ĐẠI HỌC BÁCH KHOA HÀ NỘI

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

A red and white flag with a yellow and red design

Description automatically generated

**BÁO CÁO GR1**

**Chủ đề phân cụm, phân cụm bán giám sát**

|  |  |
| --- | --- |
| Họ và tên: | **Phạm Công Hào** |
| MSSV: | **20215045** |

***Hà Nội, 6/2024***

Mục Lục

**[Phần I](#_Toc169704039)****[Giới thiệu](#_Toc169704039)** [3](#_Toc169704039)

**[Phần II Fuzzy c-Means Clustering](#_Toc169704040)** [4](#_Toc169704040)

**[1.](#_Toc169704041)****[Giới thiệu FCM](#_Toc169704041)** [4](#_Toc169704041)

**[a.](#_Toc169704042)****[Định nghĩa](#_Toc169704042)** [5](#_Toc169704042)

**[b.](#_Toc169704043)****[Lịch sử phát triển](#_Toc169704043)** [5](#_Toc169704043)

**[2.](#_Toc169704044)****[Nguyên lý cơ bản](#_Toc169704044)** [7](#_Toc169704044)

**[3.](#_Toc169704045)****[Thực hiện triển khai](#_Toc169704045)** [9](#_Toc169704045)

**[4.](#_Toc169704046)****[Kết quả thử nghiệm](#_Toc169704046)** [12](#_Toc169704046)

**[Phần III Semi Supervised Fuzzy c-Means Clustering](#_Toc169704047)** [18](#_Toc169704047)

**[1.](#_Toc169704048)****[Giới thiệu SSFCM](#_Toc169704048)** [18](#_Toc169704048)

**[a.](#_Toc169704049)****[Định nghĩa](#_Toc169704049)** [18](#_Toc169704049)

**[2.](#_Toc169704050)****[Nguyên lý cơ bản](#_Toc169704050)** [18](#_Toc169704050)

**[3.](#_Toc169704051)****[Thực hiện triển khai](#_Toc169704051)** [22](#_Toc169704051)

**[4.](#_Toc169704052)****[Kết quả thử nghiệm](#_Toc169704052)** [25](#_Toc169704052)

**[Phần IV Tài liệu tham khảo](#_Toc169704053)** [33](#_Toc169704053)

**[1.](#_Toc169704054)****[FCM](#_Toc169704054)** [33](#_Toc169704054)

**[2.](#_Toc169704055)****[SSFCM](#_Toc169704055)** [33](#_Toc169704055)

**[3.](#_Toc169704056)****[Các chỉ số đánh giá](#_Toc169704056)** [33](#_Toc169704056)

**Phần I** **Giới thiệu**

Phân loại mẫu là một trong những lĩnh vực quan trọng của học máy, nghiên cứu về cách tự động phân loại các đối tượng dữ liệu vào từng nhóm dựa trên các đặc trưng và thuộc tính. Mục đích chính là học một mô hình từ dữ liệu huấn luyện có sẵn và sau đó sử dụng mô hình này để dự đoán nhãn cho các điểm dữ liệu mới, không được nhãn trước. Phân loại mẫu có ứng dụng rộng rãi trong nhiều lĩnh vực, từ xử lý ảnh và nhận dạng giọng nói đến dự báo tài chính và chẩn đoán y tế. Phân loại mẫu có hai phương pháp phân loại, một là phân loại giám sát và một là phân loại không giám sát.

Phân loại không giám sát, gọi là phân cụm và phân loại dữ liệu ngoại trừ tiêu chí bên ngoài, rất hữu ích trong các phương pháp phân loại mẫu dữ liệu vì vậy nó đã được áp dụng trong nhiều lĩnh vực. Phân cụm cứng c-means (HCM) và phân cụm mờ c-means (FCM) là các phương pháp điển hình của phân cụm không phân cấp.Trong bài báo cáo này sẽ giới thiệu về phân cụm mờ c-means (FCM) và được trình bày chi tiết ở Phần II.

Phân loại có giám sát giúp mô hình học được từ các ví dụ có sẵn và có khả năng dự đoán chính xác hơn trên dữ liệu mới. Vì có thông tin nhãn rõ ràng, các mô hình phân loại giám sát thường có thể đạt được độ chính xác cao hơn so với các phương pháp không giám sát. Nhưng trong thực tế lượng dữ liệu có nhãn ít hơn rất nhiều dữ liệu không có nhãn do đó để có thể tận dụng tối đa nguồn dữ liệu có sẵn một phương pháp mới được tạo ra bằng cách kết hợp hai phương pháp phân loại có giám sát và phân loại không giám sát là phân cụm bán giám sát Semi-Supervised Fuzzy c-Means (SSFCM) sẽ được trình bày chi tiết ở phần III.

Ý nghĩa của phân cụm là cung cấp một cách để tổ chức dữ liệu một cách có cấu trúc và hợp lý, từ đó giúp nhận diện các mẫu ẩn trong dữ liệu, tìm ra các nhóm có tính chất chung và giúp dự đoán xu hướng, xu hướng trong tương lai. Phân cụm được ứng dụng rộng rãi trong nhiều lĩnh vực của đời sống: Maketing, ngân hàng, bảo hiểm, y tế, khoa học, internet, ....

# **Phần II Fuzzy c-Means Clustering**

1. **Giới thiệu FCM**
2. **Định nghĩa**

Fuzzy C-Means (FCM) là một thuật toán phân cụm (clustering) thuộc nhóm fuzzy clustering, được sử dụng rộng rãi trong các bài toán phân tích dữ liệu và học máy. Khác với các thuật toán phân cụm truyền thống như K-means, FCM cho phép mỗi điểm dữ liệu có thể thuộc về nhiều cluster với các mức độ thành viên khác nhau. Điều này có nghĩa là thay vì gán một điểm dữ liệu vào một cluster duy nhất, FCM xác định mức độ (membership degree) mà điểm dữ liệu thuộc về từng cluster.

FCM tối ưu hóa một hàm mục tiêu nhằm giảm thiểu khoảng cách giữa các điểm dữ liệu và các tâm cluster, đồng thời điều chỉnh các mức độ thành viên để phản ánh sự gần gũi giữa điểm dữ liệu và các tâm cluster.

1. **Lịch sử phát triển**

Thuật toán Fuzzy C-Means được phát triển bởi J.C. Bezdek vào năm 1981. Bezdek mở rộng công trình của Jim Bezdek và các nhà nghiên cứu khác trong lĩnh vực clustering và lý thuyết mờ (fuzzy theory). Lý thuyết mờ, được giới thiệu bởi Lotfi A. Zadeh vào năm 1965, là cơ sở nền tảng cho sự phát triển của FCM. Zadeh đưa ra khái niệm tập mờ (fuzzy set), trong đó mỗi phần tử có một mức độ thành viên liên tục từ 0 đến 1, thay vì chỉ có thể thuộc hoặc không thuộc như trong tập hợp cổ điển.

FCM xây dựng trên ý tưởng này bằng cách áp dụng nguyên tắc mờ vào quá trình phân cụm dữ liệu. Bezdek và các cộng sự đã phát triển và tinh chỉnh thuật toán qua nhiều nghiên cứu, chứng minh tính hiệu quả của FCM trong việc phân cụm dữ liệu phức tạp và không rõ ràng. Từ đó, FCM đã trở thành một công cụ quan trọng trong các lĩnh vực như xử lý ảnh, phân tích dữ liệu y khoa, khai phá dữ liệu, và nhiều ứng dụng khoa học khác.

Thuật toán FCM đã trải qua nhiều cải tiến và điều chỉnh nhằm tăng cường hiệu suất và khả năng ứng dụng. Những phiên bản và biến thể khác nhau của FCM đã được phát triển để xử lý các dữ liệu có kích thước lớn, dữ liệu không đồng nhất, và các bài toán phân cụm đòi hỏi độ chính xác cao hơn. Các nghiên cứu tiếp tục mở rộng khả năng của FCM, ứng dụng nó vào nhiều lĩnh vực khác nhau và cải thiện hiệu suất thông qua các phương pháp tối ưu hóa hiện đại.

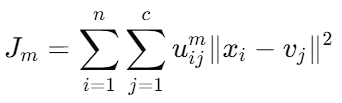
1. **Nguyên lý cơ bản**

Giả sử X={x1, x2, x3, ..., xn} là một bộ dữ liệu có n điểm dữ liệu, mỗi điểm được biểu diễn bằng một vecto p chiều (p quan sát được trong không gian Euclide p chiều) x*k,j* là đặc trưng thứ *j* của điểm dữ liệu thứ k trong X. X được phân chia thành c cụm V={v1, v2, v3, ...v*c*}, v*i* là tâm của cụm *i* được biểu diễn bằng một vecto p chiều. Mỗi điểm dữ liệu có c giá trị thành viên tương ứng với từng cụm. Các giá này nằm trong khoảng [0, 1] biểu diễn sự tương đồng của điểm dữ liệu đó với mỗi cụm. Giá trị thành viên gần bằng 1 biểu thị mức độ tương đồng cao giữa điểm dữ liệu và cụm, trong khi thành viên gần bằng 0 ngụ ý ít tương đồng giữa điểm dữ liệu và cụm. Ma trận thành viên U có kích thước (n x c) sẽ lưu giá trị của mỗi điểm đối với từng cụm.

**Mục tiêu**: Chia dữ liệu thành các nhóm mà mỗi điểm dữ liệu có một mức độ thành viên nhất định thuộc về mỗi cụm, tìm ra các tâm cụm V và ma trận thành viên U sao cho hàm mục tiêu J*m*​ được tối thiểu hóa. Điều này đảm bảo rằng các điểm dữ liệu gần với các tâm cụm mà chúng có độ thành viên cao nhất, tạo ra các cụm chặt chẽ và hợp lý.

Ý tưởng thực hiện là dựa trên hàm mục tiêu, thuật toán FCM lặp lại quá trình cập nhật ma trận thành viên U và tâm cụm V​. Quá trình này tiếp tục cho đến khi hàm mục tiêu hội tụ (không thay đổi đáng kể nữa), tức là khi đạt được sự phân cụm ổn định và tối ưu.

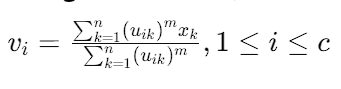
Hàm mục tiêu:



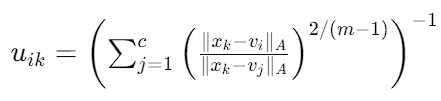
Trong đó:

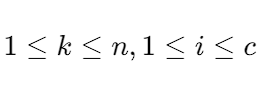
* n là số lượng điểm dữ liệu trong tập dữ liệu X.
* c là số lượng cụm.
* u*ij* là giá trị thành viên của điểm x*i* trong cụm v*j*.
* m là tham số fuzziness thường có giá trị nằm trong khoảng [1, 3].
* x*i* là điểm dữ liệu thứ i trong tập dữ liệu X, biểu diễn bằng vecto p chiều.
* v*j* là tâm cụm thứ *j*, biểu diễn bằng vecto p chiều.
* ∥x*i* ​− v*j*​∥ là khoảng cách Euclid giữa điểm dữ liệu x*i​* và tâm cụm v*j​*

Công thức tính tâm cụm:

 (1)

Công thức tính ma trận thành viên:

 (2)



1. **Thực hiện triển khai**

**B1:** Dựa vào các tham số được truyền vào gồm {c(số cụm), m(hệ số mờ), MAX\_ITER số lần lặp tối đa, epsilon} khởi tạo ma trận thành viên ban đầu U có kích thước n x c. n là số lượng dữ liệu có trong tập dữ liệu.

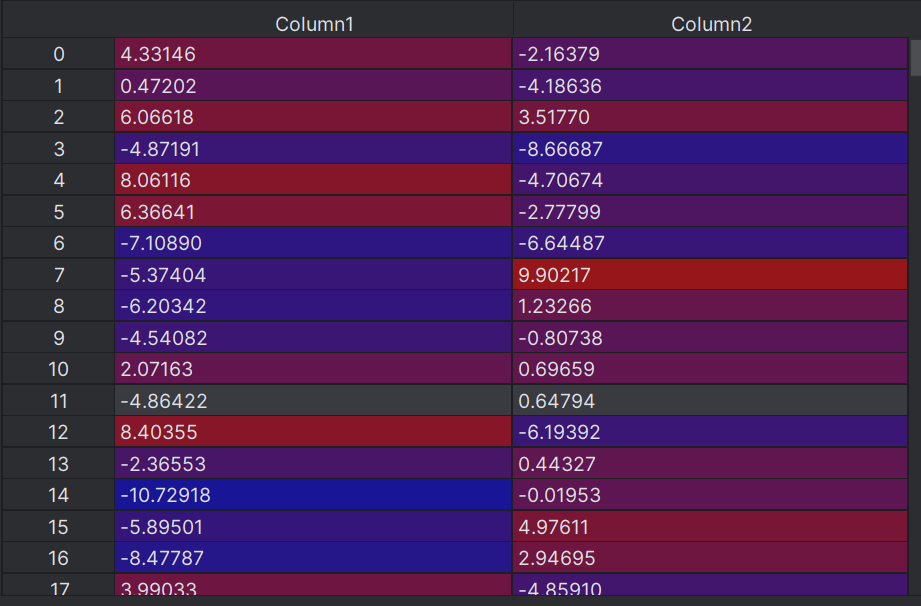
**B2:** Tại bước k (k=1, 2, ......, MAX\_ITER), tính toán lại các tâm cụm v*i* với công thức 1.

**B3:** Tính toán và cập nhật cho ma trận thành viên U bằng công thức 2.

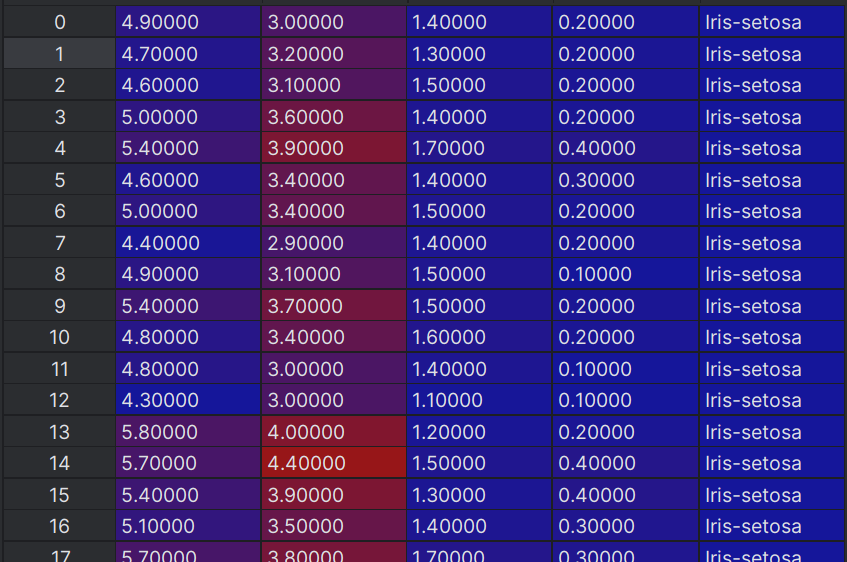
**B4:** So sánh U*(k+1)* với U*(k)* nếu

||U*(k+1)*-u *(k)*|| < epsilon hoặc k >MAX\_ITER dừng lại. Nếu không đặt U(k) = U(k+1) và quay lại **B2**.

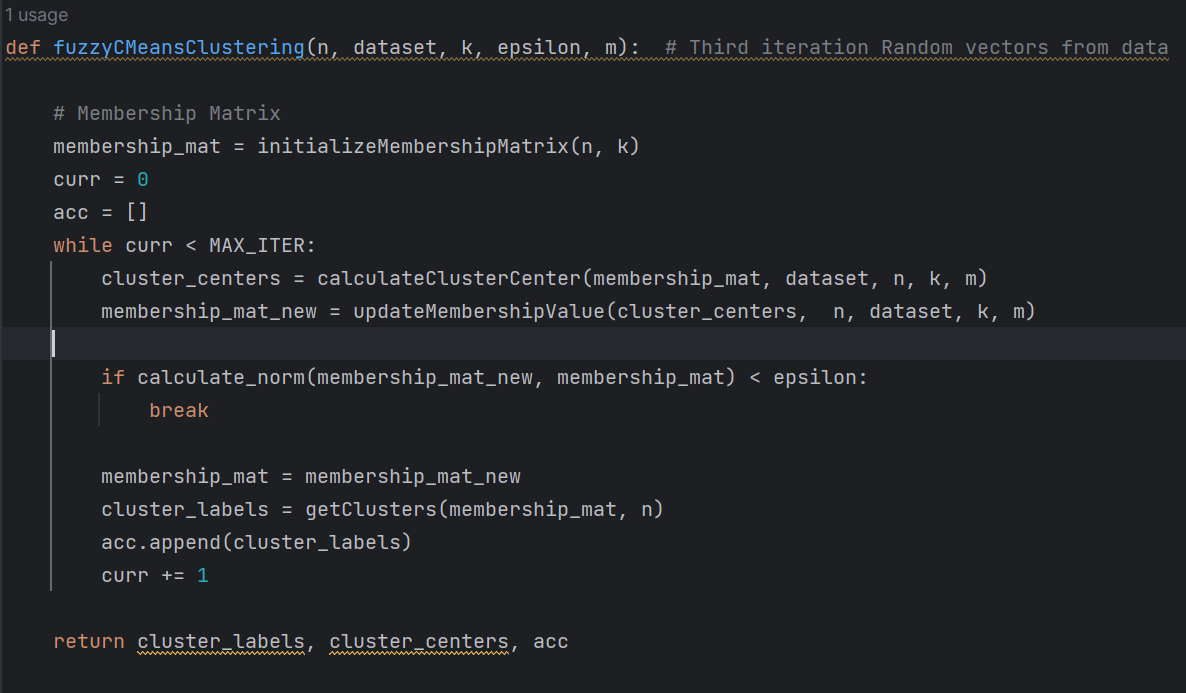
Bộ dữ liệu được sử dụng cho thuật toán này gồm dataset(x, y) và Iris. Dataset(x, y) có 250 điểm dữ liệu. Mỗi điểm được biểu diễn bằng một vecto 2 chiều. Các điểm này chưa có nhãn và sẽ được thuật toán FCM gán nhãn. Iris là bộ dữ liệu về hoa gồm 4 thuộc tính và có 150 dữ liệu.

Dataset(x, y):

Iris:



Thuật toán được cài đặt bằng python

****

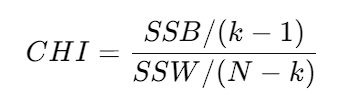
* membership\_mat là ma trận thành viên sẽ được khởi tạo thông qua hàm initializeMembershipMatrix(), được cập nhật trong mỗi lần lặp thông qua hàm updateMembershipValue().
* Hàm caculateClusterCenter() tính tâm của mỗi cụm và trả về 1 table trong đó mỗi hàng là một vecto biểu diễn tâm của 1 cụm.
* Hàm caculate\_norm() trả về kết quả của ||U*(k+1)*-U *(k)*||, sau đó thục hiện so sánh với epsilon nếu bé hơn thì kết thúc thuật toán.
* Hàm getClusters() tìm cụm của mỗi điểm trong lần lặp đó thông qua ma trận thành viên. Và trả về 1 danh sách các cụm của các điểm dữ liệu.
* Cuối cùng thuật toán sẽ trả về
  + - cluster\_labels: danh sách chứa cụm tương ứng của từng điểm.
    - cluster\_centers: tâm của các cụm.
    - acc: danh sách chứa cụm tương ứng của từng điểm trong từng lần lặp.

1. **Kết quả thử nghiệm**

Đánh giá kết quả phân cụm bằng các chỉ số đánh giá gồm: thời gian chạy thuật toán, tỷ lệ phương sai Calinski-Harabasz Index (CHI), tỷ lệ phương sai Davies-Bouldin Index (DBI).

**Tỷ lệ phương sai CHI**, còn được gọi là **Calinski-Harabasz Index**, là một thước đo chất lượng phân cụm được sử dụng rộng rãi trong phân tích dữ liệu. Giá trị CHI càng lớn thì chất lượng phân cụm càng tốt. Điều này có nghĩa là các cụm được phân biệt rõ ràng hơn, các cụm có sự khác biệt rõ rệt và các điểm dữ liệu trong cùng một cụm gần nhau.

**Công thức:**

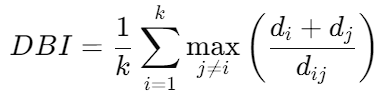
****

Trong đó:

* SSB là tổng các phương sai giữa các cụm (Between-Cluster Sum of Squares).
* SSW là tổng các phương sai trong các cụm (Within-Cluster Sum ofSquares).
* k là số lượng cụm.
* N là số lượng điểm dữ liệu.

Tỷ lệ phương sai **DBI**, hay **Davies-Bouldin Index**, là một thước đo khác để đánh giá chất lượng của các cụm. DBI đo lường mức độ tương tự giữa các cụm, tức là sự "chồng chéo" giữa các cụm. Giá trị DBI càng nhỏ thì chất lượng phân cụm càng tốt. Điều này có nghĩa là các cụm được phân biệt rõ ràng hơn và ít bị chồng chéo, các điểm dữ liệu trong cùng một cụm gần nhau.

Công thức:



Trong đó:

* d*i​* là độ lệch chuẩn của cụm *i*.
* d*j*​ là độ lệch chuẩn của cụm *j*.
* d*ij*​ là khoảng cách giữa trung tâm của cụm *i* và trung tâm của cụm *j*.
* k là số lượng cụm.

Các kết quả thu được khi chạy thử

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STT | m | k | Epsilon | CHI | DBI | Time(s) |
| 1 | 1.6 | 2 | 0.001 | 324.017 | 0.759 | 0.211 |
| 2 | 1.7 | 2 | 0.001 | 323.01 | 0.759 | 0.241 |
| 3 | 1.8 | 2 | 0.001 | 321.422 | 0.759 | 0.199 |
| 4 | 1.8 | 3 | 0.001 | 261.484 | 0.974 | 1.210 |
| 5 | 1.8 | 4 | 0.001 | 244.955 | 0.847 | 1.048 |
| 6 | 1.8 | 4 | 0.00001 | 244.950 | 0.847 | 2.613 |
| 7 | 1.8 | 4 | 0.000001 | 244.949 | 0.847 | 2.647 |

Đánh giá các kết quả thu được:

Ảnh hưởng từ m:

Khi m tăng từ 1.6 đến 1.8, chỉ số CHI có xu hướng giảm (từ 324.017 xuống 321.422). Điều này cho thấy rằng việc tăng giá trị m dẫn đến phân cụm ít rõ ràng hơn, tức là các cụm ít được phân biệt hơn.

Ảnh hưởng từ số cụm k:

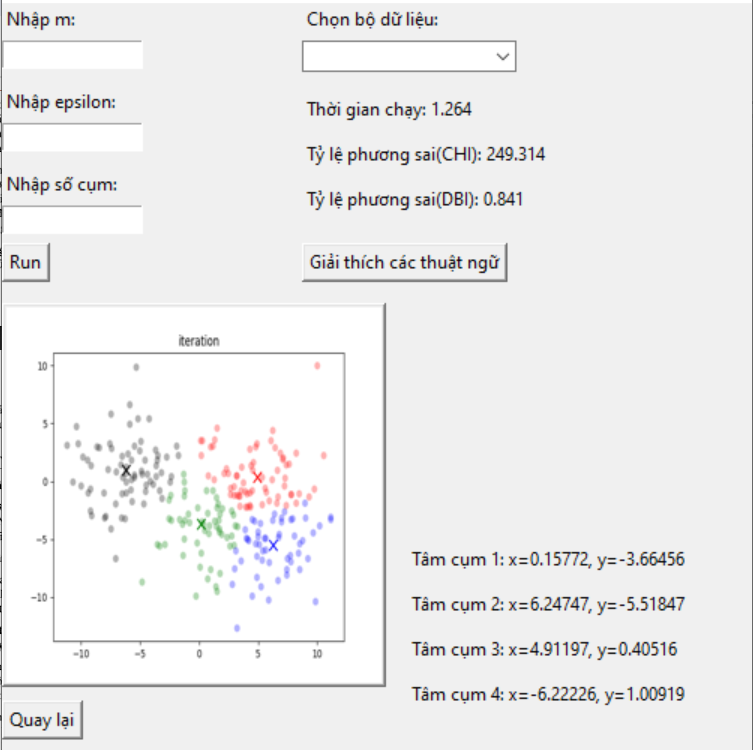
Khi số cụm k tăng từ 2 lên 4, chỉ số CHI giảm (từ 321.422 xuống 244.955). Tuy nhiên, điều này không nhất thiết là xấu vì số cụm tăng thường làm giảm tính phân biệt giữa các cụm.

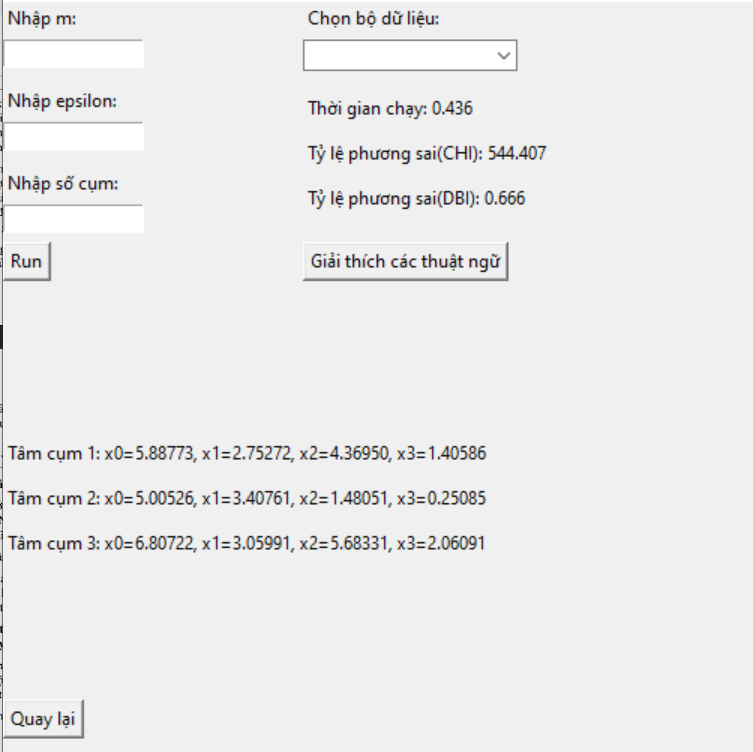
Chỉ số DBI cũng có sự thay đổi, tăng khi k tăng lên 3 (0.974) và sau đó giảm xuống khi k là 4 (0.847). Điều này cho thấy việc chọn số cụm phù hợp là rất quan trọng để giảm thiểu chồng lấn giữa các cụm.

Ảnh hưởng từ Epsilon:

Giá trị epsilon nhỏ hơn dẫn đến việc tính toán chính xác hơn, nhưng thời gian thực hiện tăng lên (ví dụ, từ 1.048s cho epsilon = 0.001 đến 2.613s cho epsilon = 0.00001). Tuy nhiên, sự khác biệt về CHI và DBI không quá lớn khi epsilon thay đổi, cho thấy rằng việc tăng độ chính xác không ảnh hưởng nhiều đến chất lượng phân cụm trong một số trường hợp.

Demo chương trình:

Dataset(x, y) 

Iris 

# **Phần III Semi Supervised Fuzzy c-Means Clustering**

1. **Giới thiệu SSFCM**
2. **Định nghĩa**

SSFCM, hay **Semi-Supervised Fuzzy c-Means**, là một biến thể của thuật toán Fuzzy c-Means (FCM) truyền thống, kết hợp dữ liệu có nhãn (labeled data) và dữ liệu không nhãn (unlabeled data) để cải thiện hiệu quả phân cụm. Thuật toán này tận dụng thông tin từ dữ liệu có nhãn để dẫn dắt quá trình phân cụm, giúp tăng độ chính xác và ý nghĩa của các cụm. Thuật toán được xây dựng dựa trên FCM và tạo một ma trận thành viên bán giám sát U, các điểm dữ liệu được giám sát sẽ được khai báo trong ma trận này.

1. **Nguyên lý cơ bản**

Giả sử X={x1, x2, x3, ..., xn} là một bộ dữ liệu có n điểm dữ liệu, mỗi điểm được biểu diễn bằng một vecto p chiều (p quan sát được trong không gian Euclide p chiều) x*k,j* là đặc trưng thứ *j* của điểm dữ liệu thứ k trong X. X được phân chia thành c cụm V={v1, v2, v3, ...v*c*}, v*i* là tâm của cụm *i* được biểu diễn bằng một vecto p chiều. Mỗi điểm dữ liệu có c giá trị thành viên tương ứng với từng cụm. Các giá này nằm trong khoảng [0, 1] biểu diễn sự tương đồng của điểm dữ liệu đó với mỗi cụm. Giá trị thành viên gần bằng 1 biểu thị mức độ tương đồng cao giữa điểm dữ liệu và cụm, trong khi thành viên gần bằng 0 ngụ ý ít tương đồng giữa điểm dữ liệu và cụm. Ma trận thành viên U có kích thước (n x c) sẽ lưu giá trị của mỗi điểm đối với từng cụm. Ma trận thành viên bán giám sát U có kích thức (n x c) lưu các giá trị để thể hiện các điểm dữ liệu được gán nhãn và thuộc về 1 cụm nào đó.

Ma trận thành viên bán giám sát U​ có các phần tử u*ki* được định nghĩa như sau:



Trong đó:

* u*ki* là mức độ thành viên của điểm dữ liệu thứ *k* trong cụm thứ *i*.
* n là số lượng điểm dữ liệu.
* c là số lượng cụm.

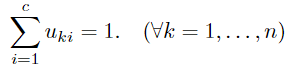
**Mục tiêu**: Chia dữ liệu thành các nhóm mà mỗi điểm dữ liệu có một mức độ thành viên nhất định thuộc về mỗi cụm, tìm ra các tâm cụm V và ma trận thành viên U sao cho hàm mục tiêu J(U, V)​ được tối thiểu hóa. Điều này đảm bảo rằng các điểm dữ liệu gần với các tâm cụm mà chúng có độ thành viên cao nhất, tạo ra các cụm chặt chẽ và hợp lý. Hàm J(U, V) được rút ra bằng cách đưa U vào hàm mục tiêu củ FCM.

Ý tưởng thực hiện là dựa trên hàm mục tiêu, thuật toán FCM lặp lại quá trình cập nhật ma trận thành viên U và tâm cụm V​. Quá trình này tiếp tục cho đến khi hàm mục tiêu hội tụ (không thay đổi đáng kể nữa), tức là khi đạt được sự phân cụm ổn định và tối ưu.

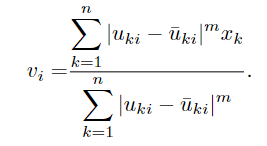
Hàm mục tiêu J(U, V):



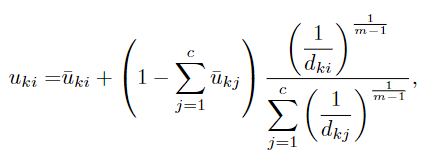
Trong đó:

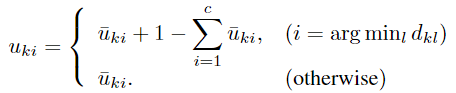
* U = {u*ki*} là ma trận thành viên.
* u*ki* là giá trị thành viên của dữ liệu được giám sát.
* m là hệ số mờ.
* ||x*k* – v*i*|| là khoảng cách Euclide giữa điểm dữ liệu x*k* và tâm cụm v*i*.
* 

Công thức tính tâm cụm:

 (3)

Công thức tính và cập nhật ma trận thành viên U:

m > 1, (4)

m = 1, (5)

Trong đó:

* d*ki* là bình phương khoảng cách Euclide giữa điểm dữ liệu x*k* và tâm cụm v*i*.
* 

1. **Thực hiện triển khai**

**B1:** Khởi tạo ma trận thành viên bán giám sát U {u*ki*}, đối với những điểm được giám sát u*ki* sẽ được gán bằng 0.51.

**B2:** Khởi tạo ma trận thành viên U

**B3:** Tính các tâm cụm V{v*i*}

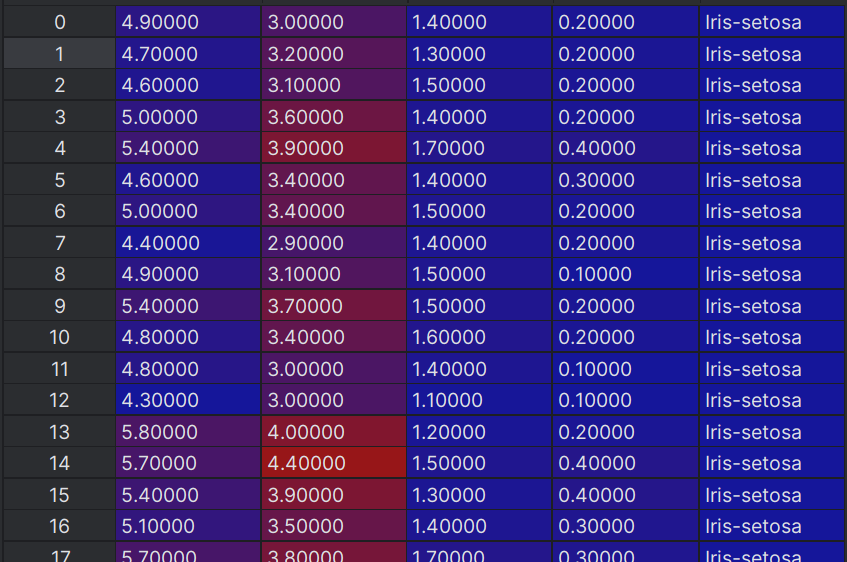
**B4:** Tính toán và cập nhật lại các giá trị của ma trận thành viên

**B5:** So sánh U*(k+1)* với U*(k)* nếu

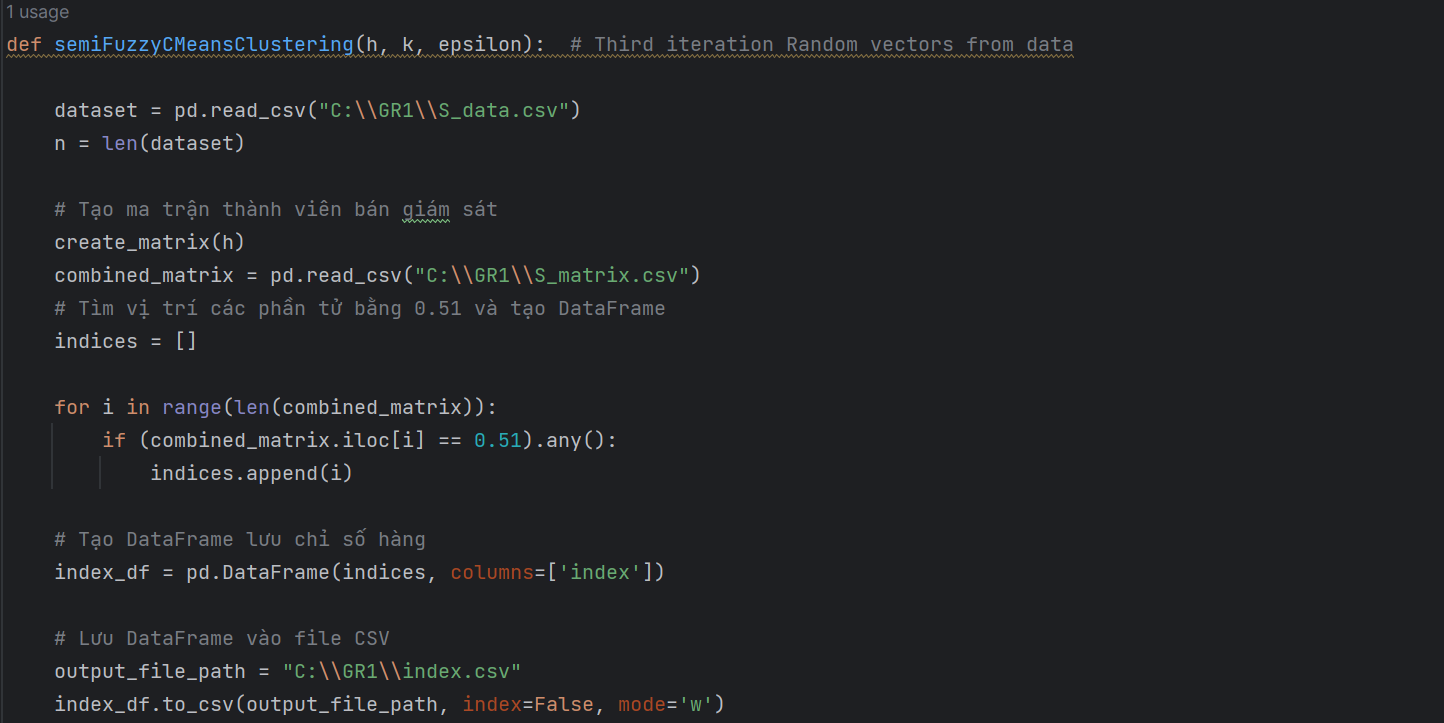
||U*(k+1)-*U*(k)*|| < epsilon hoặc k > MAX\_ITER dừng lại. Nếu không đặt U*(k)* = U*(k+1)* và quay lại **B3**.

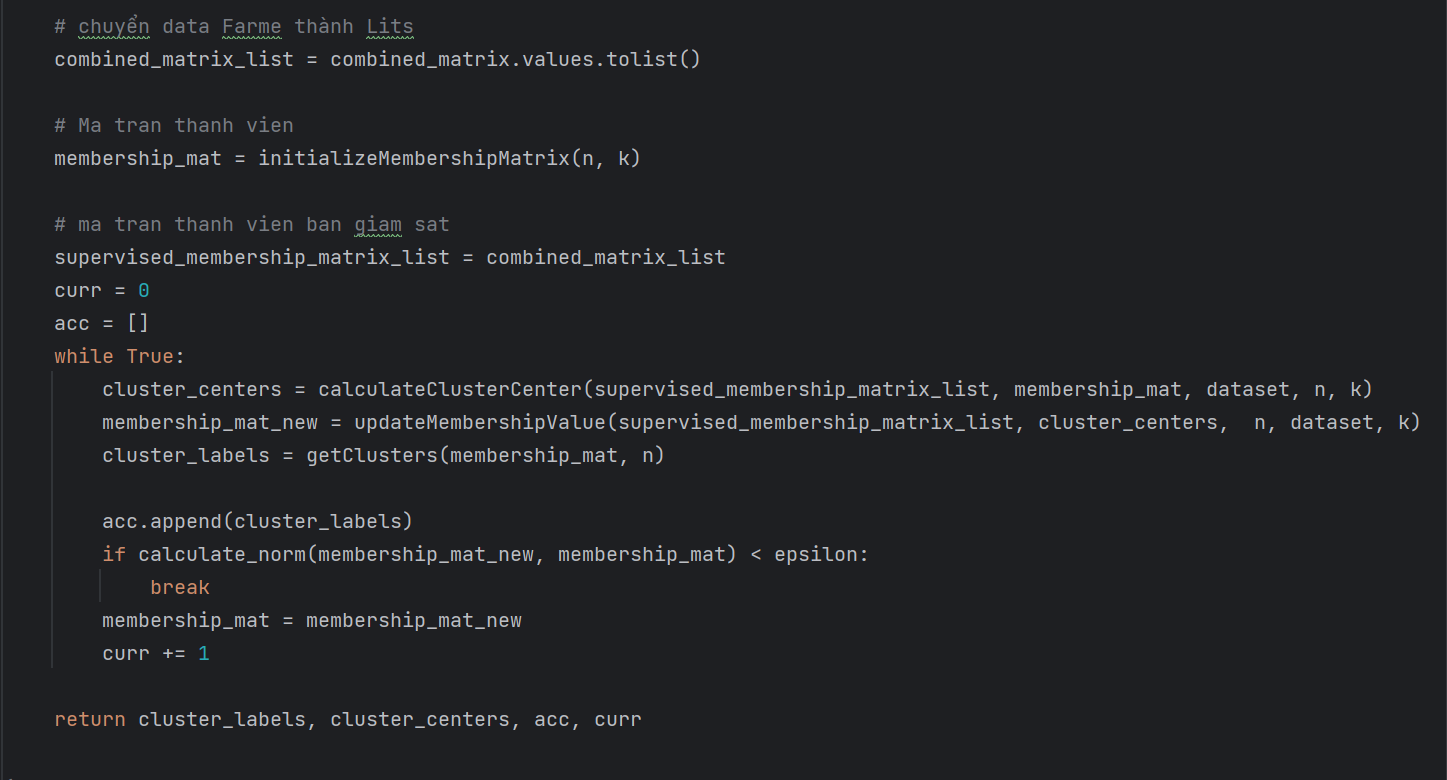
Bộ dữ liệu được sử dụng cho thuật toán này là bộ dữ liệu về các loài hoa Iris gồm có 150 dữ liệu. Mỗi điểm dữ liệu gồm 4 thuộc tính (SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWidthCm, Species) và các điểm dữ liệu đều có nhãn (Species).

Thực hiện tách nhãn của bộ dữ liệu ban đầu. Các thuộc tính được lưu vào file S\_data.csv, nhãn sẽ được lưu vào file S\_nhan.csv.

Bộ dữ liệu Iris

Cài đặt thuật toán bằng python





* Supervised\_membership\_matrix\_lits là ma trận thành viên bán giám sát U. Được tạo bởi hàm create\_matrix() có đầu vào là phần trăm dữ liệu được chọn gán nhãn.
* Hàm caculateClusterCenter() tính tâm của mỗi cụm và trả về 1 table trong đó mỗi hàng là một vecto biểu diễn tâm của 1 cụm.
* Hàm updateMembershipValue() tính toán lại các giá trị thành viên và trả về một ma trận thành viên mới U*(k+1)*.
* Hàm getClusters() thực hiện cập nhật lại cụm cho mỗi điểm dữ liệu trả về 1 list danh sách cụm của các điểm dữ liệu.
* Hàm caculate\_norm() trả về kết quả của ||U*(k+1)*-U *(k)*||, sau đó thục hiện so sánh với epsilon nếu bé hơn thì kết thúc thuật toán.
* Cuối cùng thuật toán trả về
  + - cluster\_labels: danh sách chứa cụm tương ứng của từng điểm.
    - cluster\_centers: tâm của các cụm.
    - acc: danh sách chứa cụm tương ứng của từng điểm trong từng lần lặp.
    - curr: số lần lặp

1. **Kết quả thử nghiệm**

Đánh giá kết quả phân cụm bằng các chỉ số đánh giá gồm: thời gian chạy thuật toán, Rand index, Accuracy, Precision, Recall, F1 và các thông số đánh giá độ đo trong (CHI, DBI) đã giới thiệu ở thuật toán Fuzzy c-Means.

Rand Index đánh giá sự tương đồng giữa hai phép phân cụm bằng cách tính toán số lượng cặp điểm dữ liệu được phân cụm giống nhau trong cả hai phép phân cụm và số lượng cặp điểm dữ liệu được phân cụm khác nhau trong cả hai phép phân cụm.

Rand Index không cần đồng bộ nhãn cụm và nhãn lớp.

Công thức:



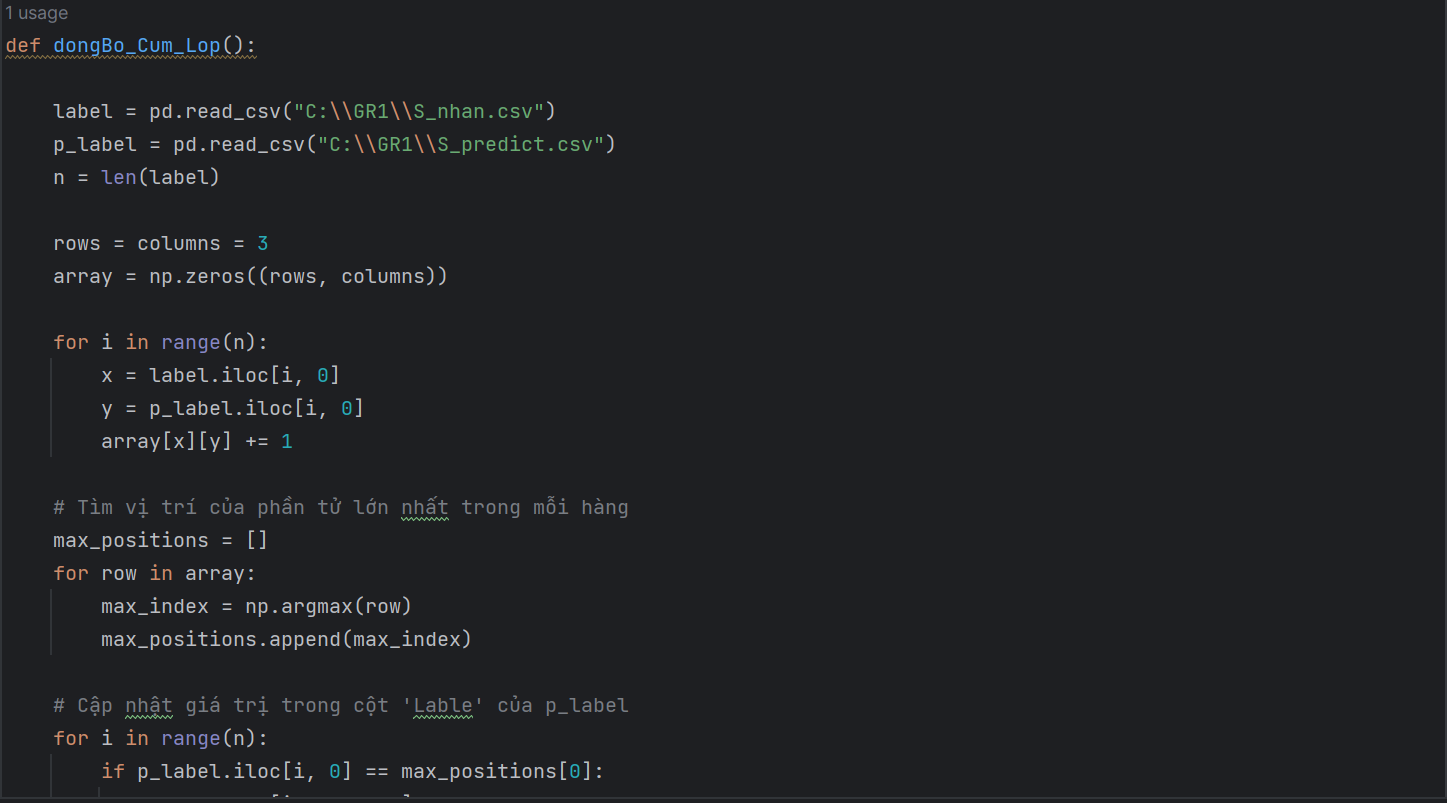
Trong đó:

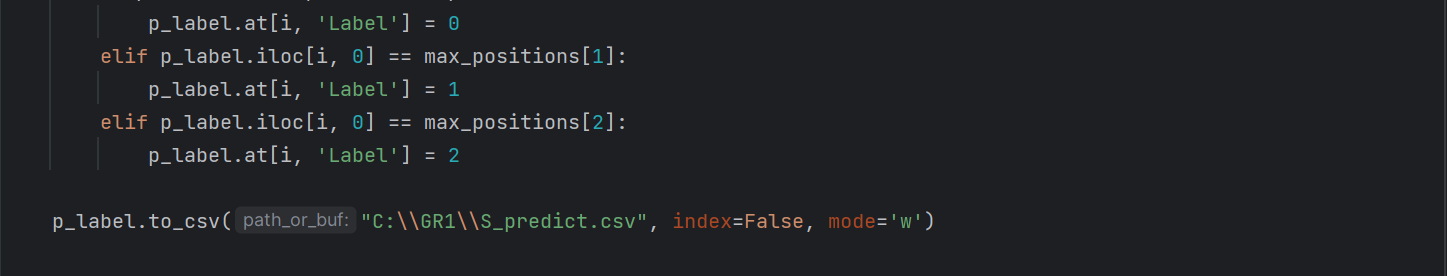
* a: Số cặp điểm dữ liệu cùng nhãn lớp và cùng nhãn cụm.
* b: Số cặp điểm dữ liệu khác nhãn lớp và khác nhãn cụm.
* c: Số cặp điểm dữ liệu cùng nhãn lớp nhưng khác nhau nhãn cụm.
* d: Số cặp điểm dữ liệu khác nhãn lớp nhưng cùng nhãn cụm.

Với các chỉ số khác như Accuracy, Precision, Recall, F1 cần phải đồng bộ nhãn lớp và nhãn cụm. Để đồng bộ nhãn lớp và nhãn cụm ta sẽ cần lập ma trận (c x c), c là số cụm. Duyệt qua từng điểm dữ liệu kiểm tra sự thay đổi giữa nhãn lớp và nhãn cụm để cập nhật cho ma trận.

Dựa vào những vị trí có trọng số lớn nhất trong mỗi hàng để đồng bộ nhãn cụm và nhãn lớp.

Cài đặt thuật toán để đồng bộ nhãn cụm và nhãn lớp bằng python



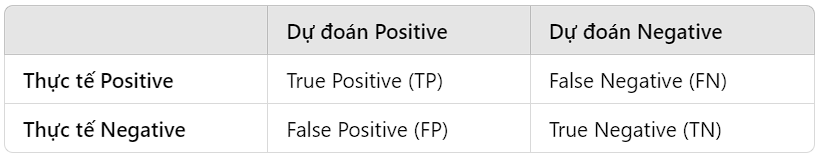


Accuracy là một thước đo hiệu suất quan trọng trong các mô hình phân loại, dùng để đánh giá tỷ lệ dự đoán đúng trên tổng số dự đoán. Nó được định nghĩa là tỷ lệ giữa số lượng dự đoán đúng và tổng số lượng dự đoán. Chỉ số Accuracy của từng lớp là nhue nhau .

Công thức:



Trong đó:



Với các chỉ số như Precision, Recall, F1 thì mỗi lớp sẽ có 1 giá trị riêng.

Precision: đo độ chính xác đối với class được quan tâm.

Recall: đo độ bao phủ đối với các dự đoán về class được quan tâm.

F1: trung bình điều hòa giữa Precision và Recall.

Công thức:

Precision = TP/(TP+FP)

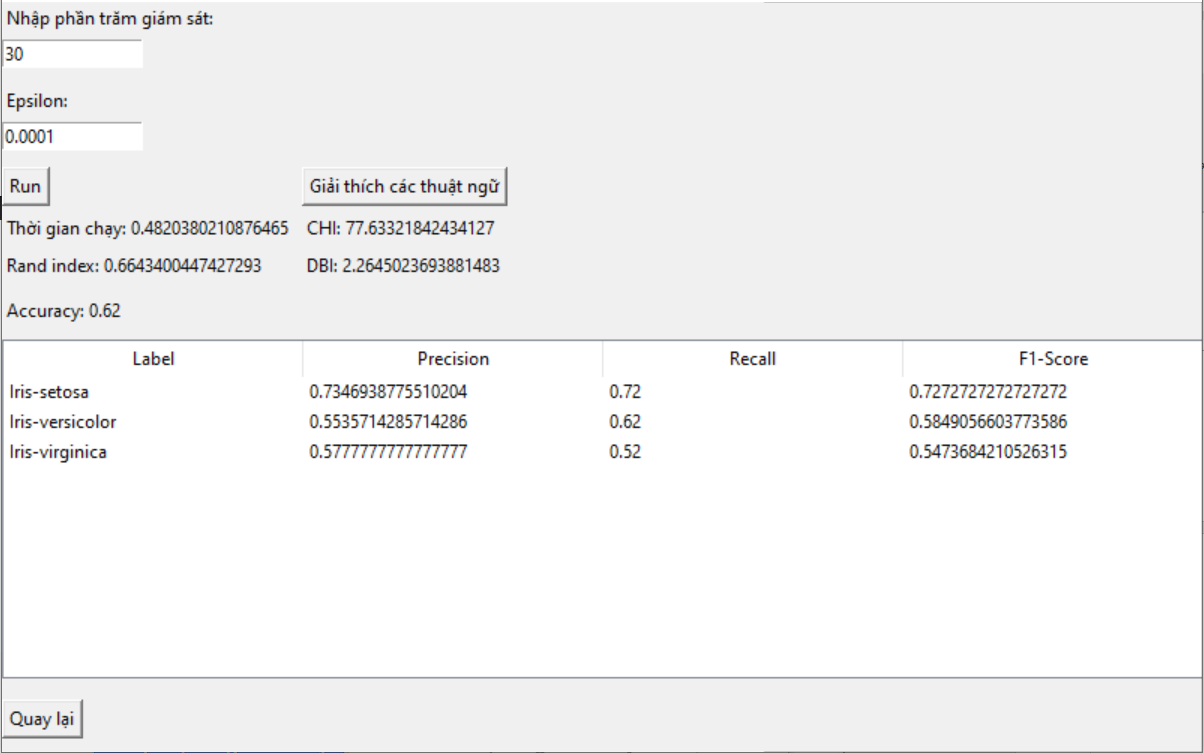
Recall = TP/(TP+FN)

F1 = (2\*Pre\*Re)/(Pre+Re)

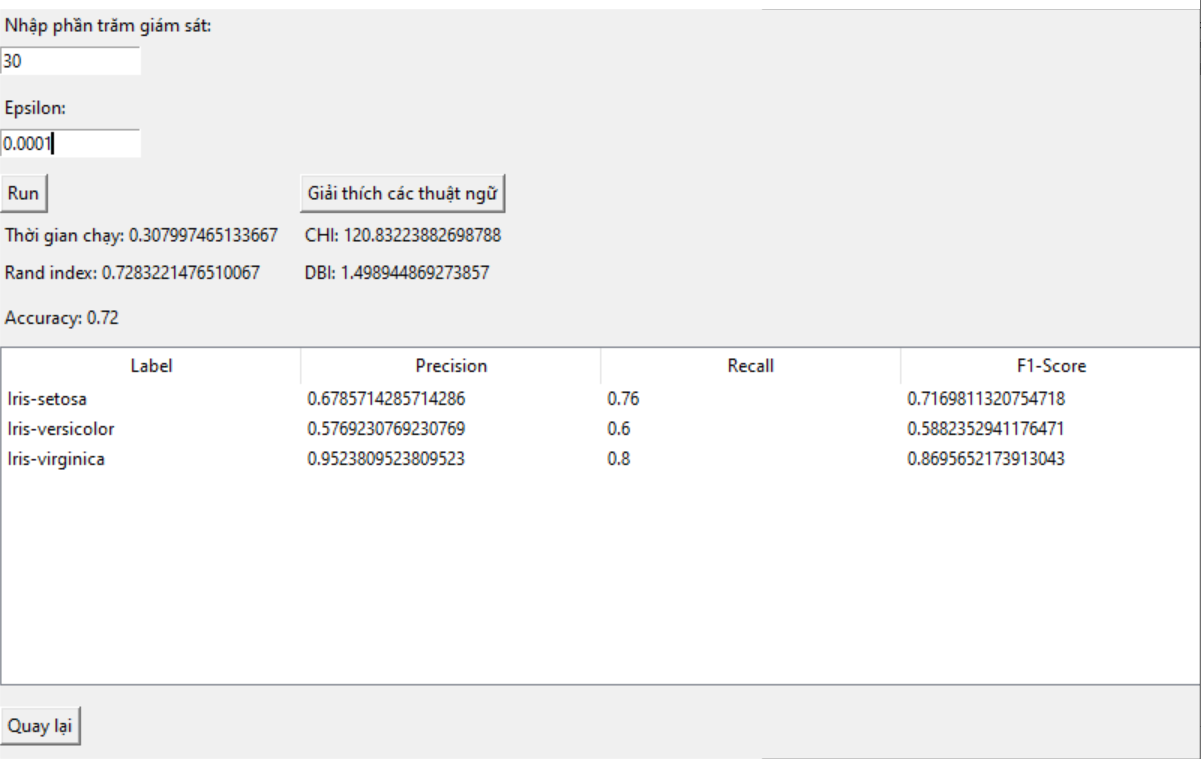
Các kết quả thu được khi chạy thử

Với 30% có nhãn, Epsilon = 0.0001

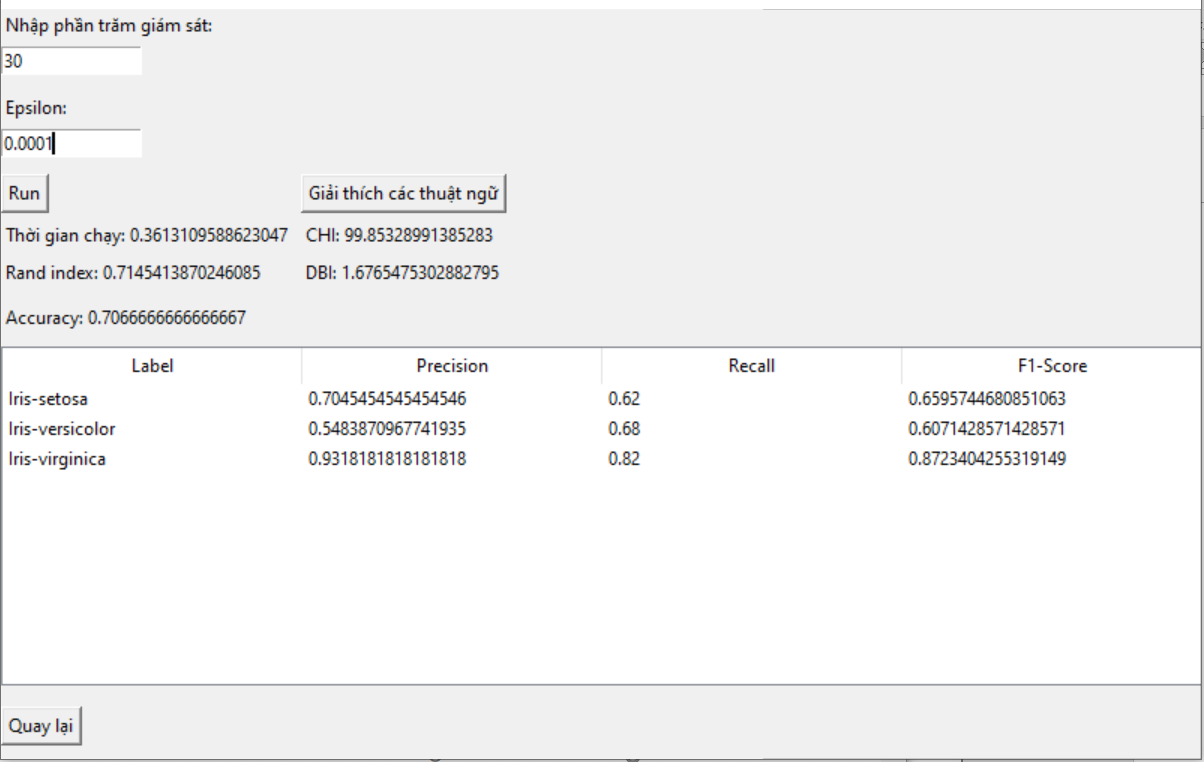
Chạy lần 1



Chạy lần 2

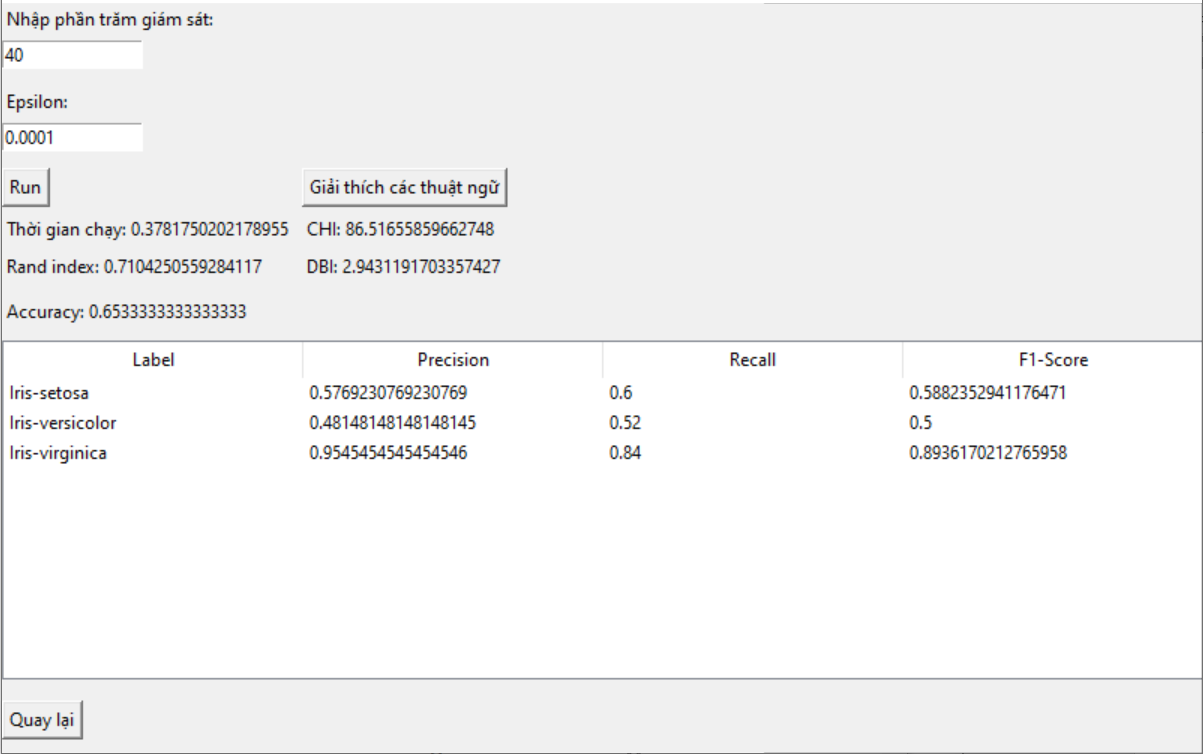


Chạy lần 3

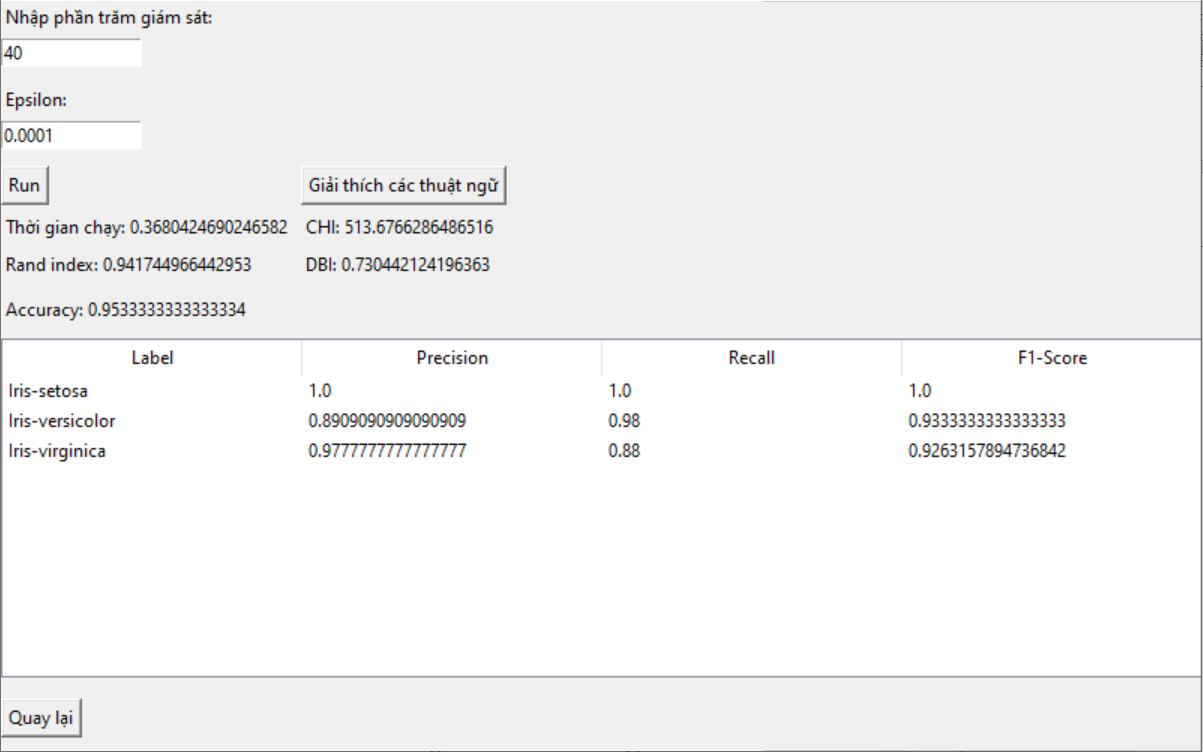


Với 40% có nhãn, Epsilon = 0.0001

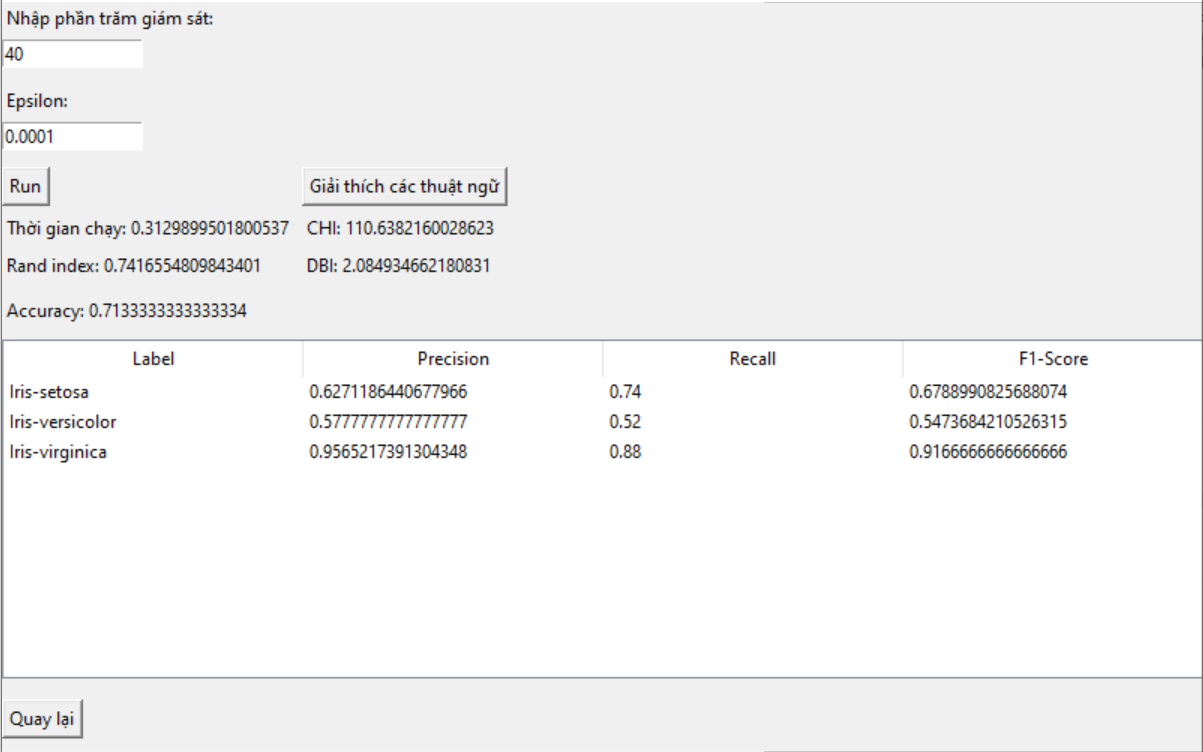
Chạy lần 1



Chạy lần 2



Chạy lần 3



# **Phần IV Tài liệu tham khảo**

1. **FCM**

Một số tài liệu do thầy hướng dẫn cung cấp.

Tài liệu từ [Wikipedia](https://en.wikipedia.org/wiki/Fuzzy_clustering).

Tài liệu từ [Geeksforgeeks](https://www.geeksforgeeks.org/ml-fuzzy-clustering/).

1. **SSFCM**

Một số tài liệu do thầy hướng dẫn cung cấp.

1. **Các chỉ số đánh giá**

[Rand index](https://en.wikipedia.org/wiki/Rand_index), [Accuracy, Precision, Recall, F1](https://www.miai.vn/2020/06/16/oanh-gia-model-ai-theo-cach-mi-an-lien-chuong-2-precision-recall-va-f-score/).

[Calinski-Harabasz Index (CHI).](https://en.wikipedia.org/wiki/Calinski%E2%80%93Harabasz_index)

[Davies-Bouldin Index (DBI).](https://en.wikipedia.org/wiki/Davies%E2%80%93Bouldin_index)