**ĐẠI HỌC THỦY LỢI**

KHOA CÔNG NGHỆ THÔNG TIN



**BÀI TẬP LỚN**

**MÔN KHAI PHÁ DỮ LIỆU**

***Đề tài:***

**KHAI PHÁ DỮ LIỆU VỀ CHÁY RỪNG BẰNG K-MEAN VÀ K-MEDOIDS**

* **Nhóm sinh viên thực hiện:** Nhóm 18 – lớp 59TH3
* **Thành viên nhóm:**
* Nguyễn Thanh Tùng 175A071423
* Đào Ninh Thái 175A071427
* Vũ Thị Ngọc Anh 175A071277
* Ngô Việt Hoàng 175A071295
* **Giảng viên hướng dẫn:** Trần Mạnh Tuấn

*Hà Nội, tháng 5 năm 2021*

**BẢNG PHÂN CÔNG CÔNG VIỆC**

|  |  |
| --- | --- |
| **SINH VIÊN** | **CÔNG VIỆC PHÂN CÔNG** |
| Nguyễn Thanh Tùng | * Tìm hiểu về thuật toán K-Mean * Triển khai K-Mean trên WEKA * Triển khai mã nguồn K-Mean bằng Python * Đóng góp ý kiến và hỗ trợ các thành viên khác |
| Đào Ninh Thái | * Tiền xử lý dữ liệu * Thực hiện Power Point * Đóng góp ý kiến và hỗ trợ các thành viên khác |
| Vũ Thị Ngọc Anh | * Tìm hiểu về khai phá dữ liệu * Tìm hiểu về thuật toán K-Mean * Đóng góp ý kiến và hỗ trợ các thành viên khác |
| Ngô Việt Hoàng | * Tìm hiểu về thuật toán K-Medoids * Triển khai mã nguồi K-Medoids bằng Python * Tham chiếu dữ liệu trên bản đồ thế giới bằng Qgis * Đóng góp ý kiến và hỗ trợ các thành viên khác |

PHỤ LỤC

[PHỤ LỤC 3](#_Toc73372468)

[LỜI NÓI ĐẦU 4](#_Toc73372469)

[I. GIỚI THIỆU CHUNG 5](#_Toc73372470)

[1. Phát hiện tri thức 5](#_Toc73372471)

[2. Quy trình khám phá tri thức từ CSDL 5](#_Toc73372472)

[3. Phân cụm dữ liệu 6](#_Toc73372473)

[4. Mô tả bài toán 6](#_Toc73372474)

[II. TIỀN XỬ LÝ DỮ LIỆU 9](#_Toc73372475)

[1. Làm sạch dữ liệu 9](#_Toc73372476)

[2. Tích hợp dữ liệu 9](#_Toc73372477)

[3. Biến đổi dữ liệu 9](#_Toc73372478)

[III. THUẬT TOÁN K-MEAN 14](#_Toc73372479)

[1. Giới thiệu về K-Mean 14](#_Toc73372480)

[2. Các bước thực hiện thuật toán K-Mean 14](#_Toc73372481)

[3. Thực hiện bài toán bằng thuật toán K-Mean trên WEKA 17](#_Toc73372482)

[4. Thực hiện bài toán bằng thuật toán K-Mean trên Python 18](#_Toc73372483)

[IV. THUẬT TOÁN K-MEDOIDS 20](#_Toc73372484)

[1. Giới thiệu về K-Medoids 20](#_Toc73372485)

[2. Các bước thực hiện thuật toán K-Medoids 21](#_Toc73372486)

[3. Ưu điểm và nhược điểm của thuật toán K-Medoids 25](#_Toc73372487)

[4. Thực hiện bài toán bằng thuật toán K-Medoids trên Python 26](#_Toc73372488)

[V. Kết luận và hướng phát triển 29](#_Toc73372489)

[1. Kết luận 29](#_Toc73372490)

[2. Hạn chế 29](#_Toc73372491)

[3. Phát triển 29](#_Toc73372492)

[TÀI LIỆU THAM KHẢO 30](#_Toc73372493)

LỜI NÓI ĐẦU

Hiện nay với tình hình nóng lên toàn cầu và hiệu ứng nhà kính nguy cơ cháy rừng là ngày 1 tăng nhất là tại những quốc gia nằm gần đường xích đạo với khí hậu khô cằn là điều kiện lý tưởng để các đợt cháy rừng bùng phát.

Khu rừng Montesinho là một khu rừng cực kỳ nổi tiếng trên thế giới nằm ở đông bắc Bồ Đào Nha với địa hình nằm ngay đường xích đạo, và là rừng nhiệt đới nhưng nơi đấy rất hay xảy ra các vụ cháy rừng hàng năm, thật đáng tiếc cho những quần thể sinh vật sống tại đây khi có nạn cháy.

Để có thể phòng chống cháy rừng, ta cần chia nhỏ các khu vực nhỏ để có thể quản lý dễ dàng, phản ứng kịp thời, nhanh chóng khi các khu vực cháy xuất hiện và tăng cường biện pháp phòng cháy, chữa cháy ở những nơi thường xuyên xảy ra cháy nhằm tránh ảnh hưởng đến ra rộng hơn. Hơn nữa việc quản lý, phòng chống trong các khu vực nhỏ thường dễ hơn so với quản lý một khu vực rộng lớn và không phải chỗ nào cũng bảo vệ như nhau mà cần ưu tiên bảo vệ ở những khu vực thường xảy ra cháy.

Để có thể phân cụm các khu vực một cách hiệu quả, ta cần một thuật toán để tính toán phân cụm các khu vực theo các đặc điểm về thời tiết, địa điểm, đặc điểm khu vực. Nhưng với số lượng dữ liệu lớn như vậy, ta không thể ngồi tính toán bằng cách thủ công. Vậy tại sao chúng ta không sử dụng công nghệ thông tin với lợi thế tính toán nhanh, xử lý dữ liệu một cách nhanh chóng. Trong bài tập lớn này, chúng tôi sẽ sử dụng 2 thuật toán là K-Mean và K-Medoids để thực hiện phân cụm dữ liệu đưa vào.

Cuối cùng, từ kết quả của bài tập lớn này, ta có thể áp dụng vào phòng chống cháy rừng ở Việt Nam vì Việt Nam cũng nằm trong miền nhiệt đới, gần đường xích đạo của địa cầu, cũng như chịu ảnh hưởng của gió mùa Châu Á.

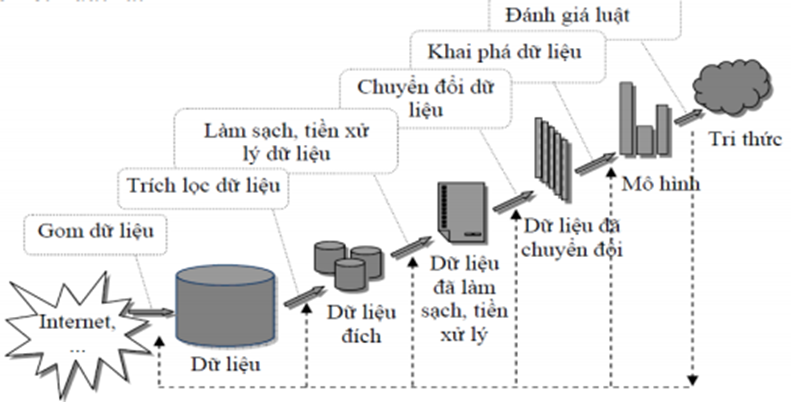
Trong quá trình hoàn thành bài tập lớn, chúng em đã cố gắng tìm hiểu và tham khảo các tài liệu liên quan. Tuy nhiên, thời gian có hạn nên chúng em sẽ không tránh khỏi những thiếu sót, rất mong nhận được sự đóng góp ý kiến của thầy cô và các bạn để nâng cao kỹ năng của bọn em, cũng như bài tập lớn này có thế áp dụng được trong thực tiễn.

Bọn em xin cảm ơn sự hướng dẫn tận tình trong lúc nghiện cứu của thầy Trần Mạnh Tuấn.

1. GIỚI THIỆU CHUNG
2. Phát hiện tri thức

Phát hiện tri thức (*Knowledge Discovery*) trong các cơ sở dữ liệu là một qui trình nhận biết các mẫu hoặc các mô hình trong dữ liệu với các tính năng: hợp thức, mới, khả ích, và có thể hiểu được.

Khai phá dữ liệu (Data mining) được định nghĩa như sau: “*Data mining là một quá trình tìm kiếm, phát hiện các tri thức mới, tiềm ẩn, hữu dụng trong CSDL lớn*”.

1. Quy trình khám phá tri thức từ CSDL

*Hình 1.2.1: Quy trình khai phá tri thức từ CSDL*

**Quy trình khai phá dữ liệu:**

* **Làm sạch dữ liệu:** Các nhiễu và dữ liệu không nhất quán sẽ được loại bỏ.
* **Tích hợp dữ liệu:** Dữ liệu từ nhiều nguồn khác có thể được tổ hợp lại.
* **Lựa chọn dữ liệu:** Những dữ liệu thích hợp với nhiệm vụ phân tích sẽ được trích rút ra từ CSDL.
* **Chuyển đổi dữ liệu:** Dữ liệu sau khi được chọn lọc sẽ được chuyển đổi hay hợp nhất về dạng thích hợp cho việc khai phá.
* **Khai phá dữ liệu:** Quá trình cốt lõi, tất yếu trong đó các phương pháp thông minh sẽ được áp dụng nhằm trích rút ra các mẫu dữ liệu.
* **Đánh giá mẫu:** Các nhà phân tích dữ liệu sẽ dựa trên một số độ đo nào đó để xác định lợi ích thực sự, độ quan trọng của các mẫu biểu diễn tri thức.
* **Biểu diễn tri thức:** Giai đoạn này các kĩ thuật biểu diễn và hiển thị tri thức sẽ được sử dụng để đưa tri thức đã lấy ra đến người dùng.

1. Phân cụm dữ liệu

Phân cụm dữ liệu (data clustering) là hình thức học không giám sát (unsupervised learning) trong đó các mẫu học chưa được gán nhãn. Mục đích chính của phân cụm dự liệu là tìm những mẫu đại diện hoặc gom những dữ liệu tượng tự nhau (theo một chuẩn đánh giá nào đó) thành những cụm. Các điểm dữ liệu trong các khác nhau có độ tương tự thấp hơn các điểm dữ liệu trong cùng một cụm.

Một số ứng dụng của phân cụm dữ liệu:

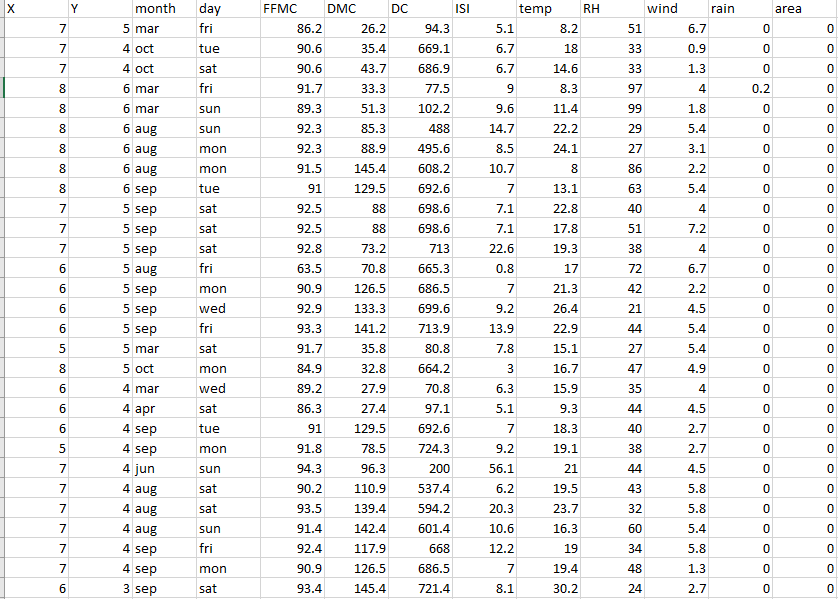
* Xem xét phân bố dữ liệu
* Tiền xử lý cho các thuật toán khác
* Khám phá thói quen và nhu cầu của khách hàng để có phương pháp tiếp thị thích hợp
* Phân loại đất theo công năng hoặc thực tế sử dụng để có chính sách quy hoạch phù hợp
* Phân loại bệnh nhân
* Phân loại nhà theo vị trí, giá trị
* …

1. Mô tả bài toán

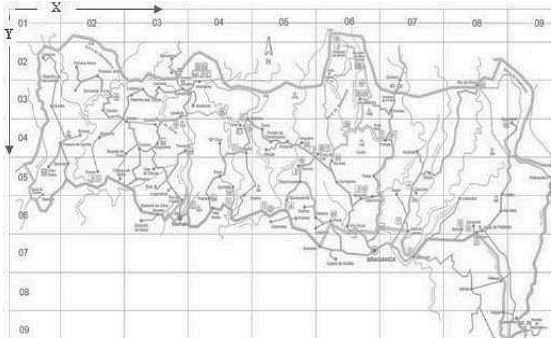
* Đầu vào của bài toán: Bộ dữ liệu về cháy rừng của công viên quốc gia Montesinho, một khu bảo tồn ở đông bắc Bồ Đào Nha.
* Yêu cầu thực tiễn: Các nhà chức trách cần xây dựng các biện pháp phòng chống cháy rừng trong các khu vực thường xảy ra cháy ví dụ các đặt trụ cứu hỏa, xây dựng hồ chứa nước hoặc xây dựng tháp quan sát và cứu hộ để phản ứng kịp thời, nhanh chóng với các khu cháy. Để làm được điều đó ta cần phân chia các khu vực thường xảy ra cháy thành các cụm khu nhỏ để dễ quản lý khu vực và có thể nhanh chóng phản ứng kịp thời với các vụ cháy, tránh để lan rộng.
* Yêu cầu bài toán: Thực hiện tiền xử lý và khai phá dữ liệu nhằm thực hiện việc phân nhóm dữ liệu. Từ đó, ta có thể phát biểu tri thức từ dữ liệu khai phá được.
* Ý tưởng của bài toán: Dựa vào dữ liệu đưa vào thực hiện phân tích, chuẩn hóa dữ liệu. Sau đó, sử dụng các thuật toán học máy về phân cụm K-Mean và K-Medoids để thực hiện việc phân cụm dữ liệu.
* Đầu ra mong muốn: Phân cụm được dữ liệu. Từ đó, thực hiện tham chiếu trên bản đồ của công viên quốc gia Montesinho.

**Dữ liệu đưa vào bài toán:**

Nguồn dữ liệu: nycdatascience.com

* Hiểu dữ liệu: Dữ liệu bao gồm nhiều thuộc tính liên quan tới cháy rừng cũng như diện tích độ cháy và tọa độ điểm cháy đối chiếu lên rừng Montesinho
* Dữ liệu gồm: Dữ liệu bao gồm 517 bản ghi với 13 thuộc tính dùng để dự đoán

*Hình 1.3.1: Dữ liệu đưa vào bài toán*

Dữ liệu đầu vào gồm có 13 thuộc tính sau:

*Hình 1.3.2: Bản đồ công viên Montesinho ứng với tọa độ dữ liệu đưa vào bài toán*

|  |  |
| --- | --- |
| **Tên thuộc tính** | **Chú thích** |
| X | Tọa độ X trên bản đồ công viên Montesinho: từ 1 đến 9 |
| Y | Tọa độ Y trên bản đồ công viên Montesinho: từ 2 đến 9 |
| Month | Tháng |
| Day | Ngày |
| FFMC | Chỉ số của các nguyên liệu có khả năng gây cháy từ 18 - 96 |
| DMC | Chỉ số của các chất hữu cơ có khả năng gây cháy từ 7 - 900 |
| DC | Chỉ số của các nguyên liệu, chất hữu cơ gây cháy nằm dưới lòng đất từ 0 – 56 |
| ISI | Chỉ số lây lán của ngọn lửa |
| Temp | Nhiệt độ |
| RH | Độ ẩm |
| Wind | Tốc độ gió |
| Rain | Lượng mưa |
| Area | Phạm vi cháy (hecta) |

1. TIỀN XỬ LÝ DỮ LIỆU
2. Làm sạch dữ liệu

Là quá trinh nhận dạng dữ liệu đã có để tiến hành xử lý các dữ liệu bị thiếu (missing data) xử lý dữ liệu bị nhiễu (noisy data).

1. **Missing data**

Thay thế các giá trị bị thiếu bằng chức năng ReplaceMissingValue

1. **Noisy data**

Thuộc tính *Day* không quá quan trọng nên sẽ bị loại bỏ, Montesinho 2 dữ liệu có độ chênh lệnh quả lớn ở mức Area là 1090.84 và 746.28 quá lớn so với phần còn lại nên sẽ bị loại bỏ để tăng độ chính xác.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X | Y | Month | Day | FFMC | DMC | DC | ISI | Temp | RH | Wind | Rain | Area |
| 6 | 5 | Sep | Sat | 92.5 | 121.1 | 674.4 | 8.6 | 25.1 | 27 | 4 | 0 | 1090.84 |
| 8 | 6 | Aug | Thu | 94.8 | 222.4 | 698.6 | 13.9 | 27.5 | 27 | 4.9 | 0 | 746.28 |

1. Tích hợp dữ liệu

Là quá trình trộn dữ liệu từ các nguồn khác nhau vào một kho dữ liệu sẵn sàng cho quá trình khai phá dữ liệu.

* Dữ liệu của khu rừng **Montesinho, Montesinho, Montesinho** không của khu vực nào khác

1. Biến đổi dữ liệu

Là quá trình biến đổi hay kết hợp dữ liệu vào những dạng thích hợp cho quá trình khai phá dữ liệu.

Các chiến lược thu giảm:

* Làm trơn dữ liệu.
* Kết hợp dữ liệu.
* Tổng quát hóa dữ liệu.
* Chuẩn hóa dữ liệu.
* Xây dựng thuộc tính đặc tính.

1. **Thực hiện chuẩn hóa dữ liệu**

Trong đó: *v* = [minA, maxA] là giá trị cũ

*v’* = [0, 1] là giá trị mới

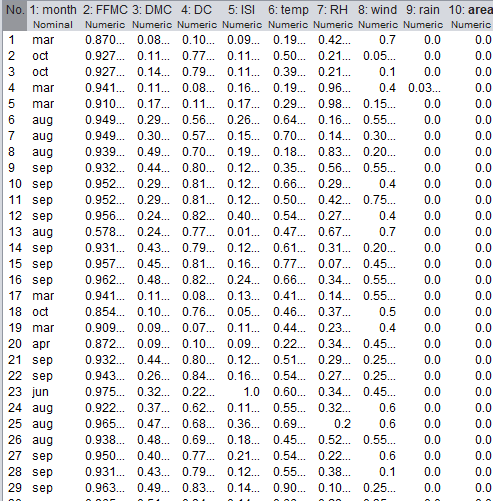
Ví dụ:

Thuộc tính RH: = 45

Chuẩn hóa về giá trị: v’ = [0, 1].

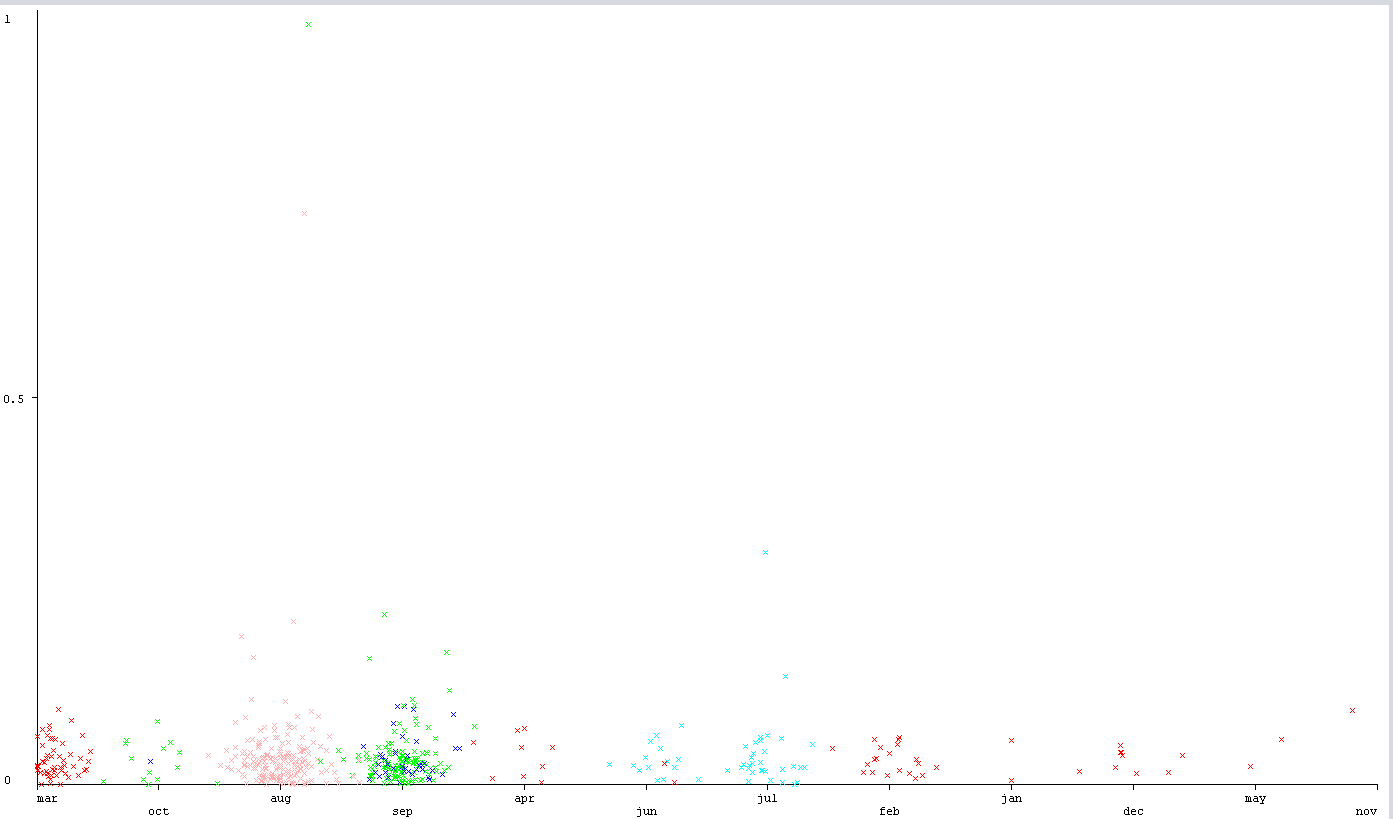
Với v = 45.

Tiến hành chuẩn hóa các thuộc tính số về đoạn [0, 1] bằng phương pháp chuẩn hóa min-max bằng bộ lọc Normalize, lưu file lại dưới định dạng csv.

* Lưu ý độ quan trọng của tất cả thuộc tính còn lại là như nhau

*Hình 2.3.1: Dữ liệu sau khi chuẩn hóa*

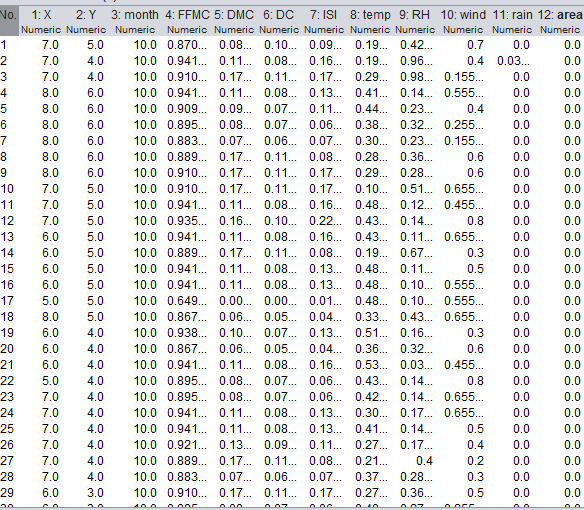
1. **Chuẩn hóa Month**

Ta thấy rằng thuộc tính *Month* đang ở dạng (NOM) rời rạc do đó để thuận tiện cho việc tính toán sau này ta sẽ đưa nó về dạng Numeric.

Hình ảnh trên phán chiếu với chục X là tháng của các vụ cháy và Y là biểu diện tần suất xuất hiện của các vụ cháy.

Ta sẽ đánh gia số liệu tương ứng sau đó chuẩn hóa về Numeric:

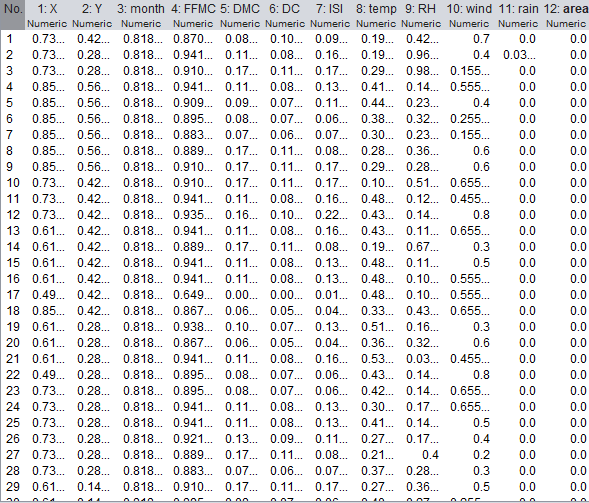
|  |  |
| --- | --- |
| **Tháng** | **Độ nghiêm trọng** |
| January | 3 |
| Febuary | 8 |
| March | 10 |
| April | 4 |
| May | 2 |
| June | 7 |
| July | 9 |
| Autugus | 12 |
| September | 11 |
| October | 6 |
| November | 1 |
| December | 5 |

1. **Chuẩn hóa X, Y**

*Hình 2.3.2: Dữ liệu trước khi chuẩn hóa X, Y*

Month sau khi đã chuẩn hóa

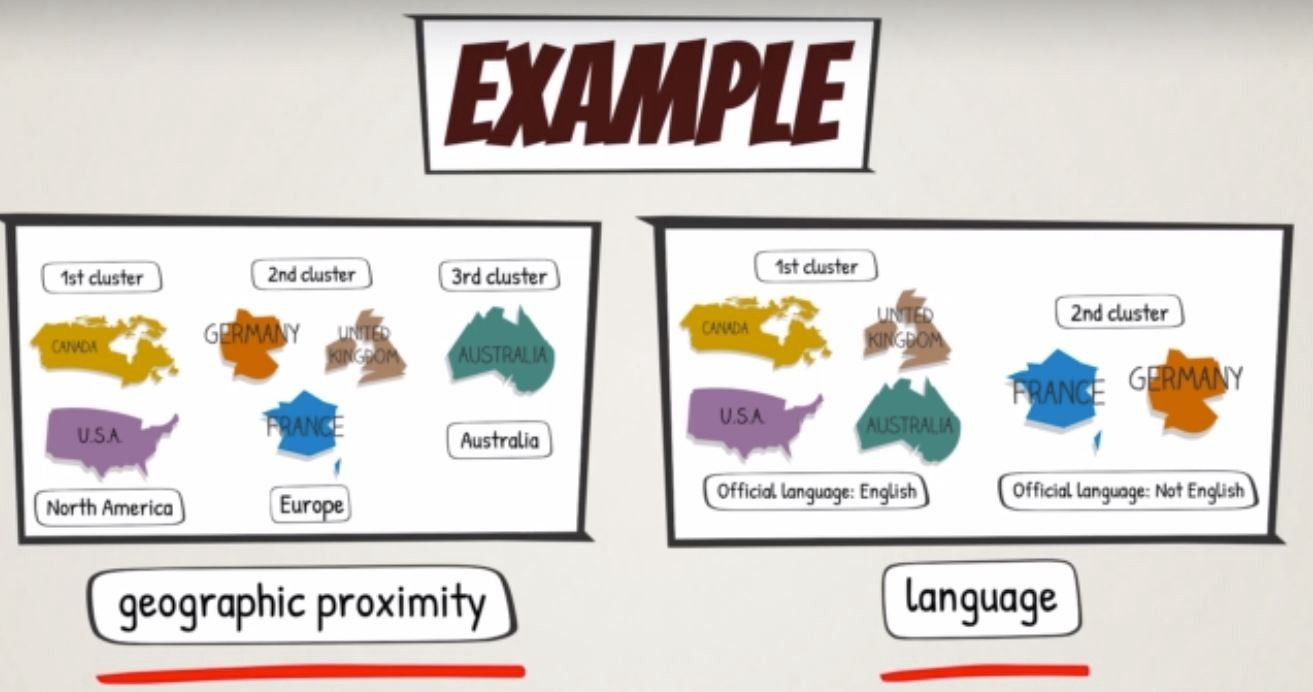
Với *X* và *Y* là tọa độ của Montesinho do đó để tính chính xác khu vực cháy ta cho *X* và *Y* cộng với area sau đó thực hiện chuẩn hóa.



*Hình 2.3.3: Dữ liệu sau khi chuẩn hóa X, Y*

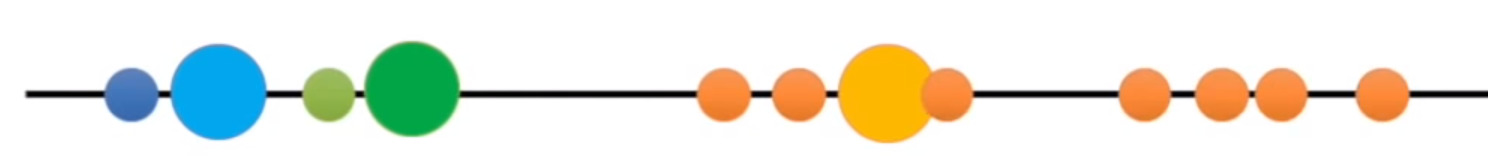
1. THUẬT TOÁN K-MEAN
2. Giới thiệu về K-Mean

Tưởng tượng chúng ta có 1 tập dữ liệu mà có thể sử dụng và chúng ta hiểu rằng chúng ta cần phải chia nó ra vào các cụm khác nhau, vì 3 cụm này là thuộc tính của 3 giá trị hoàn toàn khác nhau

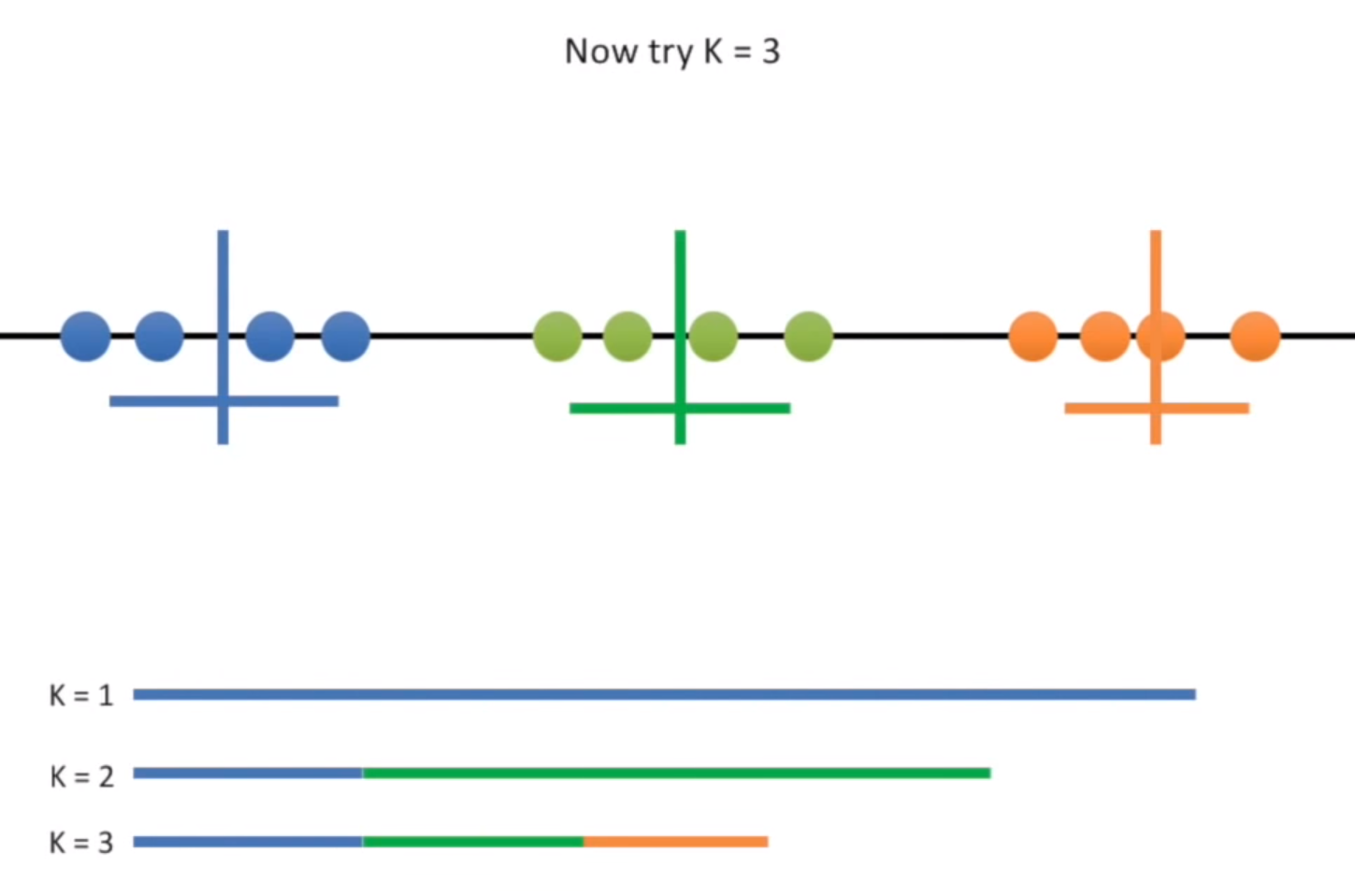


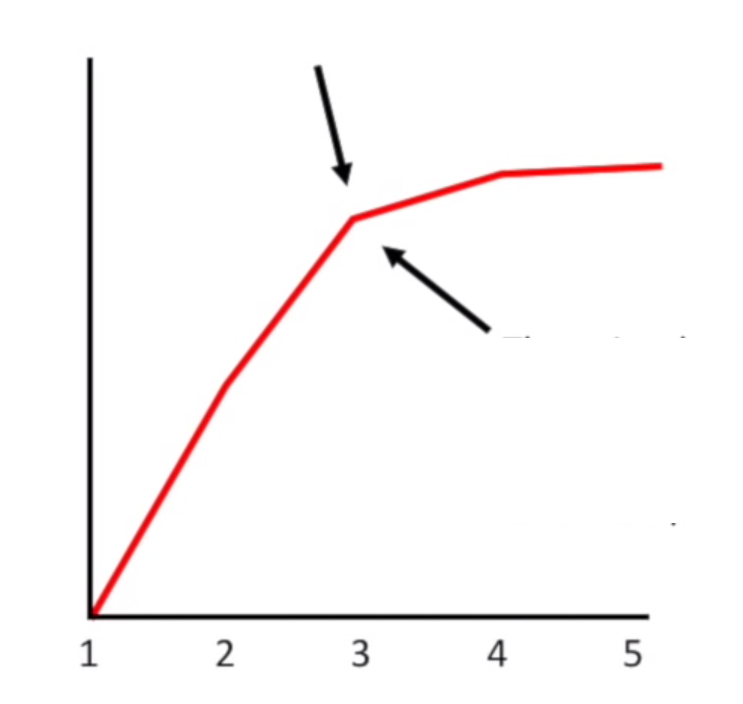
Ở trường hợp này giá trị trên là phân cụm, nhưng thay vì phỏng đoán theo đôi mắt hãy xem máy tính nhận dạng các cụm này như nào. Và đó là lý do **K-Mean** ra đời.

1. Các bước thực hiện thuật toán K-Mean
2. **Các bước thực hiện thuật toán K-Mean**

* **Bước 1:** Chọn số lượng cụm mong muốn để chia, hãy nói về việc chọn bao nhiêu cụm là tốt ở mục tiếp.
* **Bước 2:** Chọn ngẫu nhiên các cụm K tương ứng với cụm ban đầu.
* **Bước 3:** Đo lường khoảng cách của các biến xung quanh tới các cụm và thực hiện phân cụm.
* **Bước 4:** Phân các cụm gần với tâm nhất.
* **Bước 5:** Thực hiện lại với cụm mới ở đây dùng công thức tính khoảng cách.

Lặp lại với tâm mới đến khi tâm mới không thay đổi hoặc kết thúc bước lặp.

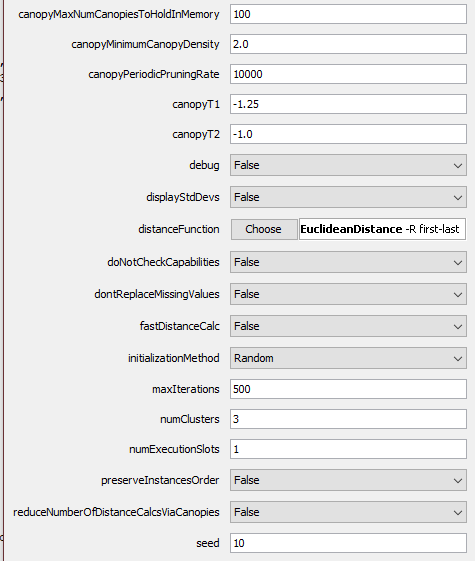
1. **Chọn K**

* Với K=1 cho kết quả kém
* Với K=2 cho kết quả tốt hơn K=1
* Với K=3 kết quả phân cụm tốt và đều nhau hơn
* Với K càng lớn thì các biến sẽ càng ít và khi K bằng với giá trị thì biến sẽ bằng0
* **Lưu ý chọn K**

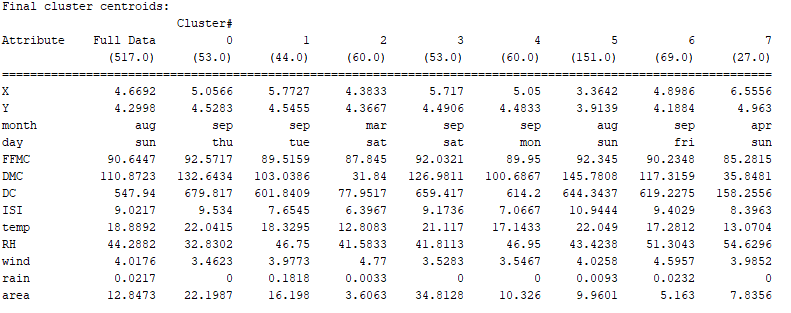
Với X là số lượng K và Y là mức độ giảm của biến ta nhận thấy rằng trong trường hợp này khi K=3 thì độ giảm của biến ở mức K=4 và K=5 là không đáng kể ta có thể gọi K = 3 là elbow và chọn nó làm K lý tưởng để phân cụm

Dùng công thức inertia để tính với K tương ứng:

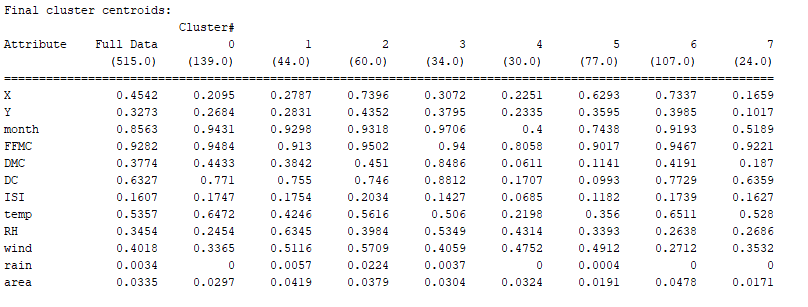
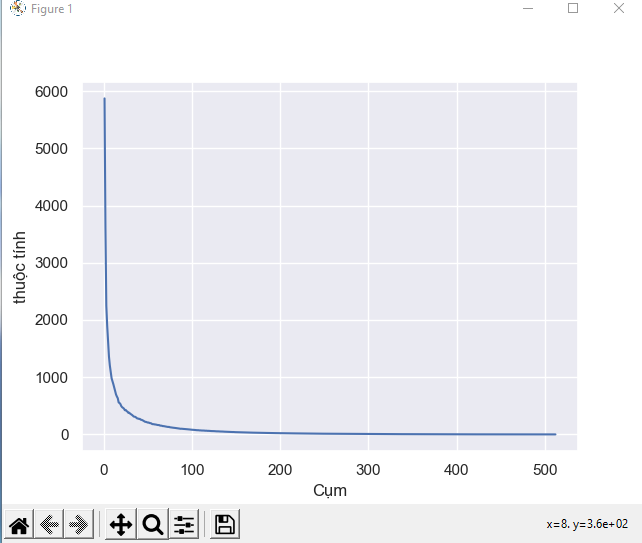
1. Thực hiện bài toán bằng thuật toán K-Mean trên WEKA
2. **Dữ liệu chưa chuẩn hóa**



**DistanceFunction**: chọn Euclidendistance

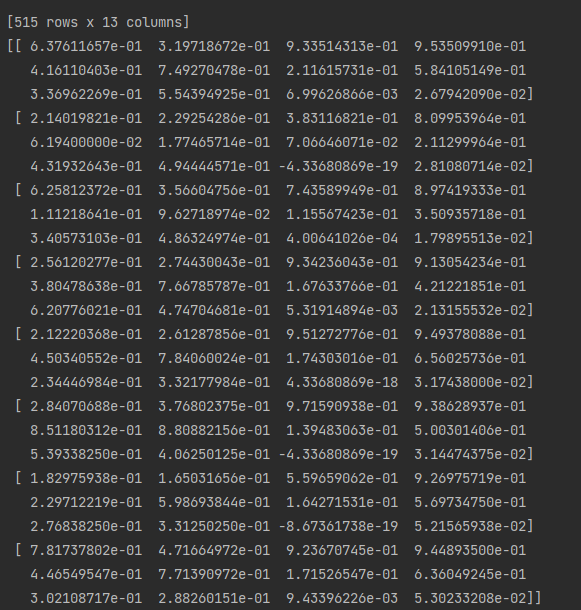
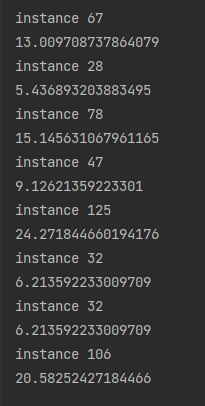
**numCluster**: chọn 8

**Nhận xét kết quả:** nhận thấy kết quả cá vụ chạy diễn ra tại tháng 9 nhiều và lớn.

1. **Dữ liệu đã chuẩn hóa**
2. Thực hiện bài toán bằng thuật toán K-Mean trên Python

Chọn K = 8

Biểu diện sự liên quan giữa tháng và khu vực cháy.

Kết quả cuối cùng:

**Nhận xét:** Dữ Liệu chia làm 8 cụm có sai số với Weka nhưng không cao.

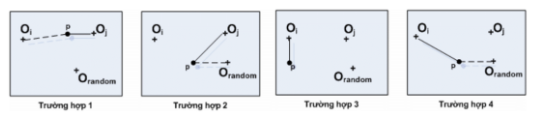
1. THUẬT TOÁN K-MEDOIDS

Thuật toán K-Medoids lấy đối tượng trong cụm làm trọng tâm của cụm (gọi là đối tượng tâm). Thuật toán này dựa trên nguyên tắc làm cự tiểu sự khác nhau giữa các đối tượng trong cùng một cụm.

1. Giới thiệu về K-Medoids

Ý tưởng của thuật toán K-Medoids như sau: Để tìm ra K cụm với n đối tượng thì K-Medoids chọn ngầu nhiên K đối tượng vào K cụm, coi mỗi đối tượng này là trọng tâm của cụm. Phân bố các đối tượng còn lại vào cụm mà sự khác nhau của nó với đối tượng trọng tâm của cụm là ít nhất (gần nhất). Sau đó lặp lại quá trình: thay đổi đối tượng trọng tâm của mỗi cụm sao cho chất lượng cụm được cải thiện. Chất lượng của cụm được lượng giá bởi một hàm đo sự khác nhau giữa một đối tượng với đối tượng trọng tâm chứa nó. Quá trình lặp cho đến khi không còn sự thay đổi nào về lực lượng cũng như hình dạng các cụm.

Để chọn một đối tượng không là đối tường trọng tâm Orandom thay thế tốt cho một đối tượng trọng tâm Oi thì mỗi đối tượng p xét theo 4 trường hợp sau đây:

* p đang thuộc vào cụm có trọng tâm Oi. Nếu Oj được thay thế bởi Orandom và p gần nhất với Oi (i j) thì p được gán lại vào Oi.
* p đang thuộc vào Oi. Nếu Oi được thay thế bởi Orandom và p gần nhất với Orandom thì p được gán lại vào Orandom.
* p đang thuộc Oj. Nếu Oj được thay thế bởi Orandom và p vẫn gần nhất với Oj thì không thay đổi gì cả. Tức p vẫn thuộc Oj.
* p đang thuộc vào Oj (i j). Nếu Oj được thay thế bởi Orandom và p gần nhất với Orandom thì p được gán lại vào Orandom.

*Hình 4.1.1: Các trường hợp chọn đối tượng thay thế*

Thuật toán phân cụm K-Medoids điển hình là PAM hoạt động hiệu quả đối với các tập dữ liệu nhỏ nhưng không hoạt động tốt đối với các tập dữ liệu lớn. Để xử lý các tập dữ liệu lớn hơn, một phương pháp dựa trên lấy mẫu thử được gọi là CLARA (Clustering LARge Applications) có thể được sử dụng. Thay vì đưa toàn bộ tập dữ liệu vào xem xét, CLARA sử dụng một mẫu thử ngẫu nhiên của tập dữ liệu. Thuật toán PAM sau đó được áp dụng để tính toán các medoid tốt nhất từ mẫu. Tốt nhất, mẫu thử phải gần với đại diện của tập dữ liệu gốc. Trong nhiều trường hợp, một mẫu thử lớn hoạt động tốt nếu nó được tạo ra cho nên mỗi đối tượng có xác suất được chọn vào mẫu thử bằng nhau. Các đối tượng mang tính đại diện (Medoid) được chọn sẽ có khả năng tương tự như những đối tượng đã được chọn từ toàn bộ tập dữ liệu. CLARA xây dựng các nhóm từ nhiều mẫu ngẫu nhiên và trả về phân cụm tốt nhất làm đầu ra. Sự phức tạp của việc tính toán các medoid trên một mẫu ngẫu nhiên là O(ks2 + k(n-k)), trong đó s là kích thước của mẫu, k là số của cụm, và n là tổng số đối tượng. CLARA có thể xử lý các tập dữ liệu lớn hơn PAM. Hiệu quả của CLARA phụ thuộc vào cỡ của mẫu thử. Lưu ý rằng PAM tìm kiếm K-Medoids tốt nhất trong tập dữ liệu cho trước, trong khi CLARA tìm kiếm k-medoid tốt nhất trong số mẫu thử được chọn của tập dữ liệu. CLARA không thể tìm thấy 1 cụm tốt nào nếu các mẫu medoids tốt nhất xa khỏi các k-medoid tốt nhất Nếu một đối tượng là một trong những loại k-medoid tốt nhất nhưng không được chọn trong quá trình lấy mẫu thử, CLARA sẽ không bao giờ tìm thấy phân cụm tốt nhất. Đầu tiên, nó chọn ngẫu nhiên k đối tượng trong tập dữ liệu làm medoid hiện có. Sau đó nó chọn ngẫu nhiên một medoid hiện có x và một đối tượng y không phải là một trong số medoid hiện có. Liệu thay x bằng y có thể cải thiện tiêu chí về sai số tuyệt đối không? Nếu có, việc thay thế được thực hiện. CLARANS thực hiện cuộc tìm kiếm ngẫu nhiên như vậy 1 lần. Tập hợp của medoid hiện có sau bước 1 được coi là tối ưu cục bộ. CLARANS lặp lại quá trình ngẫu nhiên này m lần và trả về kết quả cuối cùng là tối ưu cục bộ tốt nhất.

Trong tài liệu này, chúng tôi sẽ tập trung vào thuật toán K-Medoids điển hình là PAM.

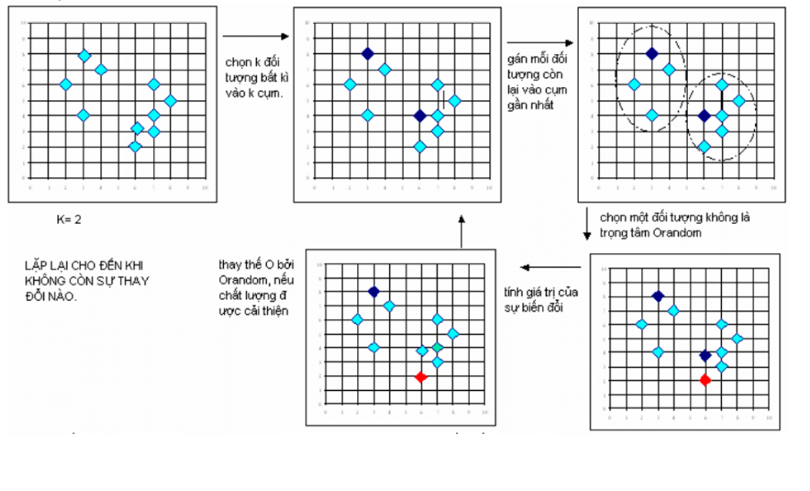
1. Các bước thực hiện thuật toán K-Medoids
2. **Các bước thực hiện thuật toán K-Medoids**

**Đầu vào:** tập CSDL gồm n đối tượng cần phân cụm và số nguyên K.

**Các bước thực hiện thuật toán K-Medoids:**

* **Bước 1:** chọn K đối tượng bất kỳ vào K cụm. Coi mỗi đối tượng này là tâm của nhóm.
* **Bước 2:** Vòng lặp
  + **Bước 2.1:** Gán mỗi đối tượng còn lại vào 1 cum mà nó gần với đối tượng tâm của cụm nhất.
  + **Bước 2.2:** Chọn ngấu nhiên 1 đối tượng không là đối tượng tâm Orandom.
* **Bước 3:** Tính lại giá trị, S, đối với việc đổi Oj với Orandom.
  + **Bước 3.2:** Nếu S < 0 thì đổi Oj với Orandom để tạo ra một tập với đối tượng tâm mới.
  + **Bước 3.2:** Đến khi không có sự thay đổi nào nữa thì mới dừng.

**Đầu ra:** Một tập gồm k cụm mà tổng giá trị của sự khác nhau của các đối tượng đến đối tượng trọng tâm của cụm chúa nó là nhỏ nhất.



*Hình 4.2.1: Các bước thực hiện thuật toán K-Medoids (với K = 2)*

1. **Ví dụ minh họa thuật toán K-Medoids**

Giả sử ta có bảng dữ liệu sau:

|  |  |  |
| --- | --- | --- |
| x1 | 2 | 6 |
| x2 | 3 | 4 |
| x3 | 3 | 8 |
| x4 | 4 | 7 |
| x5 | 6 | 2 |
| x6 | 6 | 4 |
| x7 | 7 | 3 |
| x8 | 7 | 4 |
| x9 | 8 | 5 |
| x10 | 7 | 6 |

Ta chọn K = 2 và chọn ngẫu nhiên K tâm cụm là: x2(3,4) và x8(7,4)

Suy ra ta được 2 cụm: C1(3,4) và C2(7,4)

Bây giờ ta tính toán khoảng cách của những điểm còn lại với tâm 2 cụm bằng khoảng cách Euclid:

Ta được khoảng cách từ các điểm đến tâm cụm như sau:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Điểm** | **Tọa độ điểm** | | **Tọa độ tâm cụm C1** | | **Tọa độ tâm cụm C2** | | **Khoảng cách đến tâm cụm C1** | **Khoảng cách đến tâm cụm C2** | **Thuộc cụm** |
| x1 | 2 | 6 | 3 | 4 | 7 | 4 | 2.2361 | 5.3852 | C1 |
| x2 | 3 | 4 | 3 | 4 | 7 | 4 | 0 | 4 | C1 |
| x3 | 3 | 8 | 3 | 4 | 7 | 4 | 4 | 5.6569 | C1 |
| x4 | 4 | 7 | 3 | 4 | 7 | 4 | 3.1623 | 4.2426 | C1 |
| x5 | 6 | 2 | 3 | 4 | 7 | 4 | 3.6056 | 2.2361 | C2 |
| x6 | 6 | 4 | 3 | 4 | 7 | 4 | 3 | 1 | C2 |
| x7 | 7 | 3 | 3 | 4 | 7 | 4 | 4.1231 | 1 | C2 |
| x8 | 7 | 4 | 3 | 4 | 7 | 4 | 4 | 0 | C2 |
| x9 | 8 | 5 | 3 | 4 | 7 | 4 | 5.099 | 1.4121 | C2 |
| x10 | 7 | 6 | 3 | 4 | 7 | 4 | 4.4721 | 2 | C2 |

* Ta được:
* Cụm 1 lúc này C1 = {1, 2, 3, 4} hoặc theo tọa độ C1= {(2,6), (3,4), (3,8), (4,7)}
* Cụm 2 lúc này C2 = {5, 6, 7, 8, 9, 10} hoặc theo tọa độ

C2 = {(6,2), (6,4), (7,3), (7,4), (8,5), (7,6)}

Bây giờ ta tính tổng gái trị (total cost) của các điểm có trung tâm là c1 và c2.

Công thức tính giá trị giữa các điểm là

Trong đó:

x: là tất cẩ các điểm không phải là trọng tâm

c: là các điểm trọng tâm

d: là số cụm (trong trường hợp này là 2)

* Total cost = {Cost(1, C1) + Cost(3, C1) + Cost(4, C1)} +

{Cost(5, C2) + Cost(6, C2) + Cost(7, C2) + Cost(9, C2) + cost(10,C2)}

= (2.2361 + 4 + 3.1623) + (2.2361 + 1 + 1 + 1.4121 + 2)

= 17.0466

Với tâm cụm C1 = x2 (3, 4). Ta xét các điểm không phải là trọng tâm.

Xét x7(7, 3):

Lúc này, ta có 2 trọng tâm mới là C1 = x2(3, 4) và C2 = x7(7, 3)

Ta tính lại khoảng cách từ các điểm đến tâm 2 cụm:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Điểm** | **Tọa độ điểm** | | **Tọa độ tâm cụm C1** | | **Tọa độ tâm cụm C2** | | **Khoảng cách đến tâm cụm C1** | **Khoảng cách đến tâm cụm C2** | **Thuộc cụm** |
| x1 | 2 | 6 | 3 | 4 | 7 | 3 | 2.2361 | 5.831 | C1 |
| x2 | 3 | 4 | 3 | 4 | 7 | 3 | 0 | 4.1231 | C1 |
| x3 | 3 | 8 | 3 | 4 | 7 | 3 | 4 | 6.4031 | C1 |
| x4 | 4 | 7 | 3 | 4 | 7 | 3 | 3.1623 | 5 | C1 |
| x5 | 6 | 2 | 3 | 4 | 7 | 3 | 3.6056 | 1.4142 | C2 |
| x6 | 6 | 4 | 3 | 4 | 7 | 3 | 3 | 1.4142 | C2 |
| x7 | 7 | 3 | 3 | 4 | 7 | 3 | 4.1231 | 0 | C2 |
| x8 | 7 | 4 | 3 | 4 | 7 | 3 | 4 | 1 | C2 |
| x9 | 8 | 5 | 3 | 4 | 7 | 3 | 5.099 | 2.2361 | C2 |
| x10 | 7 | 6 | 3 | 4 | 7 | 3 | 4.4721 | 3 | C2 |

Total cost 1 = (2.2361 + 4 + 3.1623) + (1.4142 + 1.4142 + 1 + 3 + 2.2361) = 18.4629

Ta thấy: Total cost 1 – Total cost = 18.4629 - 17.0466 = 1.4163 > 0

* Không thay đổi vị trí của điểm x7

Tương tự ta xét tất cả các điểm x còn lại và tính khoảng cách nhỏ nhất với Total Cost ban đầu.

Kết quả cho ta là dữ liệu vẫn không thay đổi 🡪 Chúng ta chỉ gom được 2 cụm như ban đầu.

Cụm 1:C1 = {x1, x2, x3, x4}

Cụm 2: C2 = {x5, x6, x7, x8, x9, x10}

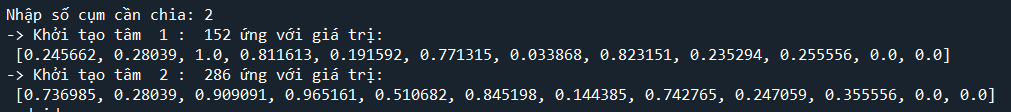
1. Ưu điểm và nhược điểm của thuật toán K-Medoids
2. **Ưu điểm của thuật toán K-Medoids**

* Dễ dàng thực hiện, đơn giản và dễ hiểu
* K-Medoids xuất ra các cụm cuối cùng trong một số lần lặp cố định

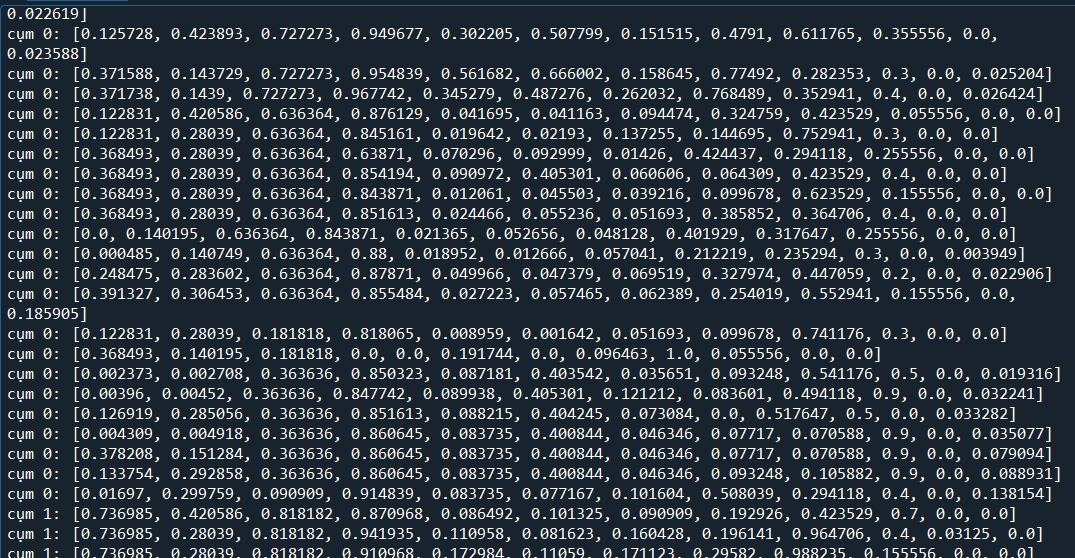
1. **Nhược điểm của thuật toán K-Medoids**

* Ta có thể thu được kết quả khác nhau cho các lần chạy khác nhau trên cùng một tập dữ liệu vì các tâm cụm đầu tiên được chọn ngẫu nhiên.
* K-Medoids cố định số lượng K ban đầu nhập vào, vì vậy ta không thể biết thuật toán có phân cụm chính xác không.
* Thời gian tính toán và phân cụm phụ thuộc vào số lượng cụm nhập vào ban đầu
* Kém hiệu quả

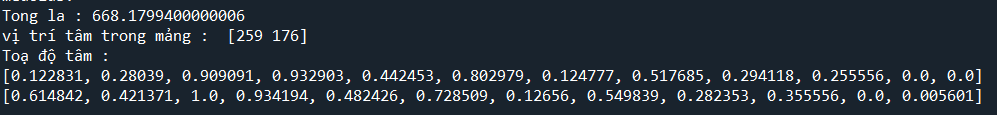
1. Thực hiện bài toán bằng thuật toán K-Medoids trên Python

* Đầu vào:
* Tập dữ liệu cháy rừng của công viên quốc gia Montesinho trước khi chuẩn hóa X, Y.
* Tập dữ liệu cháy rừng của công viên quốc gia Montesinho sau khi kết thúc chuẩn hóa.
* Công cụ sử dụng:
* Ngôn ngữ lập trình: Python
* Công cụ lập trình: Spyder (Anaconda)
* Công cụ tham chiếu bản đồ: Qgis
* Bản đồ sử dụng trong khi tham chiếu:
* Bản đồ thế giới từ plugin OpenStreetMap, mã bản đồ: **EPSG:3857**
* Bản đồ sử dụng đất đai của Bồ Đào Nha từ geofabrik
* Kết quả thực hiện trên Python
* Kết quả thực hiện khởi tại ngẫu nhiên:

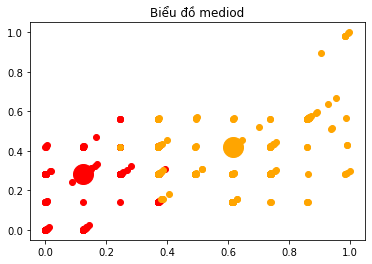
*Hình 4.4.1: Kết quả thực hiện khởi tạo ngẫu nhiên*

* Kết quả của quá trình phân cụm:

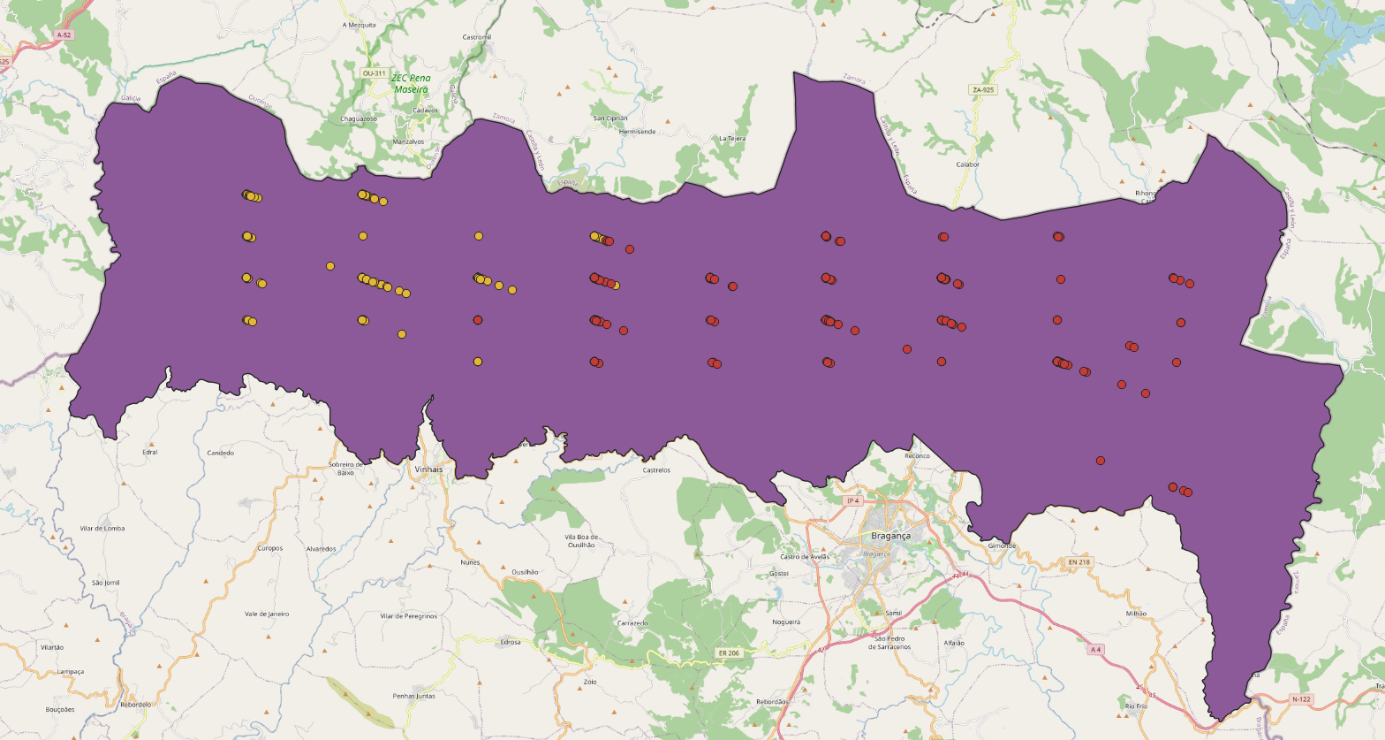
*Hình 4.4.2: Kết quả của quá trình phân cụm*

* Kết quả sau khi kết thúc thuật toán:

*Hình 4.4.3: Kết quả sau khi kết thúc thuật toán*

* Biểu đồ phân cụm dữ liệu:

*Hình 4.4.4: Biểu đồ phân cụm dữ liệu*

* Đầu ra:
  + File dữ liệu “.csv” của các cụm sau khi khôi phục X và Y theo chỉ số bản đồ thế giới trên Qgis.
  + File dữ liệu “.shp” của các cụm sau khi khôi phục X và Y theo chỉ số bản đồ thế giới trên Qgis.
* Kết quả tham chiếu trên Qgis:

*Hình 4.4.4: Kết quả tham chiếu trên Qgis*

1. Kết luận và hướng phát triển
2. Kết luận

Phương pháp phân cụm là một phương pháp không có nhãn bọn em đã nắm được lý thuyết và triển khai nó vào bài học dùng để ứng dụng vào thực tế. Nhóm em đã đạt được một số kết quả như sau:

* Tìm hiểu tổng quan về khai phá dự liệu, bài toán phân lớp, phương pháp hồi quy tuyến tính và thuật toán của 2 thuật toán K-Mean và K-Medoids.
* Thu thập dữ liệu về cháy rừng tiền xử lý bằng WEKA sau đó chạy bằng python và so sánh.
* Thực hiện tham chiếu dữ liệu trên Qgis.

1. Hạn chế

* Dữ liệu chưa thật sự tối ưu và còn ít để đưa ra mô hình chính xác.
* Kết quả trên WEKA chưa ăn khớp với Python.
* Kết quả tham chiếu trên Qgis chưa chính xác trong thực tế.

1. Phát triển

* Xây dựng, cải tiến mô hình với các loại phương pháp phân cụm khác.
* Tiếp tục nâng cấp và thêm dữ liệu.
* Tiếp tục cải tiến biểu thức tính để tham chiếu trên Qgis có tỷ lệ chính xác cao trong thực tế và thực hiện nâng cấp để tiết diện của mỗi điểm trong tham chiếu có tiết diện ứng với diện tích cháy.

TÀI LIỆU THAM KHẢO

* Bài giảng môn khai phá dữ liệu của thầy Trần Mạnh Tuấn
* Giáo trình khai phá dữ liệu – Nguyễn Hà Nam, Nguyễn Trí Thành, Hà Quang Thụy
* Hướng dẫn chuyển đổi CSV sang SHP:

[CSV to Shapefile with pyshp | Geospatiality (glenbambrick.com)](https://glenbambrick.com/2016/01/09/csv-to-shapefile-with-pyshp/)

* Hướng dẫn sử dụng Qgis:

<http://www.qgistutorials.com/vi/>

* Bản đồ công viên quốc gia Montesinho:

[Rua Da Gasparona, Montesinho Natural Park, 5320, Portugal (findlatitudeandlongitude.com)](https://www.findlatitudeandlongitude.com/l/Rua+da+Gasparona%2C+Montesinho+Natural+Park%2C+5320%2C+Portugal/2260516/)

* Lý thuyết về K-Medoids:
* <https://www.facebook.com/1589166614717569/posts/2403530413281181/>
* <https://ichi.pro/vi/k-medoid-clustering-la-gi-tai-sao-va-lam-the-nao-198896056049168>
* <https://iq.opengenus.org/k-medoids-clustering/>
* Bản đồ Bồ Đào Nha dùng trong Qgis:

<http://download.geofabrik.de/europe/portugal.html>