**Thuật Toán K-means Clustering**

**Câu 1**: Thế nào là thuật toán gôm cụm ?

Thuật toán gôm cụm (Clustering) là một kỹ thuật Machine Learning có liên quan đến việc nhóm các điểm dữ liệu. Với một tập hợp các điểm dữ liệu, chúng ta có thể sử dụng thuật toán phân cụm để phân loại từng điểm dữ liệu thành một nhóm cụ thể. Về lý thuyết, các điểm dữ liệu nằm trong cùng một nhóm phải có các thuộc tính và các tính năng tương tự, trong khi các điểm dữ liệu trong các nhóm khác nhau phải có các thuộc tính và tính năng không giống nhau. Clustering là một phương pháp học tập không giám sát và là một kỹ thuật phổ biến để phân tích dữ liệu thống kê được sử dụng trong nhiều lĩnh vực.

Thuật toán clustering được chia làm 2 nhóm chính:

* distance-based clustering algorithms: thuật toán phân nhóm dựa trên khoảng cách
* density-based clustering algorithms: thuật toán phân nhóm dựa trên mật độ

**Câu 2**:

K-mean nó là 1 thuật toán để phân nhóm 1 tập các đối tượng dựa trên các thuộc tính của chúng.

K-mean thuộc nhóm distance-based clustering algorithms: thuật toán phân nhóm dựa trên khoảng cách

Mục tiêu của thuật toán này là tìm nhóm trong dữ liệu, với số lượng các nhóm đại diện bởi các biến K. Thuật toán hoạt động lặp lại để gán mỗi điểm dữ liệu cho một trong các nhóm K dựa trên các tính năng được cung cấp. Các điểm dữ liệu được phân cụm dựa trên tính tương tự về tính năng. Kết quả của thuật toán phân cụm K–means là:

Các trung tâm của các cụm K, có thể được sử dụng để gắn nhãn dữ liệu mới

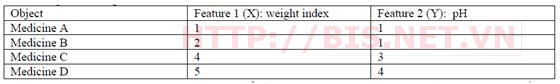
Nhãn cho dữ liệu thử (mỗi điểm dữ liệu được gán cho một cụm duy nhất)

Mỗi trọng tâm của một cụm là một tập hợp các giá trị tính năng mà nó xác định các nhóm kết quả. Kiểm tra trọng số tính năng trọng tâm có thể được sử dụng để phân tích chất lượng loại của nhóm từng đại diện.

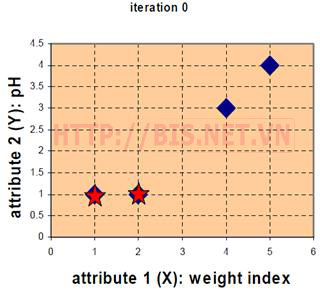
**Câu 3**: Cho ví dụ về thuật toán được phân loại được cho:

**Ví dụ 1: Phân loại thuốc:**

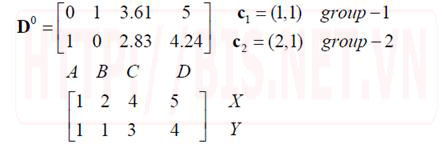
Giả sử ta có 4 loại thuốc A,B,C,D, mỗi loại thuộc được biểu diễn bởi 2 đặc trưng X và Y như sau. Mục đích của ta là nhóm các thuốc đã cho vào 2 nhóm (K=2) dựa vào các đặc trưng của chúng.



**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)



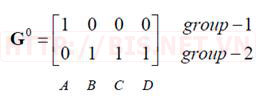
Mỗi cột trong ma trận khoảng cách (D) là một đối tượng (cột thứ nhất tương ứng với đối tượng A, cột thứ 2 tương ứng với đối tượng B,). Hàng thứ nhất trong ma trận khoảng cách biểu diễn khoảng cách giữa các đối tượng đến tâm của nhóm thứ nhất (c1) và hàng thứ 2 trong ma trận khoảng cách biểu diễn khoảng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

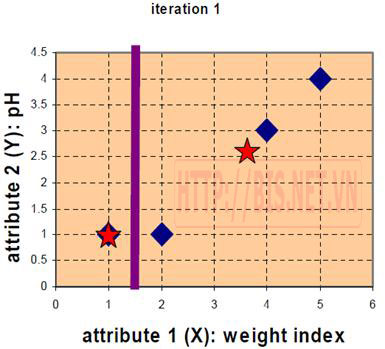
**Bước 3.** Nhóm các đối tượng vào nhóm gần nhất



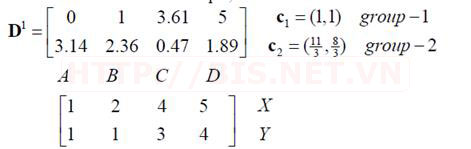
Ta thấy rằng nhóm 1 sau vòng lặp thứ nhất gồm có 1 đối tượng A và nhóm 2 gồm các đối tượng còn lại B,C,D.

**Bước 4.** 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:

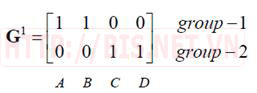
http://bis.net.vn/photos/storage/ClusteringTechnique009.jpg



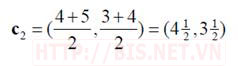
**Bước 5.** Tính lại khoảng cách từ các đối tượng đến tâm mới

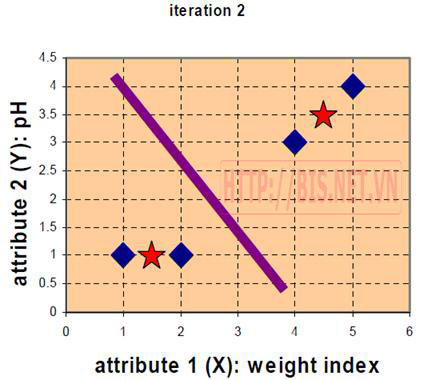


**Bước 6.** Nhóm các đối tượng vào nhóm

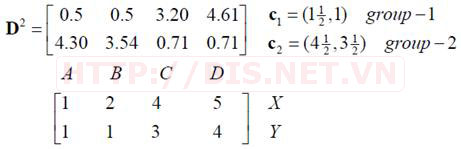


**Bước 7.** Tính lại tâm cho nhóm mới

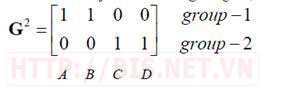
http://bis.net.vn/photos/storage/ClusteringTechnique013.jpg



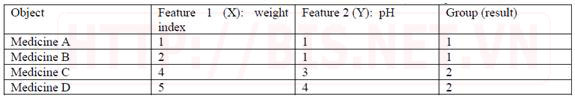
**Bước 8.** Tính lại khoảng cách từ các đối tượng đến tâm mới



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



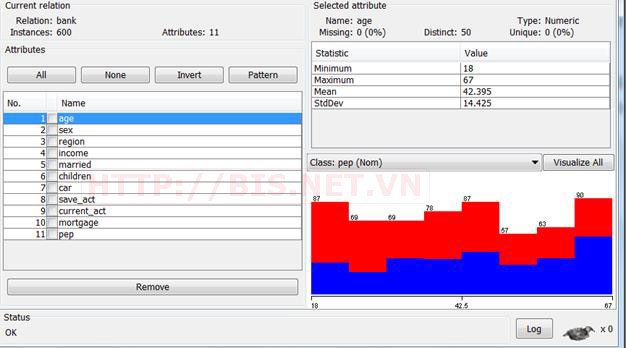
Ta thấy G2 = G1 (Không có sự thay đổi nhóm nào của các đối tượng) nên thuật toán dừng và kết quả phân nhóm như sau:



 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.

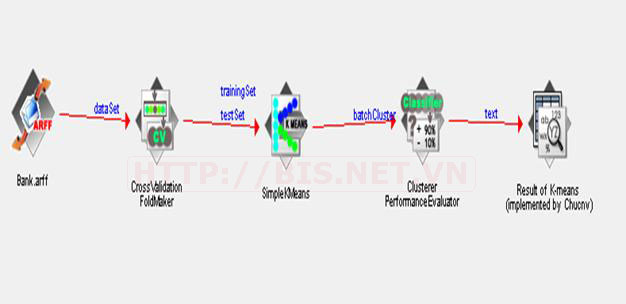
**Ví dụ 2: Phân loại khách hàng:**

Trong ví dụ này,dữ liệu dùng để phân cụm trong ví dụ này là dữ liệu dùng để phân loại khách hàng của ngân hàng. bank.arff gồm có 11 thuộc tính và 600 khách hàng (instances). Dưới đây là cấu trúc và phân bố dữ liệu của

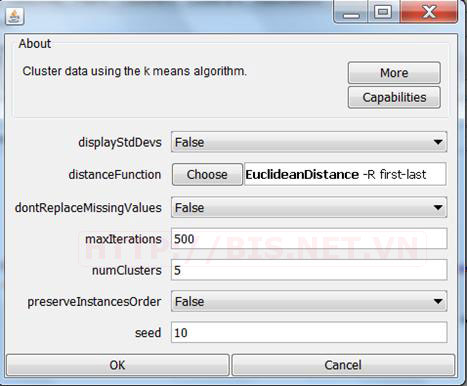


Nhiệm vụ của chúng ta là dùng thuật toán K-Means để phân nhóm các khách hàng vào K nhóm (trong ví dụ này K=5) dựa vào sự tương tự (similar) trên11 thuộc tính của họ.

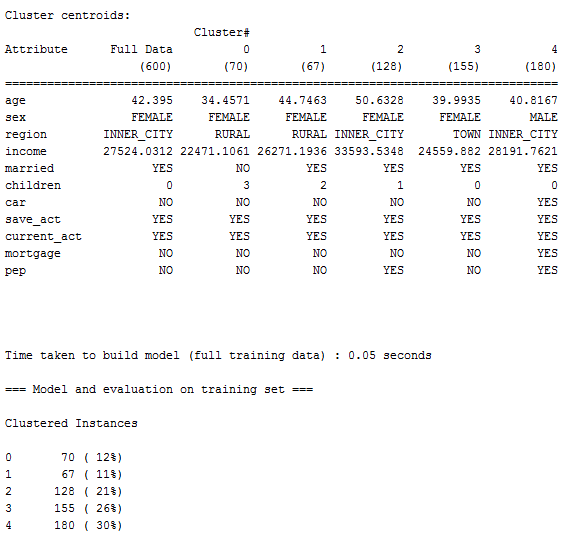
 Ta xây dựng một KnowledgeFlow trong WeKa như sau:



Thiết lập các tham số cho thuật toán K-Means như số cụm (trong ví dụ này K=5), Cách tính khoảng cách (trong ví dụ này dùng khoảng cách Euclidean),…



**Kết quả phân cụm chi tiết như sau:**

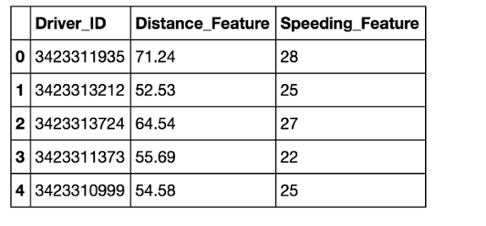


**Ví dụ 3: Phân loại đội lái xe giao hàng nhanh**

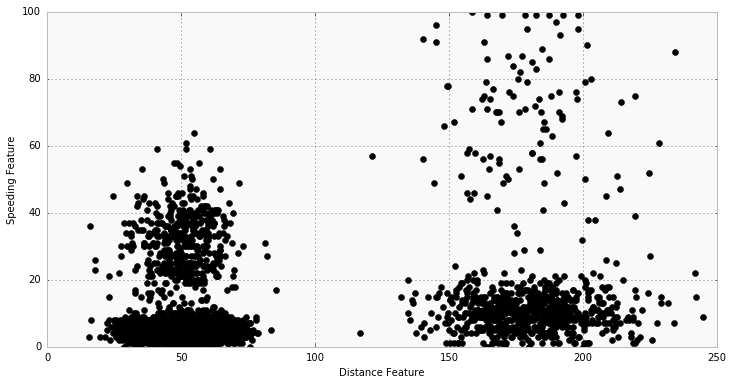
Ví dụ, chúng tôi sẽ chỉ ra cách thuật toán *K* -means hoạt động với một bộ dữ liệu mẫu của đội lái xe giao hàng nhanh. Để đơn giản hơn, chúng ta sẽ chỉ xem xét hai tính năng của ngời lái xe: khoảng cách trung bình của người giao hàng mỗi ngày và tỷ lệ phần trăm trung bình của thời gian người lái xe trên 5 dặm một giờ so với giới hạn tốc độ. Nói chung, thuật toán này có thể được sử dụng cho bất kỳ số lượng đối tượng địa lý nào, miễn là số lượng mẫu dữ liệu lớn hơn nhiều so với số lượng đối tượng địa lý.

**Bước 1.** Làm sạch và chuyển đổi dữ liệu

Trong ví dụ này, chúng ta sẽ đã làm sạch và hoàn thành một số phép biến đổi dữ liệu đơn giản. Mẫu dữ liệu dưới dạng **pandas DataFrame** được hiển thị bên dưới.



Biểu đồ bên dưới hiển thị tập dữ liệu cho 4.000 người lái xe, với tính năng khoảng cách trên trục x và tính năng tăng tốc trên trục y.



Bước 2. Chọn K và Chạy thuật toán

Bắt đầu bằng cách chọn  *K* = 2. Đối với ví dụ này, sử dụng các gói Python [scikit-learn](http://scikit-learn.org/stable/) và [NumPy](http://www.numpy.org/) để tính toán như hình dưới đây:

**import numpy as np**

**from sklearn.cluster import KMeans**

**### For the purposes of this example, we store feature data from our**

**### dataframe `df`, in the `f1` and `f2` arrays. We combine this into**

**### a feature matrix `X` before entering it into the algorithm.**

**f1 = df['Distance\_Feature'].values**

**f2 = df['Speeding\_Feature'].values**

**X=np.matrix(zip(f1,f2))**

**kmeans = KMeans(n\_clusters=2).fit(X)**

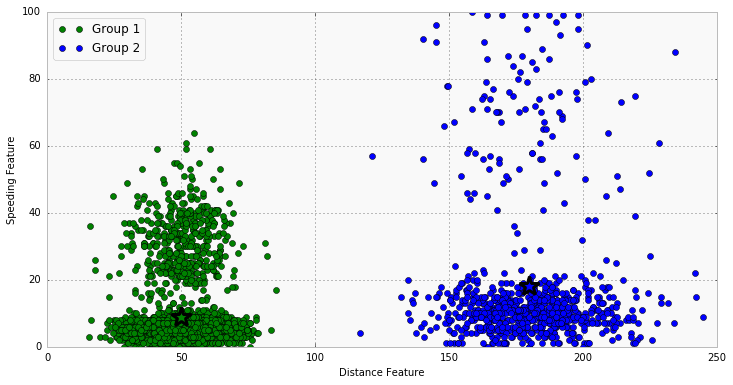
Các nhãn cụm được trả về trong **kmeans.labels\_**.

**Bước 3.** Xem lại kết quả

Biểu đồ dưới đây cho thấy kết quả. Trực quan, bạn có thể thấy rằng thuật toán *K* -means chia tách hai nhóm dựa trên tính năng khoảng cách. Mỗi centroid cụm được đánh dấu bằng một ngôi sao.

* Nhóm 1 Centroid = (50, 5.2)
* Nhóm 2 Centroid = (180,3, 10,5)

Sử dụng kiến ​​thức miền của tập dữ liệu, chúng ta có thể suy ra rằng Nhóm 1 là trình điều khiển đô thị và Nhóm 2 là trình điều khiển nông thôn.



**Bước 4.** Lặp lại một số giá trị của K

Kiểm tra các kết quả tìm *K* = 4 như thế nào . Để làm điều này, tất cả những gì bạn cần thay đổi là số lượng các cụm mục tiêu trong **KMeans()** hàm.

**kmeans = KMeans(n\_clusters=4).fit(X)**

Biểu đồ dưới đây cho thấy các cụm kết quả. Chúng ta thấy rằng bốn nhóm riêng biệt đã được xác định bằng thuật toán; bây giờ nhóm người lái xe quá tốc độ đã được tách ra khỏi những người lái xe theo giới hạn tốc độ, ngoài phân chia nông thôn và đô thị. Nhóm lái xe đô thị có ngưỡng tăng tốc thấp hơn so với người lái xe nông thôn, có thể do người lái xe đô thị dành nhiều thời gian hơn trong giao lộ và giao thông dừng lại và đi.

