TRƯỜNG ĐẠI HỌC ĐIỆN LỰC

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CHUYÊN ĐỀ HỌC PHẦN**

**NHẬP MÔN HỌC MÁY**

**ĐỀ TÀI:**

**PHÂN LOẠI HOA IRIS BẰNG THUẬT TOÁN K-MEANS**

|  |  |
| --- | --- |
|  |  |
| |  |  | | --- | --- | | **Sinh viên thực hiện** | **: LÊ CHÍ CÔNG TRƯỜNG** | | **Giảng viên hướng dẫn** | **NGÔ TRƯỜNG GIANG** | | | **Ngành** | **: CÔNG NGHỆ THÔNG TIN** | | | **Chuyên ngành** | **: CÔNG NGHỆ PHẦN MỀM** | | | **Lớp** | **: D12CNPM1** | | |  |
|  |  |

***Hà Nội, tháng 06 năm 2020***

**PHIẾU CHẤM ĐIỂM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ và tên sinh viên** | **Nội dung thực hiện** | **Điểm** | **Chữ ký** |
| 1 | Hoàng Minh Phương |  |  |  |
| 2 | Lê Chí Công Trường |  |  |  |

|  |  |  |
| --- | --- | --- |
| **Họ và tên giảng viên** | **Chữ ký** | **Ghi chú** |
| Giảng viên chấm 1: |  |  |
| Giảng viên chấm 2: |  |  |

Contents

[CHƯƠNG 1. GIỚI THIỆU THUẬT TOÁN 4](#_Toc44726699)

[**1.1.** **Giới thiệu thuật toán phân cụm trong học máy** 4](#_Toc44726700)

[**1.2.** **Thuật toán K-means** 4](#_Toc44726701)

[CHƯƠNG 2. ÁP DỤNG THUẬT TOÁN 10](#_Toc44726702)

[**2.1. Mô tả bài toán** 10](#_Toc44726703)

[**2.2. Mô tả tập dữ liệu** 10](#_Toc44726704)

[**2.3. Áp dụng thuật toán k-means vào bài toán** 13](#_Toc44726705)

[KẾT LUẬN 21](#_Toc44726706)

[Tài liệu tham khảo 22](#_Toc44726707)

# CHƯƠNG 1. GIỚI THIỆU THUẬT TOÁN

* 1. **Giới thiệu thuật toán phân cụm trong học máy**

Phân cụm là kỹ thuật rất quan trọng trong khai phá dữ liệu, nó thuộc lớp các phương pháp **Unsupervised Learning** trong Machine Learning. Có rất nhiều định nghĩa khác nhau về kỹ thuật này, nhưng về bản chất ta có thể hiểu *phân cụm là các qui trình tìm cách nhóm các đối tượng đã cho vào các cụm (clusters), sao cho các đối tượng trong cùng 1 cụm tương tự (similar) nhau và các đối tượng khác cụm thì không tương tự (Dissimilar) nhau*.

Mục đích của phân cụm là tìm ra bản chất bên trong các nhóm của dữ liệu. Các thuật toán phân cụm (Clustering Algorithms) đều sinh ra các cụm (clusters). Tuy nhiên, không có tiêu chí nào là được xem là tốt nhất để đánh hiệu của của phân tích phân cụm, điều này phụ thuộc vào mục đích của phân cụm như: data reduction, “natural clusters”, “useful” clusters, outlier detection

*Kỹ thuật phân cụm có thể áp dụng trong rất nhiều lĩnh vực như:*

* Marketing: Xác định các nhóm khách hàng (khách hàng tiềm năng, khách hàng giá trị, phân loại và dự đoán hành vi khách hàng,…) sử dụng sản phẩm hay dịch vụ của công ty để giúp công ty có chiến lược kinh doanh hiệu quả hơn;
* Biology: Phận nhóm động vật và thực vật dựa vào các thuộc tính của chúng;
* Libraries:  Theo dõi độc giả, sách, dự đoán nhu cầu của độc giả…;
* Insurance, Finance: Phân nhóm các đối tượng sử dụng bảo hiểm và các dịch vụ tài chính, dự đoán xu hướng (trend) của khách hàng, phát hiện gian lận tài chính (identifying frauds);
* WWW:  Phân loại tài liệu (document classification);  phân loại người dùng web (clustering weblog);…
  1. **Thuật toán K-means**

K-Means là thuật toán rất quan trọng và được sử dụng phổ biến trong kỹ thuật phân cụm. Tư tưởng chính của thuật toán K-Means là tìm cách phân nhóm các đối tượng (objects) đã cho vào K cụm (K là số các cụm được xác đinh trước, K nguyên dương) sao cho tổng bình phương khoảng cách giữa các đối tượng đến tâm nhóm (centroid ) là nhỏ nhất.

***Thuật toán K-Means thực hiện qua các bước chính sau:***

1.    Chọn ngẫu nhiên K tâm (centroid) cho K cụm (cluster). Mỗi cụm được đại diện bằng các tâm của cụm.

2.    Tính khoảng cách giữa các đối tượng (objects) đến K tâm (thường dùng khoảng cách Euclidean)

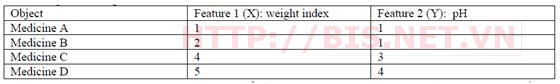
3.    Nhóm các đối tượng vào nhóm gần nhất

4.    Xác định lại tâm mới cho các nhóm

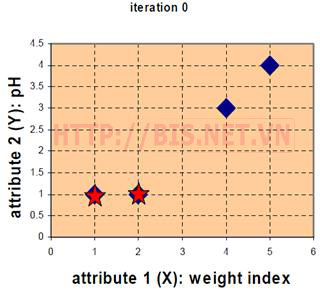
5.    Thực hiện lại bước 2 cho đến khi không có sự thay đổi nhóm nào của các đối tượng

***Ví dụ minh họa thuật toán K-Mean:***

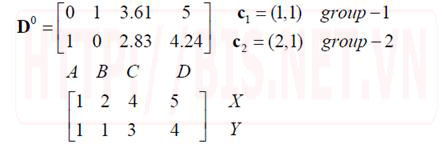
Giả sử ta có 4 loại thuốc A,B,C,D, mỗi loại thuộc được biểu diễn bởi 2 đặc trưng X và Y như sau. Mục đích của ta là nhóm các thuốc đã cho vào 2 nhóm (K=2) dựa vào các đặc trưng của chúng.



**Bước 1.** Khởi tạo tâm (centroid) cho 2 nhóm. Giả sử ta chọn A là tâm của nhóm thứ nhất (tọa độ tâm nhóm thứ nhất c1(1,1)) và B là tâm của nhóm thứ 2 (tạo độ tâm nhóm thứ hai c2 (2,1)).



**Bước 2.** Tính khoảng cách từ các đối tượng đến tâm của các nhóm (Khoảng cách Euclidean)



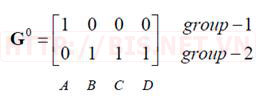
Mỗi cột trong ma trận khoảng cách (D) là một đối tượng (cột thứ nhất tương ứng với đối tượng A, cột thứ 2 tương ứng với đối tượng B,…). Hàng thứ nhất trong ma trận khoảng cách biểu diễn khoảng cách giữa các đối tượng đến tâm của nhóm thứ nhất (c1) và hàng thứ 2 trong ma trận khoảng cách biểu diễn khoảng cách của các đối tượng đến tâm của nhóm thứ 2 (c2).

Ví dụ, khoảng cách từ loại thuốc C=(4,3) đến tâm c1(1,1) là 3.61  và đến tâm c2(2,1) là 2.83 được tính như sau:

http://bis.net.vn/photos/storage/ClusteringTechnique006.jpg

http://bis.net.vn/photos/storage/ClusteringTechnique007.jpg

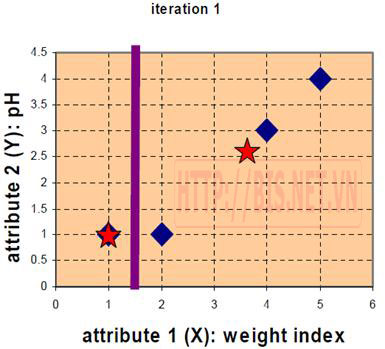
**Bước 3.** Nhóm các đối tượng vào nhóm gần nhất



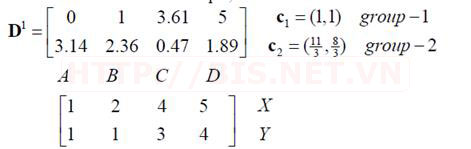
Ta thấy rằng  nhóm 1 sau vòng lặp thứ nhất gồm có 1 đối tượng A và nhóm 2 gồm các đối tượng còn lại B,C,D.

**Bước 5.** Tính lại tọa độ các tâm cho các nhóm mới dựa vào tọa độ của các đối tượng trong nhóm. Nhóm 1 chỉ có 1 đối tượng A nên tâm nhóm 1 vẫn không đổi, c1(1,1). Tâm nhóm 2 được tính như sau:

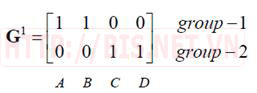
http://bis.net.vn/photos/storage/ClusteringTechnique009.jpg



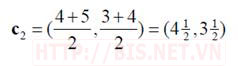
**Bước 6.** Tính lại khoảng cách từ các đối tượng đến tâm mới

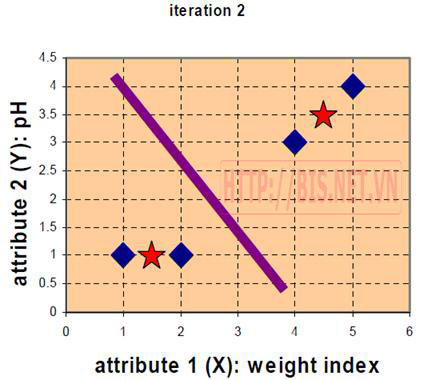


**Bước 7.** Nhóm các đối tượng vào nhóm

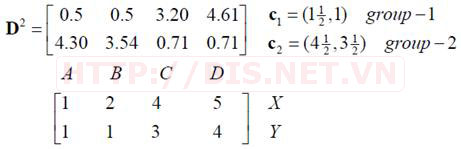


**Bước 8**. Tính lại tâm cho nhóm mới

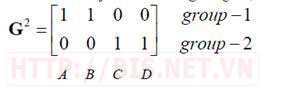
http://bis.net.vn/photos/storage/ClusteringTechnique013.jpg



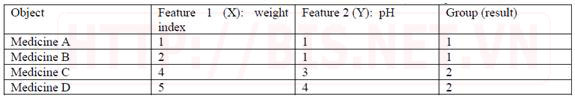
**Bước 8.** Tính lại khoảng cách từ các đối tượng đến tâm mới



**Bước 9.** Nhóm các đối tượng vào nhóm



Ta thấy G2 = G1 (Không có sự thay đổi nhóm nào của các đối tượng) nên thuật toán dừng và kết quả phân nhóm như sau:



 Thuật toán K-Means có ưu điểm là đơn giản, dễ hiểu và cài đặt. Tuy nhiên, một số hạn chế của K-Means là hiệu quả của thuật toán phụ thuộc vào việc chọn số nhóm K (phải xác định trước) và chi phí cho thực hiện vòng lặp tính toán khoảng cách lớn khi số cụm K và dữ liệu phân cụm lớn.

# CHƯƠNG 2. ÁP DỤNG THUẬT TOÁN

## **2.1. Mô tả bài toán**

Trên thế giới có nhiều loại hoa Iris, cho đến nay, người ta có thể phân biệt các loài hoa này bằng mắt thường một cách dễ dàng, nhưng chuyện đó rất mất công sức và thời gian nếu lượng mẫu vật quá lớn. Nhờ nghiên cứu kĩ lưỡng, hiện nay người ta đã có thể tìm ra các đặc trưng cụ thể của tửng loài hoa, mà theo đó có thể dễ dàng dùng để phân biệt các loại hoa này. Hơn thế nữa, các dữ liệu này hoàn toàn có thể được xử lý đưa vào trong máy tính để phân tích, nghĩa là giờ đây, ta có thể giúp máy tính hiểu được bộ dữ liệu và phân biệt được các loài hoa này khi có đủ dữ liệu của chúng được đưa vào. Nội dung báo cáo này chính là thuật toán có thể giúp máy tính có thể phân biệt được các loại hoa Iris( ở đây là 3 loại) khi đưa vào bộ dữ liệu phù hợp.

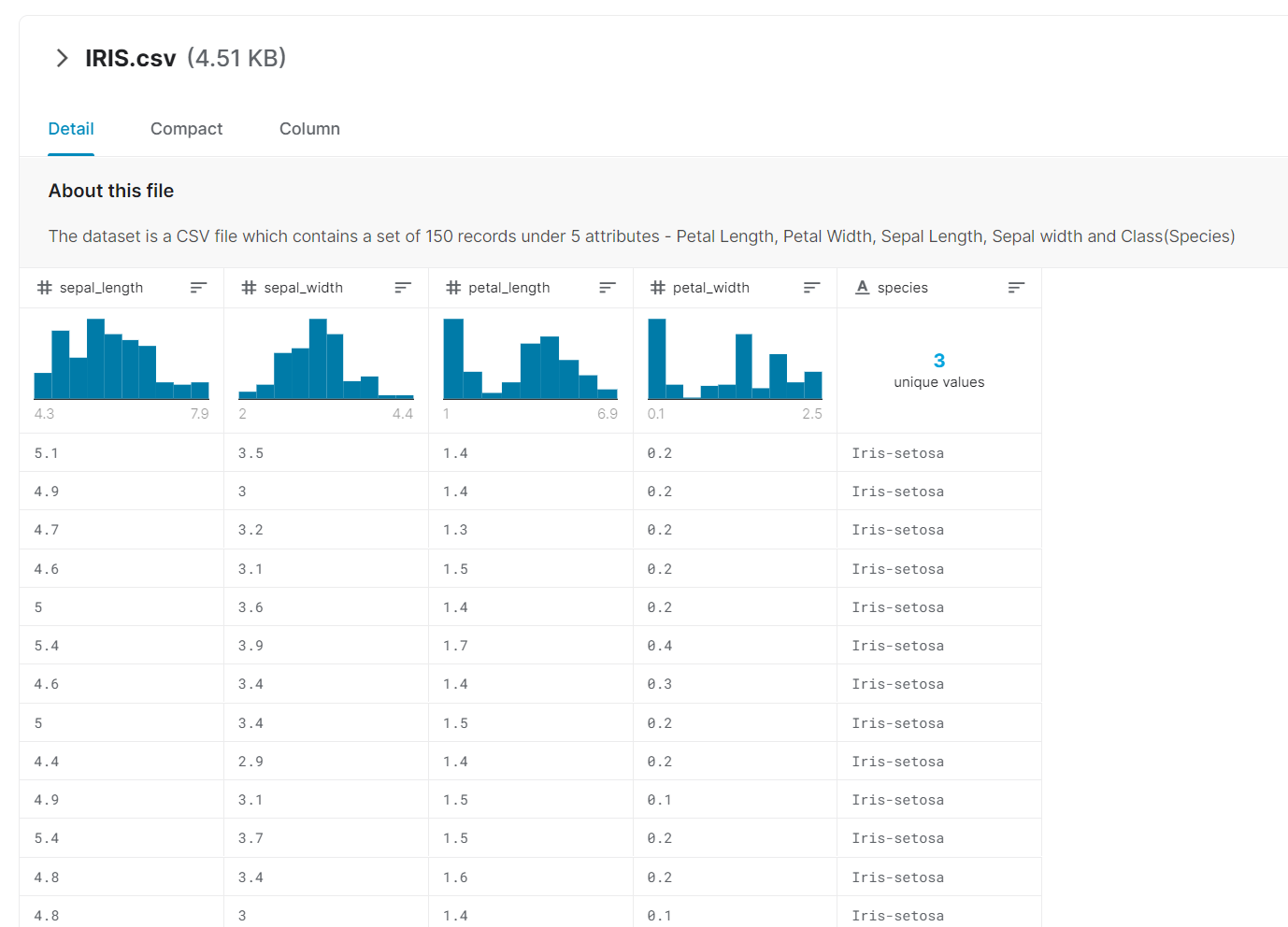
**2.2. Mô tả tập dữ liệu**

Bộ dữ liệu hoa Iris là một bộ dữ liệu đa biến được giới thiệu bởi nhà thống kê và nhà sinh vật học người Anh Ronald Fisher trong bài báo năm 1936 của ông Việc sử dụng nhiều phép đo trong các vấn đề phân loại. Đôi khi nó được gọi là tập dữ liệu Iris của Anderson vì Edgar Anderson đã thu thập dữ liệu để định lượng sự biến đổi hình thái của hoa Iris của ba loài liên quan.

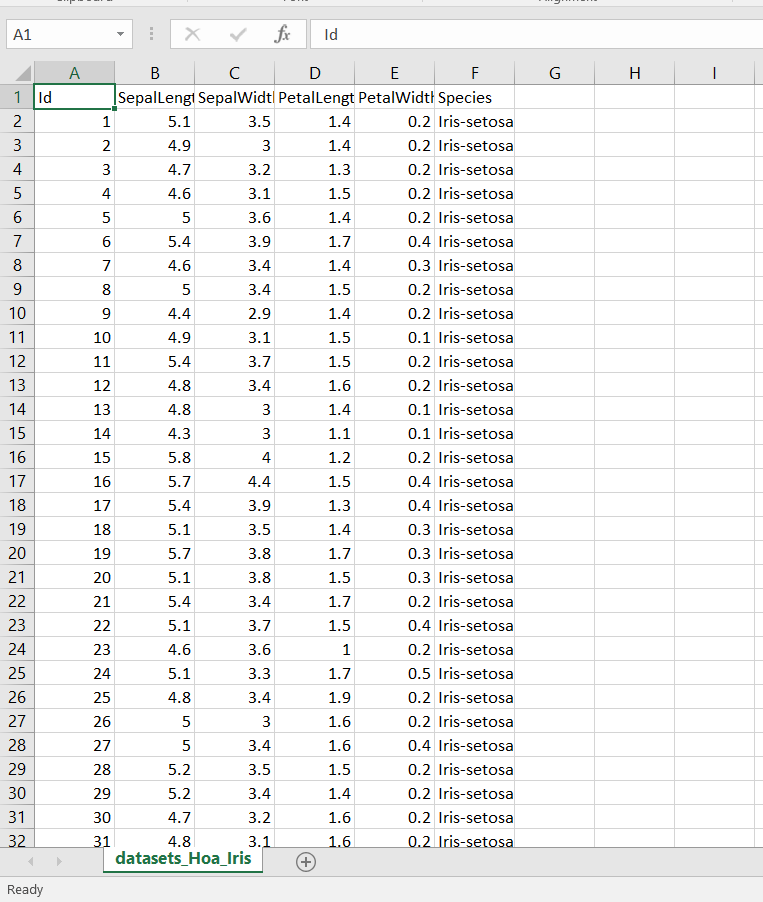
Bộ dữ liệu bao gồm 50 mẫu từ mỗi ba loài Iris (Iris Setosa, Iris virginica và Iris Versolor). Bốn đặc điểm được đo từ mỗi mẫu: chiều dài và chiều rộng của cánh hoa và cánh hoa, tính bằng centimet.

Bộ dữ liệu được lấy trên trang kaggle.

Link tải: <https://www.kaggle.com/arshid/iris-flower-dataset>



Bộ dữ liệu sau khi tải về được lưu dưới dạng file csv:



## **2.3. Áp dụng thuật toán k-means vào bài toán**

Chương trình được viết bằng ngôn ngữ python 3, trên nền tảng anaconda 3.7 và viết trên Spyder 4. Các thư viện cần thiết sử dụng trong chương trình bao gồm:

* Pandas: dùng để đọc file csv
* numpy: xử lí mảng và toán học
* scipy: dùng để gọi hàm cdist, một hàm tính khoảng cách

Các thư viện được thêm vào qua các lệnh:

import pandas as pd

import numpy as np

from scipy.spatial.distance import cdist

Dữ liệu đưa vào chương trình sẽ được đọc trực tiếp qua file csv ở trên. Vì cột đầu tiên của bộ dữ liệu( Id) là không cần thiết nên ta có thể bỏ đi. Việc đọc file và xử lý dữ liệu thừa được thực hiện qua câu lệnh:

datas = pd.read\_csv('G:\Datas\HocTap\MachineLearning\BaoCao\datasets\_Hoa\_Iris.csv').drop('Id', axis = 1).values

Vì ở đây chỉ có 3 loài hoa, nên ta sẽ chia làm 3 cụm. Các hàm cần thiết sử dụng trong chuong trình:

* Hàm Random\_cluster(train, n\_cluster): chọn ra ngẫu nhiên số cụm = n\_cluster( trong bài này là 3), các cụm được chọn là các mẫu đã có trong tập dữ liệu.
* Hàm Choose\_cluster(train, centers): tính khoảng cách từ các điểm dữ liệu đến các tâm cụm, chọn ra tâm cụm gần nó nhất để trả về( giá trị trả về là index của tâm cụm trong mảng các tâm cụm, ở đây có 3 tâm cụm nên trả về 1 trong 3 giá trị 0, 1, 2 đồng thời là tên cụm luôn). Đồng thời giá trị đó chính là nhãn cho điểm dữ liệu.
* Hàm Update\_centers(train, labels, n\_cluster): tìm lại tâm cụm mới.
* Hàm Check\_update(centers, new\_centers): kiểm tra xem cácđiểm dữ liệu sau khi cập nhật tâm cụm có thay đổi với lần trước không, nếu không có nghĩa thuật toán đã tìm được tâm cụm cần tìm
* Hàm Kmeans(init\_centes, init\_labels, train, n\_cluster): Hàm chính của chương trình, chạy các hàm khác để tìm ra lời giải, trả về tọa độ các tâm cụm, số lần lặp để tìm ra tâm cụm, và tâm cụm tương ứng của từng điểm dữ liệu

Code thuật toán:

import pandas as pd

import numpy as np

from scipy.spatial.distance import cdist # ham tinh khoang cach

datas = pd.read\_csv('G:\Datas\HocTap\MachineLearning\BaoCao\datasets\_Hoa\_Iris.csv').drop('Id', axis = 1).values

train\_full = datas

train = datas[:, :4]

n\_cluster = 3

# tao tam cum ngau nhien trong tap train

def Random\_cluster(train, n\_cluster):

return train[np.random.choice(train.shape[0], n\_cluster, replace=False)]

#chon tam cum gam nhat cho diem du lieu, return index tam cum gan nhat, lay lam nhan luon

def Choose\_cluster(train, centers):

D = cdist(train, centers)

return np.argmin(D, axis = 1)

#Cap nhat lai tam cum

def Update\_centers(train, labels, n\_cluster):

centers = np.zeros((n\_cluster, train.shape[1]))

for k in range(n\_cluster):

Xk = train[labels == k, :]

centers[k,:] = np.mean(Xk, axis = 0)

return centers

#kiem tra 2 lan cap nhat tam cum co giong nhau khong

def Check\_update(centers, new\_centers):

return (set([tuple(a) for a in centers]) == set([tuple(a) for a in new\_centers]))

def Kmeans(init\_centes, init\_labels, train, n\_cluster):

centers = init\_centes

labels = init\_labels

times = 0

while True:

labels = Choose\_cluster(train, centers)

new\_centers = Update\_centers(train, labels, n\_cluster)

if Check\_update(centers, new\_centers):

break

centers = new\_centers

times += 1

return (centers, labels, times)

init\_centers = Random\_cluster(train, n\_cluster)

init\_labels = np.zeros(train.shape[0])

centers, labels, times = Kmeans(init\_centers, init\_labels, train, n\_cluster)

# In ra ket qua

print("Cac tam cum tim duoc:")

print(centers)

print('So lan lap tim tam cum la ', times)

labels = labels.reshape(-1, 1)

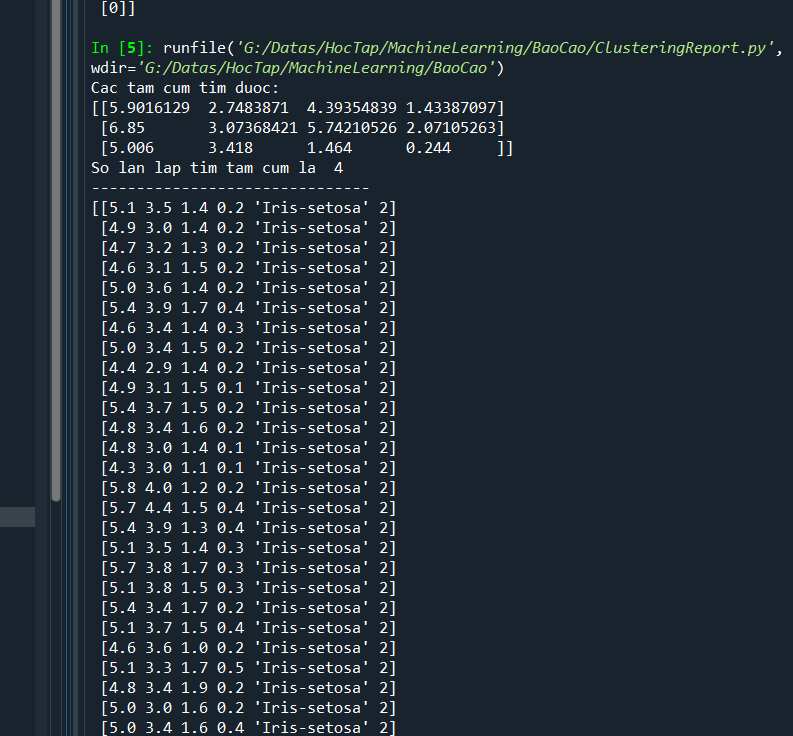
train\_full = np.append(train\_full, labels, axis = 1)

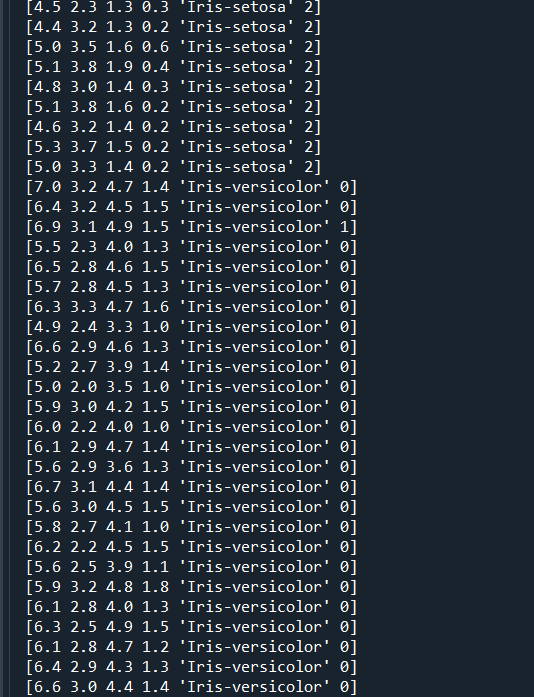
print("-------------------------------")

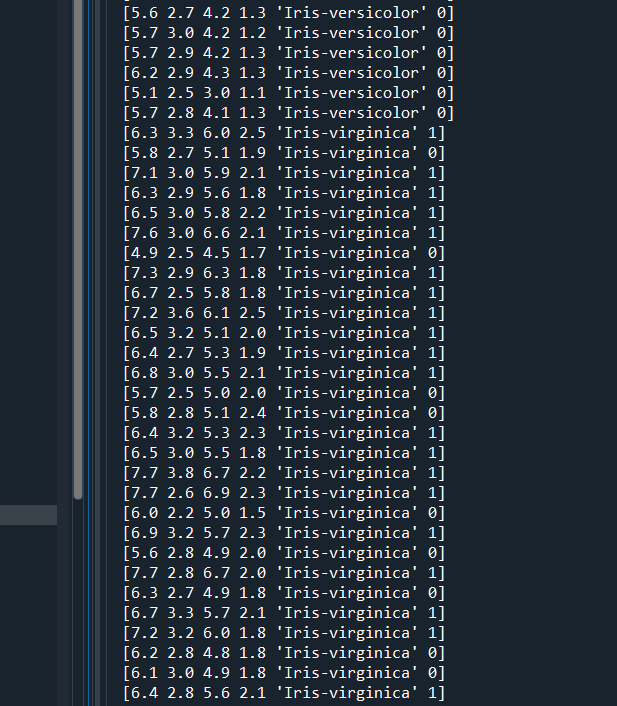
print (train\_full)

2.4. Đánh giá kết quả

Dưới đây là kết quả chạy thuật toán bằng Spyder:







**Nhận xét:**

Thuật toán chạy khá ổn với bộ dữ liệu, đặc biệt với loại hoa setosa với tỷ lệ chính xác 100%, với loại hoa versicolor thì tỉ lệ này thấp hơn một chút: 96%. Đặc biệt với loại hoa virginica, thuật toán tỏ ra không được hiệu quả lắm khi nhầm lẫn khá nhiều với loại hoa phía trên, độ chính xác đạt 72%.

Nguyên nhân có thể là do từ môi trường thực tế, 2 loài hoa này có thể có các dữ liệu tương đương nhau nhiều hơn, cũng có thể do thuật toán chưa tương thích được với bộ dữ liệu này, cần phải xử lý dữ liệu nhiều hơn hoặc điều chỉnh thuật toán cho phù hợp.

Bộ dữ liệu cũng có giới hạn số lượng chỉ 150 mẫu thử, khá ít cho 1 chương trình học máy, nên khó tránh khỏi việc sai sót, việc có thể đạt được kết quả như trên cũng có thể coi là không tồi.

# KẾT LUẬN

Bài báo cáo được xây dựng với mục đích giúp mọi người có thể tìm hiểu cũng như có them hiểu biết về thuật toán Kmeans nói chung, và cụ thể có thể có 1 ứng dụng thực tế về phân loại loài hoa Iris trong tự nhiên với bộ dữ liệu chuẩn. Bài báo cáo vẫn còn nhiều thiếu xót và còn có thể phát triển them để kết quả có thể được cải thiện hơn như:

* Tiền xử lý dữ liệu tốt hơn
* Lấy thêm nhiều dữ liệu hơn cho hệ thống học
* Cải thiện thuật toán, thử với các hàm tính khoảng cách khác, …

Trên đây là bài báo cáo của em, do thời gian và lượng kiến thức có hạn, việc sai sót là khó tránh khỏi, mong thầy cô có thể bỏ qua cho.

Một lần nữa em xin cảm ơn thầy Ngô Trường Giang đã nhiệt tình chỉ dạy em trong quá trình học tập môn nhập môn học máy để em có thể hoàn thành báo cáo này.

# Tài liệu tham khảo

[1]: Tài liệu của thầy Ngô Trường Giang

[2]: <https://machinelearningcoban.com>

[3]: <http://bis.net.vn/forums/t/374.aspx>

[4]: <https://codetudau.com/de-dang-hieu-phuong-phap-k-means-qua-hinh-ve/index.html>