

**ĐẠI HỌC THỦY LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----



**BÀI TẬP LỚN**

**MÔN HỌC MÁY**

**Đề tài: Sử dụng phương pháp học kết hợp dự đoán giá Bitcoin**

**Giảng viên hướng dẫn:** PGS.TS Nguyễn Hữu Quỳnh

**Nhóm sinh viên thực hiện:** Đặng Quang Vinh

Đỗ Thị Ngọc Ánh

Nguyễn Trung Hiếu

**Năm học: 2021-2022**

# Phần 1: Tổng quan

## 1. Giới thiệu về học máy

Khái niệm: Machine Learning là một lĩnh vực của [trí tuệ nhân tạo](https://vi.wikipedia.org/wiki/Tr%C3%AD_tu%E1%BB%87_nh%C3%A2n_t%E1%BA%A1o), một lĩnh vực nhỏ của Khoa Học Máy Tính liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống "học" tự động từ dữ liệu đưa vào để giải quyết những vấn đề cụ thể mà không cần phải được lập trình cụ thể.

Lịch sử thế giới đã trải qua 3 cuộc Cách mạng công nghiệp. Con người đang đi trên cuộc cách mạng công nghiệp lần thứ 4, mà nổi lên trong đó là Trí tuệ nhân tạo (Artificial intelligence – AI). Ta có thể thấy trí tuệ nhân tạo dường như len lỏi khắp nơi trong cuộc sống của ta.

Vai trò của Machine Learning trong thực tiễn:

* Dịch vụ tài chính
* Chính phủ
* Chăm sóc sức khỏe
* Dầu khí
* Giao thông vận tải

Phân nhóm các thuật toán Machine Learning (theo phương thức học)

* Supervised learning
* Unsupervised learning
* Học có giám sát ( Supervised Learning )
* Đặc điểm
* Supervised learning là nhóm phổ biến nhất trong các thuật toán Machine Learning
* Là thuật toán dự đoán đầu ra (outcome) của một dữ liệu mới (new input) dựa trên các cặp (data,label) đã biết từ trước
* Phân loại
* Classification (Phân lớp): Một bài toán được gọi là Classification nếu các label của input data được chia thành một số hữu hạn nhóm
* Regression (Hồi quy): Nếu label không được chia thành các nhóm mà là một giá trị thực cụ thể
* Ưu điểm
* Học tập có giám sát cho phép thu thập dữ liệu và tạo ra dữ liệu đầu ra từ những kinh nghiệm trước đó.
* Giúp tối ưu hóa các tiêu chí hiệu suất với sự trợ giúp của kinh nghiệm
* ML có giám sát giúp giải quyết nhiều loại vấn đề tính toán trong thế giới thực.
* Hạn chế
* Phân loại dữ liệu lớn có thể là một thách thức.
* Đào tạo cho việc học có giám sát cần rất nhiều thời gian tính toán.
* Học không giám sát (Unsupervised Learning)
* Đặc điểm
* Unsupervised Learning là khi chúng ta chỉ có dữ liệu vào X mà không có biết nhãn Y
* Thuật toán unsupervised learning sẽ dựa vào cấu trúc của dữ liệu để thực hiện một công việc nào đó
* Phân loại

Các bài toán Unsupervised learning được tiếp tục chia nhỏ thành hai loại:

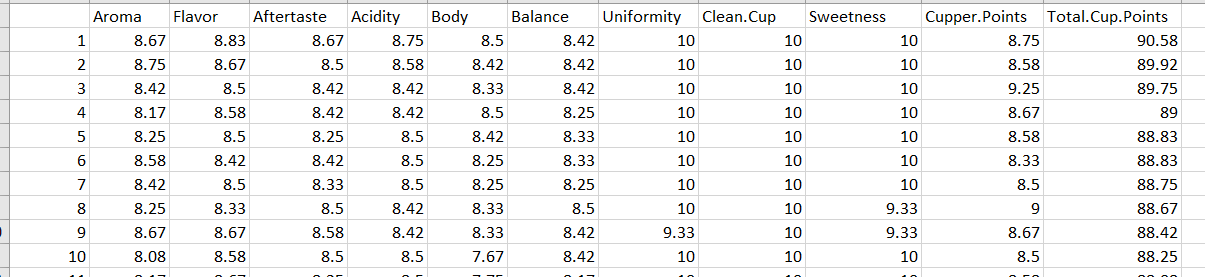
* Clustering (phân cụm): một bài toán phân nhóm toàn bộ dữ liệu X thành các nhóm nhỏ dựa trên sự liên quan giữa các dữ liệu trong mỗi nhóm.
* Association (kết hợp): là bài toán khi chúng ta muốn khám phá ra một quy luật dựa trên nhiều dữ liệu cho trước.
* Ưu điểm
* Có các thuật toán để phân cụm: Clustering, K-mean và Association (kết hợp)
* Dữ liệu không cần có nhãn
* Nhược điểm
* Dữ liệu đầu vào không gắn nhãn nên khó khăn trong việc huấn luyện và đánh giá
* Cần có sự can thiệp của con người để đánh giá kết quả đầu ra

## 2.Trình bày bài toán

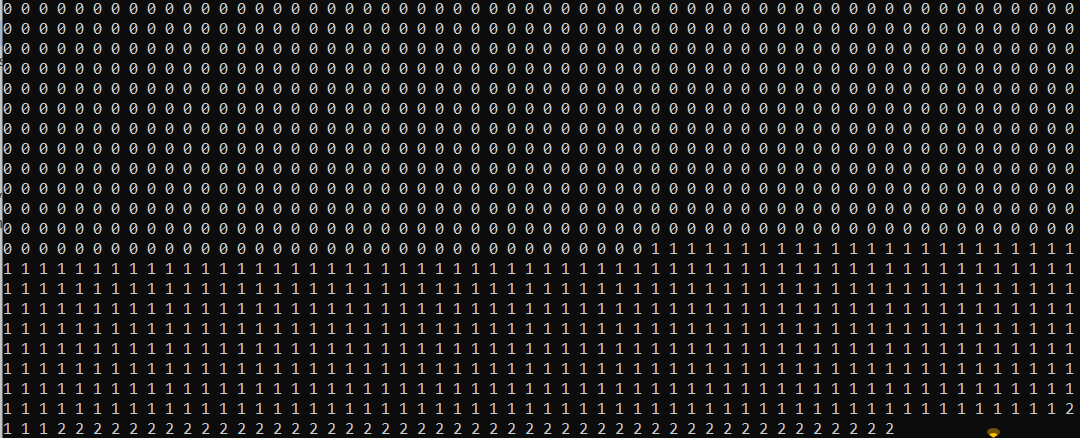
Sử dụng phương pháp phân cụm để phân dữ liệu về hạt cafe arabica vào từng cụm với đặc trưng riêng. Nhóm lựa chọn phương pháp phân cụm K-Means.

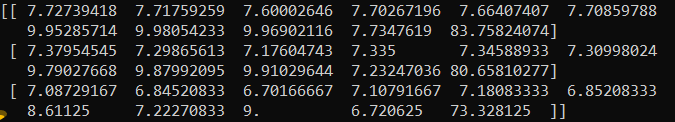
Với đầu vào là bộ dữ liệu về điểm đánh giá của các hạt café arabica từ các vùng khác nhau. Các đặc điểm bao gồm:

* Aroma: Mùi hương
* Flavor: Hương vị
* Aftertaste: Dư vị
* Acidity: Độ chua
* Body: Hình dáng
* Balance: Cân đối
* Uniformity: Đồng nhất
* Clean.Cup: cặn, khuyết tật
* Sweetness: Ngọt
* Cupper.Points: điểm hạt
* Total.Cup.Points: Tổng điểm hạt



Đầu ra là các tâm cụm và các điểm dữ liệu đã được phân vào cụm tương ứng.





## 3. Phương pháp học máy sử dụng trong bài tập lớn

### Tổng quan

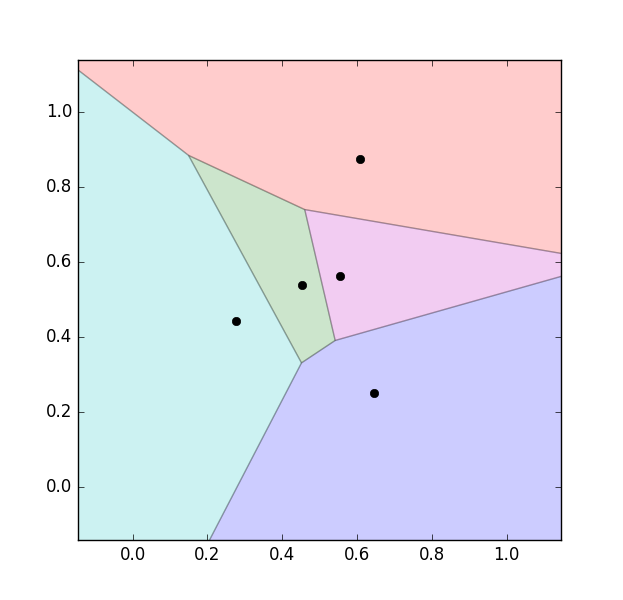
Trong thuật toán K-means clustering, chúng ta không biết nhãn (label) của từng điểm dữ liệu. Mục đích là làm thể nào để phân dữ liệu thành các cụm (cluster) khác nhau sao cho dữ liệu trong cùng một cụm có tính chất giống nhau.

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). Hình bên dưới là một ví dụ về 3 cụm dữ liệu (từ giờ tôi sẽ viết gọn là cluster).



Giả sử mỗi cluster có một điểm đại diện (center) màu vàng. Và những điểm xung quanh mỗi center thuộc vào cùng nhóm với center đó.

Hình dưới đây là một hình minh họa cho việc phân chia.



Chúng ta thấy rằng đường phân định giữa các vùng là các đường thẳng (chính xác hơn thì chúng là các đường trung trực của các cặp điểm gần nhau). Vì vậy, vùng sẽ là một hình đa giác.

Cách phân chia này trong toán học được gọi là [Voronoi Diagram](https://en.wikipedia.org/wiki/Voronoi_diagram).

Trong không gian ba chiều sẽ là một đa diện. Trong không gian nhiều chiều hơn, chúng ta sẽ có *siêu đa diện* (hyperpolygon).

### Phân tích toán học

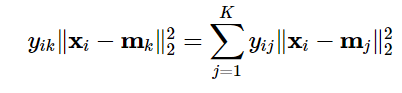
Mục đích cuối cùng của thuật toán phân nhóm này là: từ dữ liệu đầu vào và số lượng nhóm chúng ta muốn tìm, hãy chỉ ra center của mỗi nhóm và phân các điểm dữ liệu vào các nhóm tương ứng. Giả sử thêm rằng mỗi điểm dữ liệu chỉ thuộc vào đúng một nhóm.

* Hàm mất mát và bài toán tối ưu

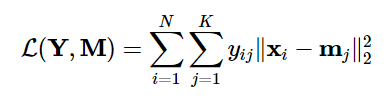
Nếu ta coi center mk là center (hoặc representative) của mỗi cluster và ước lượng tất cả các điểm được phân vào cluster này bởi mk, thì một điểm dữ liệu xi được phân vào cluster k sẽ bị sai số là (xi−mk)(xi−mk). Chúng ta mong muốn sai số này có trị tuyệt đối nhỏ nhất nên ta sẽ tìm cách để đại lượng sau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



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

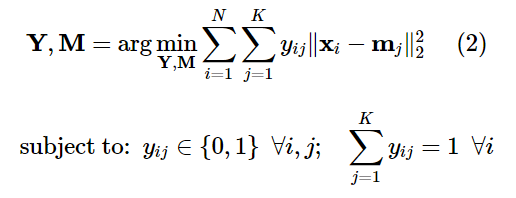


Sai số cho toàn bộ dữ liệu sẽ là:



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 nêu trong phương trình (1).

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



* Thuật toán tối ưu hàm mất mát

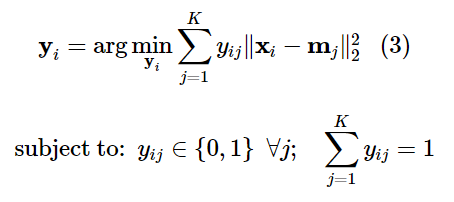
Bài toán (2) là một bài toán khó tìm điểm tối ưu vì nó có thêm các điều kiện ràng buộc. Bài toán này thuộc loại mix-integer programming (điều kiện biến là số nguyên) - là loại rất khó tìm nghiệm tối ưu toàn cục (global optimal point, tức nghiệm làm cho hàm mất mát đạt giá trị nhỏ nhất có thể).Tuy nhiên, 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 (2) là xen kẽ giải Y và M khi biến còn lại được cố định. Đây là một thuật toán lặp, cũng là kỹ thuật phổ biến khi giải bài toán tối ưu. Chúng ta sẽ lần lượt giải quyết hai bài toán sau đây:

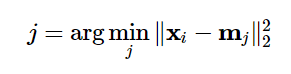
Cố định M, tìm Y

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 xixi như sau:



Vì chỉ có một phần tử của label vector yi bằng 1 nên bài toán (3) có thể tiếp tục được viết dưới dạng đơn giản hơn:

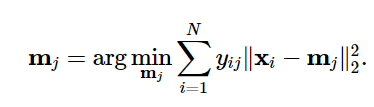


Vì chính là bình phương khoảng cách tính từ điểm xi tới center mj, ta có thể kết luận rằng **mỗi điểm**xi**thuộc vào cluster có center gần nó nhất!** Từ đó ta có thể dễ dàng suy ra label vector của từng điểm dữ liệu.

**Cố định Y, tìm M**

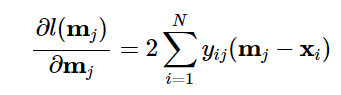
**Giả sử đã tìm được cluster cho từng điểm, hãy tìm center mới cho mỗi cluster để hàm mất mát đạt giá trị nhỏ nhất.**

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:

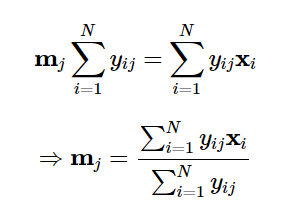


Tới đây, ta có thể tìm nghiệm bằng phương pháp giải đạo hàm bằng 0, vì hàm cần tối ưu là một hàm liên tục và có đạo hàm xác định tại mọi điểm*.*Và quan trọng hơn, hàm này là hàm convex (lồi) theo mjmj nên chúng ta sẽ tìm được giá trị nhỏ nhất và điểm tối ưu tương ứng. Sau này nếu có dịp, tôi sẽ nói thêm về tối ưu lồi (convex optimization) - một mảng cực kỳ quan trọng trong toán tối ưu*.*

Đặt l(mj) là hàm bên trong dấu argmin, 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. Hay nói một cách đơn giản hơn nhiều: mj **là trung bình cộng của các điểm trong cluster**j.

### Tóm tắt thuật toán

Đầu vào: Dữ liệu X và số lượng cluster cần tìm K.

Đầu ra: Các center M và label vector cho từng điểm dữ liệu Y.

1. Chọn K điểm bất kỳ làm các center ban đầu.
2. Phân mỗi điểm dữ liệu vào cluster có center gần nó nhất.
3. Nếu việc gán dữ liệu vào từng cluster ở bước 2 không thay đổi so với vòng lặp trước nó thì ta dừng thuật toán.
4. Cập nhật center cho từng cluster bằng cách lấy trung bình cộng của tất các các điểm dữ liệu đã được gán vào cluster đó sau bước 2.
5. Quay lại bước 2.

Chúng ta có thể đảm bảo rằng thuật toán sẽ dừng lại sau một số hữu hạn vòng lặp. Thật vậy, vì hàm mất mát là một số dương và sau mỗi bước 2 hoặc 3, giá trị của hàm mất mát bị giảm đi. Theo kiến thức về dãy số trong chương trình cấp 3: nếu một dãy số giảm và bị chặn dưới thì nó hội tụ! Hơn nữa, số lượng cách phân nhóm cho toàn bộ dữ liệu là hữu hạn nên đến một lúc nào đó, hàm mất mát sẽ không thể thay đổi, và chúng ta có thể dừng thuật toán tại đây.

# Phần 2: Thực nghiệm

## 1.Mô tả tập ví dụ huấn luyện và tập test

Những dữ liệu này bao gồm các đánh giá về 1312 hạt cà phê arabica và 28 hạt cà phê robusta từ những người đánh giá được đào tạo của Viện Chất lượng Cà phê. Các tính năng bao gồm:

* Aroma: Điểm đánh giá về mùi hương
* Flavor: Điểm đánh giá về hương vị
* Aftertaste: Điểm đánh giá về dư vị
* Acidity: Điểm đánh giá về dộ chua, tính acid trong hạt
* Body: Điểm đánh giá hình dáng của hạt
* Balance: Điểm đánh giá độ cân đối của hạt
* Uniformity: Điểm đánh giá về tính đồng nhất của các hạt với nhau
* Clean.Cup: Điểm đánh giá độ cặn, xác suất khuyết tật trên các hạt
* Sweetness: Điểm về độ ngọt ngào
* Cupper.Points: Điểm đánh giá tổng thể hạt
* Total.Cup.Points: Tổng điểm đánh giá hạt

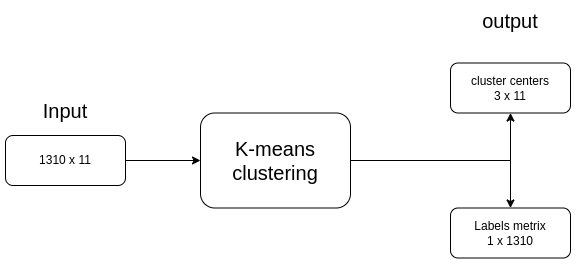
Thư mục [data] (https://github.com/jldbc/coffee-quality-database/tree/master/data) chứa cả dữ liệu thô và dữ liệu đã làm sạch. Dữ liệu thô chính xác như được tìm thấy trên trang CQI. Những dữ liệu do con người ghi lại này sử dụng nhiều loại mã hóa, chữ viết tắt và đơn vị đo lường khác nhau cho tên trang trại, độ cao, khu vực và các trường khác.

Trang web đã bị cắt bằng cách sử dụng trình duyệt không có đầu Selenium và Beautiful Soup. Để sao chép hoặc thu thập dữ liệu cập nhật, hãy đăng nhập vào trang CQI và nhập lời giới thiệu của bạn vào [scraper]

(<https://github.com/jldbc/coffee-quality-database/tree/master/scraper>)

Những dữ liệu này được thu thập từ [các trang đánh giá] của Viện Chất lượng Cà phê (https://database.coffeeinstitution.org/) vào tháng 1 năm 2018.

## 2.Mô tả chương trình demo



Hình 1: Tổng quan model

Chương trình được viết bằng ngôn ngữ python, sử dụng mô hình phấn cụm k-means (k-means clustering) được cung cấp bởi Sklearn cùng với các thư viện sau:

* Numpy
* Pandas
* Pickle
* Sklearn
* Matplotlib

Chương trình gồm có những file:

* config.py: File các thông só cơ bản của chương trìn
* arabica\_data\_extracted.csv: File chứa dữ liệu sử dụng để làm đầu vào cho chương trình
* extract\_data.py: File thực hiện trích xuất dữ liệu, tạo ra file arabica\_data\_extracted.csv
* training.py: File chứa các hàm thực hiện huấn luyện mô hình và lưu mô hình
* model.pkl: File chứa mô hình đã được huấn luyện
* show.py: File chứa các hàm hiển thị
* main.py: File chính của chương trình thực hiện gọi các hàm trong các file khác

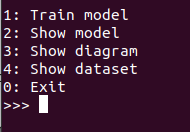
Các hàm bao gồm:

* training():
  + - Hàm sử dụng để huấn luyện và lưu mô hình vào file model.pkl
    - Không có tham số truyền vào
    - Trả về model đã được huấn luyện hoặc flase khi gặp lỗi
    - Nằm trong file training.py
* inputCustomizeParameter():
* Hàm lấy các tham số mong muốn của người dùng để phục vụ cho việc huấn luyện model
* Không có tham số truyền vào
* Trả về mảng tham số mà người dùng nhập hoặc flase khi gặp lỗi
* Nằm trong file training.py
* showModel(model):
  + - Hàm hiển thị các thông số của model
    - Tham số truyền vào là một model
    - Trả về boolean
    - Nằm trong file show.py
* showDiagram(lbls):
  + - Hàm hiển thị biểu đồ scatter giữa 2 cột dữ liệu chỉ định
    - Tham số truyền vào là mảng nhãn của từng điểm dữ liệu
    - Trả về boolean
    - Nằm trong file show.py
* drawScatter(feature\_name, labels):
  + - Hàm vẽ biểu đồ scatter giữa 2 cột dữ liệu chỉ định
    - Tham số truyền vào là mảng tên các cột cần vẽ và nhãn của từng điểm dữ liệu
    - Trả về boolean
    - Nằm trong file show.py
* showData():
  + - Hàm hiển thị dữ liệu được sử dụng
    - Không có tham số truyền vào
    - Trả về boolean
    - Nằm trong file show.py

Khi chạy chương trình cần chú ý thay đổi cái thông số trong file config.py cho phù hợp với cấu hình của máy.

Chương trình khởi chạy hiển thị các tùy chọn mà người dùng có thể chọn. Người dùng nhập vào tùy chọn muốn sử dụng các tùy chọn bao gồm:

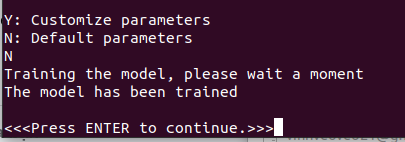
* + - Train model
    - Show model
    - Show diagram
    - show data
    - exit



Hình 2: Hình menu chính của chương trình

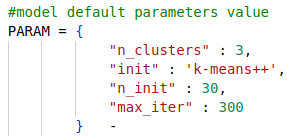
* Train model

Có thể chọn sử dụng tham số mặc định của chương trình hoặc tham số tùy chỉnh do người dùng nhập vào từ bàn phím



Hình 3: Hình ảnh model được huấn luyện với các tham số mặc định

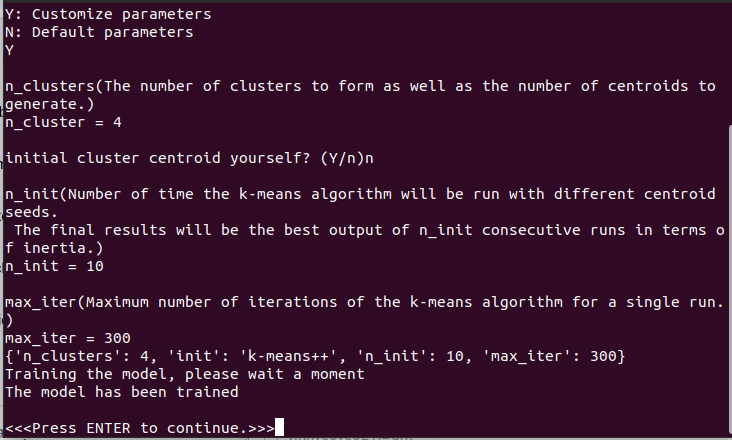
Khi chọn sử dụng tham số mặc định chương trình sẽ sử dụng các thông số măc định đã được cài đặt sẵn để huấn luyện model



Hình 4: Hình ảnh các tham số mặc định

Với việc chọn tham số tùy chỉnh thì người dùng sẽ được nhập vào các thông tin về:

* + - Số tâm cụm cần tìm (n\_cluster)
    - Số lần chương trình sẽ khởi tạo lại các điểm khởi tạo (n\_init)
    - Số vòng lặp tối đa mà chương trình có thể được thực hiện (n\_inter)
    - Tùy chọn việc khởi tạo các điểm khởi tại là ngẫu nhiên hay được nhập vào từ bàn phím theo ý người dùng



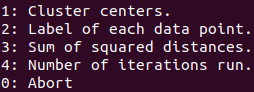
Hình 5: Hình ảnh huấn luyện model với tham số tùy chỉnh

và điểm khởi tạo là ngẫu nhiên

* Show model

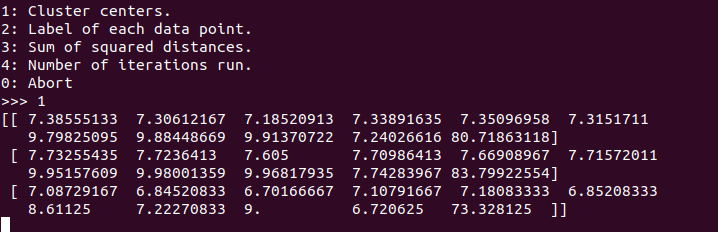
Hiển thị tùy chọn cho xem chi tiết thông tin model gồm:

* Cluster centers.
* Label of each data point.
* Sum of squared distances.
* Number of iterations run.
* Abort



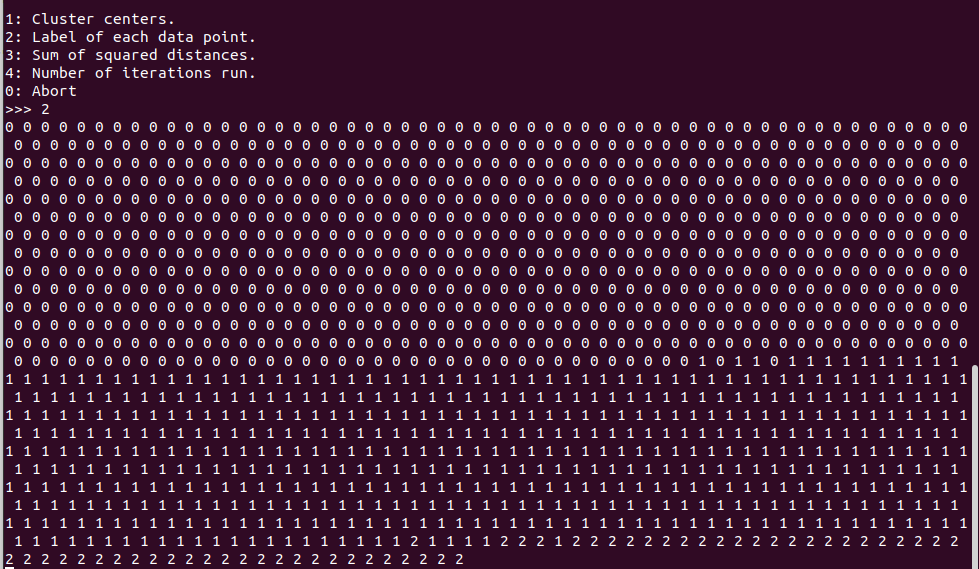
Hình 6: Hình ảnh các tùy chọn xem chi tiết model

Cluster centers: hiển thị các tâm cụm được tìm thấy



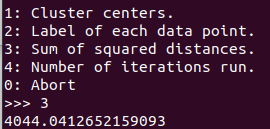
Hình 7: Hình ảnh hiển thị các tâm cụm

Label of each data point: hiển thị nhãn cho từng điểm dữ liệu



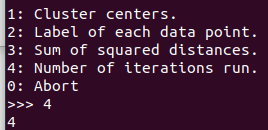
Hình 8: Hình ảnh hiển thị nhãn của từng điểm dữ liệu

Sum of squared distances: hiển thị tổng bình phương khoảng cách từ các điểm tới tâm gần nó nhất



Hình 9: Hình ảnh cho tổng bình phương

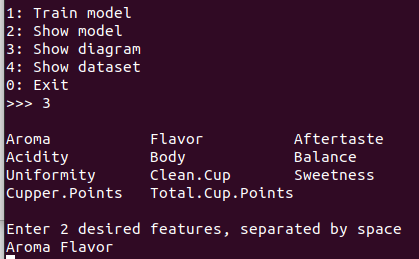
Number of iterations run: hiển thị số vòng lặp đã thực hiện



Hình 10: Hình ảnh số vòng lặp

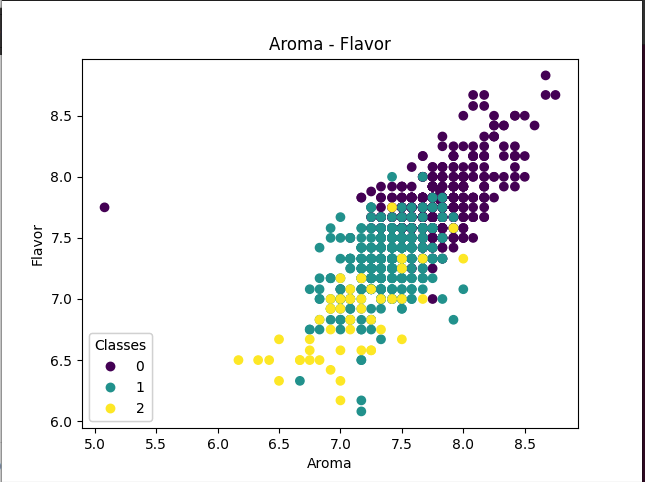
* Show diagram

Hiển thị danh sách các cột có thể lựa chọn để vẽ biểu đồ. Người dùng nhập vào tên của 2 cột muốn vẽ biểu đồ



Hình 11: Hình ảnh nhập tên cột

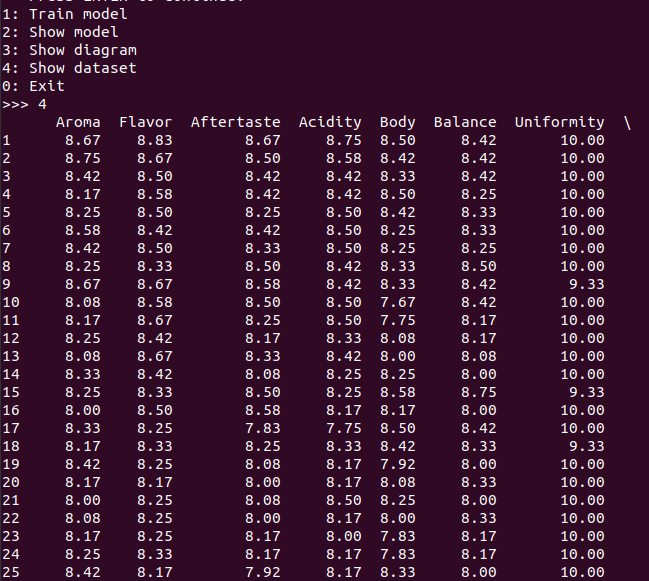
Chương trình hiển thị biểu đồ scatter của 2 cột đã lựa chọn



Hình 12: Hình ảnh biểu đồ scatter của cột Aroma và Flavor

* Show data

Hiển thị tập dữ liệu đang sử dụng trong chương trình



Hình 13: Hình ảnh tập dữ liệu

## 3. Mô tả kết quả dự đoán

# Kết luận

# Tài liệu tham khảo