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

- Giải thuật K-Means là một thuật toán học máy không giám sát phổ biến dùng để giải quyết bài toán phân cụm (clustering). Mục tiêu chính của nó là chia một tập hợp dữ liệu gồm N điểm thành K cụm (cluster) đã được xác định trước, sao cho tổng bình phương khoảng cách từ mỗi điểm dữ liệu đến tâm cụm (centroid) mà nó thuộc về là nhỏ nhất. Ý tưởng là các điểm trong cùng một cụm sẽ có sự tương đồng cao hơn các điểm ở các cụm khác.
- Các bước chính:
    + *Khởi tạo (Initialization)*

      Xác định K: Đầu tiên, ta cần chọn số lượng cụm mong muốn là K (số nguyên dương).
      Chọn K tâm cụm ban đầu: Chọn ngẫu nhiên K điểm dữ liệu bất kỳ trong tập dữ liệu để làm các tâm cụm (centroid) khởi tạo ban đầu, ký hiệu là $c1,c2,...,cK$

    + *Gán nhãn/Phân cụm (Assignment Step)*

      Đối với mỗi điểm dữ liệu xi trong tập dữ liệu, tính khoảng cách (thường là khoảng cách Euclidean) từ điểm đó đến tất cả K tâm cụm hiện tại $(c1,c2,...,cK)$.
      Gán điểm dữ liệu xi vào cụm có tâm cụm gần nhất với nó.

    + *Cập nhật tâm cụm (Update Step)*

      Sau khi tất cả các điểm đã được gán vào các cụm, ta tính toán lại vị trí của **tâm cụm mới** cho mỗi cụm.
      Tâm cụm mới của mỗi cụm j là **trung bình cộng** (mean) của tất cả các điểm dữ liệu đã được gán vào cụm đó.

    + Kiểm tra điều kiện dừng (Convergence Check)

Lặp lại **Bước 2** và **Bước 3** cho đến khi thuật toán hội tụ. Điều kiện dừng thường là:

- Các tâm cụm **không còn thay đổi** vị trí đáng kể so với vòng lặp trước.
- Không có điểm dữ liệu nào **thay đổi cụm** so với vòng lặp trước.
- Đạt đến số lần lặp tối đa (Max Iterations) đã định trước.

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 lượng cụm K như một tham số đầu vào** bởi vì:
- **Tính chất của thuật toán:** K-Means được thiết kế để tìm ra K tâm cụm (K centroids) sao cho tổng bình phương khoảng cách từ các điểm dữ liệu đến tâm cụm gần nhất của chúng (còn gọi là **Within-Cluster Sum of Squares - WCSS** hoặc độ biến thiên

trong cụm) là nhỏ nhất. Để bắt đầu quá trình lặp (tức là bước Khởi tạo), thuật toán cần biết nó sẽ phải khởi tạo và tìm kiếm bao nhiêu tâm cụm.

- **Quá trình Khởi tạo:** Bước đầu tiên của K-Means là **chọn ngẫu nhiên K điểm** làm tâm cụm. Nếu không có giá trị K, thuật toán sẽ không thể thực hiện bước khởi tạo này.
- **Cơ chế Phân vùng:** K-Means hoạt động bằng cách chia tập dữ liệu thành K tập con **rời rạc** và **không chồng chéo**. Số lượng tập con này chính là K.

- Việc chọn K tối ưu rất quan trọng vì nó ảnh hưởng trực tiếp đến chất lượng và tính hữu ích của kết quả phân cụm. Có hai phương pháp phổ biến nhất để xác định giá trị K tối ưu:

a. Phương pháp Elbow (Phương pháp Khuỷu tay)

Phương pháp này dựa trên việc đo lường sự biến thiên trong cụm (**WCSS**).

1. **Chạy K-Means:** Thực hiện thuật toán K-Means với một loạt các giá trị K khác nhau (ví dụ: từ K=1 đến K=10 hoặc một ngưỡng nào đó).
2. **Tính WCSS:** Đối với mỗi giá trị K, tính tổng bình phương khoảng cách từ mỗi điểm dữ liệu đến tâm cụm của nó. WCSS sẽ giảm khi K tăng (vì mỗi cụm sẽ trở nên nhỏ hơn).
3. **Vẽ biểu đồ:** Vẽ đồ thị thể hiện **WCSS** theo số lượng cụm **K**.
4. **Xác định "Elbow" (Khuỷu tay):** Quan sát đồ thị và tìm điểm mà sự giảm WCSS bắt đầu **chậm lại đáng kể**, tạo thành hình dạng giống như một "khuỷu tay". Giá trị K tại điểm "khuỷu tay" đó thường được coi là tối ưu, vì việc tăng thêm K từ điểm này trở đi không mang lại sự cải thiện đáng kể về mặt phân cụm (giảm WCSS) nhưng lại tăng thêm độ phức tạp mô hình.

b. Hệ số Silhouette (Silhouette Score)

Hệ số Silhouette đo lường **chất lượng** của các cụm bằng cách so sánh **khoảng cách trong cụm** (sự gắn kết) và **khoảng cách giữa các cụm** (sự tách biệt).

1. **Chạy K-Means:** Tương tự, chạy thuật toán K-Means cho một loạt các giá trị K.
2. **Tính Hệ số Silhouette:** Đối với mỗi giá trị K, tính **Hệ số Silhouette trung bình** cho toàn bộ dữ liệu.
    - Hệ số Silhouette nằm trong khoảng từ **-1 đến +1**.
    - Giá trị **gần +1** cho thấy điểm dữ liệu nằm rất gần cụm của nó và rất xa các cụm khác (phân cụm tốt).
    - Giá trị **gần 0** cho thấy điểm dữ liệu nằm gần ranh giới giữa hai cụm (phân cụm kém).
    - Giá trị **gần -1** cho thấy điểm dữ liệu có thể đã bị gán nhầm cụm.
3. **Xác định K tối ưu: Giá trị K cho ra Hệ số Silhouette trung bình cao nhất** thường được coi là tối ưu.

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, còn được gọi là hàm biến dạng (distortion function) hoặc **Tổng bình phương khoảng cách trong cụm (Within-Cluster Sum of**

Squares - WCSS), đo lường sự khác biệt giữa các điểm dữ liệu và tâm cụm của chúng. Mục tiêu của thuật toán là **giảm thiểu** giá trị của hàm này.

- WCSS được tính bằng cách lấy tổng bình phương khoảng cách Euclidean từ mỗi điểm dữ liệu đến tâm cụm mà nó thuộc về.
- Ý nghĩa của hàm mục tiêu: Hàm mục tiêu này đo lường **độ chặt chẽ** hay **độ gắn kết** của các cụm.

- **Giá trị J nhỏ**: Điều này cho thấy các điểm dữ liệu trong mỗi cụm **rất gần** với tâm cụm của chúng, do đó các cụm được coi là **gắn kết** và có sự phân tách rõ ràng.
- **Giá trị J lớn**: Điều này cho thấy các điểm dữ liệu nằm **xa** tâm cụm, các cụm lỏng lẻo và có thể không được phân cụm hiệu quả.

Trong mỗi bước lặp của thuật toán K-Means, quá trình cập nhật tâm cụm và gán lại các điểm dữ liệu được thực hiện nhằm mục đích liên tục **giảm giá trị của J** cho đến khi nó đạt giá trị tối thiểu cục bộ và thuật toán hội tụ. Vì lý do này, K-Means luôn đảm bảo sự hội tụ sau một số hữu hạn các bước lặp.

-

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

Hạn chế	Giải thích
Cần xác định trước K	Người dùng phải chỉ định số lượng cụm (K) trước khi chạy thuật toán. Việc chọn sai K sẽ dẫn đến kết quả phân cụm không chính xác hoặc không có ý nghĩa.
Nhạy cảm với Khởi tạo ban đầu	Kết quả phân cụm cuối cùng có thể thay đổi đáng kể tùy thuộc vào việc chọn K tâm cụm ngẫu nhiên ban đầu. Nếu các tâm cụm ban đầu được chọn không tốt, thuật toán có thể hội tụ về một cực tiểu cục bộ (local minimum) thay vì cực tiểu toàn cục (global minimum) của hàm mục tiêu.
Nhạy cảm với Ngoại lai (Outliers)	K-Means tính toán tâm cụm bằng cách lấy giá trị trung bình cộng (mean) của các điểm. Các điểm ngoại lai (giá trị cực đoan) có thể kéo tâm cụm về phía chúng, làm biến dạng hình dạng và vị trí của cụm, dẫn đến kết quả phân cụm không chính xác.
Giả định về Hình dạng cụm	K-Means hoạt động tốt nhất khi các cụm có hình dạng hình cầu (globular) và có kích thước tương tự nhau. Nó sử dụng khoảng cách Euclidean (khoảng cách "đường chim bay"), vốn giả định hình dạng hình học này.

- K-Means có thể hoạt động kém hiệu quả hoặc thất bại trong việc tìm ra các cụm ý nghĩa trong các trường hợp sau:

a. Cụm có hình dạng phức tạp (Non-Globular Shapes)

Nếu các cụm có hình dạng bất thường, chẳng hạn như hình **vòng tròn** (concentric circles), hình **bán nguyệt** (crescent shapes), hoặc hình dạng kéo dài, K-Means sẽ không thể tách chúng ra một cách chính xác. Thay vào đó, nó sẽ cố gắng chia các cụm này thành các vùng hình cầu dựa trên khoảng cách.

Ví dụ: Dữ liệu tạo thành hai hình bán nguyệt lồng vào nhau. K-Means sẽ chia chúng thành hai cụm hình cầu, cắt ngang qua các hình bán nguyệt thay vì tách chúng ra.

b. Cụm có mật độ và kích thước khác nhau

K-Means giả định rằng tất cả các cụm đều có **mật độ và kích thước tương đương**. Nếu dữ liệu có:

- **Các cụm có mật độ khác nhau:** Một cụm rất dày đặc và một cụm rất thưa thớt.
- **Các cụm có kích thước khác nhau:** Một cụm rất lớn và một cụm rất nhỏ.

Thuật toán có xu hướng chia cụm lớn thành nhiều phần và gộp các cụm nhỏ lại, hoặc làm cho ranh giới phân cụm bị lệch, dẫn đến việc gán nhãn sai.

c. Dữ liệu không tách biệt tốt (Non-Linearly Separable Data)

Nếu các cụm bị chồng chéo nhiều hoặc được phân tách bằng một ranh giới phức tạp, K-Means (là một thuật toán tuyến tính) sẽ gặp khó khăn trong việc phân tách chúng một cách hiệu quả.

d. Dữ liệu có nhiễu hoặc Ngoại lai cao

Như đã đề cập, sự hiện diện của **ngoại lai** sẽ làm lệch tâm cụm, khiến ranh giới cụm bị dịch chuyển và làm giảm chất lượng tổng thể của mô hình.

Trong các trường hợp này, các thuật toán phân cụm khác như **DBSCAN** (phù hợp với các cụm có mật độ khác nhau và hình dạng phức tạp) hoặc **Phân cụm phân cấp (Hierarchical Clustering)** có thể là lựa chọn tốt hơn.

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

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

```
from sklearn.metrics import silhouette_score
wcss = []
silhouette_scores = []
# Thử các giá trị K từ 2 đến 10
for k in range(2, 11):
    kmeans = KMeans(n_clusters=k, init='k-means++', random_state=42)
```

```
    kmeans.fit(X_scaled)
    wcss.append(kmeans.inertia_)  # inertia_ = WCSS
    silhouette_scores.append(silhouette_score(X_scaled, kmeans.labels_))
# Biểu đồ Elbow Method
plt.figure(figsize=(6,5))
plt.plot(range(2, 11), wcss, marker='o')
plt.title('Elbow Method for Optimal K')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('WCSS')
plt.show()
# Biểu đồ Silhouette Score
plt.figure(figsize=(6,5))
plt.plot(range(2, 11), silhouette_scores, marker='o', color='orange')
plt.title('Silhouette Score for Different K')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('Silhouette Score')
plt.show()
```

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++)?

- Phương pháp K-Means++ là một thuật toán khởi tạo thông minh được thiết kế đặc biệt để giải quyết vấn đề này. Thay vì chọn các tâm cụm ngẫu nhiên hoàn toàn, K-Means++ chọn các tâm cụm ban đầu một cách có chiến lược.
- Cách hoạt động của K-means++:
    + **Chọn tâm cụm đầu tiên:** Chọn ngẫu nhiên một điểm dữ liệu bất kỳ làm tâm cụm đầu tiên.
    + **Tính khoảng cách:** Đối với mỗi điểm dữ liệu còn lại, tính bình phương khoảng cách gần nhất của nó đến một tâm cụm đã được chọn.
    + **Chọn tâm cụm tiếp theo:** Chọn điểm dữ liệu tiếp theo làm tâm cụm mới với xác suất tỉ lệ thuận với bình phương khoảng cách của nó. Điều này đảm bảo rằng các điểm nằm ở xa các tâm cụm hiện tại có khả năng được chọn làm tâm cụm tiếp theo cao hơn.
    + **Lặp lại:** Lặp lại các bước 2 và 3 cho đến khi chọn đủ K tâm cụm.
- Một cách tiếp cận đơn giản nhưng hiệu quả khác là chạy thuật toán K-Means nhiều lần với các điểm khởi tạo ngẫu nhiên khác nhau.
- Để có kết quả tốt nhất, ta có thể kết hợp cả hai phương pháp trên. Bắt đầu bằng cách khởi tạo bằng K-Means++, sau đó chạy thuật toán một vài lần để đảm bảo tính ổn định và chọn kết quả tốt nhất.

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)?