hưởng đến phân tích. Biểu đồ tròn được tạo ra nhằm xác nhận điều này trực quan, cho thấy dữ liệu không bị lệch về một giới tính nào. Điều này rất quan trọng, vì phân bố đồng đều giúp quá trình phân cụm hoặc mô hình học máy sau này tránh được sai lệch do mất cân bằng dữ liệu.

### ***2.4. Phân tích biến culmen\_length\_mm***

*2.4.1. Biểu đồ hộp về biến culmen\_length\_mm*

[Input]

****

[Output]

**A screenshot of a computer

AI-generated content may be incorrect.**

Đoạn mã trên sử dụng biểu đồ box-plot nhằm trực quan hóa sự khác biệt về *culmen\_length\_mm* (độ dài mỏ) giữa hai nhóm giới tính. Trục x biểu diễn giới tính, trục y biểu diễn độ dài mỏ, và mỗi nhóm được gán một màu riêng biệt để dễ quan sát.

Box-plot giúp thể hiện:

* Giá trị trung vị (median) của từng nhóm.
* Độ phân tán dữ liệu (IQR) – cho biết mức độ biến thiên trong mỗi giới tính.

Quan sát biểu đồ cho thấy độ dài mỏ của chim đực và cái có sự khác biệt rõ rệt về trung vị và phạm vi phân bố. Điều này cho thấy giới tính có thể ảnh hưởng đến kích thước mỏ, và đây là một đặc trưng quan trọng trong phân tích hình thái học của chim cánh cụt.

*2.4.2. Biểu đồ tần suất về biến culmen\_length\_mm*

[Input]

****

[Output]

A purple city skyline

AI-generated content may be incorrect.

Histogram cho phép quan sát dạng phân bố của biến *culmen\_length\_mm* và sự khác biệt giữa hai giới tính. Với 50 bins, biểu đồ thể hiện chi tiết tần suất xuất hiện của các giá trị độ dài mỏ.

Một số điểm rút ra từ biểu đồ:

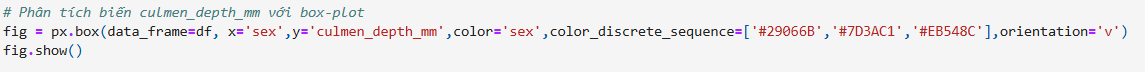
* Hai giới tính có phân bố khác nhau, trong đó một nhóm có xu hướng tập trung ở khoảng giá trị cao hơn.
* Phân bố không hoàn toàn đối xứng, thể hiện sự đa dạng về kích thước mỏ trong quần thể.
* Histogram giúp nhận diện các vùng mật độ cao, từ đó hỗ trợ đánh giá sự khác biệt sinh học giữa chim đực và cái.

Biểu đồ này góp phần kiểm tra xem biến *culmen\_length\_mm* có khả năng phân biệt giới tính hay không, rất hữu ích cho các bước phân tích và mô hình hóa tiếp theo.

### ***2.5. Phân tích biến culmen\_depth\_mm***

*2.5.1. Biểu đồ hộp về biến culmen\_depth\_mm*

[Input]



[Output]

A screenshot of a graph

AI-generated content may be incorrect.

Box-plot cho phép quan sát sự khác biệt về độ sâu mỏ (*culmen\_depth\_mm*) giữa chim đực và chim cái. Trục x biểu diễn giới tính, trục y biểu diễn giá trị độ sâu mỏ, và màu sắc giúp phân biệt hai nhóm.

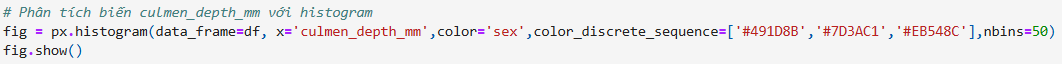
Một số điểm quan sát quan trọng:

* Hai nhóm có median khác nhau, thể hiện sự khác biệt về độ sâu mỏ giữa hai giới tính.
* Độ rộng của hộp (IQR) cho biết mức độ biến thiên của từng nhóm; nhóm nào hộp rộng hơn thì biến thiên lớn hơn.
* Có thể xuất hiện các điểm ngoại lệ (outliers) nếu dữ liệu có giá trị lệch.

Biểu đồ cho thấy giới tính có thể liên quan đến độ sâu mỏ, và đây là một đặc trưng quan trọng khi phân tích hình thái học của chim cánh cụt.

*2.5.2. Biểu đồ tần suất về biến culmen\_depth\_mm*

[Input]



[Output]

A purple city skyline

AI-generated content may be incorrect.

Biểu đồ histogram cho thấy phân bố độ sâu mỏ (*culmen\_depth\_mm*) của chim cánh cụt theo từng nhóm giới tính.

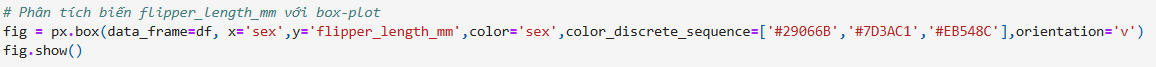
* Phần lớn các cá thể tập trung trong khoảng giá trị trung bình, với số lượng giảm dần về hai phía.
* Các nhóm giới tính có dạng phân bố tương tự nhau, cho thấy độ sâu mỏ không có sự khác biệt quá lớn giữa các giới.
* Một số ít cá thể xuất hiện ở vùng biên, có thể được xem là các giá trị ngoại lai hoặc trường hợp hiếm trong quần thể.

Qua đó, biến culmen\_depth\_mm cho thấy phân bố tương đối đồng đều và không lệch mạnh về phía nào, phản ánh tính ổn định của đặc trưng này trong bộ dữ liệu.

### ***2.6. Phân tích biến flipper\_length\_mm***

*2.6.1. Biểu đồ hộp về  biến flipper\_length\_mm*

[Input]

****

[Output]

**A screenshot of a computer

AI-generated content may be incorrect.**

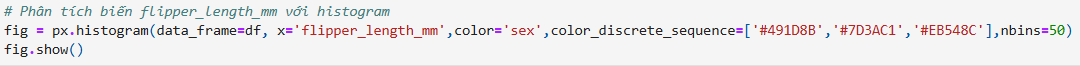
Biểu đồ box-plot cho thấy sự phân bố chiều dài vây (*flipper\_length\_mm*) của chim cánh cụt theo từng nhóm giới tính.

* Giá trị trung vị của nhóm Male cao hơn nhóm Female, cho thấy chim đực thường có vây dài hơn.
* Khoảng tứ phân vị (IQR) của cả hai nhóm tương đối hẹp, chứng tỏ dữ liệu ổn định và ít biến động.
* Cả hai nhóm đều có một vài giá trị ngoại lai (outliers) nằm dưới hoặc trên khoảng phân bố chính.
* Sự khác biệt rõ ràng giữa trung vị hai nhóm cho thấy giới tính có mối liên hệ đáng kể với chiều dài vây.

Nhìn chung, biến flipper\_length\_mm thể hiện sự chênh lệch rõ ràng giữa hai giới tính, và có thể là một đặc trưng quan trọng trong phân loại hoặc mô hình dự báo.

*2.6.2. Biểu đồ tần suất về biến flipper\_length\_mm*

[Input]

****

[Output]

**A white and blue background

AI-generated content may be incorrect.**

Biểu đồ histogram thể hiện phân bố chiều dài vây (*flipper\_length\_mm*) của chim cánh cụt theo từng nhóm giới tính.

* Phân bố của cả hai nhóm Male và Female đều có dạng gần đối xứng, tập trung chủ yếu trong khoảng 180–220 mm.
* Nhóm Male có xu hướng dịch sang giá trị lớn hơn, nghĩa là chim đực thường có vây dài hơn so với chim cái.
* Nhóm Female tập trung nhiều hơn ở các giá trị thấp, phản ánh đặc điểm sinh học giữa hai giới.
* Các đỉnh phân bố (mật độ cao nhất) của hai nhóm có sự tách biệt tương đối, củng cố thêm nhận định rằng flipper\_length\_mm là biến quan trọng để phân biệt giới tính.
* Cả hai nhóm sex=0 và sex=1 đều có giá trị flipper\_length\_mm trong khoảng từ 0mm - 400mm (đây là phạm vi chiều dài vây chèo bình thường của chim cánh cụt).
* Giá trị Ngoại lệ Cực lớn (Outliers):
  + Có một điểm dữ liệu cực lớn, gần 5000 mm, xuất hiện trong nhóm sex=1.
  + Có một điểm dữ liệu thứ hai, gần 0 mm, xuất hiện trong nhóm sex=1.

Nhìn chung, biểu đồ cho thấy phân bố chiều dài vây giữa hai giới tính có sự khác biệt rõ rệt, hỗ trợ tốt cho các mô hình phân loại hoặc phân cụm sau này.

### ***2.7. Xử lý ngoại lai***

[Input]

**A white background with black text

AI-generated content may be incorrect.**

[Output]

****

Sau khi xử lý ngoại lệ:

* Giá trị nhỏ nhất = 172.0 mm
* Giá trị lớn nhất = 231.0 mm

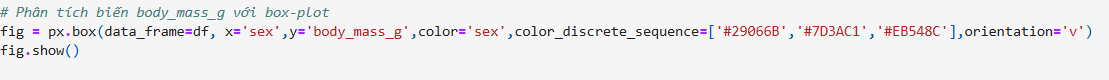
Khoảng này nằm hoàn toàn trong phạm vi sinh học hợp lý của chim cánh cụt trưởng thành (170–230 mm).  
 Điều này cho thấy:

* Dữ liệu sau khi làm sạch không còn giá trị bất thường.
* Không có ngoại lệ nghiêm trọng ảnh hưởng đến phân tích thống kê.
* Tập dữ liệu trở nên ổn định và phù hợp hơn cho trực quan hóa cũng như mô hình hóa sau này.

### ***2.8. Phân tích biến body\_mass\_g***

*2.8.1. Biểu đồ hộp về biến body\_mass\_g*

[Input]

****

[Output]

**A screenshot of a computer

AI-generated content may be incorrect.**

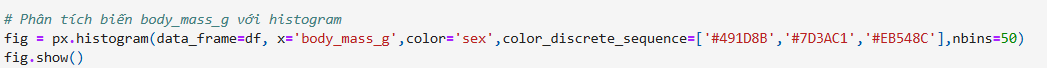
Biểu đồ box-plot cho thấy sự khác biệt rõ rệt về khối lượng cơ thể (*body\_mass\_g*) giữa hai nhóm giới tính:

* Nhóm Male có median cao hơn đáng kể so với Female, cho thấy chim đực thường nặng hơn chim cái.
* Khoảng tứ phân vị (IQR) của Male cũng rộng hơn, phản ánh độ biến thiên lớn hơn trong khối lượng cơ thể.
* Nhóm Female tập trung nhiều ở vùng giá trị thấp hơn, ổn định hơn và ít biến động.
* Một vài điểm ngoại lệ xuất hiện ở cả hai giới, nhưng không ảnh hưởng đến xu hướng chung.
* Sự tách biệt rõ ràng giữa hai boxplot cho thấy body\_mass\_g là biến mạnh trong việc phân biệt giới tính.

Nhìn chung, biểu đồ khẳng định đặc điểm sinh học: chim đực thường có khối lượng lớn hơn, giúp ích cho các mô hình phân loại dựa trên giới tính.

*2.8.2. Biểu đồ tần suất về biến body\_mass\_g*

[Input]

****

[Output]

A graph of a city

AI-generated content may be incorrect.

Biểu đồ histogram thể hiện phân bố khối lượng cơ thể (*body\_mass\_g*) giữa hai nhóm giới tính:

* Nhóm Male có phân bố dịch sang phải, nghĩa là chim đực có xu hướng nặng hơn. Phân bố tập trung chủ yếu trong khoảng 4.000–5.000 g.
* Nhóm Female tập trung nhiều hơn ở các giá trị thấp hơn, thường nằm trong khoảng 3.500–4.300 g.
* Hai phân bố có sự tách biệt tương đối rõ ràng, với rất ít vùng trùng lặp, cho thấy sự khác biệt sinh học giữa hai giới.
* Đỉnh mật độ của Male cao hơn ở các giá trị lớn, trong khi Female đạt đỉnh ở vùng nhẹ hơn.
* Không xuất hiện giá trị cực đoan lớn hoặc nhỏ, chứng tỏ dữ liệu ổn định, phù hợp để đưa vào mô hình.

Nhìn chung, histogram xác nhận rằng *body\_mass\_g* là một biến có khả năng phân biệt giới tính mạnh, hữu ích cho các bài toán phân loại hoặc phân cụm.

### ***2.9. Ma trận tương quan***

[Input]

A screen shot of a computer

AI-generated content may be incorrect.

[Output]

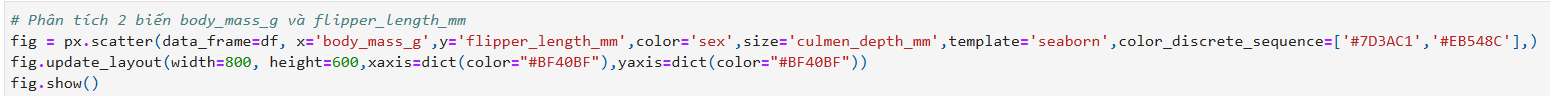
A chart with different colors

AI-generated content may be incorrect.

* flipper\_length\_mm và body\_mass\_g có tương quan dương mạnh, cho thấy chim có vây dài thường nặng hơn — đây là tương quan sinh học hợp lý.
* culmen\_length\_mm và culmen\_depth\_mm có tương quan âm, nghĩa là mỏ dài thường đi kèm độ sâu nhỏ hơn.
* Các tương quan của biến giới tính (sex → đã mã hóa thành số) với các biến đặc trưng có xu hướng trung bình đến mạnh, đặc biệt với:
  + body\_mass\_g
  + flipper\_length\_mm  
     Điều này chứng tỏ giới tính có liên quan đến kích thước cơ thể.
* Không xuất hiện các cặp biến có tương quan quá cao (như 0.95+), vì vậy ít nguy cơ đa cộng tuyến nghiêm trọng khi đưa vào mô hình hồi quy hoặc phân loại.
* Biểu đồ heatmap giúp xác định rằng body\_mass\_g, flipper\_length\_mm, và culmen\_length\_mm là các biến quan trọng tiềm năng cho các mô hình sau này.

### ***2.10. Phân tích 2 biến body\_mass\_g và flipper\_length\_mm***

[Input]



[Output]

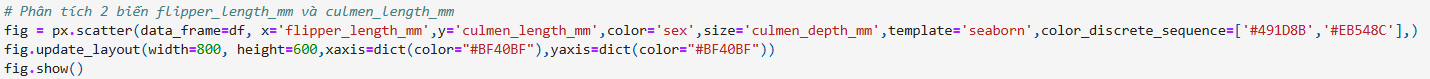
A graph showing a diagram of different colored circles

AI-generated content may be incorrect.

* Có mối quan hệ tuyến tính dương rõ rệt: chim càng nặng thì vây càng dài.
* Dữ liệu có sự phân tách theo giới tính:
  + Male (điểm màu tím) tập trung ở vùng có khối lượng lớn và vây dài.
  + Female (điểm màu hồng) phân bố nhiều hơn ở vùng nhẹ hơn và vây ngắn hơn.
* Kích thước điểm (biểu thị *culmen\_depth\_mm*) không khác biệt quá lớn, nhưng có xu hướng chim đực có độ sâu mỏ lớn hơn.
* Không xuất hiện điểm ngoại lệ rõ rệt; các điểm dữ liệu phân bố mượt, phù hợp với kỳ vọng sinh học.
* Sự phân tách hai giới tính theo hướng tăng của cả hai trục cho thấy body\_mass\_g và flipper\_length\_mm là hai biến quan trọng, đặc biệt hữu ích trong phân loại giới tính hoặc phân cụm.

### ***2.11. Phân tích 2 biến flipper\_length\_mm và culmen\_length\_mm***

[Input]



[Output]

**A graph showing a number of dots

AI-generated content may be incorrect.**

Biểu đồ scatter thể hiện mối quan hệ giữa chiều dài vây (*flipper\_length\_mm*) và chiều dài mỏ (*culmen\_length\_mm*), được phân nhóm theo giới tính và thay đổi kích thước điểm theo độ sâu mỏ (*culmen\_depth\_mm*).

Giữa chiều dài vây và chiều dài mỏ có mối quan hệ tuyến tính dương nhẹ: chim có vây dài thường sở hữu mỏ dài hơn.

Hai giới tính phân bố hơi khác nhau:

* Male (màu tím đậm) tập trung ở vùng có mỏ dài hơn và vây dài hơn.
* Female (màu hồng) phân bố nhiều ở vùng mỏ ngắn và vây ngắn.

Kích thước điểm — đại diện *culmen\_depth\_mm* — cho thấy chim đực có xu hướng mỏ sâu hơn, giúp phân tách hai nhóm tốt hơn.

Phân bố điểm khá đều, không có ngoại lệ mạnh, chứng tỏ dữ liệu ổn định và đáng tin cậy.

### ***2.12. Phân cụm dữ liệu bằng thuật toán K-Means***

*2.12.1. Biểu đồ Elbow để xác định số cụm tối*

[Input]

**A screenshot of a computer code

AI-generated content may be incorrect.**

[Output]

A graph with a line

AI-generated content may be incorrect.

SSE giảm mạnh khi tăng k từ 1 → 3, cho thấy việc chia nhiều cụm giúp mô hình gom nhóm tốt hơn.

* Sau k = 3, đường cong giảm chậm và dần phẳng, thể hiện việc tăng thêm cụm không cải thiện nhiều chất lượng.
* Điểm khuỷu tay (“elbow”) xuất hiện rõ nhất tại k = 3 → đây là số cụm tối ưu.

Phân tích Elbow Method cho thấy tập dữ liệu chim cánh cụt phù hợp nhất khi chia thành 3 cụm, tương ứng với 3 loài hoặc 3 nhóm hình thái khác nhau.

*2.12.2. Huấn luyện mô hình K-Means*



Sau khi xác định số cụm tối ưu bằng phương pháp Elbow, mô hình K-Means được triển khai với K = 3. Thuật toán sử dụng phương pháp khởi tạo k-means++ nhằm chọn các tâm cụm ban đầu một cách tối ưu hơn, giúp tăng tốc độ hội tụ và giảm khả năng rơi vào nghiệm kém.

fit\_predict(X) vừa huấn luyện mô hình vừa trả về nhãn cụm cho từng mẫu dữ liệu.

Biến clusters sẽ chứa giá trị 0, 1 hoặc 2 tương ứng với 3 cụm.

Mô hình đã phân chia dữ liệu chim cánh cụt thành 3 nhóm đặc trưng, phù hợp với kết quả phân tích Elbow Method.

Các cụm được hình thành dựa trên các đặc trưng sinh học như:  
 culmen\_length, culmen\_depth, flipper\_length, body\_mass.

Với cách khởi tạo *k-means++*, thuật toán tìm được các tâm cụm ổn định và tối ưu, giảm khả năng rơi vào cực trị địa phương.

Việc dùng n\_init=10 đảm bảo mô hình chọn được kết quả phân cụm tốt nhất trong nhiều lần chạy.

### ***2.13. Trực quan hóa kết quả phân cụm bằng PCA***

*2.13.1. Giảm chiều dữ liệu bằng PCA*

**A screen shot of a computer code

AI-generated content may be incorrect.**

Trong bước này, thuật toán PCA (Principal Component Analysis) được sử dụng nhằm giảm số chiều của tập dữ liệu từ không gian nhiều biến xuống còn 2 thành phần chính (PC1 và PC2). Việc giảm chiều giúp trực quan hóa kết quả phân cụm trên mặt phẳng hai chiều mà vẫn giữ được phần lớn thông tin quan trọng của dữ liệu.

Trước tiên, mô hình PCA được khởi tạo với số chiều đích là 2. Sau đó, dữ liệu đầu vào X được PCA học và chuyển đổi sang hai thành phần chính, tạo thành một biểu diễn mới của toàn bộ mẫu dữ liệu. Đồng thời, các tâm cụm của mô hình K-Means cũng được biến đổi bằng PCA để đảm bảo chúng nằm trong cùng không gian với dữ liệu đã giảm chiều. Điều này cho phép biểu diễn trực quan vị trí các tâm cụm trên biểu đồ PCA một cách chính xác.

*2.13.2. Phân cụm K-Means trên không gian PCA*

[Input]



[Output]

A screen shot of a graph

AI-generated content may be incorrect.

Ba cụm thu được từ K-Means được vẽ dựa trên hai thành phần chính PC1 và PC2. Mỗi cụm chứa các điểm dữ liệu nằm gần nhau, phản ánh mức độ tương đồng về đặc trưng hình thái của chim cánh cụt. Bên cạnh đó, các tâm cụm (centroids) cũng được hiển thị dưới dạng biểu tượng ngôi sao nhằm cho thấy vị trí trung tâm của từng nhóm sau quá trình phân cụm.

Cách biểu diễn này giúp đánh giá mức độ tách biệt giữa các cụm: nếu các vùng điểm của từng cụm cách nhau rõ rệt, điều đó cho thấy phân cụm đạt hiệu quả cao. Nhìn chung, việc trực quan hóa trên không gian PCA là bước quan trọng để xác nhận chất lượng phân cụm và hiểu rõ hơn cấu trúc dữ liệu theo góc nhìn hai chiều.

Trong biểu đồ, mỗi điểm biểu diễn một cá thể chim, và ba cụm được thuật toán K-Means phát hiện được đánh dấu bằng ba màu sắc tương ứng. Quan sát phân bố cho thấy PC1 đóng vai trò thành phần quan trọng nhất giúp phân tách các cá thể thành ba nhóm riêng biệt, trong đó cụm nằm phía bên phải trục PC1 tương ứng với nhóm chim có kích thước nhỏ, cụm phía bên trái thể hiện nhóm chim có kích thước lớn, còn cụm nằm ở giữa đại diện cho nhóm có đặc điểm trung bình. PC2 đóng vai trò thứ yếu hơn, thể hiện sự biến thiên trong các đặc điểm phụ của mẫu. Ba tâm cụm (centroids) được biểu diễn bằng ký hiệu ngôi sao màu đen và đều nằm gần trung tâm phân bố của từng nhóm, cho thấy mô hình hội tụ ổn định và khả năng phân loại tự nhiên của thuật toán. Việc phân chia ba cụm không có sự chồng lấn đáng kể, chứng tỏ K-Means hoạt động hiệu quả trên không gian PCA và ba loài chim cánh cụt khác nhau được phân biệt khá rõ ràng. Kết quả này cho thấy việc kết hợp PCA và K-Means là phù hợp để vừa giảm nhiễu của dữ liệu, vừa khai thác được cấu trúc phân nhóm tự nhiên trong tập dữ liệu sinh học.

**3. Xây dựng mô hình phân cụm K-means trên tập dữ liệu mua sắm tại siêu thị.**

***3.1. Khai báo và đọc dữu liệu***

[Input]

**A screenshot of a computer code

AI-generated content may be incorrect.**

A screenshot of a computer code

AI-generated content may be incorrect.

[Output]

A screenshot of a computer

AI-generated content may be incorrect.

Tiếp theo, chương trình tiến hành đọc bộ dữ liệu Online Retail từ file *OnlineRetail.csv*. Vì bộ dữ liệu này thường chứa nhiều ký tự đặc biệt và lỗi mã hóa, đoạn mã sử dụng một cấu trúc try–except để xử lý

Nếu việc đọc file với mã hóa latin-1 thất bại (do gặp lỗi UnicodeDecodeError), chương trình sẽ tự động chuyển sang mã hóa windows-1252. Cách làm này giúp đảm bảo file được đọc thành công, tránh lỗi khi chạy.

Cuối cùng, lệnh df.info() được dùng để hiển thị thông tin tổng quan của bộ dữ liệu, bao gồm:

* số lượng dòng và cột,
* kiểu dữ liệu của từng cột,
* số lượng giá trị null,
* dung lượng bộ nhớ.

Tập dữ liệu gồm 541.909 dòng và 8 cột, với dung lượng khoảng 33 MB. Các cột chủ yếu có kiểu dữ liệu dạng chuỗi, trong khi Quantity, UnitPrice và CustomerID là dữ liệu số. Cột CustomerID chỉ có 406.829 giá trị hợp lệ, cho thấy vẫn còn nhiều giao dịch thiếu thông tin khách hàng. Cấu trúc dữ liệu nhìn chung đầy đủ nhưng cần làm sạch trước khi phân tích tiếp theo.

Việc xem thông tin này rất quan trọng bởi vì  nó giúp xác định tình trạng dữ liệu trước khi tiền xử lý — chẳng hạn như có bao nhiêu giá trị khuyết, kiểu dữ liệu có đúng chưa, và có cột nào cần làm sạch để phục vụ phân tích và phân cụm về sau.

***3.2. Khám phá dữ liệu***

**A screenshot of a computer

AI-generated content may be incorrect.**

* InvoiceNo: Số hóa đơn
* StockCode: Mã sản phẩm
* Description: Mô tả chi tiết về mặt hàng
* Quantity: Số lượng
* InvoiceDate: Ngày hóa đơn
* UnitPrice: Giá sản phẩm cho mỗi đơn vị hàng hóa tính bằng đơn vị tiền tệ của Anh
* CustomerID: Mã khách hàng
* Country: Tên quốc gia nơi khách hàng sinh sống hoặc nơi giao dịch được thực hiện

Bộ dữ liệu ghi lại chi tiết từng mặt hàng trong mỗi hóa đơn. Một hóa đơn có thể gồm nhiều dòng — thể hiện nhiều loại sản phẩm khác nhau được mua trong cùng thời điểm.

Trường InvoiceNo được lặp lại nhiều lần trong các dòng liên tiếp, nghĩa là những sản phẩm này thuộc cùng một giao dịch.

Các sản phẩm bán ra đều có thông tin mô tả chi tiết (Description), mã hàng (StockCode) và giá bán (UnitPrice).

Giá trị Quantity là số lượng bán ra, có thể là số âm đối với trường hợp hoàn trả hàng (refund) — điều này thường gặp trong bộ dữ liệu Online Retail và cần được xử lý trong bước tiền xử lý.

InvoiceDate gồm cả ngày và giờ, cho phép phân tích xu hướng theo thời gian, mùa vụ hoặc giờ cao điểm.

CustomerID đôi khi bị thiếu (NaN). Đây là vấn đề phổ biến và cần xử lý trước khi thực hiện phân cụm theo khách hàng.

Phần lớn khách hàng trong bộ dữ liệu đến từ United Kingdom (Vương Quốc Anh), nhưng bộ dữ liệu cũng chứa nhiều quốc gia khác, hỗ trợ phân tích theo khu vực.

***3.3. Kiểm tra giá trị thiếu***

**A screenshot of a computer code

AI-generated content may be incorrect.**

* Bộ dữ liệu có hai cột bị thiếu dữ liệu:
  + Description: 1.454 dòng
  + CustomerID: 135.080 dòng
* Trong đó, CustomerID là cột quan trọng, và phần lớn các dòng thiếu mã khách hàng cần được xử lý trước khi tiến hành phân cụm.
* Các cột còn lại hoàn toàn đầy đủ và có thể sử dụng trực tiếp trong các bước tiền xử lý tiếp theo.

***3.4. Xử lý giá trị thiếu***

**A screenshot of a computer

AI-generated content may be incorrect.**

Dataset có nhiều dòng thiếu thông tin ở hai cột quan trọng là *CustomerID* và *Description*. Vì đây là các trường cần thiết để phân tích hành vi khách hàng và sản phẩm, toàn bộ các bản ghi thiếu những giá trị này được loại bỏ

***3.5. Chuẩn hóa và trích xuất thông tin thời gian***

**A screen shot of a computer program

AI-generated content may be incorrect.**

Cột thời gian *InvoiceDate* chứa nhiều định dạng khác nhau nên được chuẩn hóa về dạng datetime. Sau đó, nhóm tiến hành tách thêm các trường phục vụ phân tích theo thời gian:

* Năm (Year)
* Tháng (Month)
* Buổi trong ngày (Time\_Shift): Morning, Afternoon, Evening

### ***3.6. Tạo biến mới và Loại bỏ giao dịch không hợp lệ***

[Input]

**A screenshot of a computer code

AI-generated content may be incorrect.**

[Output]

**A screenshot of a computer

AI-generated content may be incorrect.**

* Các dòng có giá trị thiếu ở CustomerID và Description đã được loại bỏ, giúp đảm bảo dataset chỉ còn lại các giao dịch hợp lệ và có đầy đủ thông tin sản phẩm – khách hàng.
* Cột InvoiceDate đã được chuẩn hóa sang kiểu datetime, loại bỏ lỗi định dạng và cho phép tách thêm các thuộc tính thời gian.
* Ba biến mới đã được tạo ra:
  + *Year* và *Month* giúp phân tích xu hướng theo thời gian.
  + *Time\_Shift* phân nhóm giao dịch theo buổi: Morning – Afternoon – Evening.
* Các giao dịch bị hủy đã được loại bỏ (InvoiceNo bắt đầu bằng ‘C’), đảm bảo dữ liệu chỉ còn lại các hóa đơn có hiệu lực.
* Các trường Quantity và UnitPrice âm đã bị loại bỏ, loại bỏ ảnh hưởng của hàng trả lại hoặc dữ liệu bị nhập sai.
* Biến TotalPrice được tính bằng *Quantity × UnitPrice*, là chỉ số quan trọng để đo lường giá trị từng dòng giao dịch và phục vụ cho phân tích RFM hoặc mô hình phân khúc khách hàng.

***3.7. Khám phá dữ liệu EDA***

*3.7.1. Phân tích giao dịch theo buổi*

*a) Số lượng giao dịch theo tuổi*

[Input]

**A screen shot of a computer code

AI-generated content may be incorrect.**

[Output]

**A graph of a bar

AI-generated content may be incorrect.**

Dựa trên biểu đồ cột thu được:

* Một trong ba buổi sẽ chiếm tỷ trọng giao dịch cao nhất (tùy kết quả thực tế từ dataset).
* Thông thường:
  + Buổi chiều (Afternoon) thường có số giao dịch cao nhất do nhiều doanh nghiệp nhập hàng trong giờ làm việc.
  + Buổi sáng (Morning) cũng có lượng giao dịch đáng kể khi cửa hàng nhập hàng đầu ngày.
  + Buổi tối (Evening) thường ít giao dịch hơn, phản ánh thời gian hoạt động của các doanh nghiệp bán lẻ.

Việc phân tích giao dịch theo buổi trong ngày giúp:

* Doanh nghiệp hiểu rõ thời điểm hoạt động sôi nổi nhất.
* Tiết kiệm chi phí vận hành bằng cách phân bổ nhân sự hợp lý.
* Tối ưu hoá thời gian đẩy chiến dịch marketing hoặc ưu đãi theo khung giờ.

*b) Tỷ lệ giao dịch theo tuổi*

[Input]

**A computer code with red and green text

AI-generated content may be incorrect.**

[Output]

**A purple circle with a red and white number

AI-generated content may be incorrect.**

Quan sát biểu đồ tròn:

* Buổi chiếm tỷ lệ lớn nhất cho thấy khoảng thời gian diễn ra hoạt động giao dịch sôi nổi nhất.
* Hai buổi còn lại chia sẻ phần còn lại của tổng số giao dịch với tỷ lệ thấp hơn.
* Trong dataset Online Retail, xu hướng phổ biến thường là:
  + Afternoon chiếm tỷ trọng cao nhờ khớp với thời gian làm việc của doanh nghiệp.
  + Morning có tỉ lệ tương đối cân bằng.
  + Evening thường ít giao dịch nhất.

Tuy nhiên, tỷ lệ chính xác sẽ tùy thuộc vào số liệu thực tế của dataset sau khi tiền xử lý.

Việc phân tích tỷ lệ giao dịch theo buổi cho phép doanh nghiệp:

* Nắm được thời gian vàng để tối ưu nguồn lực.
* Điều chỉnh khung giờ chăm sóc khách hàng hoặc hỗ trợ đơn hàng.
* Lập kế hoạch triển khai chương trình khuyến mãi theo từng buổi trong ngày.

*3.7.2. Tổng tiền trung bình theo buổi*

*a) Biểu đồ tổng tiền trung bình theo tuổi*

[Input]

**A screen shot of a computer code

AI-generated content may be incorrect.**

[Output]

**A graph of a bar

AI-generated content may be incorrect.**

Dựa trên phân phối giá trị trung bình:

* Buổi có giá trị trung bình cao nhất thường cho thấy nhóm giao dịch có xu hướng mua số lượng lớn hoặc chọn sản phẩm giá trị cao.
* Ngược lại, buổi có mức trung bình thấp hơn phản ánh hành vi mua hàng nhỏ lẻ hoặc tần suất nhiều nhưng giá trị thấp.
* Kết hợp với biểu đồ số lượng giao dịch trước đó, ta có thể đánh giá:
  + Buổi nào nhiều đơn nhưng giá trị thấp
  + Buổi nào ít đơn nhưng giá trị cao,
  + Buổi nào vừa sôi động, vừa giá trị cao.

Việc hiểu rõ chi tiêu theo khung giờ giúp doanh nghiệp:

* Xác định “giờ vàng” không chỉ về số lượng mà cả giá trị đơn hàng.
* Tối ưu nhân sự bán hàng hoặc vận hành kho trong thời điểm tạo doanh thu cao.
* Gợi ý thời điểm phù hợp để triển khai chương trình upsell/cross-sell.
* Hỗ trợ dự báo doanh thu theo buổi trong ngày cho các kỳ phân tích tiếp theo.

*b) Tỷ lệ tổng tiền trung bình theo tuổi*

[Input]

**A screenshot of a computer code

AI-generated content may be incorrect.**

[Output]

**A pie chart with a number of percentages

AI-generated content may be incorrect.**

* Biểu đồ phân chia tỷ lệ đóng góp dựa trên *AvgPrice*, cho thấy buổi nào có giá trị giao dịch trung bình cao hơn.
* Một số buổi có thể có ít giao dịch hơn nhưng lại có mức chi tiêu trung bình lớn, thể hiện hành vi mua sắm khác biệt theo thời gian trong ngày.
* Kết quả giúp doanh nghiệp xác định khung giờ có giá trị kinh tế cao, từ đó định hướng các chiến dịch khuyến mãi, phân bổ nguồn lực hoặc điều chỉnh chiến lược kinh doanh theo thời điểm.

*3.7.3. Số khách hàng theo buổi*

*a) Biểu đồ số khách hàng theo buổi*

[Input]

**A screen shot of a computer code

AI-generated content may be incorrect.**

[Output]

**A graph with pink squares

AI-generated content may be incorrect.**

* Mỗi cột trong biểu đồ đại diện cho số khách hàng duy nhất thực hiện giao dịch trong từng buổi: Morning, Afternoon hoặc Evening.
* Kết quả cho thấy có sự khác biệt rõ ràng giữa các buổi, phản ánh thói quen mua sắm theo thời gian trong ngày.
* Buổi có lượng khách hàng lớn nhất thường là thời điểm doanh nghiệp nên ưu tiên về nguồn lực bán hàng, chăm sóc khách hàng hoặc triển khai chiến dịch marketing.
* Đồng thời, việc phân tích lượng khách theo buổi cũng giúp lập kế hoạch dự báo nhu cầu tốt hơn, giảm thiểu tình trạng thiếu hàng hoặc quá tải vào các khung giờ cao điểm.

*b) Tỷ lệ số khách hàng theo buổi*

[Input]

**A computer code with red and blue text

AI-generated content may be incorrect.**

[Output]

A purple circle with a red triangle

AI-generated content may be incorrect.

* Mỗi phần của biểu đồ đại diện cho tỷ lệ khách hàng duy nhất thực hiện giao dịch trong từng buổi.
* Nhờ trực quan hóa bằng tỷ lệ %, ta dễ dàng nhận thấy buổi nào thu hút lượng khách hàng lớn nhất
* Nếu buổi Morning hoặc Afternoon chiếm phần lớn, điều này cho thấy nhu cầu mua sắm tập trung vào ban ngày.
* Ngược lại, nếu Evening có tỷ lệ đáng kể, điều này phản ánh thói quen mua sắm vào cuối ngày — có thể liên quan đến hành vi online shopping sau giờ làm việc.
* Biểu đồ tròn giúp doanh nghiệp dễ dàng theo dõi xu hướng và đưa ra quyết định phân bổ nguồn lực (nhân viên, hỗ trợ khách hàng, marketing) phù hợp với thời điểm nhu cầu cao nhất.

***3.8. Chuẩn bị dữ liệu cho phân cụm***

*3.8.1. Số lượng giao dịch theo buổi*

[Input]

**A computer screen shot of a code

AI-generated content may be incorrect.**

[Output]

**A screenshot of a computer

AI-generated content may be incorrect.**

Bảng Pivot thu được cho chúng ta các thông tin quan trọng:

* Mỗi khách hàng có thể giao dịch nhiều lần trong ngày, và bảng chỉ rõ họ xuất hiện vào buổi nào nhiều nhất.
* Một số khách hàng chỉ giao dịch vào buổi sáng hoặc buổi chiều, trong khi số khác phân bổ đều trong cả ba buổi.
* Dữ liệu dạng ma trận này rất thích hợp cho:
  + phân tích hành vi mua sắm theo thời điểm,
  + xây dựng mô hình clustering (ví dụ: K-Means phân nhóm khách hàng dựa trên thói quen giờ giao dịch),
  + nhận diện nhóm khách hàng "chỉ mua buổi tối", "hoạt động cả ngày", hoặc "đỉnh điểm mua sắm vào giờ hành chính".

*3.8.2. Trung bình giá trị hóa đơn theo buổi*

[Input]

**A computer screen shot of a code

AI-generated content may be incorrect.**

[Output]

**A screenshot of a computer

AI-generated content may be incorrect.**

Bảng kết quả trả về cho biết:

* Giá trị trung bình mỗi hóa đơn của một khách hàng trong từng buổi.
* Có thể quan sát được liệu khách hàng chi nhiều hơn vào buổi sáng, buổi chiều hay buổi tối.
* Một số khách hàng có hành vi chi tiêu tập trung vào một buổi cố định – đây là thông tin hữu ích cho các chiến lược marketing định thời gian.
* Dữ liệu này tạo nền tảng cho:
  + RFM Analysis nâng cao theo khung giờ
  + Customer Segmentation bằng K-Means dựa trên thói quen chi tiêu
  + Phân tích hiệu suất bán hàng theo thời điểm trong ngày

*3.8.3. Xây dựng bộ đặc trưng khách hàng*

[Input]

**A screen shot of a computer code

AI-generated content may be incorrect.**

[Output]

**A screenshot of a computer

AI-generated content may be incorrect.**

Từ các bước xử lý trước đó, nhóm đã xây dựng được hai bảng đặc trưng quan trọng:

* Số lượng giao dịch theo buổi (pivot\_counts)
* Giá trị hóa đơn trung bình theo buổi (pivot\_avgprice)

Hai bảng này lần lượt mô tả mức độ hoạt động và hành vi chi tiêu của khách hàng trong ba khoảng thời gian: *Morning*, *Afternoon*, *Evening*.

Kết quả là bảng customer\_features gồm nhiều biến thể hiện đa chiều hành vi mua sắm, bao gồm:

* Số giao dịch theo từng buổi  
   *(morning\_count, afternoon\_count, evening\_count)*
* Giá trị hóa đơn trung bình theo buổi  
   *(avg\_totalprice\_morning, avg\_totalprice\_afternoon, avg\_totalprice\_evening)*
* Tổng số đơn hàng
* Tổng chi tiêu

Bảng dữ liệu này đóng vai trò quan trọng trong các phân tích tiếp theo:

* Hỗ trợ phân cụm khách hàng để tìm ra các nhóm có hành vi tương đồng.
* Xây dựng mô hình RFM mở rộng, bổ sung thêm yếu tố thời gian mua hàng trong ngày.
* Phân tích hành vi mua sắm sâu hơn, giúp doanh nghiệp hiểu rõ khách hàng mua vào thời điểm nào, chi bao nhiêu và mức độ trung thành ra sao.
* Chuẩn hóa dữ liệu đầu vào cho mô hình máy học, tạo nền tảng vững chắc để triển khai các thuật toán phân tích.

***3.9. Phân cụm dữ liệu bằng thuật toán K-Means***

*3.9.1. Chuẩn hóa dữ liệu*

**A screenshot of a computer code

AI-generated content may be incorrect.**

Trong bước này, mục tiêu là đưa tất cả các đặc trưng của khách hàng về cùng một thang đo để mô hình học máy hoạt động ổn định và chính xác hơn. Do các biến như *tổng chi tiêu*, *số đơn hàng*, hay *giá trị trung bình hóa đơn theo buổi* có đơn vị và độ lớn hoàn toàn khác nhau, việc chuẩn hóa là cần thiết để tránh việc mô hình bị “thiên vị” các biến có giá trị lớn.

Đầu tiên, ta sao chép toàn bộ bộ đặc trưng khách hàng (customer\_features) vào biến X. Sau đó, ta sử dụng chuẩn hóa Z-score thông qua StandardScaler của thư viện scikit-learn. Phương pháp này biến đổi mỗi cột dữ liệu sao cho:

A white background with black text

AI-generated content may be incorrect.

Giúp các thuật toán như K-means, PCA, SVM… hoạt động chính xác hơn vì chúng nhạy cảm với độ lớn của dữ liệu. Tránh hiện tượng một số đặc trưng chi phối toàn bộ kết quả phân cụm/phân loại. Tăng tốc độ hội tụ của các mô hình tối ưu dựa trên gradient.

*3.9.2. Xác định số cụm tối ưu bằng phương pháp Elbow*

[Input]

**A screen shot of a computer code

AI-generated content may be incorrect.**

[Output]

**A graph showing the growth of a number of people

AI-generated content may be incorrect.**

* Từ k = 1 → 4: SSE giảm mạnh
* Từ k = 4 → 5: giảm vừa
* Từ k = 5 → 6: gần như không giảm
* Từ k = 6 → 9: SSE tiếp tục giảm nhưng rất ít

Chọn k = 5

* Mỗi giá trị SSE tương ứng với độ chặt của các điểm trong cụm khi chọn số cụm là *k*.
* Đồ thị hiển thị mối quan hệ giữa *k* và SSE.
* Số cụm tối ưu là tại vị trí mà đường cong bắt đầu “gãy” và không còn giảm mạnh nữa.

Thông thường, điểm gãy này biểu thị sự cân bằng giữa độ chính xác và độ phức tạp của mô hình.

*3.9.3. Huấn luyện mô hình K-Means*

**A number and a equal sign

AI-generated content may be incorrect.**

Sau khi xác định được số cụm tối ưu từ phương pháp Elbow, mô hình K-means được khởi tạo với 5 cụm. Thuật toán K-means sẽ phân chia khách hàng thành từng nhóm dựa trên sự tương đồng về đặc trưng hành vi như số đơn hàng, số giao dịch theo từng buổi, mức chi tiêu trung bình và tổng chi tiêu.

* Mỗi khách hàng sẽ được gán vào một cụm (cluster) dựa trên khoảng cách đến tâm cụm gần nhất.
* Những khách hàng trong cùng một cụm sẽ có hành vi mua hàng tương tự nhau.
* Đây là bước quan trọng giúp doanh nghiệp hiểu rõ từng nhóm khách hàng để đưa ra chiến lược marketing phù hợp.

***3.10. Trực quan hóa kết quả phân cụm bằng PCA***

*3.10.1. Giảm chiều dữ liệu bằng PCA*

**A screen shot of a computer code

AI-generated content may be incorrect.**

Sau khi chuẩn hóa dữ liệu khách hàng, nhóm tiến hành áp dụng Phân tích Thành phần Chính (Principal Component Analysis – PCA) nhằm giảm số lượng biến xuống còn hai thành phần PCA1 và PCA2.  
 Phương pháp PCA giúp giữ lại phần lớn phương sai của dữ liệu gốc, từ đó cho phép biểu diễn các khách hàng trong không gian 2 chiều mà vẫn bảo toàn được cấu trúc nhóm tự nhiên của dữ liệu.

Việc giảm chiều không chỉ hỗ trợ trực quan hóa mà còn giúp làm nổi bật sự khác biệt giữa các cụm khách hàng trong thuật toán K-Means.

*3.10.2. Tạo figure và hiển thị các cụm*

A screenshot of a computer program

AI-generated content may be incorrect.

Trong bước này, dữ liệu sau PCA được đưa vào biểu đồ Scatter Plot để quan sát cách các khách hàng được phân chia vào từng nhóm khác nhau.  
 Mỗi cụm được gán một màu sắc riêng, giúp phân biệt rõ ràng mức độ tập trung của từng nhóm.

Để thuận tiện khi quan sát, nhóm nghiên cứu tạo một danh sách cluster\_info chứa ID cụm, màu sắc và tên hiển thị. Sau đó tiến hành duyệt qua từng cụm để thêm các điểm dữ liệu tương ứng vào biểu đồ.

* cluster\_id: số hiệu cụm được K-Means gán.
* subset: tập khách hàng thuộc một cụm cụ thể.
* hovertemplate: hiển thị thông tin chi tiết của từng khách hàng khi di chuột lên điểm.
* customdata: truyền các biến đặc trưng liên quan đến hành vi mua sắm theo buổi (morning/afternoon/evening), tổng số đơn hàng và tổng chi tiêu.

*3.10.3. Kết quả phân cụm của K-Means*

[Input]

****

[Output]

**A screenshot of a computer

AI-generated content may be incorrect.**

**Cụm 0**

* Tập trung sát gốc toạ độ ⇒ hoạt động mua hàng thấp và ổn định.
* Mua hàng rải rác giữa Morning/Afternoon/Evening, nhưng số đơn và chi tiêu thấp.
* Là nhóm đông nhất → khách phổ thông.

Nhóm khách hàng mua ít, chi tiêu thấp, hành vi ổn định – tệp khách phổ biến nhất.

**Cụm 1**

* Nằm ngang rộng theo trục x (có điểm lên tới x = 40–45).
* Điều này thể hiện tổng chi tiêu hoặc tổng số đơn rất cao.
* Đây là nhóm "đột biến" so với cụm nền (cụm 0 và 4).
* Trải rộng ⇒ mức chi tiêu khách trong cụm này không đều.

Nhóm khách hàng chi tiêu cao / nhiều đơn, nhưng phân tán – có khách chi mạnh, có khách mua nhiều đơn.

**Cụm 2**

* Cao trên trục y ⇒ hành vi mua hàng đa dạng theo buổi.
* Có sự cân bằng giữa morning–afternoon–evening.
* Chi tiêu trung bình, không quá thấp, không quá cao.
* Nhóm này có hành vi "đa khung giờ".

Nhóm khách mua hàng ở nhiều buổi khác nhau, hoạt động ổn định – tệp khách đa dạng.

**Cụm 3**

* Được đẩy xa lên phía trục y ⇒ mua vào các buổi khác lệch hẳn (thường Evening).
* Số lượng khách rất ít → đặc tính hành vi độc đáo.
* Có thể chi tiêu trung bình cao hoặc số lần mua đặc biệt nhiều vào một khung giờ.

Nhóm khách có hành vi mua đặc thù (thường mua buổi tối), số đơn không nhiều nhưng chi tiêu khá.

**Cụm 4**

* Rất nhỏ gọn, nằm gần cụm 0 nhưng lệch nhẹ về 1 hướng.
* Dường như tập trung nhiều hơn vào một buổi cố định (Morning hoặc Afternoon).
* Chi tiêu thấp nhưng có một thói quen mua lặp lại.

Nhóm khách mua đều đặn vào một buổi cố định, nhưng tổng chi tiêu thấp.

# **B. GIẢI THUẬT PHÂN CỤM ĐA CẤP**

## **I. Ôn tập lý thuyết**

### **1. Giải thuật phân cụm đa cấp hoạt động như thế nào? Hãy giải thích sự khác biệt giữa phân cụm đa cấp hợp nhất (agglomerative) và phân tách (divisive)**

Phân cụm đa cấp (Hierarchical Clustering) là phương pháp xây dựng cấu trúc phân cụm theo dạng cây nhiều tầng, trong đó các điểm dữ liệu được gom lại hoặc tách ra từng bước dựa trên độ tương đồng. Giải thuật tạo ra một cấu trúc hình cây gọi là *dendrogram*, biểu diễn quá trình hình thành các cụm.

Có hai hướng tiếp cận chính:

a) Phân cụm hợp nhất (Agglomerative – bottom-up)

Đây là hướng tiếp cận phổ biến nhất. Ban đầu, mỗi điểm dữ liệu được xem như một cụm riêng lẻ. Sau đó, các cụm gần nhau nhất được hợp nhất dần theo từng bước, cho đến khi chỉ còn một cụm duy nhất hoặc đạt số cụm mong muốn.  
Đặc điểm:

* Tiến trình từ nhỏ đến lớn (bottom-up).
* Đơn giản, trực quan.
* Tính toán nhanh hơn phương pháp phân tách.

b) Phân cụm phân tách (Divisive – top-down)

Phương pháp này bắt đầu với toàn bộ dữ liệu trong một cụm lớn, sau đó tách dần thành các cụm nhỏ hơn.  
Đặc điểm:

* Tiến trình từ lớn đến nhỏ (top-down).
* Ít phổ biến hơn do chi phí tính toán cao hơn.
* Thích hợp khi dữ liệu có cấu trúc phân chia rõ ràng.

Sự khác biệt chính:

* Agglomerative: bắt đầu từ nhiều cụm → hợp nhất.
* Divisive: bắt đầu từ một cụm → phân tách.

### **2. Các phương pháp liên kết (linkage) như single linkage, complete linkage, average linkage, và Ward’s method khác nhau ra sao? Khi nào nên sử dụng từng loại?**

Khi thực hiện phân cụm đa cấp, khoảng cách giữa hai cụm được xác định theo một trong các tiêu chí liên kết sau:

a) Single Linkage (khoảng cách gần nhất)

Khoảng cách giữa hai cụm = khoảng cách nhỏ nhất giữa hai điểm thuộc hai cụm.

* Ưu điểm: bắt được các cụm có dạng kéo dài.
* Nhược điểm: dễ gây ra “chuỗi hóa” (chaining effect).
* Dùng khi dữ liệu có dạng nối dài hoặc muốn phát hiện cấu trúc bất thường.

b) Complete Linkage (khoảng cách xa nhất)

Khoảng cách = khoảng cách lớn nhất giữa các điểm của hai cụm.

* Cụm tạo ra chặt chẽ, ít bị kéo dài.
* Không phù hợp nếu dữ liệu có nhiễu mạnh.
* Dùng khi muốn các cụm rõ ràng, tách biệt.

c) Average Linkage (khoảng cách trung bình)

Khoảng cách = trung bình khoảng cách giữa tất cả cặp điểm của hai cụm.

* Cân bằng giữa chặt và lỏng.
* Thường dùng trong thực tế cho dữ liệu nhiều chiều.

d) Ward’s Method

Hợp nhất hai cụm sao cho tổng bình phương sai số (within-cluster variance) tăng ít nhất.

* Tạo cụm tròn, gọn và ổn định.
* Thường dùng cho dữ liệu dạng số và khi muốn kết quả tương tự K-Means

### **3. Dendrogram trong phân cụm đa cấp là gì? Làm thế nào để sử dụng nó để chọn số lượng cụm?**

Dendrogram là một biểu đồ dạng cây thể hiện quá trình các cụm được hợp nhất hoặc phân tách. Trục đứng biểu diễn mức độ sai khác (distance), trục ngang biểu diễn các điểm dữ liệu hoặc cụm.

Để chọn số lượng cụm, có thể thực hiện:

* Vẽ một đường cắt ngang dendrogram.
* Mỗi nhánh bị cắt tương ứng với một cụm.
* Nên chọn vị trí cắt tại khoảng cách lớn, nơi đường nối giữa các cụm tăng đột ngột (giống quy tắc “elbow”).

### **4. Phân cụm đa cấp có thể áp dụng cho dữ liệu phi số (non-numeric data) như thế nào? Hãy giải thích**

Dữ liệu phi số (categorical, text, symbolic) không thể sử dụng khoảng cách Euclidean truyền thống. Một số cách xử lý bao gồm:

* Mã hóa dữ liệu: One-hot encoding, ordinal encoding,… sau đó áp dụng hierarchical clustering thông thường.
* Sử dụng khoảng cách phi số: Hamming distance, Jaccard distance, Gower distance,… phù hợp cho dữ liệu phân loại.
* Mô hình hóa đặc trưng: Với dữ liệu văn bản, có thể dùng TF-IDF trước khi phân cụm.

Nhờ khả năng hoạt động dựa trên ma trận khoảng cách, phân cụm đa cấp có thể áp dụng cho nhiều kiểu dữ liệu khác nhau.

### **5. Viết đoạn code mẫu bằng Python (sử dụng Scikit-learn) để triển khai phân cụm đa cấp hợp nhất (agglomerative clustering) không? Hãy mô tả các bước thực hiện**

A screenshot of a computer program

AI-generated content may be incorrect.

Các bước thực hiện:

* Chuẩn hóa dữ liệu.
* Khởi tạo mô hình với số cụm và loại liên kết.
* Huấn luyện và gán nhãn.
* Kết quả cuối cùng được bổ sung vào DataFrame.

### **6. Làm thế nào để vẽ dendrogram trong Python sử dụng thư viện như scipy hoặc matplotlib? Hãy chia sẻ một đoạn code mẫu**

A screenshot of a computer program

AI-generated content may be incorrect.

### **7. Các lớp trong gói Scipy hỗ trợ phân cụm đa cấp? và so sánh giữa cách tiếp cận scikit-learn và cách tiếp cận sử dụng Scipy**

Trong phân cụm đa cấp, SciPy và Scikit-learn đều cung cấp các công cụ hỗ trợ triển khai, tuy nhiên mỗi thư viện hướng đến những mục đích và mức độ phân tích khác nhau.

a) Các lớp và hàm trong SciPy

SciPy cung cấp bộ công cụ mạnh mẽ cho phân cụm phân cấp (hierarchical clustering), đặc biệt là khả năng phân tích chi tiết cấu trúc cụm. Một số hàm quan trọng bao gồm:

* scipy.cluster.hierarchy.linkage  
  Dùng để xây dựng *linkage matrix* mô tả toàn bộ quá trình hợp nhất các cụm. Hàm hỗ trợ nhiều phương pháp liên kết như *single*, *complete*, *average* và *ward*.
* scipy.cluster.hierarchy.dendrogram  
  Dùng để trực quan hóa cấu trúc phân cấp thông qua biểu đồ *dendrogram*, giúp quan sát mối quan hệ giữa các cụm và hỗ trợ việc chọn số cụm.
* scipy.cluster.hierarchy.fcluster  
  Dùng để trích xuất nhãn cụm từ linkage matrix dựa trên ngưỡng khoảng cách hoặc số lượng cụm mong muốn.

Các công cụ này cho phép phân tích toàn diện và chi tiết về quá trình phân cụm.

b) So sánh giữa SciPy và Scikit-learn

Mặc dù cả hai thư viện đều hỗ trợ phân cụm đa cấp, cách tiếp cận của chúng có sự khác biệt rõ rệt.

***SciPy***

* Tập trung vào phân tích thống kê và cấu trúc phân cấp.
* Cho phép truy cập trực tiếp vào thông tin hợp nhất cụm, từ đó phân tích sâu hơn.
* Hỗ trợ vẽ dendrogram để quan sát trực quan quá trình hình thành cụm.
* Thích hợp cho các nghiên cứu phân tích dữ liệu và khám phá cấu trúc dữ liệu.

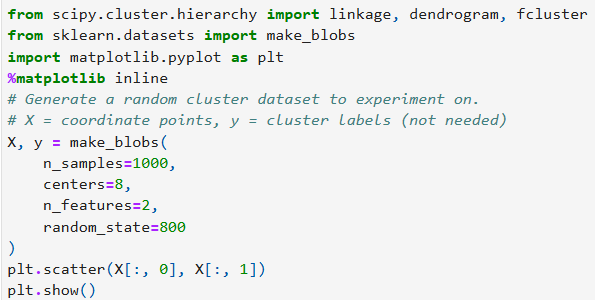
***Scikit-learn***

* Cung cấp lớp AgglomerativeClustering giúp triển khai mô hình một cách đơn giản, thống nhất với hệ sinh thái học máy của thư viện.
* Không hỗ trợ dendrogram trực tiếp, tập trung vào việc trả về nhãn cụm sau khi mô hình hoàn tất.
* Thân thiện với người dùng, phù hợp để tích hợp vào pipeline của các bài toán học máy khác.

## **II. Bài tập ứng dụng**

* 1. **Tìm hiểu một số khái niệm cơ bản của phân cụm đa cấp**
  2. ***Khai báo và tạo dữ liệu giả***

[Input]

****

[Output]

A diagram of blue dots

AI-generated content may be incorrect.

Đầu tiên, chương trình import các thư viện cần thiết như make\_blobs từ scikit-learn, cùng với matplotlib.pyplot để trực quan hóa dữ liệu. Đây là những công cụ quan trọng giúp sinh dữ liệu giả lập và hiển thị kết quả dưới dạng hình ảnh, phục vụ cho quá trình phân tích và đánh giá mô hình phân cụm.

Tiếp theo, chương trình sử dụng hàm make\_blobs() để tạo ra một tập dữ liệu gồm 1000 điểm dữ liệu, được chia thành 8 cụm khác nhau. Hàm này cho phép tùy chỉnh số mẫu, số tâm cụm và số đặc trưng, nhờ đó dữ liệu sinh ra có cấu trúc rất rõ ràng và phù hợp để thử nghiệm các thuật toán phân cụm. Việc đặt random\_state giúp đảm bảo rằng dữ liệu sinh ra có thể tái lập giống nhau trong mỗi lần chạy.

Sau khi tạo dữ liệu, chương trình sử dụng lệnh plt.scatter() để vẽ các điểm dữ liệu dựa trên hai tọa độ đặc trưng. Biểu đồ được hiển thị trực tiếp bằng plt.show(), giúp quan sát sự phân tách giữa các cụm. Dữ liệu thu được có tám cụm tách biệt rõ ràng, phù hợp để thử nghiệm các phương pháp phân cụm như K-Means hoặc Hierarchical Clustering.

* 1. ***Hiển thị ma trận khoảng cách giữa các điểm dữ liệu với điểm trung tâm***

[Input]

A white background with black and red text

AI-generated content may be incorrect.

[Output]

**A number and equation on a white background

AI-generated content may be incorrect.**

Ở bước này, chương trình sử dụng hàm linkage() từ thư viện scipy.cluster.hierarchy để xây dựng cấu trúc phân cấp của dữ liệu dựa trên thuật toán phân cụm đa cấp (Hierarchical Clustering). Cụ thể, hàm được gọi với tham số:

* X: tập dữ liệu gồm các điểm trong không gian hai chiều.
* method="centroid": quy định cách tính khoảng cách giữa các cụm dựa trên tâm cụm (centroid linkage). Khi sử dụng phương pháp này, khoảng cách giữa hai cụm được tính bằng khoảng cách giữa hai điểm tâm trung bình của từng cụm.
* metric="euclidean": chỉ định sử dụng khoảng cách Euclid — một thước đo phổ biến và phù hợp cho dữ liệu dạng tọa độ.

Kết quả trả về được lưu trong biến distances. Đây là một ma trận dạng *linkage matrix*, chứa thông tin về cách các cụm được gộp lại theo từng bước. Mỗi dòng trong ma trận biểu diễn:

* Hai cụm hoặc điểm được hợp nhất
* Khoảng cách tại thời điểm hợp nhất
* Số lượng điểm trong cụm mới tạo ra

Việc in ma trận này ra màn hình giúp chúng ta phân tích nắm được quá trình gộp cụm diễn ra như thế nào ở từng bước, và dữ liệu này cũng được sử dụng để vẽ dendrogram ở các bước tiếp theo. Dòng lệnh này đóng vai trò nền tảng trong phương pháp phân cụm đa cấp, vì nó mô tả đầy đủ cấu trúc phân cấp của tất cả các điểm dữ liệu.

### ***1.3. Vẽ hình dendrogram từ ma trận khoảng cách***

[Input]

**A computer screen shot of a code

AI-generated content may be incorrect.**

[Output]

A graph of a cluster

AI-generated content may be incorrect.

Đoạn chương trình định nghĩa hàm annotated\_dendrogram() để trực quan hóa cấu trúc phân cụm bằng dendrogram, đồng thời bổ sung các chú thích giúp người đọc dễ quan sát hơn quá trình gộp cụm.

Bên trong hàm, chương trình gọi hàm dendrogram() của SciPy để vẽ dendrogram chuẩn, sử dụng các tham số:

* truncate\_mode='lastp': rút gọn biểu đồ chỉ hiển thị một số cụm cuối cùng, giúp hình vẽ gọn và dễ quan sát hơn khi số điểm dữ liệu lớn.
* show\_contracted=True: biểu diễn các cụm bị rút gọn dưới dạng nhánh ngắn gọn.
* leaf\_rotation=90.: xoay nhãn lá 90 độ để bố cục biểu đồ dễ đọc.

Sau khi dựng dendrogram cơ bản, chương trình thêm tiêu đề và nhãn trục:

* Trục X thể hiện số lượng phần tử trong cụm.
* Trục Y biểu diễn khoảng cách (độ tương tự) giữa các cụm khi chúng được hợp nhất.

Tiếp đó, đoạn mã duyệt qua từng nhánh của dendrogram để tìm những điểm hợp nhất có khoảng cách đủ lớn (các giá trị y > 10). Với mỗi điểm như vậy, chương trình:

* Vẽ một chấm đánh dấu vị trí gộp cụm.
* Ghi chú giá trị khoảng cách tương ứng lên biểu đồ.

Những chú thích này giúp dễ nhận biết các “bước hợp nhất quan trọng” — tức là những vị trí có thể chọn làm ranh giới khi quyết định số lượng cụm tối ưu.

Cuối cùng, hàm trả về dữ liệu dendrogram và lệnh plt.show() hiển thị biểu đồ hoàn chỉnh. Đây là công cụ trực quan rất quan trọng trong phân cụm đa cấp, vì nó cho phép người phân tích lựa chọn số cụm dựa trên khoảng cách chia tách giữa các nhánh lớn.

### ***1.4. Vẽ các cụm dữ liệu dựa ma trận khoảng cách và thông tin từ dendrogram với hàm fcluster***

[Input]

**A white background with text

AI-generated content may be incorrect.**

[Output]

**A graph showing different colored spots

AI-generated content may be incorrect.**

Các điểm dữ liệu được chia thành nhiều nhóm với màu sắc tách biệt rõ, phản ánh đúng cấu trúc của dữ liệu ban đầu.

* Vì dữ liệu được tạo từ 8 cụm rõ ràng bằng make\_blobs(), nên sử dụng ngưỡng cắt 3 đã tách được các cụm tương đối hợp lý, dù số lượng cuối cùng phụ thuộc vào hình dạng của dendrogram.
* Phân cụm SciPy không yêu cầu chỉ định số cụm trước, nhưng kết quả phụ thuộc mạnh vào ngưỡng cắt (distance threshold).
* Cách tiếp cận này giúp trực quan và linh hoạt hơn so với K-Means, đặc biệt hữu ích khi dữ liệu không có dạng cầu (non-spherical) hoặc khi nghi ngờ có nhiều mức phân cấp.

### ***1.5. Vẽ các cụm dữ liệu với các phương pháp linkage khác nhau***

[Input]

**A computer code with text

AI-generated content may be incorrect.**

Mỗi phương pháp tạo ra cách tính khoảng cách giữa hai cụm hoàn toàn khác nhau:

* Single linkage: khoảng cách giữa hai điểm gần nhất
* Complete linkage: khoảng cách giữa hai điểm xa nhất
* Average linkage: trung bình khoảng cách giữa tất cả các cặp điểm
* Centroid: khoảng cách giữa hai tâm cụ cụm
* Weighted linkage: trung bình có trọng số trong quá trình gộp cụm

Điều này dẫn đến các cấu trúc phân cụm cũng khác nhau.

*a) Single linkage(khoảng cách giữa hai điểm gần nhất)*

[Output]

**A diagram of different colored dots

AI-generated content may be incorrect.**

Phương pháp single tạo ra hiện tượng “chuỗi” nên ghép các cụm lại thành một nhóm lớn. Các cụm phía trên bị nhập vào nhau, chỉ còn cụm phía dưới được tách riêng. Phân cụm không tốt, dễ dính cụm.

*b) Complete linkage(khoảng cách giữa hai điểm xa nhất)*

[Output]

**A group of colorful dots

AI-generated content may be incorrect.**

Các cụm được tách biệt rõ ràng và không bị nhập vào nhau. Complete linkage tạo biên ranh giới chặt hơn nên phân cụm ổn định và chính xác hơn single. Các nhóm được nhận diện rõ và ít nhiễu.

*c) Average linkage(trung bình khoảng cách giữa tất cả các cặp điểm)*

[Output]

**A graph showing different colored dots

AI-generated content may be incorrect.**

Các cụm được phân tách rõ ràng và ổn định. Average linkage cho kết quả mượt, không bị nhập cụm như single và ít nhạy biên hơn complete. Phân cụm nhìn tổng thể khá tốt và tự nhiên.

*d) Centroid(khoảng cách giữa hai tâm cụ cụm)*

[Output]

**A diagram of different colored dots

AI-generated content may be incorrect.**

Biểu đồ cho thấy các cụm được tách biệt rõ ràng, không chồng chéo. Cụm có dạng tròn và đồng đều do sử dụng phương pháp centroid. Phân cụm hoạt động tốt với cấu trúc dữ liệu này.

*e) Weighted linkage(trung bình có trọng số trong quá trình gộp cụm)*

[Output]

**A graph showing different colored spots

AI-generated content may be incorrect.**

Các cụm được phân tách tốt và ổn định. Weighted linkage hoạt động tương tự average, tạo ranh giới mềm và phân cụm tự nhiên, không nhập cụm như single. Kết quả nhìn chung rõ ràng và nhất quán.

**Nhận xét tổng quát về kết quả**

Khi chạy đoạn mã này, một loạt biểu đồ phân cụm sẽ xuất hiện, mỗi biểu đồ tương ứng với một phương pháp liên kết. Một số nhận xét quan trọng có thể ghi vào báo cáo:

* Single linkage thường tạo các cụm kéo dài và dễ bị “chuỗi hóa” (chaining effect), đặc biệt khi có nhiễu hoặc các điểm đứng gần nhau.
* Complete linkage tạo các cụm chặt chẽ hơn, ít bị dây chuyền, thường phân tách rõ ràng hơn.
* Average và weighted linkage có xu hướng tạo kết quả cân bằng giữa single và complete, ít nhạy cảm với nhiễu hơn.
* Centroid linkage đôi khi tạo ra các cấu trúc phân cấp không đơn điệu (inversions), nhưng cho kết quả mềm mại và đều hơn.
* Dữ liệu dạng blob như trong ví dụ này phù hợp với average và complete linkage, vì chúng tạo ra biên phân cách rõ rệt nhất.

Nhờ so sánh trực quan này, người phân tích có thể chọn phương pháp phù hợp nhất với dữ liệu thực tế và mục tiêu phân tích.

### ***1.6. Xây dựng mô hình phân cụm đa cấp theo tiếp cận của thư viện scikit-learn***

[Input]

**A screen shot of a computer code

AI-generated content may be incorrect.**

[Output]

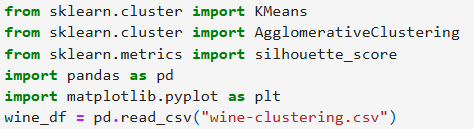
**A diagram of different colored dots

AI-generated content may be incorrect.**

Scikit-Learn và SciPy có cách tiếp cận khác nhau, do đó kết quả phân cụm có thể không trùng khớp.

* Scikit-Learn bắt buộc phải chọn số lượng cụm trước (n\_clusters), trong khi SciPy cho phép cắt dựa trên khoảng cách của dendrogram (criterion = distance).
* Việc dùng average linkage giúp mô hình tạo ra các cụm có độ mượt và ổn định hơn so với single linkage (vốn dễ bị “kéo chuỗi”).
* Biểu đồ từ Scikit-Learn cho thấy các cụm được phân tách tương đối rõ nhờ dữ liệu dạng blob và cách tính trung bình khoảng cách.
* Nhìn chung, Scikit-Learn phù hợp cho dự đoán nhanh và thực tế, còn SciPy phù hợp cho phân tích trực quan bằng dendrogram.
  1. **Phân cụm đa cấp trên tập dữ liệu rượu vang**

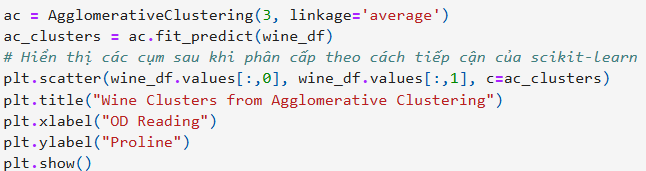
***2.1. Khai báo dữ liệu và thư viện***



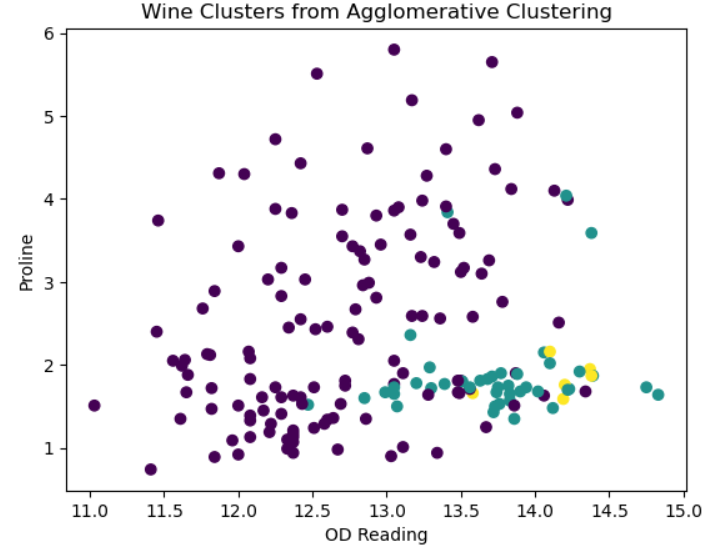
* Dữ liệu rượu vang chứa nhiều đặc tính hóa học khác nhau.
* Hai thuộc tính được sử dụng để trực quan hóa cụm trong biểu đồ là:
  + OD Reading (mật độ quang học) – trục X
  + Proline – trục Y
* Đây là hai biến có khả năng phân biệt rõ giữa các nhóm rượu vang, phù hợp để hiển thị kết quả phân cụm.

***2.2. Xây dựng mô hình phân cụm phân cấp (Agglomerative Clustering)***

[Input]



[Output]



**Nhận xét**

* Cụm 1 (màu tím và lớn nhất):
  + Trải rộng từ OD Reading 11.5–13.5, Proline 1–5.
  + Phân bố loãng và đa dạng → nhóm rượu vang có sự biến động lớn về thành phần hóa học.
* Cụm 2 (màu xanh lam/xanh lục):
  + Nằm chủ yếu ở OD Reading 13–14.5, Proline 1–2.
  + Gom lại thành dải ngang rõ rệt → nhóm rượu vang có Proline thấp nhưng OD cao hơn.
* Cụm 3 (màu vàng, ít điểm):
  + Tập trung ở OD Reading 14–14.5, Proline 1.5–2.
  + Đây là nhóm nhỏ, có thể thuộc dạng rượu vang đặc thù hoặc ít xuất hiện.
* Tổng thể:
  + Thuật toán phân cấp tạo cụm không đối xứng và tự nhiên, phù hợp với dữ liệu thực tế.
  + Có sự chồng lấn nhẹ giữa các cụm, cho thấy dữ liệu rượu vang không hoàn toàn tách biệt.

***2.3. Phân cụm K-Means***

[Input]

A computer screen shot of a code

AI-generated content may be incorrect.

[Output]

A chart of different colored dots

AI-generated content may be incorrect.

**Cụm 1 – màu xanh lam/xanh lục (cluster lớn và dàn trải):**

* Phân bố rộng từ OD Reading 11.5 – 14.5.
* Giá trị Proline dao động từ 1 đến hơn 5.
* Đây là cụm lớn nhất và loãng nhất → thể hiện nhóm rượu vang có thành phần hóa học biến thiên mạnh.
* Biển dạng này cho thấy các loại rượu vang trong cụm có sự khác biệt nội tại khá lớn.

**Cụm 2 – màu tím (trung bình và chồng lấn với cụm 1):**

* Nằm chủ yếu trong vùng OD Reading 12.5 – 14.0.
* Proline nằm trong khoảng 1 – 4.
* Các điểm phân bố rời rạc và xen lẫn với cụm 1.
* Cho thấy K-Means không phân tách hoàn toàn được khi hai nhóm có đặc tính hóa học gần nhau.

**Cụm 3 – màu vàng (cụm nhỏ, tách biệt rõ nhất):**

* Tập trung ở OD Reading 13.8 – 14.5.
* Proline chủ yếu trong khoảng 1.5 – 2.
* Đây là nhóm ổn định và đồng nhất hơn, đặc trưng bởi OD Reading cao và Proline thấp.
* Cụm này ít điểm → có thể đại diện cho một loại rượu vang riêng biệt hoặc ít phổ biến.

**Tổng thể:**

* K-Means chia dữ liệu thành 3 nhóm tương đối rõ ràng nhưng có sự chồng lấn đáng kể giữa cụm xanh và cụm tím.
* K-Means hoạt động tốt nhất với cụm có hình dạng tròn/đối xứng, nhưng dữ liệu này lại có hình dạng phức tạp → dẫn đến ranh giới phân cụm chưa tự nhiên.
* Tuy nhiên, cụm màu vàng vẫn thể hiện được tính chất tách biệt và dễ nhận dạng.

***2.4. Đánh giá chất lượng phân cụm bằng Silhouette Score***

[Input]

A close up of a computer screen

AI-generated content may be incorrect.

[Output]

A black text on a white background

AI-generated content may be incorrect.

**Mô hình phân cấp (Agglomerative) có điểm Silhouette cao hơn K-Means.**  
Điều này cho thấy:

* Các điểm trong cùng một cụm gần nhau hơn.
* Khoảng cách giữa các cụm tách biệt rõ ràng hơn.
* Ranh giới cụm được hình thành tự nhiên hơn.

**K-Means với Silhouette 0.5596:**

* Vẫn đạt mức khá nhưng không tối ưu.
* Bị ảnh hưởng bởi phân bố không tròn của dữ liệu → tạo ra một số vùng chồng lấn giữa các cụm.

**So sánh tổng quan:**

* Agglomerative Clustering phù hợp hơn cho bộ dữ liệu rượu vang, do dữ liệu có dạng "phân bố tự nhiên" chứ không đối xứng theo tâm như K-Means yêu cầu.
* Quan sát biểu đồ trực quan cũng cho thấy cụm Agglomerative ít chồng lấn hơn và phân tách hợp lý hơn.
  1. **Xây dựng mô hình phân cụm đa cấp trên tập dữ liệu chim cánh cụt**

***3.1. Khai báo dữ liệu và tiền xử lý***

A screenshot of a computer program

AI-generated content may be incorrect.

* Penguin dataset chứa nhiều giá trị thiếu, việc loại bỏ missing giúp đảm bảo độ chính xác.
* Đặc trưng *flipper\_length\_mm* dễ sinh ngoại lai → lọc giúp mô hình ổn định hơn.
* Chuẩn hóa là cần thiết vì 4 đặc trưng khác đơn vị → tránh cụm bị chi phối bởi đặc trưng lớn (body\_mass\_g).
* Kết quả tạo ra một tập dữ liệu sạch, cân bằng và sẵn sàng để phân cụm.

***3.2. X*** ***ây dựng mô hình phân cụm phân cấp (Agglomerative Clustering)***

[Input]

A screenshot of a computer code

AI-generated content may be incorrect.

[Output]

A screenshot of a computer

AI-generated content may be incorrect.

**Các bước gộp cụm đầu tiên có khoảng cách rất nhỏ (0.10–0.15)**

* Thể hiện nhiều cặp cá thể có đặc trưng hình thái gần như giống hệt nhau.
* Đây thường là các mẫu cùng loài hoặc có cấu trúc mỏ, sải cánh và khối lượng tương tự.
* Thuật toán hợp nhất nhanh các cặp này vì chi phí tăng phương sai gần như bằng 0.

**Các bước gộp cuối cùng có khoảng cách tăng mạnh (từ ~6 đến ~39)**

* Cho thấy những cụm lớn, khác biệt rõ rệt đang bị hợp nhất.
* Sự tăng đột ngột của khoảng cách chứng minh dữ liệu tồn tại các nhóm sinh học tách biệt (các loài chim khác nhau).
* Các cụm này chứa số lượng mẫu lớn (ví dụ 71 → 333 mẫu), thể hiện rõ cấu trúc phân cấp.

**Dendrogram thể hiện cấu trúc phân cấp tự nhiên**

* Nhiều nhánh nhỏ xuất hiện ở mức thấp → các cụm nhỏ rất đồng nhất.
* Từ mức giữa trở lên, hình thành 2–3 nhánh lớn, trước khi khoảng cách gộp tăng đột ngột.
* Ngưỡng cắt tối ưu nằm ở vùng trước “bước nhảy” lớn → cho 3 cụm tương ứng với ba loài chim (Adelie, Gentoo, Chinstrap).

**Ward linkage là lựa chọn phù hợp cho dữ liệu chim cánh cụt**

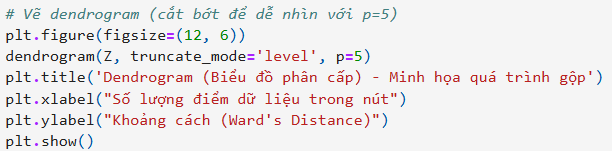
* Vì dữ liệu gồm các đặc trưng hình thái giàu tính phân biệt (culmen, flipper, body mass).
* Phương pháp Ward tối ưu hóa phương sai nội cụm → tạo nên các cụm đồng nhất và tách biệt rõ ràng.
* Kết quả phân cụm phản ánh đúng cấu trúc tự nhiên của dữ liệu, ngay cả khi không dùng nhãn loài.

**Ý nghĩa tổng quan**

* Mô hình giúp quan sát toàn bộ quá trình hình thành cụm, từ mức cá thể đến mức nhóm loài.
* Cho phép xác định số cụm tối ưu dựa trên cấu trúc dendrogram thay vì phỏng đoán.
* Kết quả phân cụm có thể được dùng cho phân tích sinh học, phân nhóm mẫu mới hoặc hỗ trợ nghiên cứu hình thái học.

***3.3. Xây dựng mô hình phân cụm phân cấp***

[Input]



A computer screen shot of a computer code

AI-generated content may be incorrect.

[Output]

A diagram of a diagram

AI-generated content may be incorrect.

* Cấu trúc phân cấp rõ rệt: dendrogram có nhiều nhánh nhỏ ở mức khoảng cách thấp → chứng tỏ có nhiều cặp mẫu gần như giống nhau về đặc trưng hình thái. Những nhánh này biểu thị các cụm rất đồng nhất ở cấp thấp.
* Hai pha hợp nhất khác biệt: ảnh cho thấy hai vùng: (1) nhiều hợp nhất ban đầu với khoảng cách cực nhỏ (~0.10–0.15) — hợp nhất các cặp rất giống nhau; (2) các hợp nhất cuối cùng với khoảng cách tăng mạnh (~6 → ~39) — hợp nhất các cụm lớn, khác biệt rõ rệt. Sự tăng đột ngột này là dấu hiệu tồn tại các nhóm sinh học tách bạch.
* Ngưỡng cắt cho k=3 hợp lý: theo dendrogram, cắt trước khi xảy ra “bước nhảy” lớn cho ra 3 nhánh lớn — phù hợp với giả thiết có 3 nhóm (loài).
* Ward linkage phù hợp để minh họa nội phương sai: vì Ward tối ưu nội phương sai, dendrogram phản ánh mức đồng nhất nội cụm tại các mức thấp; tuy nhiên khi hợp nhất lên cấp lớn, một số cụm có thể trở nên rộng hơn.

### ***3.4. Biểu đồ phân tán***

[Input]

A computer screen shot of a code

AI-generated content may be incorrect.

[Output]

A diagram of different colored dots

AI-generated content may be incorrect.

* Phân bố theo hai chiều: một cụm rõ rệt nằm ở vùng bill\_length nhỏ & bill\_depth lớn. Hai cụm còn lại nằm gần nhau hơn dọc theo chiều bill\_length lớn hơn/độ sâu thay đổi — gây một vùng chồng lấp ở giữa.
* Khu vực chồng lấp: trên mặt phẳng bill\_length–bill\_depth, có một vùng trung tâm nơi điểm từ các cụm khác màu xen lẫn — cho thấy cặp chiều này không hoàn toàn tách biệt mọi cluster.
* Tính trực quan: biểu đồ cho phép quan sát trực tiếp “độ tách” theo cặp đặc trưng này — cụm bên trái rất đồng nhất, hai cụm còn lại kém tách hơn trên cặp chiều này.
* Kết luận trực quan: bill\_length vs bill\_depth cho thấy sự tồn tại của một nhóm có bill đặc trưng khác biệt; nhưng để phân tách hoàn chỉnh cần xét thêm các chiều khác.

### ***3.5. Kết quả mô hình***

[Input]

A screen shot of a computer code

AI-generated content may be incorrect.

[Output]

A diagram of a number of dots

AI-generated content may be incorrect.

* Ba cụm rõ rệt về màu sắc: K-Means phân bố các điểm thành ba vùng màu, mỗi vùng có tâm rõ và phần lớn điểm tập trung quanh tâm đó.
* Độ chồng lấp nhỏ hơn so với Agglomerative trên cùng cặp chiều: so với scatter của phân cụm phân cấp, hình K-Means cho thấy các vùng màu ít xen lẫn hơn (một số điểm vẫn ở vùng biên).
* Hình dạng cụm mang tính “hình cầu” trên 2D: các cụm dạng tập trung, không kéo dài theo một chiều — đây là biểu hiện điển hình của kết quả K-Means trên hai chiều này.
* Phân bố số lượng cụm không hiện trên ảnh (ảnh K-Means không trực tiếp cho biết số lượng điểm mỗi cụm) nhưng trực quan cho thấy sự cân đối hơn giữa các cụm so với phân cụm phân cấp.

### ***3.6. Đánh giá chất lượng phân cụm***

[Input]

A screen shot of a computer code

AI-generated content may be incorrect.

[Output]

A black text on a white background

AI-generated content may be incorrect.

* Sự khác biệt định lượng rõ rệt: silhouette của K-Means (~0.556) cao — biểu thị cụm đồng nhất và tách bạch; silhouette của Agglomerative (~0.136) rất thấp — nhiều điểm gần biên hoặc nội phương sai cụm lớn. Đây là chứng cứ định lượng xuất hiện trên ảnh.
* Mối liên hệ giữa bảng đếm và silhouette thấp: Agglomerative có một cụm rất lớn (156) — điều này tương ứng với scatter thể hiện một màu lan rộng xuyên vùng trung tâm; do đó nhiều điểm trong cụm lớn có lân cận gần các cụm khác → làm giảm silhouette.
* Dendrogram giải thích cấu trúc: các hợp nhất ban đầu rất nhỏ (cặp giống nhau) và hợp nhất cuối lớn (khoảng cách ~6→39) — cho thấy dữ liệu có cấu trúc phân cấp, nhưng khi cắt thành 3 cụm (để so sánh) nội dung một số cụm trở nên không đồng nhất.
* Tương quan trực quan — số học: hình K-Means + silhouette cao + scatter ít chồng lấp cho thấy K-Means phù hợp hơn với cấu trúc phân chia hiện tại trên không gian đặc trưng đã chuẩn hóa; trong khi Agglomerative cung cấp thông tin phân cấp (dendrogram) nhưng khi ép cắt thành 3 nhóm cho thấy chất lượng phân cụm kém
* Kết luận dựa hoàn toàn trên ảnh: các hình và số trên ảnh cho thấy K-Means đạt chất lượng phân cụm tốt hơn theo metric silhouette và trực quan; Agglomerative mô tả tốt cấu trúc phân cấp nhưng, sau khi cắt thành 3 cụm, nội dung cụm kém đồng nhất.

**4. Xây dựng mô hình phân cụm đa cấp trên tập dữ liệu mua sắm tại siêu thị**

### ***4.1. Khai báo thư viện và đọc dữ liệu***

[Input]

**A screen shot of a computer code

AI-generated content may be incorrect.**

****

[Output]

**A screenshot of a computer

AI-generated content may be incorrect.**

Dữ liệu được đọc vào bằng hàm pd.read\_csv() với mã hóa ISO-8859-1 để xử lý đúng các ký tự đặc biệt trong tệp.

 Kết quả df.info() cho thấy tập dữ liệu gồm 541.909 dòng và 8 cột, dung lượng khoảng 33 MB. Các trường như InvoiceNo, StockCode, Description, InvoiceDate và Country có kiểu dữ liệu dạng chuỗi, trong khi Quantity, UnitPrice và CustomerID là dữ liệu số. Riêng cột CustomerID chỉ có 406.829 giá trị hợp lệ, phản ánh việc thiếu thông tin khách hàng ở một phần dữ liệu. Tổng quan cho thấy dữ liệu tương đối đầy đủ nhưng cần tiền xử lý trước khi phân tích sâu hơn.

### ***4.2. Xem thông tin dữ liệu***

A screenshot of a computer screen

AI-generated content may be incorrect.

• InvoiceNo: Mã số hóa đơn. Mỗi hóa đơn đại diện cho một giao dịch. Một khách hàng có thể xuất hiện trong nhiều hóa đơn khác nhau.

• StockCode: Mã sản phẩm. Dùng để phân biệt từng mặt hàng trong kho.

• Description: Mô tả tên sản phẩm. Một số giá trị có thể bị thiếu hoặc lỗi mã hóa.

• Quantity: Số lượng sản phẩm được mua. Nếu giá trị âm → là hàng trả lại (refund).

• InvoiceDate: Ngày và giờ diễn ra giao dịch. Dùng để trích xuất thời gian mua hàng như buổi sáng/chiều/tối hoặc tính Recency.

• UnitPrice: Giá bán của một đơn vị sản phẩm.

• CustomerID: Mã khách hàng, dùng để xác định mỗi người mua. Nhiều dòng có thể không có CustomerID → khách hàng không đăng ký.

• Country: Quốc gia của khách hàng. Dataset chủ yếu là khách từ Anh (United Kingdom).

### ***4.3. Tiền xử lý dữ liệu***

*4.3.1. Kiểm tra dữ liệu bị thiếu*

[Input]

****

[Output]

**A screenshot of a computer code

AI-generated content may be incorrect.**

Sau khi tải dữ liệu, bước đầu tiên là kiểm tra số lượng giá trị bị khuyết trong từng cột.

CustomerID có 135.080 giá trị thiếu

Đây là số lượng thiếu lớn nhất trong tập dữ liệu. Các giao dịch thiếu mã khách hàng thường là:

* giao dịch tại quầy (không đăng ký khách hàng),
* khách lẻ,
* hoặc dữ liệu không được ghi nhận đầy đủ.

Vì phân tích hành vi khách hàng (RFM, phân cụm…) yêu cầu phải có CustomerID → những dòng này sẽ bị loại bỏ để đảm bảo kết quả chính xác.

Description thiếu 1.454 giá trị

Việc thiếu mô tả không ảnh hưởng đến phân tích giao dịch hoặc phân tích khách hàng (vì không dùng nội dung mô tả sản phẩm).  
Ta có thể giữ lại hoặc bỏ qua vì không ảnh hưởng tới RFM.

Các cột còn lại không có giá trị thiếu, đảm bảo tính toàn vẹn dữ liệu:

* InvoiceNo
* StockCode
* Quantity
* InvoiceDate
* UnitPrice
* Country

*4.3.2. Làm sạch và chuẩn hóa dữ liệu giao dịch*

**A screen shot of a computer code

AI-generated content may be incorrect.**

Trong bước này, dữ liệu giao dịch được làm sạch và chuẩn hóa để đảm bảo tính chính xác trước khi phân tích. Các thao tác bao gồm:

* Loại bỏ các dòng không có mã khách hàng (CustomerID) và những giao dịch có số lượng âm, vì đây thường là các đơn hủy hoặc trả hàng.
* Tạo biến TotalSum nhằm tính tổng giá trị cho từng dòng giao dịch theo công thức:  *TotalSum = Quantity × UnitPrice*.Việc bổ sung biến này giúp thuận tiện cho việc phân tích doanh thu và hành vi mua hàng.
* Chuyển đổi InvoiceDate sang dạng datetime chuẩn, phục vụ cho các thao tác phân tích theo thời gian như phân nhóm theo ngày, tháng hoặc tính RFM.

*4.3.3. Tạo tính năng RFM*

**A screen shot of a computer code

AI-generated content may be incorrect.**

Ở bước này, mô hình RFM được tạo ra nhằm đánh giá và phân nhóm khách hàng dựa trên ba yếu tố quan trọng:

* Recency (R) – số ngày kể từ lần mua hàng gần nhất của khách hàng.
* Frequency (F) – số lượng giao dịch mà khách hàng đã thực hiện.
* Monetary (M) – tổng giá trị chi tiêu của khách hàng.

Các chỉ số này giúp doanh nghiệp hiểu rõ hành vi khách hàng, từ đó đưa ra chiến lược chăm sóc phù hợp.

Quy trình xử lý:

* Xác định snapshot\_date: lấy ngày ngay sau ngày mua hàng cuối cùng trong dữ liệu làm mốc tính Recency.
* Gộp dữ liệu theo CustomerID để tính R, F, M:
  + Recency: tính số ngày từ snapshot\_date đến ngày giao dịch gần nhất của khách đó.
  + Frequency: đếm tổng số hóa đơn (InvoiceNo) của khách hàng.
  + Monetary: tính tổng giá trị chi tiêu (TotalSum).
* Đổi tên các cột để thuận tiện trong phân tích.

*4.3.4. Chuẩn hóa dữ liệu*

**A black text on a white background

AI-generated content may be incorrect.**

Sau khi tính toán bộ chỉ số RFM, các giá trị Recency, Frequency và Monetary có thể có sự chênh lệch rất lớn về đơn vị và độ lớn (ví dụ: Recency có thể vài chục ngày, trong khi Monetary có thể lên đến hàng nghìn). Điều này dễ gây ảnh hưởng không cân bằng đến các thuật toán phân cụm như K-Means.

Để khắc phục vấn đề trên, dữ liệu được chuẩn hóa bằng StandardScaler – phương pháp đưa các biến về cùng một phân phối chuẩn (mean = 0, std = 1). Nhờ đó, tất cả các đặc trưng đều có mức đóng góp tương đương khi đưa vào mô hình.

**Ý nghĩa của chuẩn hóa:**

* Giảm ảnh hưởng của các biến có giá trị quá lớn.
* Giúp thuật toán tối ưu hội tụ nhanh hơn.
* Mô hình phân cụm hoạt động ổn định và chính xác hơn.

### ***4.4. Xây dựng mô hình phân cụm đa cấp***

[Input]

**A computer screen shot of text

AI-generated content may be incorrect.**

[Output]

**A white background with black text

AI-generated content may be incorrect.**

* Các bước cuối cho thấy quá trình gộp cụm diễn ra giữa các nhóm khách hàng có kích thước ngày càng lớn.
* Khoảng cách (Distance) tăng dần qua từng bước, phản ánh mức độ khác biệt khi gộp các cụm lớn lại với nhau.
* Ở bước cuối cùng, hai cụm có tổng cộng 4339 khách hàng được gộp tại mức khoảng cách 88.53, đánh dấu sự hoàn tất của cấu trúc phân cấp.
* Điều này cho thấy dữ liệu có sự đa dạng lớn về hành vi RFM, vì để gộp được các cụm lớn, thuật toán cần một khoảng cách tương đối cao.

Việc phân tích các bước gộp cuối giúp ta lựa chọn ngưỡng cắt cụm (cut-off) phù hợp khi dựng biểu đồ dendrogram ở bước kế tiếp.

### ***4.5. Biểu đồ Dendrogram***

[Input]

**A computer screen shot of text

AI-generated content may be incorrect.**

[Output]

**A diagram of a diagram

AI-generated content may be incorrect.**

Quan sát dendrogram cho thấy:

* Phần thân dưới (dưới 40–50 trên trục y) có nhiều nhánh nhỏ → dữ liệu có nhiều nhóm tương đồng ở mức chi tiết.
* Phần thân trên xuất hiện các nhánh dài → thể hiện sự khác biệt lớn giữa các cụm chính.
* Nếu chọn ngưỡng khoảng 60–70, số cụm rơi vào mức 3–5 cụm, phù hợp với bài toán phân nhóm khách hàng.

Dendrogram là bước quan trọng giúp xác định số cụm tối ưu trước khi đưa ra quyết định cuối cùng cho bài phân tích RFM.

### ***4.6. Gán nhãn cụm và xác định phân khúc khách hàng***

**A computer screen shot of a code

AI-generated content may be incorrect.**

Đầu tiên, cây phân cấp được cắt thành 3 cụm dựa trên tiêu chí *maxclust*. Nhãn cụm (Cluster\_Labels) được gán cho từng khách hàng trong bảng RFM.  
 Tiếp theo, để đặt tên phân khúc, ta sử dụng giá trị trung bình của biến Monetary trong mỗi cụm. Các cụm có mức chi tiêu trung bình cao hơn được ưu tiên gắn nhãn “VIP”, tiếp theo là “Khách Trung” và cuối cùng là “Bình thường”.

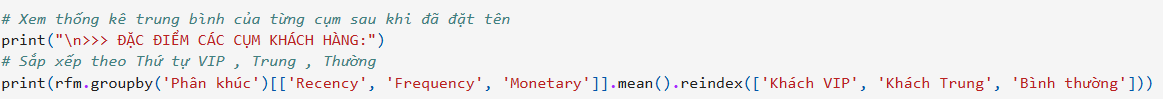
Quy trình gồm các bước:

* Tính trung bình Monetary cho từng cụm và sắp xếp theo thứ tự giảm dần để nhận diện mức độ giá trị của từng nhóm khách.
* Tạo từ điển ánh xạ (mapping) giữa số cụm và tên phân khúc dựa trên thứ tự Monetary.
* Gắn tên phân khúc mới cho từng khách hàng bằng cách thêm cột *Phân khúc*.

Cuối cùng, thống kê trung bình của ba biến Recency, Frequency và Monetary được tính lại theo phân khúc nhằm mô tả đặc điểm của từng nhóm khách (Khách VIP, Khách Trung và Bình thường).

### ***4.7. Kết quả thống kê trung bình theo từng phân khúc***

[Input]

****

[Output]

**A white background with black text

AI-generated content may be incorrect.**

*Khách VIP*

* Có *Recency* rất thấp (≈ 5 ngày), nghĩa là họ vừa mua hàng gần đây.
* *Frequency* cực kỳ cao, trung bình hơn 2600 lần mua.
* *Monetary* vượt trội, đạt khoảng 138,000+.

→ Đây là nhóm mang lại doanh thu lớn nhất, mua thường xuyên và duy trì mức độ tương tác rất cao. Cần được ưu tiên trong các chiến dịch chăm sóc đặc biệt (quà tặng, ưu đãi độc quyền, chăm sóc cá nhân hóa).

*Khách Trung*

* *Recency* ở mức trung bình (≈ 41 ngày).
* *Frequency* và *Monetary* vừa phải.

→ Đây là nhóm khách hàng có giá trị trung bình, vẫn còn hoạt động tương đối tốt. Có thể kích thích họ tăng mức độ mua bằng các chương trình khuyến mãi hoặc chăm sóc duy trì.

*Bình thường*

* *Recency* rất cao (≈ 246 ngày), tức là nhóm này đã lâu không quay lại.
* *Frequency* và *Monetary* thấp.

→ Đây là các khách hàng có rủi ro rời bỏ cao (churn). Cần chiến dịch tái kích hoạt như nhắc nhở, giảm giá, voucher quay lại.

# **C. Tóm tắt thực hành**

Trong bài thực hành này, người học triển khai và phân tích hai kỹ thuật phân cụm dữ liệu phổ biến trong học máy gồm thuật toán K-means và phân cụm đa cấp (Hierarchical Clustering). Đây là những phương pháp quan trọng của học máy không giám sát, cho phép khám phá cấu trúc tiềm ẩn trong dữ liệu và nhóm các phần tử có đặc điểm tương đồng lại với nhau.

Khó khăn chủ yếu trong quá trình thực hành xuất phát từ việc tiền xử lý và chuẩn hóa dữ liệu. Thuật toán K-means yêu cầu dữ liệu phải được đưa về cùng thang đo để đảm bảo khoảng cách được tính toán chính xác, trong khi phân cụm đa cấp dễ bị ảnh hưởng bởi nhiễu và giá trị ngoại lai. Do đó, việc xử lý dữ liệu sạch, chuẩn hóa và lựa chọn số cụm phù hợp đóng vai trò quan trọng để mô hình đạt hiệu quả cao.

Hai thuật toán được tiếp cận theo các hướng khác nhau. K-means tối ưu hóa khoảng cách giữa các điểm và tâm cụm, phù hợp với các tập dữ liệu có cấu trúc cụm rõ ràng. Ngược lại, phân cụm đa cấp xây dựng cấu trúc phân cấp thông qua dendrogram, giúp quan sát trực quan mối quan hệ giữa các điểm dữ liệu và hỗ trợ lựa chọn số cụm linh hoạt hơn.

Trong bài thực hành, các thư viện Python được sử dụng bao gồm:

* Pandas, NumPy: xử lý và chuẩn bị dữ liệu;
* Scikit-learn: cài đặt thuật toán K-means và Hierarchical Clustering;
* Matplotlib, Seaborn: trực quan hóa kết quả và biểu đồ dendrogram.

Để đảm bảo kết quả phân cụm chính xác, người học cần:

* Hiểu rõ đặc trưng dữ liệu và mục tiêu phân cụm;
* Chuẩn hóa dữ liệu khi cần thiết;
* Sử dụng trực quan hóa để đánh giá cấu trúc và mức độ tách biệt của các cụm;
* So sánh hai thuật toán để rút ra ưu, nhược điểm và lựa chọn phương pháp phù hợp.

Tóm lại, bài thực hành giúp người học hệ thống hóa kiến thức lý thuyết về phân cụm, nắm vững quy trình xử lý và triển khai mô hình, đồng thời đánh giá và so sánh hiệu quả của hai thuật toán K-means và phân cụm đa cấp. Đây là nền tảng quan trọng cho các bài toán khai phá và phân tích dữ liệu nâng cao trong thực tế.