TÌM HIỂU THUẬT TOÁN K - MEANS VÀ K - MEANS++

Trần Minh Hảia\*

a Khoa Công nghệ Thông tin, Trường Đại học Ngoại ngữ - Tin học Thành phố Hồ Chí Minh

\* Tác giả liên hệ: Email: 19dh110032@st.huflit.edu.vn | Điện thoại: 039.4862.240

**Tóm tắt**

*K-means là một thuật toán phân cụm đơn giản thuộc loại học không giám sát (tức là dữ liệu không có nhãn) và được sử dụng để giải quyết bài toán phân cụm. Ý tưởng đơn giản nhất về cluster (cụm) là tập hợp các điểm*ở gần nhau trong một không gian nào đó*(không gian này có thể có rất nhiều chiều trong trường hợp thông tin về một điểm dữ liệu là rất lớn). Trong bài viết này tôi sẽ giải quyết bài toán trong không gian 2 chiều bằng cách phân tích toán học đằng sau thuật toán K-means, áp dụng các công thức tính khoảng cách của các vector với số K-cluster đã được cho trước trong không gian 2 chiều, sử dụng ngôn ngữ lập trình Python và thư viện Pygame của Python để demo thuật toán này.Sau đó sẽ ứng dụng trong không gian 3 chiều là nén ảnh. K-means có một nhược điểm khá lớn đó là khởi tạo các trọng tâm. Tôi sẽ sử dụng K-means++ để thuật toán có thể đạt được kết quả phân cụm cao nhất.*

**Từ khoá:** K-means clustering; K-means++; Python; Pygame.

1. giới thiệu

Trong các mô hình kinh doanh, doanh nghiệp sẽ chia nhỏ tập khách hàng ra thành những nhóm đối tượng khác nhau để có thể áp dụng những chiến lược kinh doanh cụ thể cho từng nhóm đối tượng. Điều này giúp cho khách hàng được tiếp cận với các sản phẩm thật sự phù hợp với bản thân họ. Sự phù hợp đó sẽ kéo doanh số của chúng ta tăng lên. Vấn đề đặt ra là làm sao có thể chia nhỏ tập khách hàng đó ra khi mà số lượng hóa đơn là rất lớn và chúng ta không thể ngồi để phân tích từng vị khách.

Và mục tiêu của các thuật toán phân cụm là từ tập dữ liệu khổng lồ đó. Làm sao chúng ta biết có những nhóm dữ liệu đặc trưng nào trong đó? Từng dữ liệu trong đó thuộc vào nhóm nào? Đó là cái mà thuật toán phân cụm của chúng ta cần đi tìm câu trả lời.

Để giải quyết bài toán phân cụm dữ liệu này, tôi sẽ xây dựng mô hình các điểm dữ liệu trong không gian hai chiều. Mỗi điểm dữ liệu tượng trưng cho một khách hàng, mỗi k cluster đại diện cho mỗi nhóm khách hàng. Sau đó áp dụng công thức tính khoảng cách của các điểm dữ liệu (là một vector trong không gian) đến các k cluster. Ngoài ra để thuật toán đạt được kết quả phân cụm tốt nhất thì tôi sẽ sử dụng thuật toán K-means++ để có thể phân cụm k cluster một cách chính xác hơn.

1. Các công trình Nghiên cứu liên quan
   1. Toán học đằng sau thuật toán K-means
      1. Tổng quan

Thuật toán K-means [1] là phương pháp lượng tử hóa vector, có nguồn gốc từ xử lý tín hiệu số. Mục tiêu của thuật toán là phân hoạch n quan sát thành k cụm mà trong đó mỗi quan sát sẽ thuộc về một cụm có trung bình gần nhất (gọi là cluster centers hay cluster centroid)

Mục đích cuối cùng của thuật toán phân nhóm này là: từ dữ liệu đầu vào và số lượng nhóm chúng ta muốn tìm, hãy chỉ ra center của mỗi nhóm và phân các điểm dữ liệu vào các nhóm tương ứng. Giả sử thêm rằng mỗi điểm dữ liệu chỉ thuộc vào đúng một nhóm.

Giả sử có **N** điểm dữ liệu là  và K < N là số cluster chúng ta muốn phân chia. Chúng ta cần tìm các center  và label của mỗi điểm dữ liệu.[2]

Với mỗi điểm dữ liệu  đặt  là label vector của nó, trong đó nếu được phân vào cluster**k** thì  và , . Điều này có nghĩa là có đúng một phần tử của vector  là bằng 1 (tương ứng với cluster của ), các phần tử còn lại bằng 0. Ví dụ: nếu một điểm dữ liệu có label vector là  thì nó thuộc vào cluster 1, là  thì nó thuộc vào cluster 2, ……. Cách mã hóa label của dữ liệu như thế này được goi là biểu diễn one hot.

Ràng buộc của  có thể viết dưới dạng toán học như sau:

* + 1. Hàm mất mát và bài toán tối ưu

Mục tiêu là giảm thiểu tổng trong cụm bình phương (phương sai) [2], còn được gọi là sai số bình phương, trong đó mỗi sai số bình phương của một quan sát từ nhóm được xác định bởi:

Từ phương trình 1, tổng tất cả các phần tử trong một vectơ nhãn bằng 1. Sai số bình phương của một quan sát là:

Sai số toàn bộ dữ liệu sẽ là:

Trong đó , lần lượt là các ma trận được tạo bởi label vector của mỗi điểm dữ liệu và center của mỗi cluster. Hàm số mất mát trong bài toán K-means clustering của chúng ta là hàm **f(Y,M)** với ràng buộc như được nêu trong phương trình (1).

Tóm lại, chúng ta cần tối ưu bài toán sau:

* + 1. Giải quyết các vấn đề tối ưu hóa

Có 2 biến trong phương trình (5) là tâm của mỗi nhóm quan sát và vector nhãn của mỗi quan sát. Vấn đề có thể được giải quyết bằng cách sửa từng biến sau đó giảm thiểu biến kia [2]

* + 1. Cố định M, tìm Y

Giả sử đã tìm được các centers, hãy tìm các label vector để hàm mất mát đạt giá trị nhỏ nhất. Điều này tương đương với việc tìm cluster cho mỗi điểm dữ liệu.

Khi các centers là cố định, bài toán tìm label vector cho toàn bộ dữ liệu có thể được chia nhỏ thành bài toán tìm label vector cho từng điểm dữ liệu xi như sau:

Vì chỉ có một phần tử của label vector yi bằng 1 nên bài toán (6) có thể tiếp tục được viết dưới dạng đơn giản hơn:

Vì chính là bình phương khoảng cách tính từ điểm xi tới center mj, ta có thể kết luận rằng mỗi điểm xi thuộc vào cluster có center gần nó nhất! Từ đó ta có thể dễ dàng suy ra lable vector của từng điểm dữ liệu.

* + 1. Cố định Y, tìm M

**Giả sử đã tìm được cluster cho từng điểm, hãy tìm center mới cho mỗi cluster để hàm mất mát đạt giá trị nhỏ nhất.**

Một khi chúng ta đã xác định được label vector cho từng điểm dữ liệu, bài toán tìm center cho mỗi cluster được rút gọn thành:

Phương trình (7) là một đàm hàm lồi có thể phân biệt với mỗi . Do đó ta có thể giải phương trình (7) bằng cách tìm góc của đạo hàm riêng.

Giả sử : = ta có đạo hàm:

Giải phương trình đạo hàm ta có:

Suy ra:

Tử số trong công thức (9) là tổng các điểm của cluster j, còn mẫu số là số lượng các điểm trong cluster j. Tóm lại là trung bình cộng của các điểm trong cluster j.

* 1. Toán học đằng sau thuật toán K-means++
     1. Tổng quan

Thuật toán K-means bắt đầu với một tập hợp các trung tâm cụm tùy ý. David Arthur và Sergei Vassilvitskii [3] đã đề xuất một cách cụ thể để lựa chọn các trung tâm này tại bất kỳ thời gian nào, gọi là khoảng cách ngắn nhất từ điểm dữ liệu x đến trung tâm gần nhất đã được tạo ra. Được gọi là thuật toán K-means++ [3]. K-means++ có thể được hiểu là thuật toán K-means tiêu chuẩn kết hợp với việc khởi tạo center thông minh hơn.

* + 1. Xác xuất chọn trung tâm

Các bước khởi tạo thuật toán K-means++:

* Chọn ngẫu nhiên một center c1 ban đầu đồng đều từ tập dữ liệu x
* Tính khoảng cách của tất cả các điểm trong tập dữ liệu từ trung tâm đã chọn. Khoảng cách của điểm từ tâm xa nhất có thể được tính bằng:

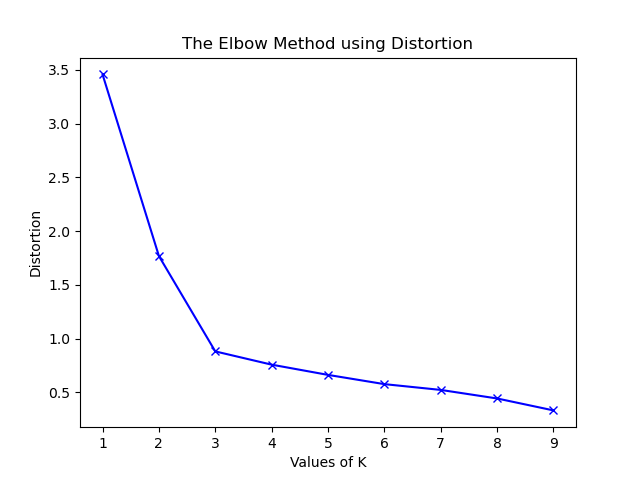
* Chọn center tiếp theo, chọn với xác suất:

* Lặp lại các bước trên cho đến khi tìm được k center

1. Cách chọn K bằng phương pháp elbow và silhouette
   1. Phương pháp elbow

Số lượng cụm (k) là tham số quan trọng trong phân nhóm K-means và K-means++. Thông thường trên không gian 2 chiều chúng ta có thể dễ dàng nhìn thấy số lượng cụm và chọn k một cách hợp lý, nhưng với những dạng bài toán được biểu diễn trên không gian 2 chiều với tập dữ liệu lớn thì ta không thể chọn k bằng tay được, phương pháp elbow [4] (còn được gọi là khuỷu tay) sẽ giúp chúng ta xác định được số k chính xác.

Trong phương pháp này, tôi sẽ chọn phạm vi của giá trị k, sau đó áp dụng thuật toán K-means và sử dụng từng giá trị k. Tìm khoảng cách trung bình của mỗi điểm trong một cụm đến tâm của nó và biểu diễn trên biểu đồ.



1. Minh họa biểu đồ Elbow

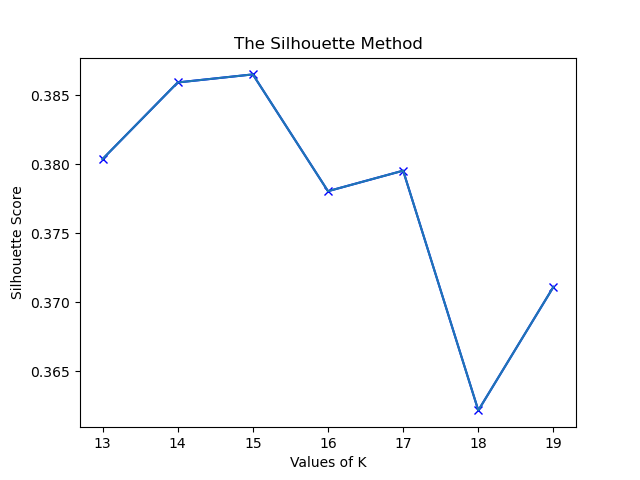
Với sự gia tăng của số lượng cụm (k) , khoảng cách trung bình giảm. Để tìm được cụm (k) tối ưu hãy quan sát biểu đồ vị trị mà độ dốc được bẻ cong ( hình dạng như khuỷu tay) là số cụm tối ưu, trong biểu đồ trên số cụm tối ưu là 3.

* 1. Phương pháp silhouette

Về dữ liệu 2 chiều, phương pháp Elbow rất thích hợp đề tìm k, nhưng đối với những bài toán có nhiều chiều hơn 2, thì Elbow không phải là phương pháp tốt nhất để tìm k cụm, thay vào đó ta sẽ sử dụng phương pháp silhouette [4]

Silhouette là một thước đo về cách một thuật toán phân cụm đã hoạt động. Sau khi tính toán hệ số hình bóng của mỗi điểm trong tập dữ liệu, hãy vẽ biểu đồ đó để có được một hình ảnh trực quan về mức độ tốt của tập dữ liệu được nhóm thành k cụm. Biểu đồ hình bóng hiển thị thước đo mức độ gần của mỗi điểm trong một cụm với các điểm trong các cụm lân cận và do đó cung cấp một cách để đánh giá các thông số như số lượng cụm một cách trực quan. Số đo này có phạm vi là [-1, 1], trong đó giá trị cao chỉ ra rằng đối tượng được đối sánh tốt với cụm của chính nó và đối sánh kém với các cụm lân cận. Nếu hầu hết các đối tượng có giá trị cao, thì cấu hình phân cụm là thích hợp. Nếu nhiều điểm có giá trị thấp hoặc âm, thì cấu hình phân cụm có thể có quá nhiều hoặc quá ít cụm.

Trong biểu đồ dưới đây tôi đã áp dụng vào bài toán nén ảnh khối u ác tính ở não người (ở mục 6) để tìm k. Số cụm mà tôi chọn sẽ từ 13 đến 19, trong khi tiến hành thực nghiệm thì đây là các số cụm cho ra kết quả khả quan nhất về hình ảnh khối u tức là có thể thấy rõ được hình ảnh của khối u. Vậy bài toán ở đây sẽ tìm ra số k thích hợp nhất để bài toán nén ảnh được giữ lại thông tin dữ liệu cũng như giảm kích thước một cách hiệu quả hơn



1. Minh họa biểu đồ Silhouette cho ảnh khối u ác tính ở người

Nhìn vào biểu đồ ta sẽ chọn k = 15 tương ứng với giá trị silhouette score cao nhất để có một ảnh nén giữ được thông tin tốt cũng như giảm được kích thước của hình ảnh và giảm thời gian chạy thuật toán tốt nhất

1. So sánh thuật toán k-means và k-means++
   1. Tổng quan

Mặt hạn chế của K-means tiêu chuẩn: Một nhược điểm của thuật toán K-means là nó nhạy cảm với việc khởi tạo các trọng tâm hoặc các điểm trung bình. Vì vậy, nếu một center được khởi tạo là một điểm "xa", nó có thể chỉ kết thúc không có điểm nào được liên kết với nó và đồng thời, nhiều hơn một cụm có thể kết thúc với một center duy nhất. Tương tự, nhiều hơn một center có thể được khởi tạo vào cùng một cụm dẫn đến phân cụm kém

Để khắc phục nhược điểm nêu trên, tôi sử dụng K-means++ [3]. Thuật toán này đảm bảo khởi tạo center thông minh hơn và cải thiện chất lượng của phân cụm. Ngoài phần khởi tạo, phần còn lại của thuật toán giống như thuật toán K-means tiêu chuẩn.

* 1. So sánh các bước khởi tạo

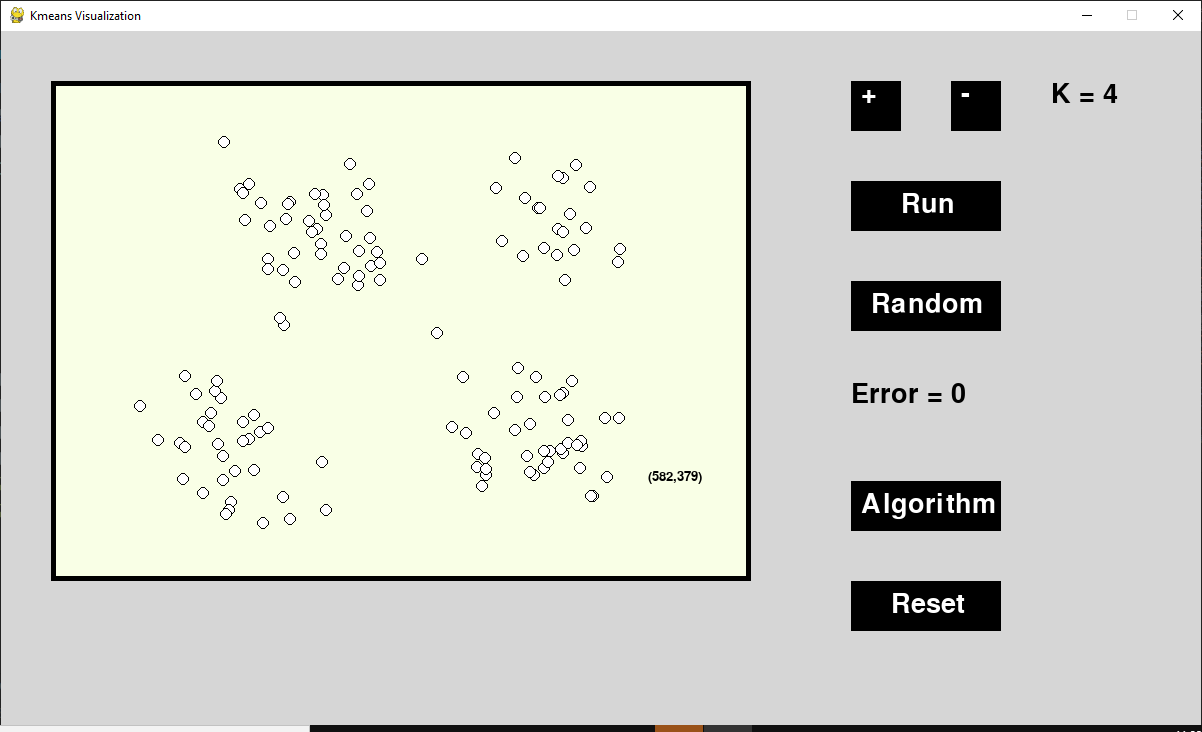
1. Các bước khởi tạo thuật toán

|  |  |
| --- | --- |
| **K - means** | **K - means++** |
| Bước 1: Tạo các điểm dữ liệu  Bước 2: Chọn ngẫu nhiên k cluster trong các điểm dữ liệu  Bước 3: Đối với mỗi điểm dữ liệu, hãy tìm khoảng cách giữa điểm đó với các cluster. Điểm sẽ được gán lable cho cụm có tâm gần nhất  Bước 4: Cập nhật giá trị và vị trí của Cluster với giá trị trung bình mới  Bước 5: Lặp lại các bước 3 và 4 khi nào đạt được giá trị trung bình nhỏ nhất giữa các điểm và cluster | Bước 1 : Tạo các điểm dữ liệu như thuật toán K-means tiêu chuẩn  Bước 2: Chọn ngẫu nhiên các center đầu tiên từ các điểm dữ liệu  Bước 3: Đối với mỗi điểm dữ liệu, tính toán khoảng cách của nó từ trung tâm gần nhất, đã chọn trước đó.  Bước 4: Chọn center tiếp theo từ các điểm dữ liệu sao cho xác suất chọn một điểm làm center tỷ lệ thuận với khoảng cách của nó từ center gần nhất, đã chọn trước đó. (nghĩa là điểm có khoảng cách tối đa từ tâm gần nhất có nhiều khả năng được chọn tiếp theo làm tâm)  Bước 5: Lặp lại bước 3 và 4 cho đến khi lấy mẫu k center |

1. Thực hiện thuật toán bằng ngôn ngữ python
   1. Thực hiện thuật toán K-Means

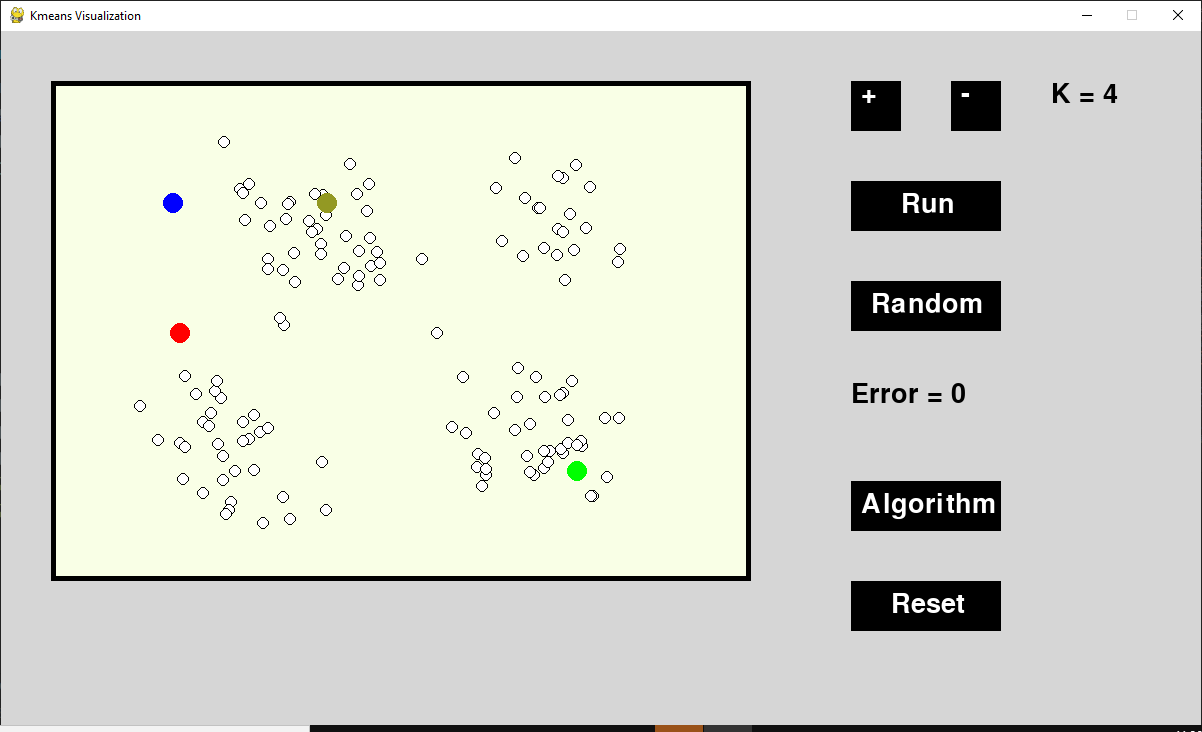
Trong bản demo này, K-means visualization được viết dựa trên ngôn ngữ python và sử dụng thư viện Pygame để vẽ lên các điểm dữ liệu và phân cụm chúng.

Bước 1: Tạo điểm dữ liệu lên bảng như ảnh 1 và chọn số K ( tức là số nhóm muốn phân cụm) bên góc phải của ảnh.



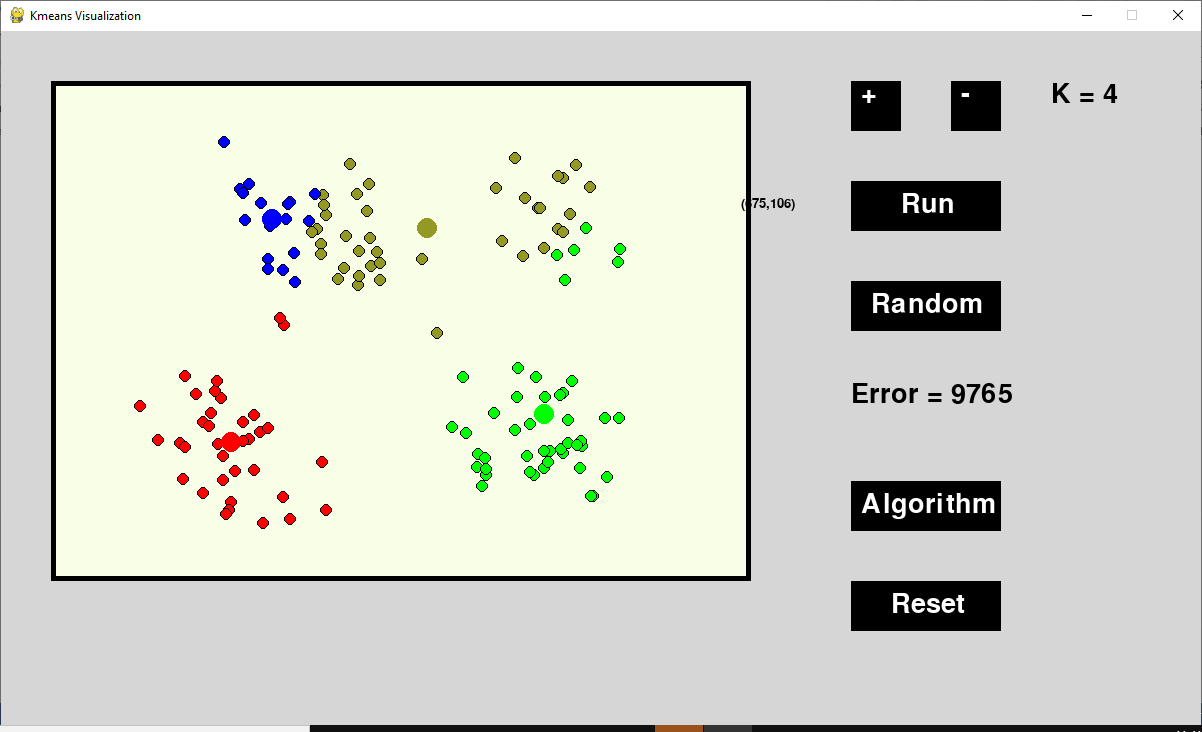
1. Tạo điểm dữ liệu và số K

Bước 2: Random các Cluster

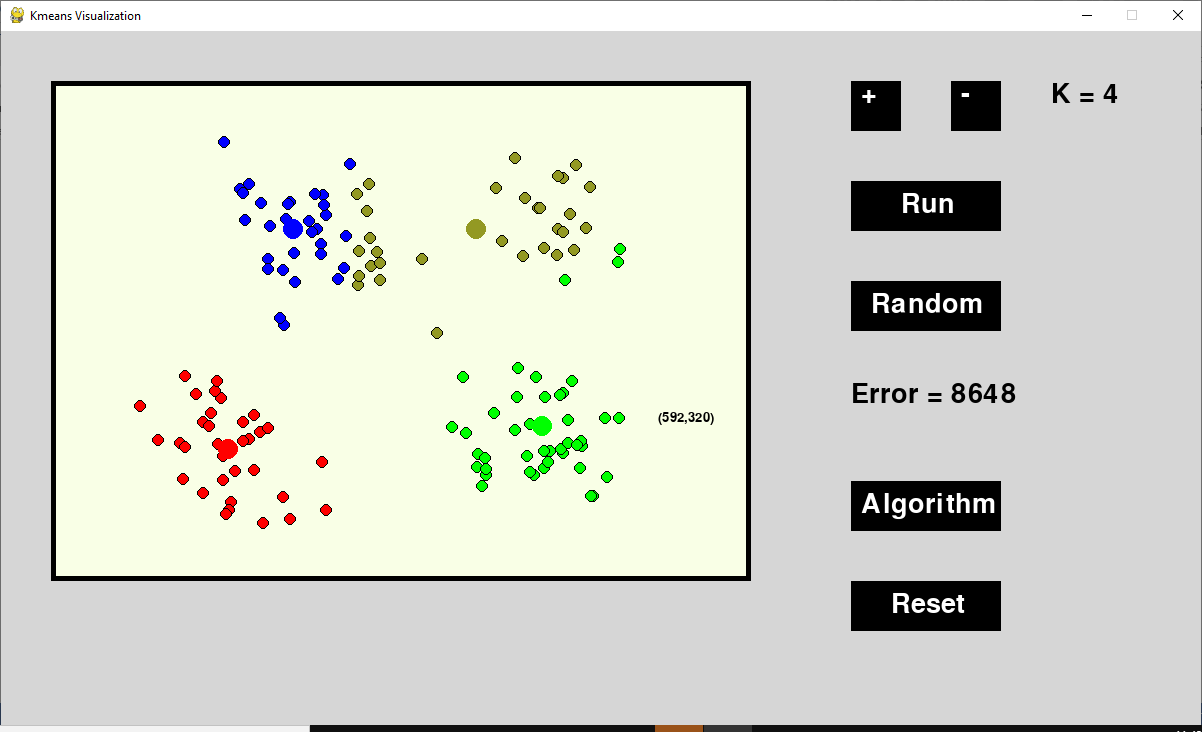


1. Random cluster

Bước 3: Chạy thuật toán cho tới khi các cụm được phân ra K cụm ( tức là 4 cụm) và các Clusters sẽ cập nhật lại ví trí của mình sao cho ở chính giữa các cụm.

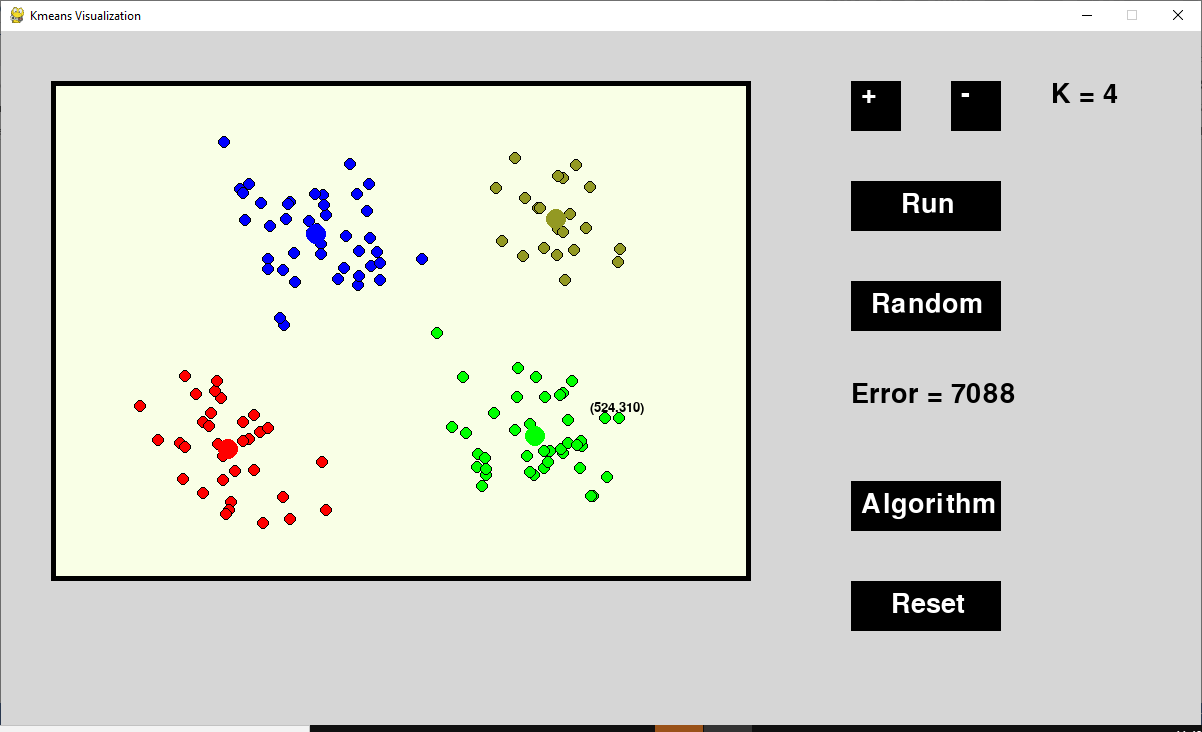


1. Lần chạy thứ 1



1. Lần chạy thứ 2

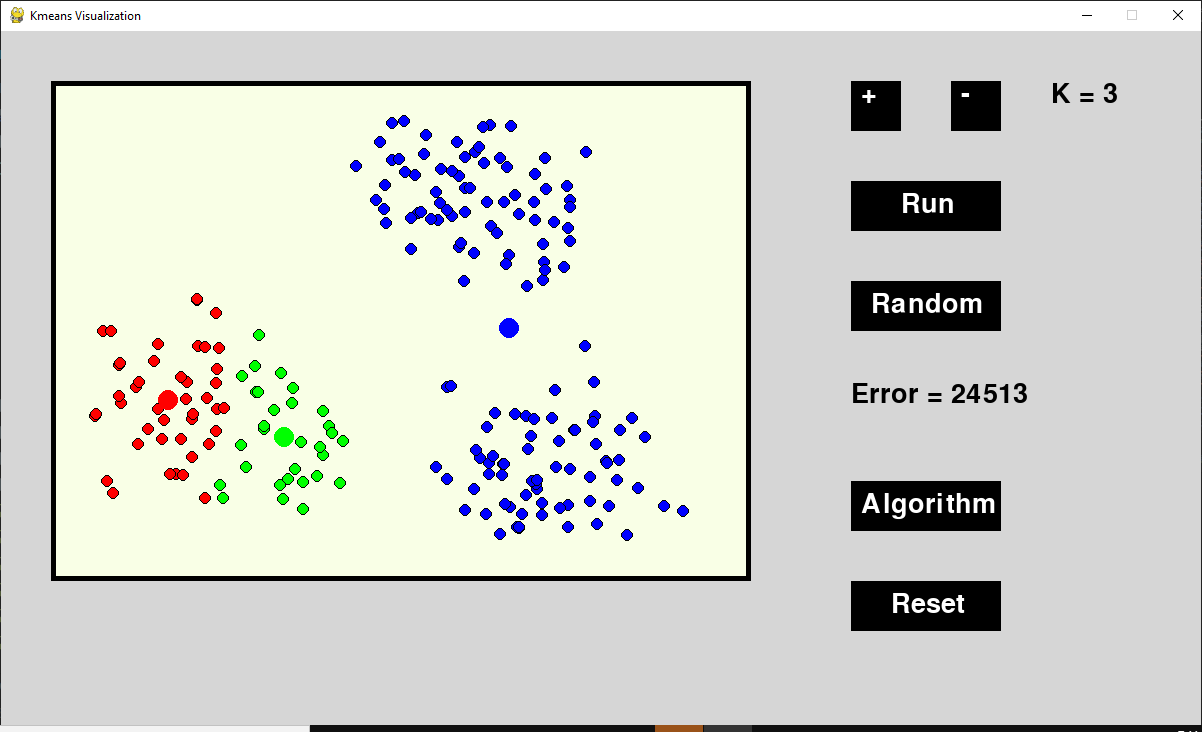
Sau 2 lần chạy thì “Error” trong lần chạy thứ 2 đã giảm đi so với lần chạy thứ nhất, vậy “Error” ở đây chính là trung bình cộng của tất cả các điểm dữ liệu tới Clusters, “Error” càng nhỏ thì thuận toán phân cụm càng chính xác hơn



1. Lần chạy thứ 3

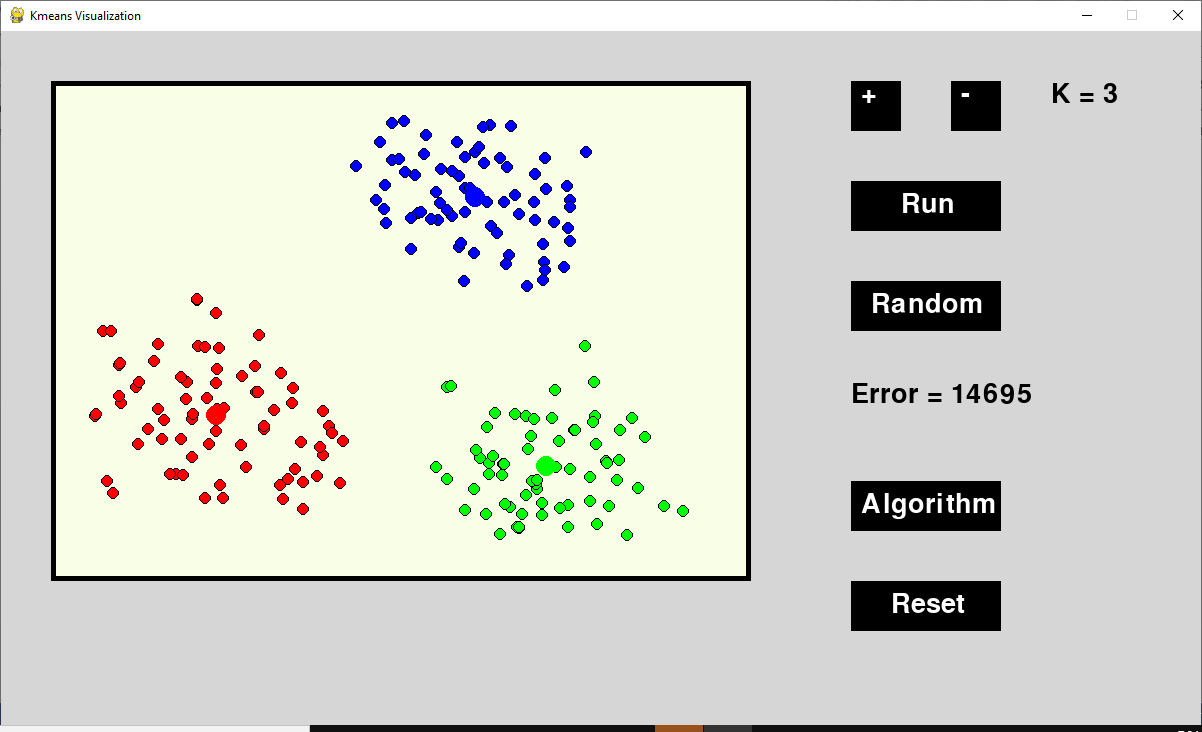
Lần chạy thứ 3 cũng là lần chạy cuối cùng, ta thấy đã phân được 4 cụm khác nhau trong hình thứ 5

Nhược điểm khi chạy thuật toán k-means là khởi tạo trung tâm kém dẫn tới việc phân cụm kém:



1. Phân cụm sai

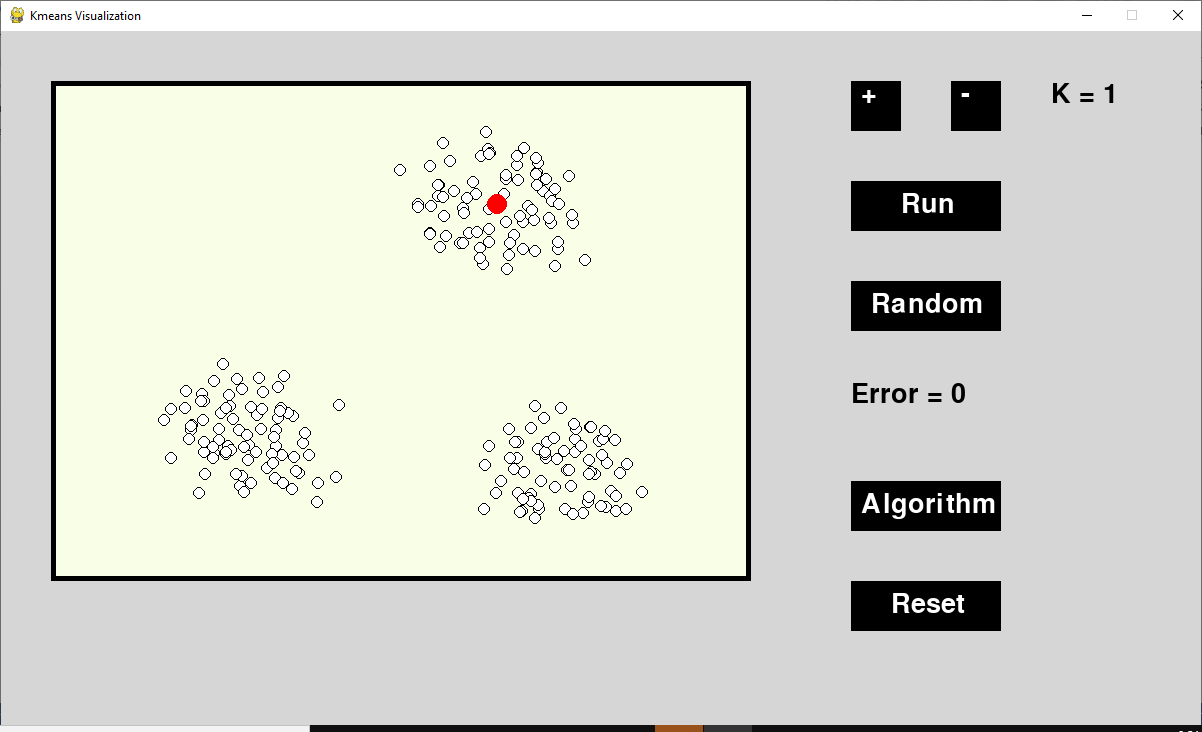
Và đây là cách phân cụm chính xác:



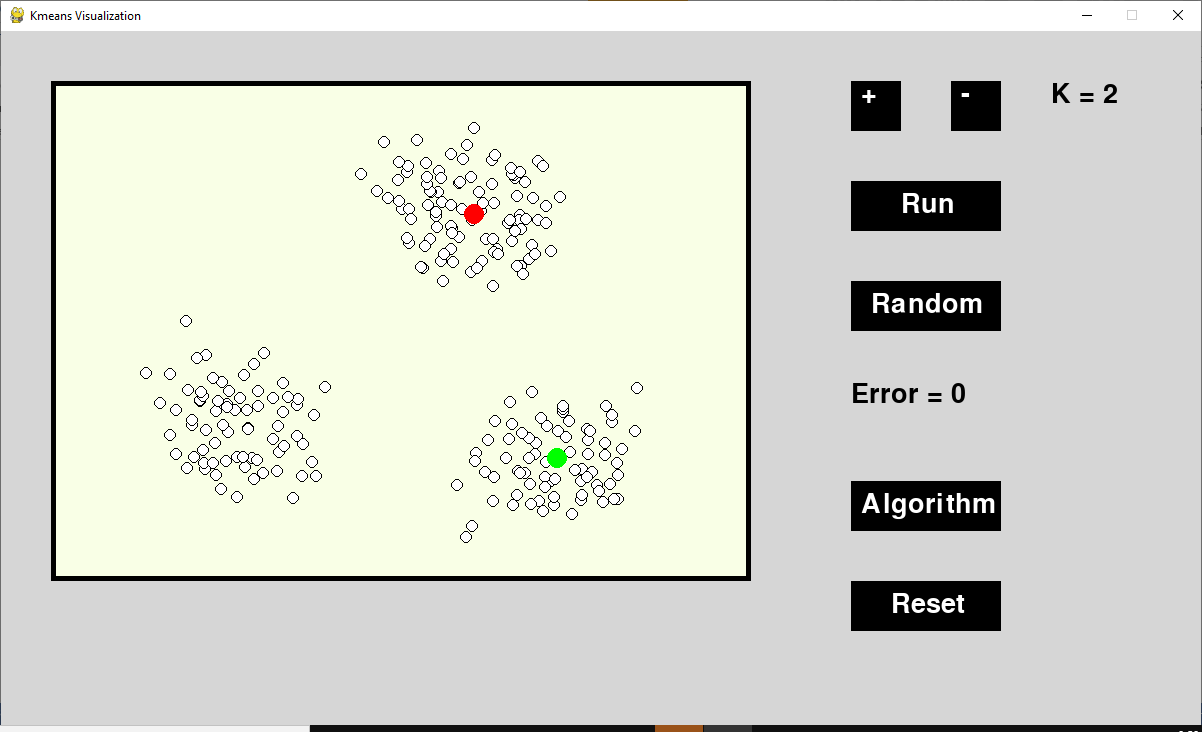
1. Phân cụm chính xác
   1. Thực hiện thuật toán K-Means ++

Để khắc phục được nhược điểm của thuật toán của K-means tiêu chuẩn, K-means ++ [3] sẽ khởi tạo các center thông minh hơn

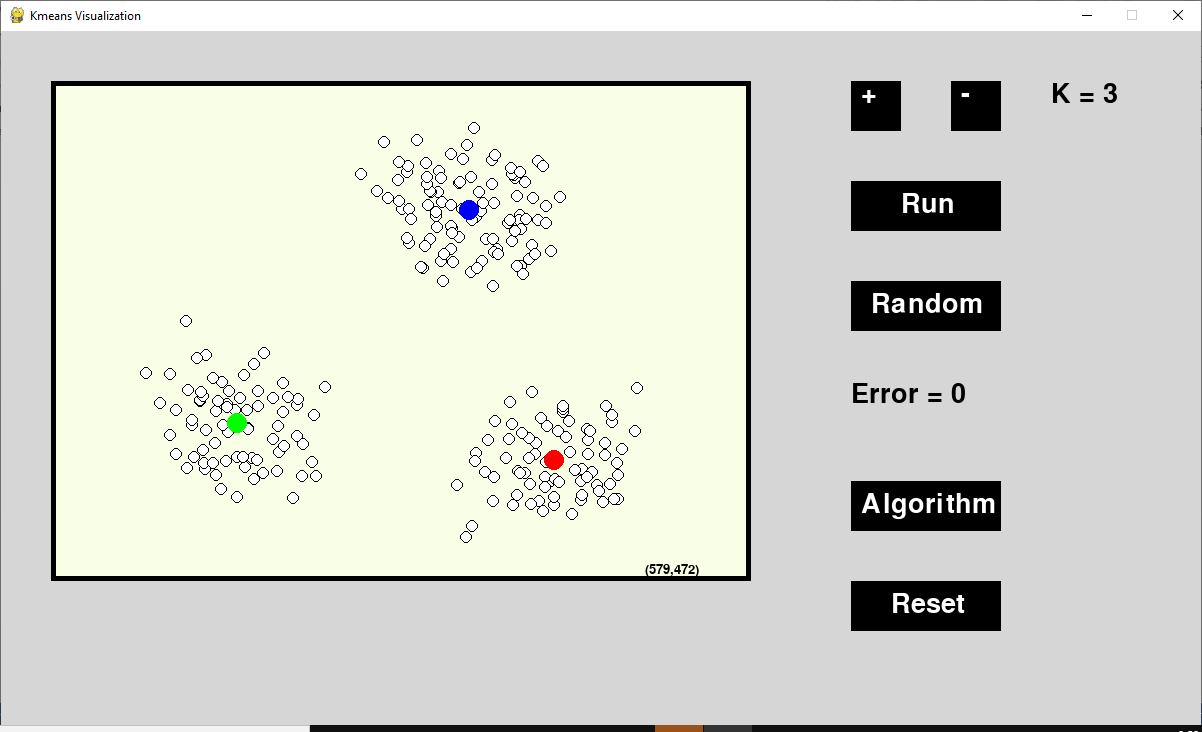
Đây là cách K-means ++ hoạt động:



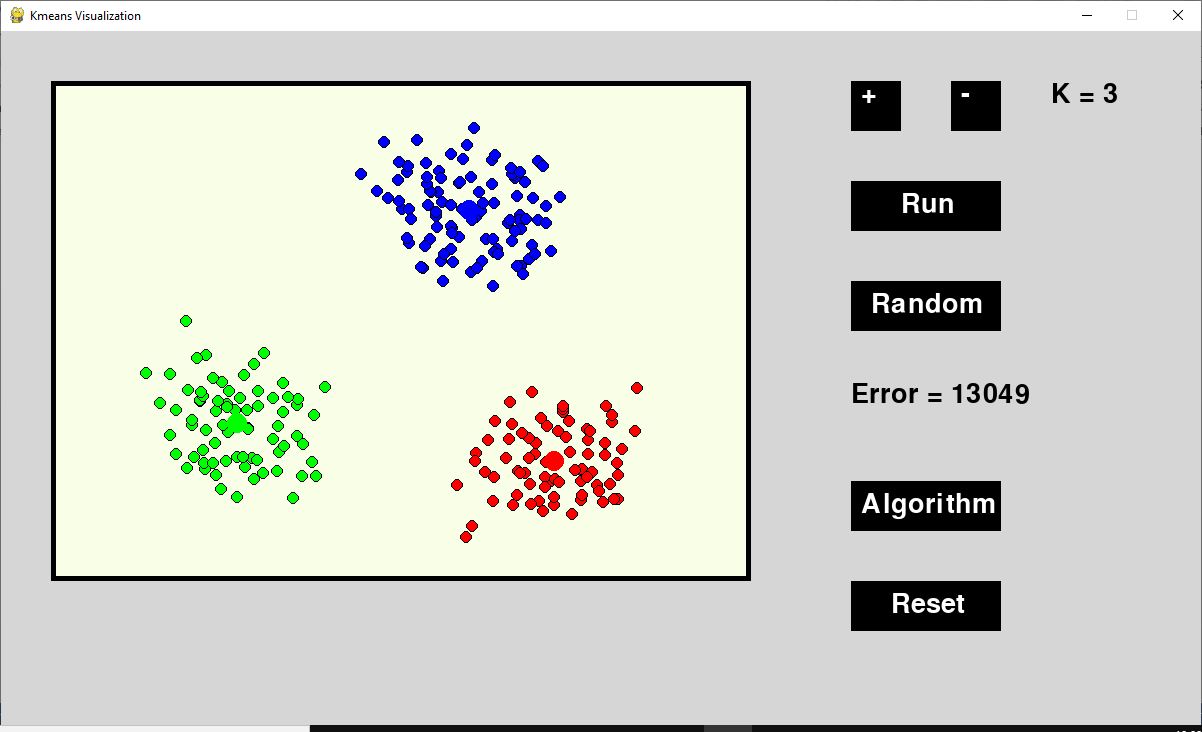
1. Lần chọn center thứ 1



1. Lần chọn center thứ 2



1. Lần chọn center thứ 3



1. Hoàn thành phân cụm

Mặc dù việc khởi tạo trong K-means ++ đắt hơn về mặt tính toán so với thuật toán K-mean tiêu chuẩn, nhưng thời gian chạy để hội tụ đến mức tối ưu đã giảm đáng kể đối với K-means ++. Điều này là do các trung tâm được chọn ban đầu có khả năng đã nằm trong các cụm khác nhau.

1. ỨNg dụng nén hình ảnh

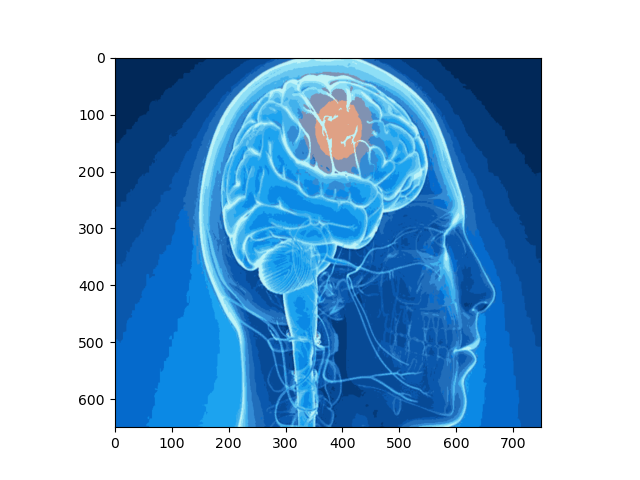
Nén hình ảnh có thể được thực hiện bằng cách loại bỏ dư thừa trong hình ảnh để giảm chi phí lưu trữ, truyền tải và xử lý. K-means và K-means++ thường được sử dụng để thực hiện phân loại hình ảnh hoặc để giảm thiểu kích thước của dữ liệu và nó đã được mô tả trong bài viết của Xing Wan [5]

Trong khi xử lý ảnh, dữ liệu đầu vào sẽ là pixel và chúng sẽ được phân cụm dựa trên thuật toán K-means và K-means++. Ngoài ra ta sẽ so sánh được kết quả khác nhau của hai thuật toán này một cách rõ ràng hơn

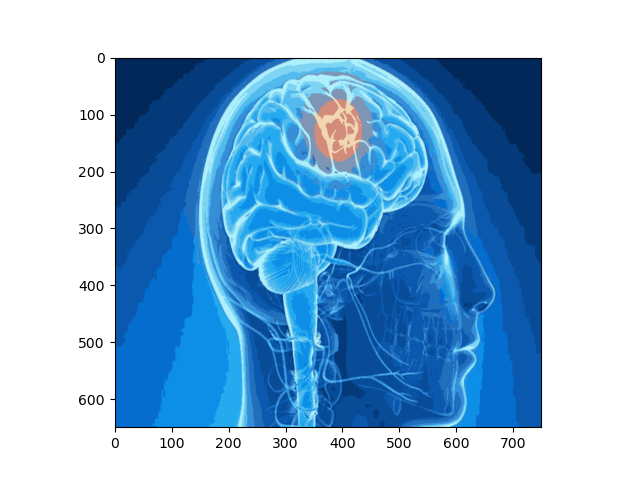
Để thực hiện nén ảnh, tôi đã sử dụng phương pháp Silhouette (ở mục 3.2) để tìm số k cho ảnh khối u ác tính ở người, và số k là 15 (hình 2). Tôi sử dụng Python viết thuật toán để thuận lợi trong việc nén ảnh và hiển thị hình ảnh.



1. Ảnh gốc khi chưa được nén



1. Ảnh đã được nén bằng thuật toán K-means



1. Ảnh đã được nén bằng thuật toán K-means++
2. So sánh thuật toán sau khi nén ảnh

|  |  |  |
| --- | --- | --- |
|  | K-Means | K-Means++ |
| Kích thước ảnh gốc | 486 (KB) | 486 (KB) |
| Kích thước ảnh sau khi nén | 164 (KB) | 160 (KB) |
| Thời gian chạy thuật toán | 53,57 (s) | 34,96 (s) |
| Hình ảnh cấu trúc của khối u | Không rõ | Rõ |

Từ hình 15,16 và sự so sánh của bảng 2 ta có thể thấy rõ kết quả khác nhau của hai thuật toán về mặt hình ảnh, kích thước và thời gian chạy của thuật toán. Về mặt hình thức nén ảnh thì cả hai thuật toán đã giảm được kích thước của hình ảnh, nhưng về thông tin hình ảnh thì thuật toán K-means++ giữ lại nhiều hơn thuật toán K-means tiêu chuẩn. Có thể thấy rõ hình dạng của khối u khi sử dụng thuật toán K-means++ còn K-means tiêu chuẩn thì không, thời gian chạy thuật toán cũng như kích thước của ảnh sau khi nén của thuật toán K-means++ vẫn tốt hơn

Việc triển khai thuật toán K-means và K-means++ đã được thực hiện thành công, khi sử dụng thuật toán K-means việc khởi tạo các trọng tấm rất quan trọng để có thể thu lại được những kết quả tốt, K-means++ sẽ khắc phục được nhược điểm của K-means

1. Kết Luận

Việc triển khai thuật toán K-means và K-means ++ vào việc phân cụm các dữ liệu đã được thực hiện thành công qua một không gian 2 chiều cũng như việc nén ảnh là một không gian 3 chiều. Các thuật toán phân cụm đã có kết quả tốt trong việc phân cụm các dữ liệu nhờ vào cách chọn k cụm qua các phương thức khác nhau phù hợp với mỗi bài toán. Qua đó ta cũng có thể so sánh được cách hoạt động của mỗi thuật toán và các phần toán học đằng sau nó

Tài liệu tham khảo

1. Forgy, Edward W.. "Cluster analysis of multivariate data: efficiency versus interpretability of classifications". Biometrics. 21 (3): 768–769. JSTOR 2528559, 1965.
2. Dung T. Lai Faculty of Science, Engineering and Technology Swinburne University of Technology. “ K-Means Algorithm and Application in Data Compression using Pascal and SwinGame API ” năm 2017.
3. Arthur, D.; Vassilvitskii, S. "k-means++: the advantages of careful seeding" (PDF). Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics Philadelphia, PA, USA. pp. 1027–1035, 2007.
4. Danny Matthew SAPUTRA, Daniel SAPUTRA, Liniyanti D. OSWARI. “Effect of Distance Metrics in Determining K-Value in KMeans Clustering Using Elbow and Silhouette Method 2019”
5. Wan, Xing. "Application of K-means Algorithm in Image Compression." IOP Conference Series: Materials Science and Engineering. Vol. 563. No. 5. IOP Publishing, 2019