**TRƯỜNG ĐẠI HỌC NGOẠI NGỮ - TIN HỌC TPHCM**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TÌM HIỂU GIẢI THUẬT GOM CỤM K-MEANS CỦA SCIKIT-LEARN ĐỂ GOM CỤM BỘ DỮ LIỆU MẪU GLASS**

**ĐỒ ÁN MÔN HỌC: TRÍ TUỆ NHÂN TẠO**

|  |  |  |
| --- | --- | --- |
| **Giảng viên hướng dẫn:** | **PGS.TS DƯƠNG TUẤN ANH** | |
| **Sinh viên thực hiện:** | **VÕ THÀNH HOÀNG SƠN** | **Mã số SV: 19DH110660** |
|  | **LA QUỐC ĐẠT** | **Mã số SV: 19DH111071** |
| **Lớp: KH1901** |  |  |
| **Khóa: 2019-2023** |  |  |

**TP.Hồ Chí Minh, tháng 11 năm 2021**

# Mục lục:

[***1. Giới thiệu***](#_c68yged205it) ***3***

[*1.1 Giới thiệu bài toán:*](#_ej9paesnojug) *3*

[*1.2 Giới thiệu về thuật toán sử dụng:*](#_jvn3bzgb790n) *3*

[*1.3 Giới thiệu nền tảng sử dụng:*](#_ku1a0ed04ufx) *4*

[*2.1 Tổng quan*](#_v6rj6qyn0b38) *4*

[*2.2 Hàm mất mát và bài toán tối ưu*](#_3408ubs8lrkb) *5*

[*2.3 Thuật toán tối ưu hàm mất mát:*](#_gnc1brb6qg19) *6*

[*2.4 Tìm M , Tìm Y*](#_743e1mbd829g) *6*

[***3. Các thuật toán và phương thức khác***](#_2tbwr67p3ggl) ***8***

[*3.1 Thuật toán giảm chiều dữ liệu PCA*](#_9h68w3l3v0w8) *8*

[*3.2 Phương pháp Elbow:*](#_lji9flox85a1) *9*

[*3.3 Phương pháp Silhouette*](#_3icpwmultltt) *10*

[***4. Áp dụng giải quyết vấn đề - Mã nguồn***](#_t03oaki0y22x) ***12***

[*4.1. Chuẩn bị vấn đề*](#_7gfqm4891fta) *12*

[*4.1.1. Đọc dữ liệu*](#_4o6iwwxpeb6s) *12*

[*4.1.2. Kiểm tra dữ liệu*](#_umo6l21n70ha) *14*

[4.2. Chuẩn bị dữ liệu](#_j8ww2orb86yu) 17

[*4.2.1. Hiển thị dữ liệu*](#_vhk4vix8swqi) *17*

[*4.2.2. Chuẩn hóa dữ liệu:*](#_jdjnf63qpjpi) *18*

[*4.3. Phân cụm*](#_rtk686uhjowg) *20*

[*4.3.1. Tìm số cụm (K)*](#_dq99t07cgd0n) *20*

[*4.3.2. Chạy thuật toán K-means*](#_99xzufew7z9o) *22*

[*4.3.3 Hiển thị đặc trưng các cụm*](#_fk4misogfmyr) *24*

[***5. Kết luận***](#_tufp2r78jnuj) ***25***

# Danh mục hình ảnh:

|  |  |  |
| --- | --- | --- |
| ***Hình 1:*** | *Minh hoạ phương pháp Elbow* | ***10*** |
| ***Hình 2:*** | *Minh hoạ phương pháp Silhouette* | ***12*** |
| ***Hình 3:*** | *Tóm tắt bộ dữ liệu* | ***13*** |
| ***Hình 4:*** | *5 dòng đầu, cuối của bộ dữ liệu* | ***14*** |
| ***Hình 5:*** | *Kiểm tra dữ liệu* | ***14*** |
| ***Hình 6:*** | *Bảng giá trị các thuộc tính* | ***15*** |
| ***Hình 7:*** | *Bảng mô tả độ tương quan các thuộc tính* | ***15*** |
| ***Hình 8:*** | *Heatmap mô tả độ tương quan các thuộc tính* | ***16*** |
| ***Hình 9:*** | *Hiển thị các tính chất dưới đồ thị histogram* | ***17*** |
| ***Hình 10:*** | *Sự phân bổ của dữ liệu* ***trước*** *khi chuẩn hoá* | ***18*** |
| ***Hình 11:*** | *Sự phân bổ của dữ liệu* ***sau*** *khi chuẩn hoá* | ***18*** |
| ***Hình 12:*** | *Tóm tắt dữ liệu sau khi giảm chiều* | ***19*** |
| ***Hình 13:*** | *Tìm K bằng thuật toán Silhouette* | ***20*** |
| ***Hình 14:*** | *Tìm K bằng thuật toán Elbow* | ***21*** |
| ***Hình 15:*** | *Mã nguồn sử dụng thuật toán KMeans* | ***22*** |
| ***Hình 16:*** | *Các centroid tìm được* | ***22*** |
| ***Hình 17:*** | *Các mẫu sau khi gom cụm* | ***23*** |
| ***Hình 18:*** | *Đặc trưng của từng cụm* | ***24*** |
|  |  |  |

# 1. Giới thiệu

## 1.1 Giới thiệu bài toán

Để hỗ trợ cho việc điều tra hiện trường của các vụ án, các điều tra viên cần phải thu thập và phân tích các mẫu kính tại hiện trường thành các nhóm khác nhau. Qua đó giúp việc điều tra trở nên thuận tiện và dễ dàng hơn.

Vì thế mục tiêu bài toán của chúng ta hôm nay là phân cụm các mẫu kính đã được thu thập. Vậy làm thế nào để chúng ta có thể phân cụm các dữ liệu? Làm thế nào để biết được các đặc trưng riêng biệt của từng loại dữ liệu?. Đó chính là những vấn đề chúng ta sẽ giải quyết mà chúng ta cần tìm câu trả lời.

## 1.2 Giới thiệu về thuật toán sử dụng

Bài toán Nhận dạng loại kính sẽ được thực hiện bằng cách sử dụng thuật toán gom cụm K-Means , một trong những thuật toán nổi tiếng nhất trong phương pháp phân cụm phân hoạch. Bên cạnh để bổ trợ cho quá trình thực hiện chúng em còn áp dụng các hàm Elbow ,Silhouette Score và thuật toán giảm chiều dữ liệu PCA.

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.

## 1.3 Giới thiệu nền tảng sử dụng

Tất cả các bước thực hiện phân tích, xử lý, chạy thuật toán đều được thực hiện trên Jupyter Lab nhằm mục đích dễ dàng quản lý và kiểm soát các bước thực hiện, toàn bộ các bước sử dụng ngôn ngữ Python. Thư viện Scikit-Learn là nền tảng chính trong việc giải quyết bài toán hôm nay. Một vài thư viện khác cần dùng là Pandas, Numpy - giúp chúng ta xử lý và quản lý bộ dữ liệu, tiếp theo là thư viện, Matplotlib và Seaborn để thuận tiện trong việc trực quan hoá các dữ liệu.

2.Thuật toán K-Means

## 2.1 Tổng quan

K-Means là một thuật toán phân cụm đơn giản thuộc loại không giám sát (Unsupervised Learning) được sử dụng giải quyết bài toán phân cụm nhằm chia bộ dữ liệu thành các cụm khác nhau , mỗi cụm đều mang một tính chất chung nhất định nào đó và các điểm trong cụm đều liên quan đến nhau, đối với máy tính thì các điểm trong cụm phải gần nhau .

Tập dữ liệu sẽ được phân cụm theo số lượng K cho trước và để đo khoảng cách giữa các điểm ta sử dụng độ đo Euclid .

Giải thuật K-Means được mô tả gồm các bước sau đây :

**Bước 1**: Chọn ngẫu nhiên k mẫu trong số n mẫu để làm các trung tâm cụm đầu tiên. Gán mỗi mẫu trong số n - k mẫu đến k cụm theo nguyên tắc mỗi mẫu được gắn vào cụm có trung tâm cụm gần với nó nhất.

**Bước 2**: Tính các trung tâm cụm theo các cụm mới hình thành sau bước gán các mẫu nêu trên.

**Bước 3**: Gán mỗi mẫu trong tập n mẫu đến cụm mà có trung tâm cụm gần nó nhất.

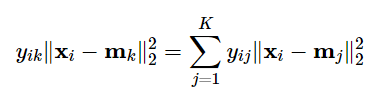
**Bước 4**; Nếu không có sự thay đổi khi gán mẫu vào các cụm giữa hai lượt lặp kế tiếp nhau thì ta dừng giải thuật. Ngược lại thì quay lại bước 2.

## 2.2 Hàm mất mát và bài toán tối ưu

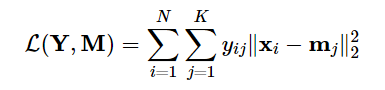
Nếu ta coi center mk là center (hoặc representative) của mỗi cluster và ước lượng tất cả các điểm được phân chia vào cluster này bởi mk, thì một điểm dữ liệu xi được phân vào cluster k sẽ bị sai số là (xi - mk). Chúng ta mong muốn sai số này có trị tuyệt đối nhỏ nhất nên ta sẽ tìm cách để đại lượng sau đây đạt giá trị nhỏ nhất:



Hơn nữa, vì xi được phân vào cluster k nên yik = 1, yij = 0, với mọi j khác k. Khi đó, biểu thức bên trên sẽ được viết lại là:

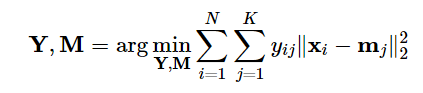


Sai số cho toàn bộ dữ liệu sẽ là:



Trong đó Y = [y1;y2;...;yN], M = [m1,m2,...mK] 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 L(Y,M) với ràng buộc như được nên trong phương trình .

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



## 2.3 Thuật toán tối ưu hàm mất mát

Bài toán trên là một bài toán khó tìm điểm tối ưu vì nó có thêm các điều kiện ràng buộc. Bài toán này thuộc loại mix-integer programming(điều kiện biến là số nguyên)- là loại rất khó tìm nghiệm tối ưu toàn cục ( global optimal point, tức nghiệm làm cho hàm mất mát đạt giá trị nhỏ nhất có thể). Tuy nhiên, trong một số trường hợp chúng ta vẫn có thể tìm được phương pháp để tìm được nghiệm gần đúng hoặc điểm cực tiểu.

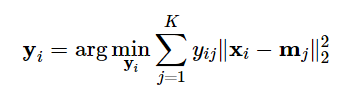
Một cách đơn giản để giải bài toán trên là xen kẽ giải Y và M khi biến còn lại được cố định. Đây là một thuật toán lặp, cung là kỹ thuật phổ biến khi giải bài toán tối ưu.

## 2.4 Tìm M , Tìm Y

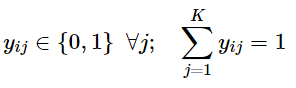
* Cố định M, tìm Y:

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.

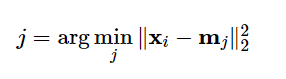
Khi các centers là cố định, bài toán tìm label vector cho toàn bộ dữ liệu có thể được chia nhỏ thành bài toán tìm label vector cho từng điểm dữ liệu xi như sau:



Với điều kiện:



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

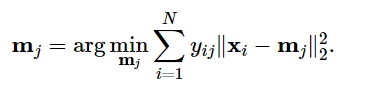


Vì ||xi - mj||22 chính là bình phương khoảng cách tính từ điểm xi tới center mj, ta có thể kết luận rằng mỗi điểm xi thuộc vào cluster có center gần nó nhất. Từ đó ta có thể dễ dàng suy ra label vector của từng điểm dữ liệu .

* Cố định Y, Tìm M

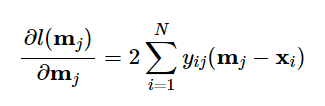
Giả sử đã tìm được cluster cho từng điểm, hãy tìm center mới cho mỗi cluster để hàm mất mát đạt giá trị nhỏ nhất.

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:

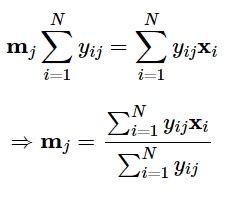


Tới đây, ta có thể tìm nghiệm bằng phương pháp giải đạo hàm bằng 0, vì hàm cần tối ưu là một hàm liên tục và có đạo hàm xác định tại mọi điểm. *Và quan trọng hơn, hàm này là hàm convex (lồi) theo* **m**j *nên chúng ta sẽ tìm được giá trị nhỏ nhất và điểm tối ưu tương ứng.*

Đặt l(mj)là hàm bên trong dấu arg min, ta có đạo hàm:



Giải phương trình đạo hàm bằng 0 ta có:



Nếu để ý một chút, chúng ta sẽ thấy rằng mẫu số chính là phép đếm số lượng các điểm dữ liệu trong cluster *j*. Còn tử số chính là tổng các điểm dữ liệu trong cluster *j*. Hay nói một cách đơn giản hơn : **m**j là trung bình cộng của các điểm trong cluster *j.*

# 3. Các thuật toán và phương thức khác

## 3.1 Thuật toán giảm chiều dữ liệu PCA

Để giúp cho việc trực quan hoá các cụm lên đồ thị 2 chiều chúng ta cần giảm chiều của bộ dữ liệu từ 9 chiều thành 2 chiều. Để làm được điều đó chúng ta cần đến thuật toán PCA

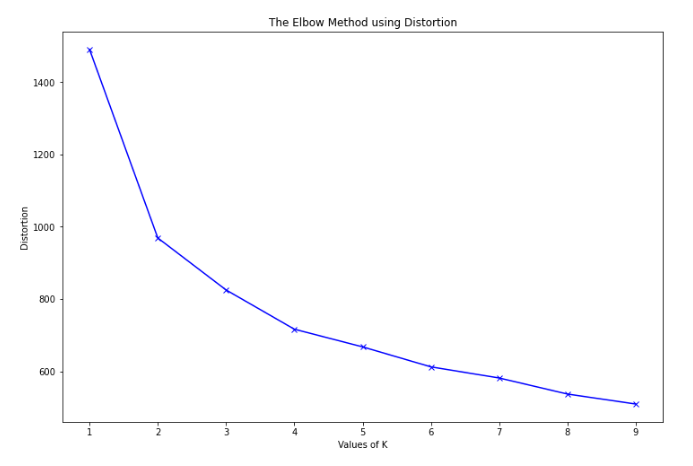
Principal component analysis (PCA): Đây là một thuật toán unsupervised giúp giảm chiều dữ liệu và vẫn giữ được nhiều thông tin nhất có thể. Thường được sử dụng với kiểu dữ liên tục. PCA xoay và chiếu điểm dữ liệu theo chiều tăng của phương sai PCA có thể hiểu là một quy trình thống kê trong đó phép biến đổi trực giao tập dữ liệu n chiều thành n chiều principal component.

PCA là thuật toán giảm chiều tuyến tính bằng cách biến đổi các biến tương quan (p) trở thành các biến k không ràng buộc (với k<p) được gọi là principal components. Các biến này sẽ tối đa hóa việc lưu giữ nhiều thông tin của tập dữ liệu nguồn.

.

## 3.2 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 (còn được gọi là khuỷu tay) sẽ giúp chúng ta xác định được số k chính xác.

Trong phương pháp này, em 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.

*Hình 1: Minh hoạ phương pháp Elbow*

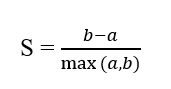
Với sự gia tăng của số lượng cụm (k) , khoảng cách trung bình giảm. Để tìm được cụm (k) tối ưu hãy quan sát biểu đồ vị trí mà độ dốc được bẻ cong ( hình dạng như khuỷu tay) là số cụm tối ưu, trong biểu đồ trên số cụm tối ưu là 2 .

## 3.3 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.

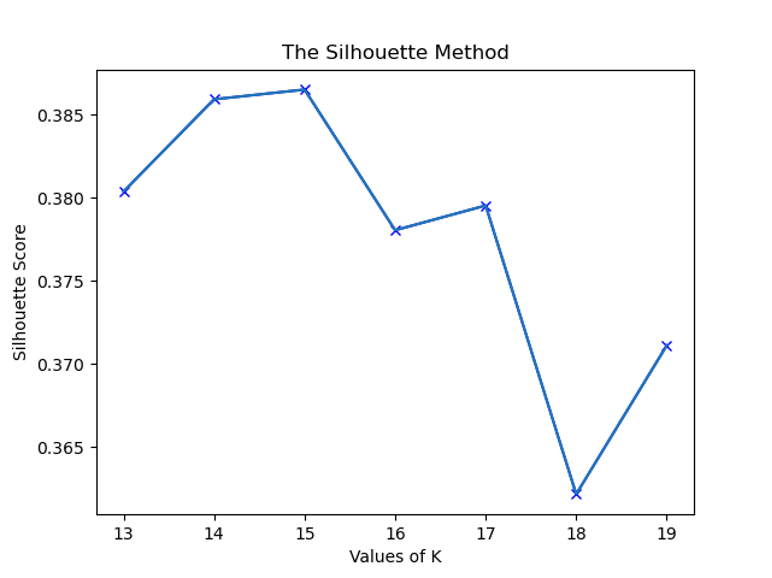
Silhouette là một thước đo về cách một thuật toán phân cụm đã hoạt động. Sau khi tính toán hệ số hình bóng của mỗi điểm trong tập dữ liệu, hãy vẽ biểu đồ đó để có được một hình ảnh trực quan về mức độ tốt của tập dữ liệu được nhóm thành k cụm. Biểu đồ hình bóng hiển thị thước đo mức độ gần của mỗi điểm 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.

Silhouette score được tính bằng công thức sau:



Trong đó:

* a là khoảng cách trung bình tới các điểm khác trong cùng 1 cụm.
* b là khoảng các trung bình tới các điểm trong cụm gần nhất.
* S là Giá trị của silhouette score



*Hình 2: Minh họa phương pháp Silhouette*

# 4. Áp dụng giải quyết vấn đề - Mã nguồn

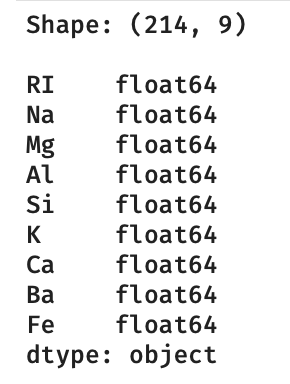
## 4.1 Chuẩn bị vấn đề

### 4.1.1 Đọc dữ liệu

Bộ dữ liệu là một tập dữ liệu gồm 214 mẫu các chất liệu và tính chất của các loại kính ta phải thực hiện thuật toán gom cụm

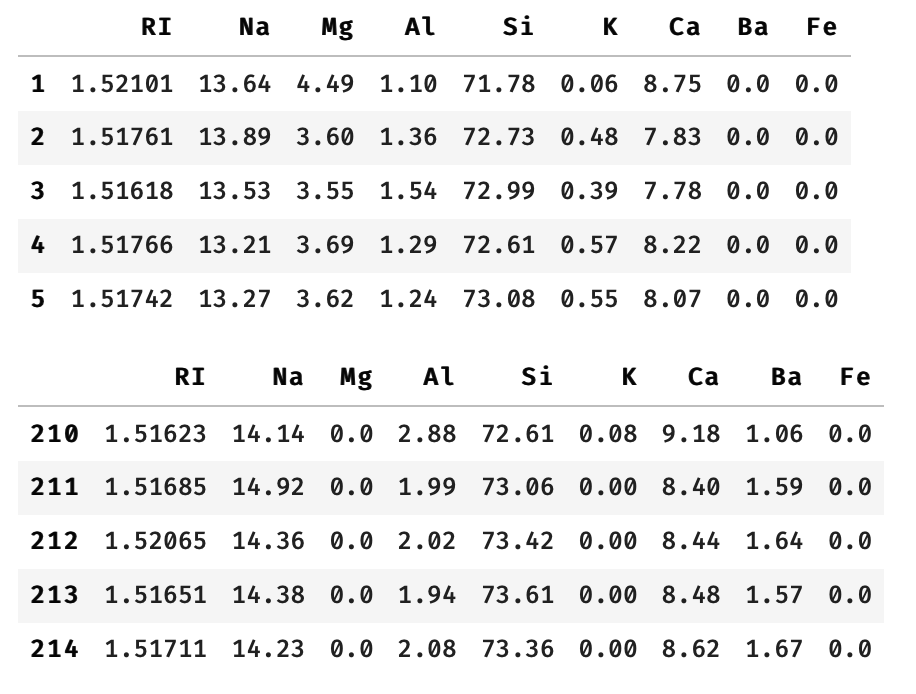
Mỗi mẫu gồm 9 thuộc tính (loại bỏ thuộc tính phân lớp: Type) như sau:

* RI : Chỉ số khúc xạ
* Na : Natri
* Mg : Magie
* Al : Nhôm
* Si : Silicon
* K : Kali
* Ca : Canxi
* Ba : Bari
* Fe : Sắt



*Hình 3: Tóm tắt bộ dữ liệu*

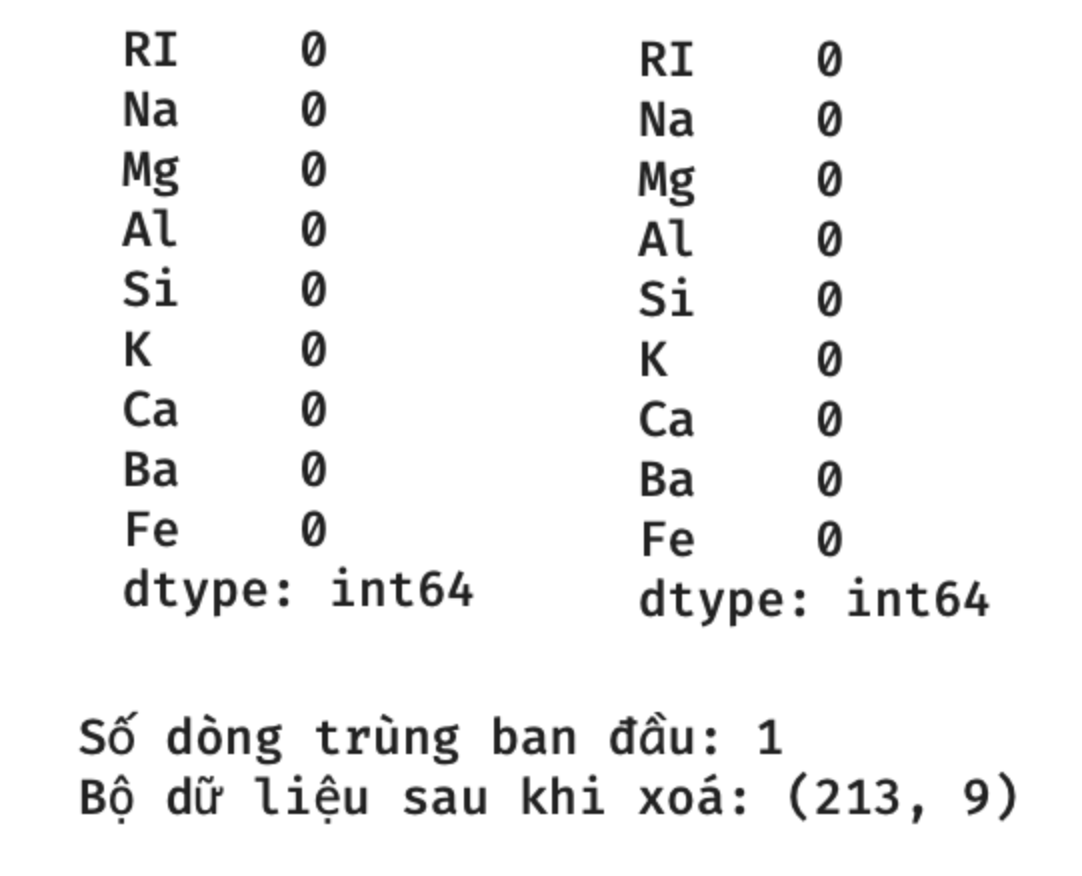
5 dòng đầu và 5 dòng cuối của bộ dữ liệu

**

*Hình 4: 5 dòng đầu, cuối của bộ dữ liệu*

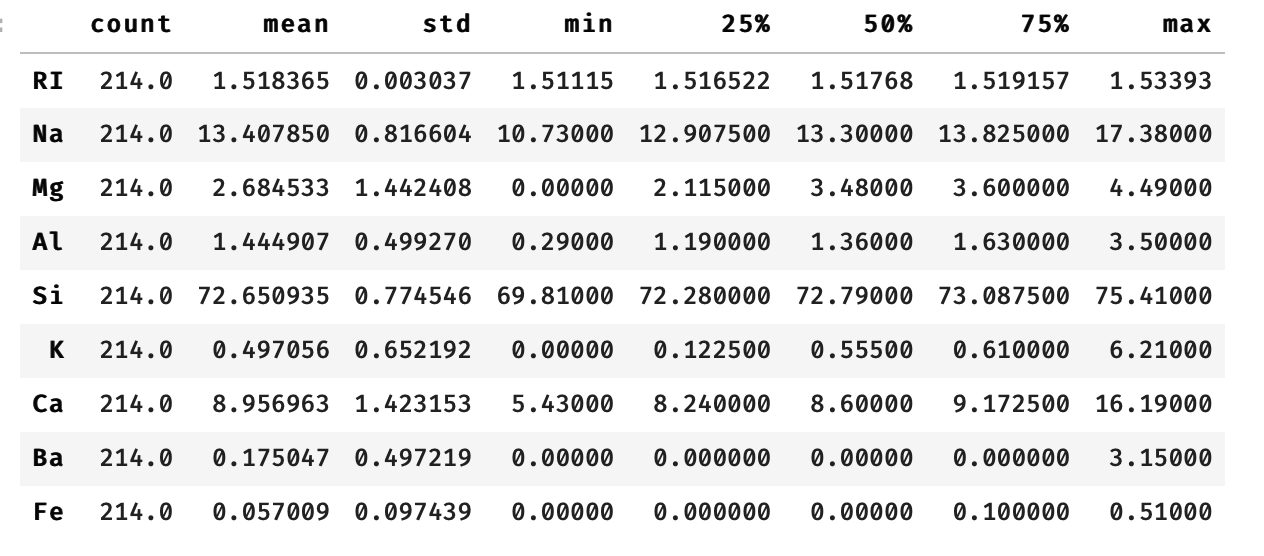
### 4.1.2 Kiểm tra dữ liệu

Bộ dữ liệu không có dữ liệu rỗng, được số hoá và có 1 dòng bị trùng lặp. Tiến hành tìm kiếm dòng trùng lặp và xóa dòng.



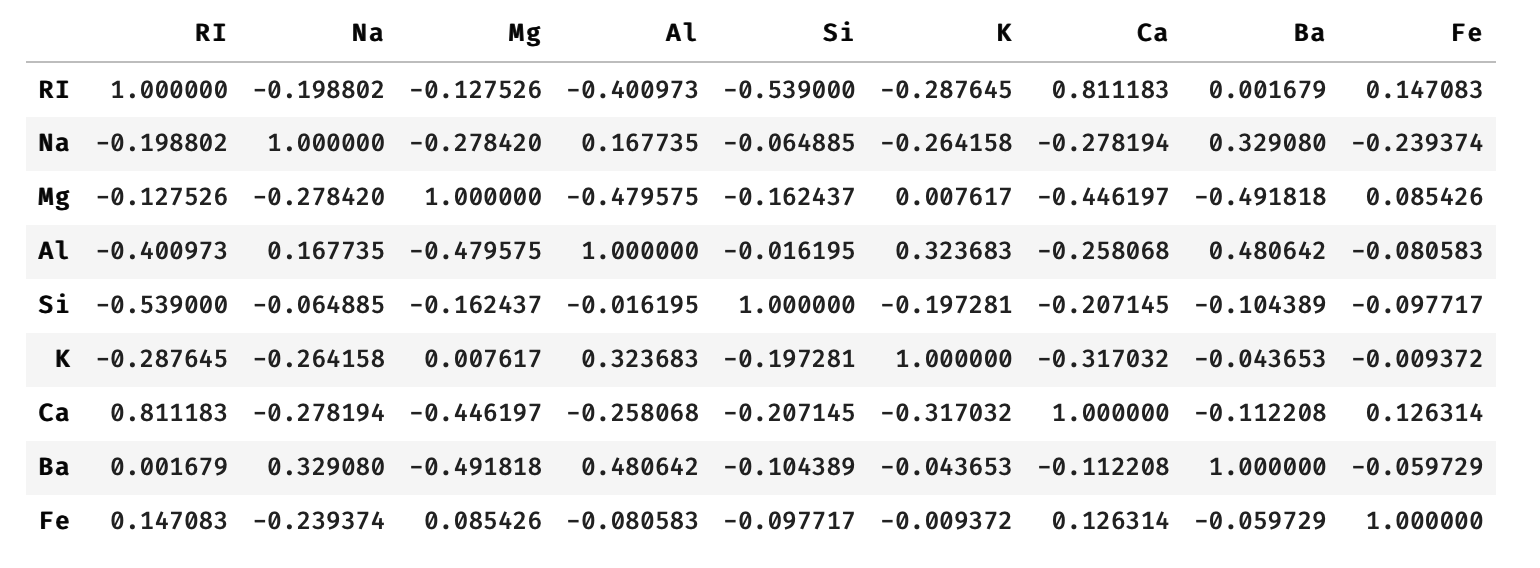
*Hình 5: Kiểm tra dữ liệu*

Giá trị của các đặc thuộc tính đối đồng đều chủ yếu nằm trong khoảng 0 - 6. Tuy nhiên có một vài đặc trưng có giá trị nằm lệch ra ngoài như Na (max = 17), Si (max = 75), Ca (max = 16). Nên chuẩn hoá (normalize) dữ liệu về tầm giá trị [0, 1] để bài toán được giải quyết một cách tốt nhất.



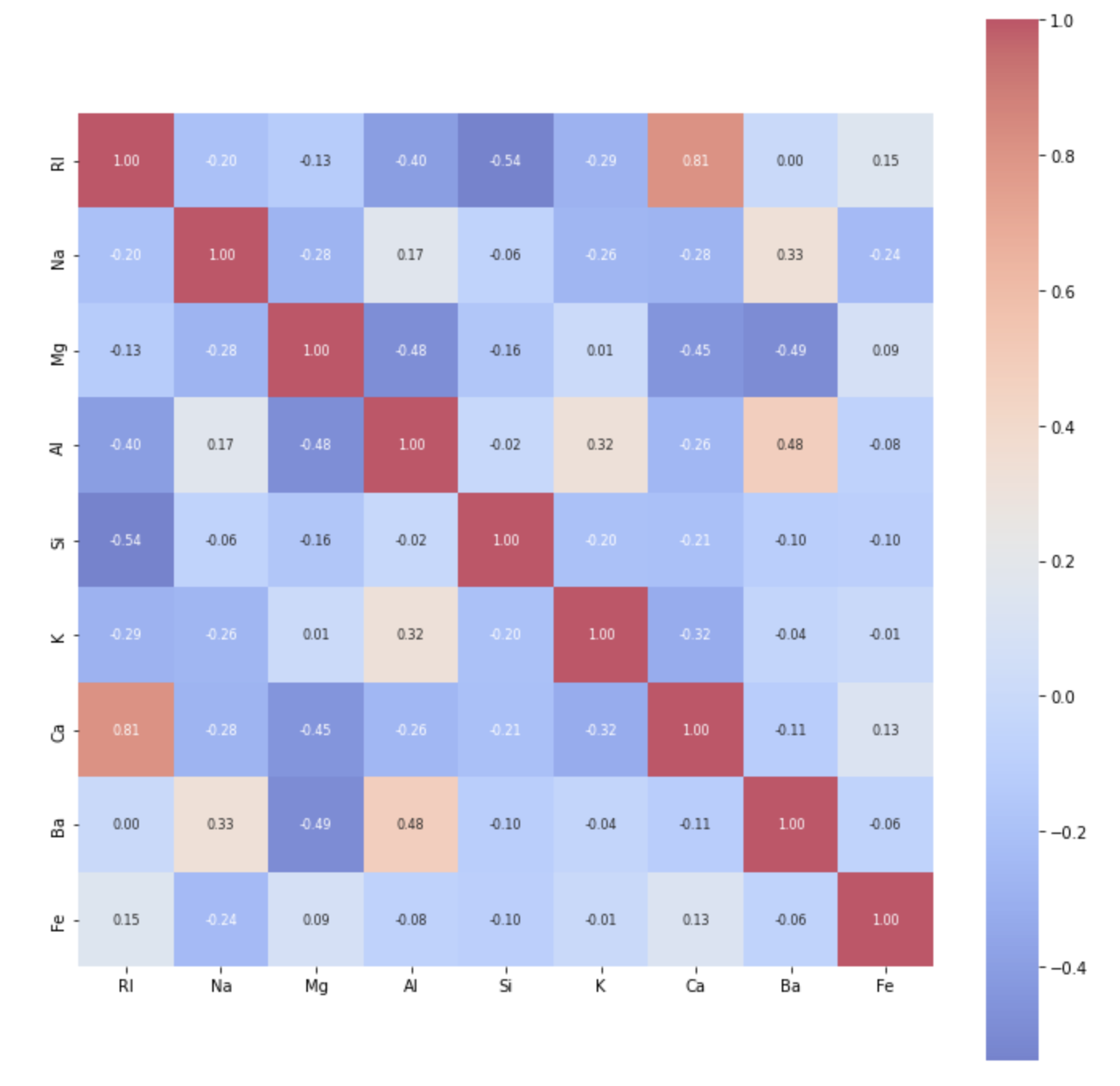
*Hình 6: Bảng giá trị các thuộc tính*

Độ tương qua của các thuộc tính rất thấp ngoại trừ 2 thuộc tính Ca và RI có độ tương quan khá cao: 0.81



*Hình 7: Bảng mô tả độ tương quan các thuộc tính*

Độ thị heatmap dưới đây sẽ cho ta thấy rõ hơn về độ tương quan giữa các thuộc tính với nhau



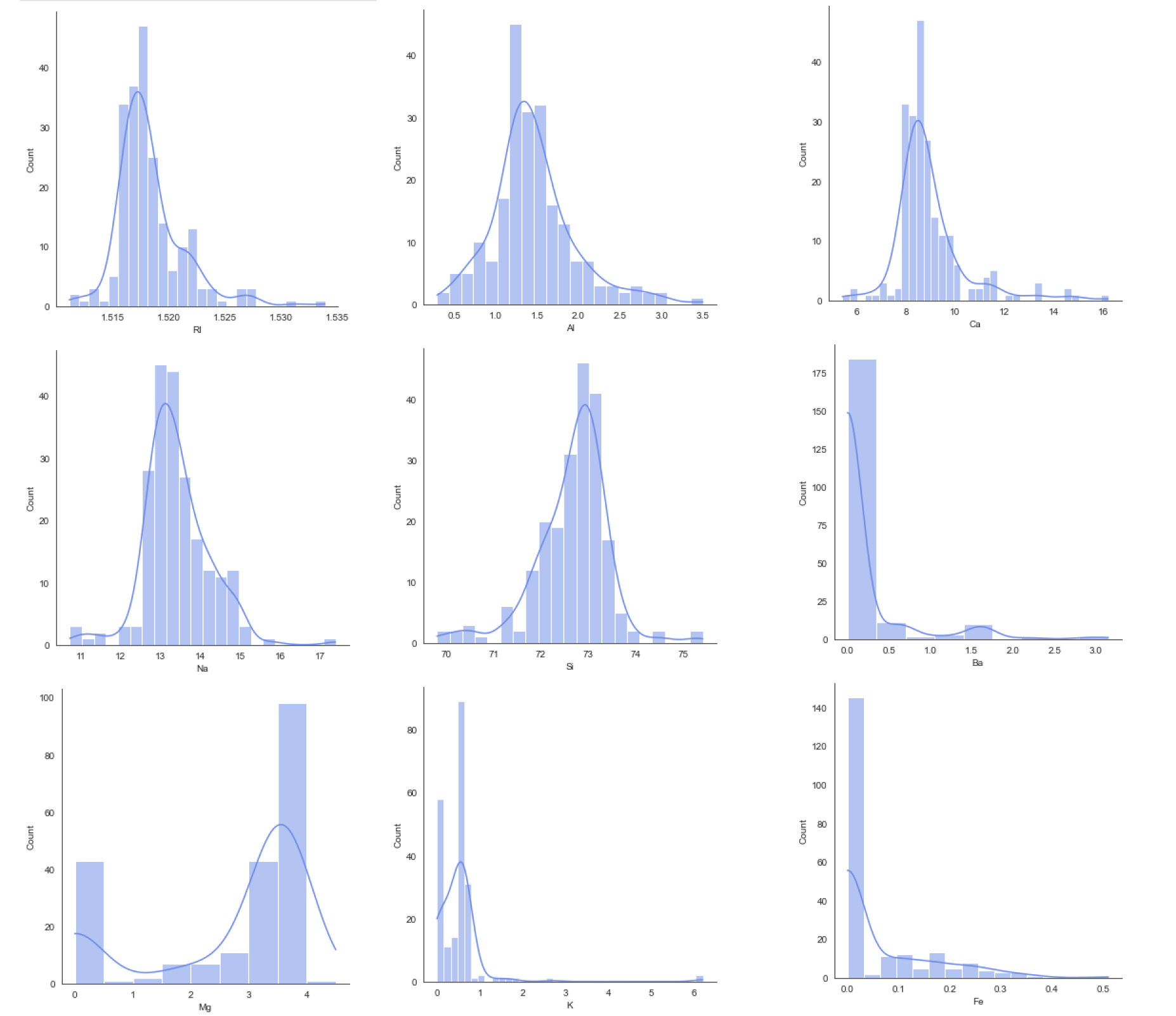
*Hình 8: Heatmap mô tả độ tương quan các thuộc tính*

## 

## 4.2 Chuẩn bị dữ liệu

### 4.2.1 Hiển thị dữ liệu

Chúng ta sẽ biểu diễn từng thuộc tính của dữ liệu dưới dạng đồ thị Histogram để xem chúng có phân phối tốt theo dạng chuẩn gauss hay không.

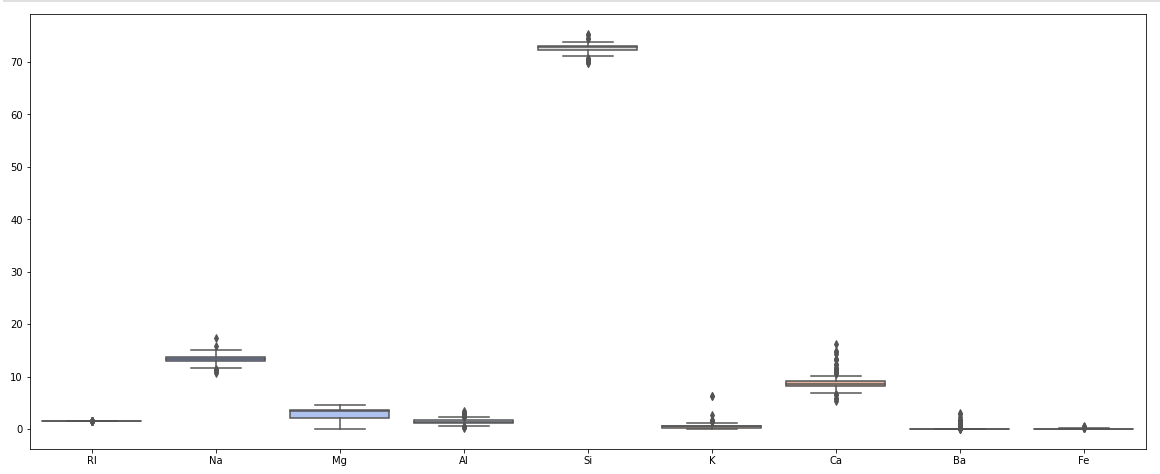


*Hình 9: Hiển thị các tính chất dưới đồ thị histogram*

Các tính chất phân phối khá tốt ngoại trừ các thuộc tính có miền giá trị thấp như Fe, Ba, K, Mg phân phối bị lệch.

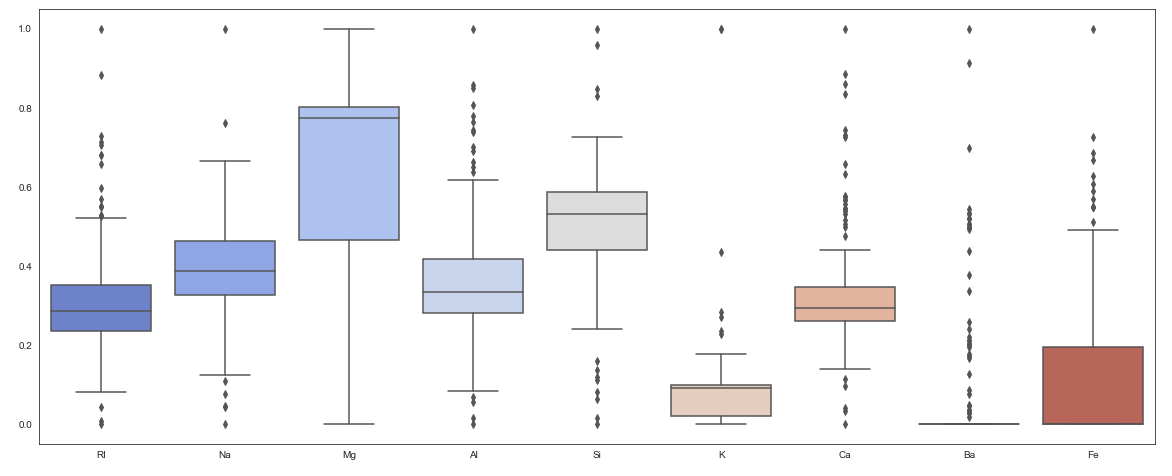
### 4.2.2 Chuẩn hóa dữ liệu

Chúng ta sẽ sử dụng phương pháp chuẩn hóa MinMax Scale để chuẩn hóa bộ dữ liệu trước khi tiến hành phân cụm.



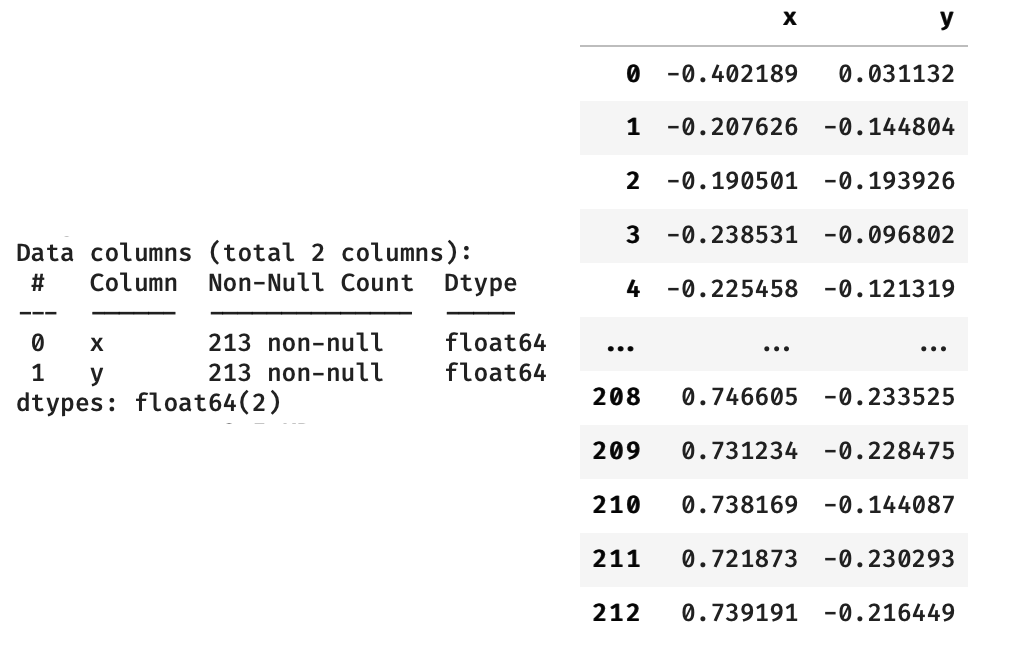
*Hình 10: Sự phân bổ của dữ liệu* ***trước*** *khi chuẩn hoá*

Sau khi chuẩn hoá bộ dữ liệu đã được đưa về miền giá trị từ [0 - 1] dễ dàng trong việc giải quyết vấn đề hơn.



*Hình 11: Sự phân bổ của dữ liệu* ***sau*** *khi chuẩn hoá*

Tiếp theo chúng ta sử dụng thuật toán PCA để giảm chiều dữ liệu từ 9 chiều thuộc tính thành 2 để có thể hiển thị rõ nét hơn về dữ liệu.

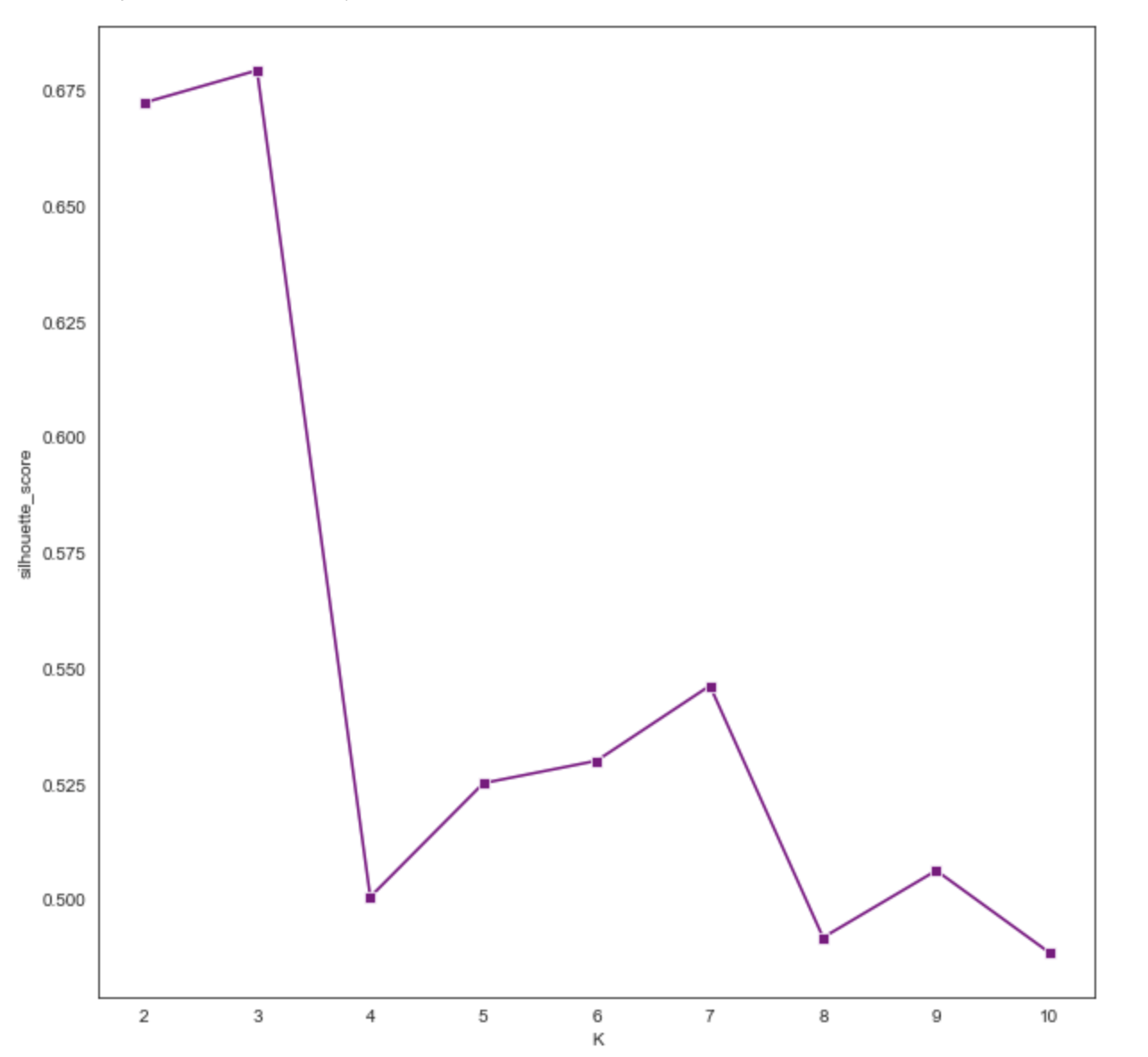


*Hình 12: Tóm tắt dữ liệu sau khi giảm chiều*

## 4.3 Phân cụm

### 4.3.1 Tìm số cụm (K)

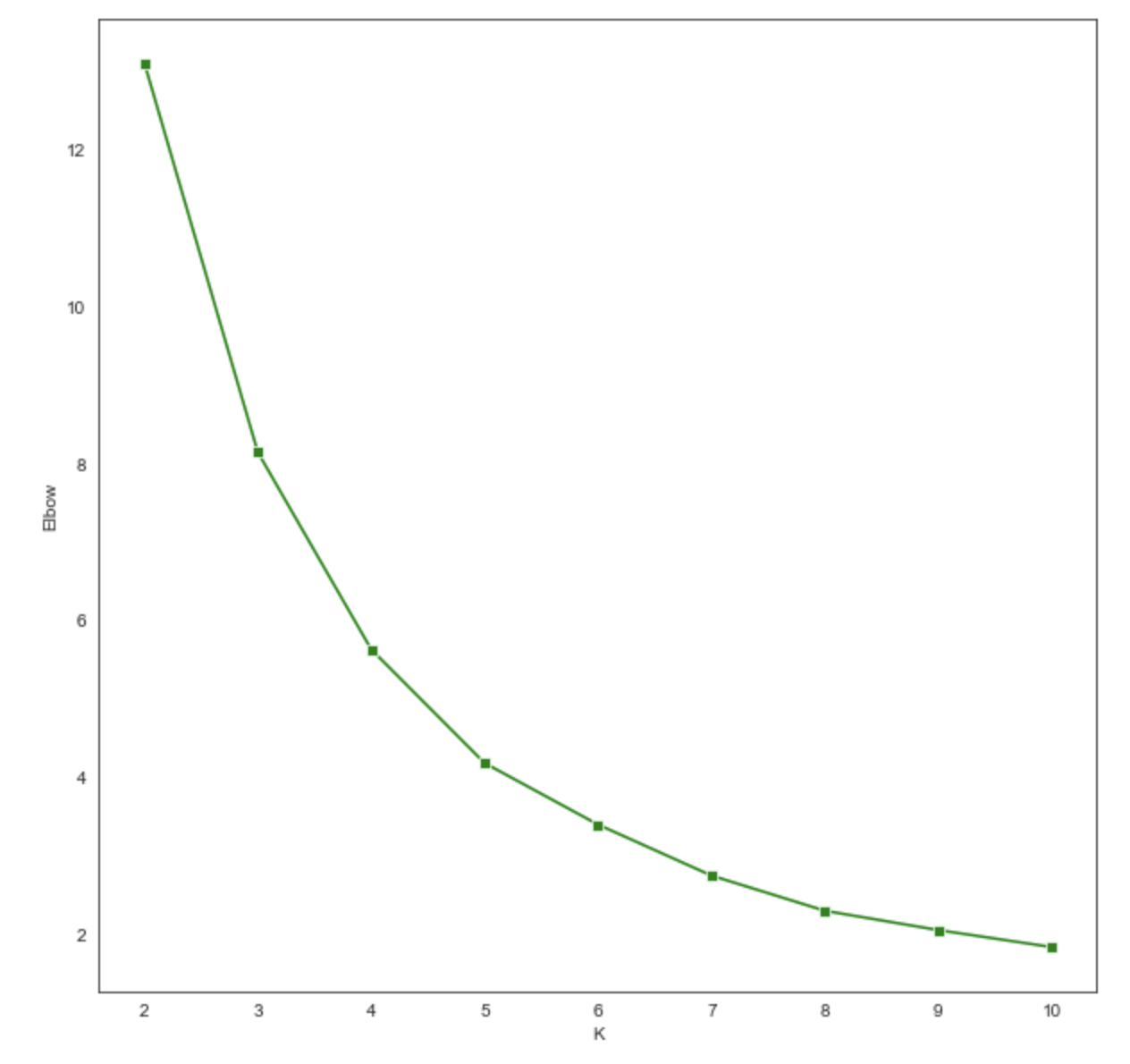
* Áp dụng phương pháp Silhouette



*Hình 13: Tìm K bằng thuật toán Silhouette*

Nhận xét: Theo thuật toán Silhouette cho ta kết quả K tốt nhất là 3.

* Áp dụng phương pháp Elbow



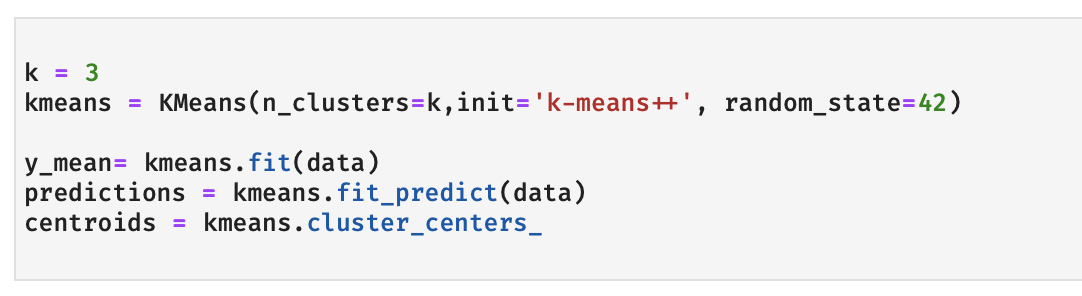
*Hình 14: Tìm K bằng thuật toán Elbow*

Nhận xét: Theo thuật toán Elbow cho ta kết quả K tốt nhất từ [3 - 6]

→ Vì vậy chúng ta sẽ chọn số cụm bằng 3

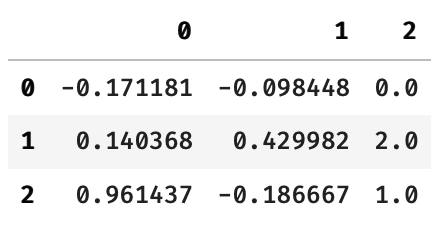
### 4.3.2 Chạy thuật toán K-means

Chúng ta tiến hành chạy thuật toán K-means bằng hàm KMeans của thư viện Sklearn với số cụm k = 3 như sau:



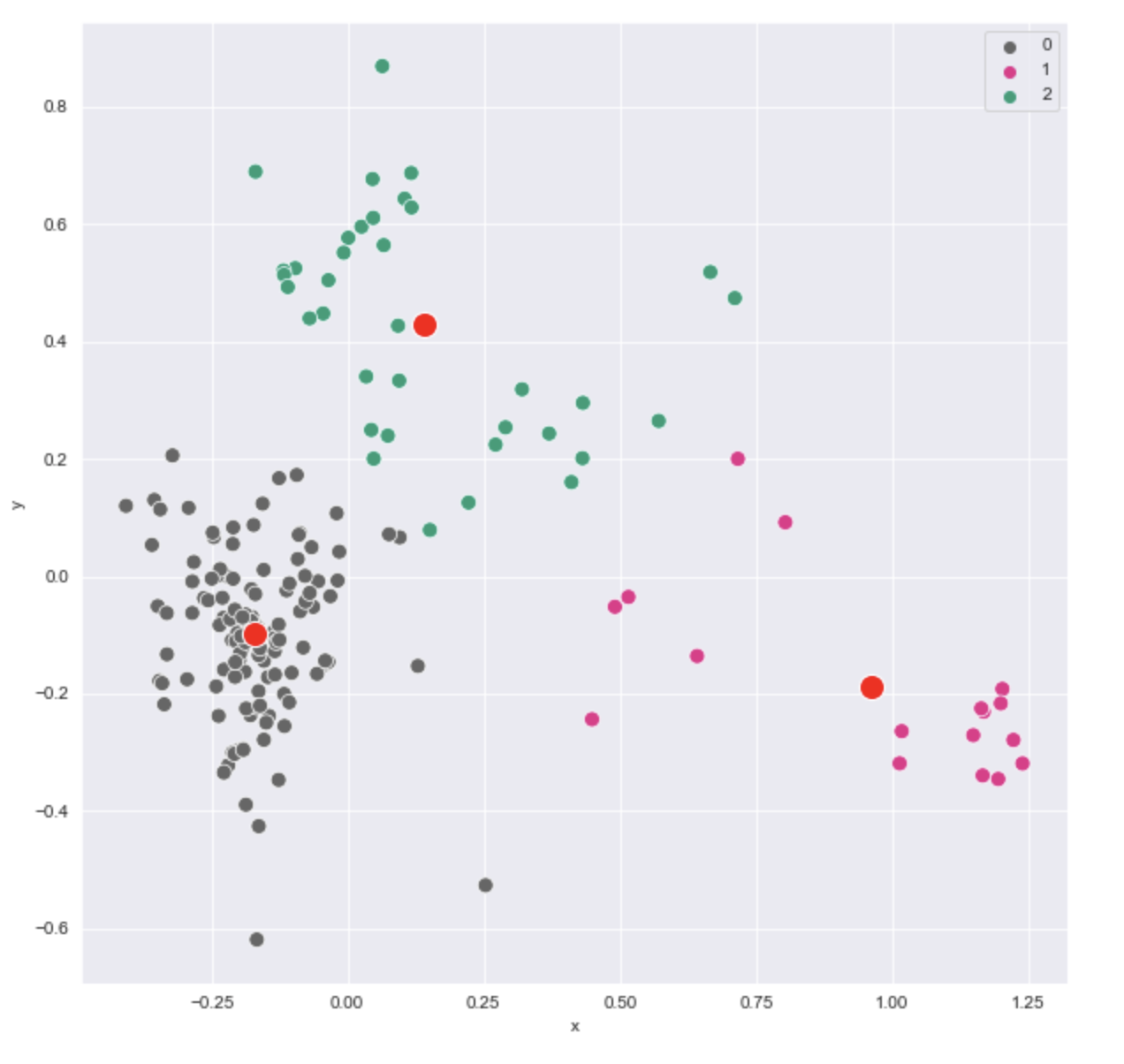
*Hình 15: Mã nguồn sử dụng thuật toán KMeans*

* Xác định các trung tâm cụm (centroid) bằng phương thức cluster\_centers\_ trong hàm KMeans



*Hình 16: Các centroid tìm được*

* Tiến hành hiển thị các điểm dữ liệu đã phân cụm và các tâm cụm



*Hình 17: Các mẫu sau khi gom cụm*

​​

Các mẫu được gom thành 3 cụm riêng biệt: các điểm màu xám thể hiện cho cụm thứ 0, các điểm màu hồng thể hiện cho cụm thứ 1 và các điểm màu xanh lá thể hiện cho cụm thứ 2. Các centroid nằm giữa các cụm, có màu đỏ và to hơn các điểm dữ liệu.

### 4.3.3 Hiển thị đặc trưng các cụm

Chúng ta tiến hành hiển thị các đặc trưng của từng cụm dữ liệu dưới dạng đồ thị boxplot để dễ dàng nhận biết đặc điểm của mỗi cụm sau khi phân loại.

### 

*Hình 18: Đặc trưng của từng cụm*

Nhận xét: Các cụm có đặc điểm riêng biệt như:

* Cụm 0: Độ RI cao nhất, giàu Fe và đặt biệt cao Ca
* Cụm 1: Đặt biệt giàu Mg và ít Al và Ca
* Cụm 2: Giàu Na, Al và Ba nhất nhưng lại ít Fe

# 5. Kết luận

Sau khi gom cụm bộ dữ liệu glass bằng thuật toán Kmean từ thư viện Scikit-Learn chúng ta nhận được 3 cụm với các đặc điểm khác nhau. Các thuật toán và phương thức dùng trong bài toán 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. Và nhờ giảm chiều dữ liệu bằng thuật toán PCA qua đó ta cũng có thể so sánh được cách hoạt động của mỗi thuật toán và hiển thị các kết quả một cách tốt nhất.

# 

# *Tài liệu tham khảo*

[1] - PGS.TS Dương Tuấn Anh, Slide bài giảng Môn Trí tuệ nhân tạo, Chương 7

[2] - Blink, Các kỹ thuật Dimensionality Reduction, <https://viblo.asia/p/cac-ky-thuatdimensionality-reduction-OeVKB98A5kW>

[3] - Nguyễn Văn Hiếu, Thuật toán K-Means (K-Means clustering) và ví dụ, <https://nguyenvanhieu.vn/thuat-toan-phan-cum-k-means>

[4] - Machine Learning Cơ Bản, Bài 4: K-means Clustering, <https://machinelearningcoban.com/2017/01/01/kmeans/#-gioi-thieu>

[5] - Nathan, Xây Dựng Clustering Model Bằng Giải Thuật K-Means Với Thư Viện Scikit-Learn Skills AI, <https://insights.magestore.com/posts/xay-dung-clustering-model-bang-giai-thuat-k-means-voi-thu-vien-scikit-learn-skills-ai>