K-Means

K-means is an algorithm that divides the unlabeled data into clusters in such way that data in same cluster show similar properties and data in different clusters whos dissimilar properties.

K- means clustering is an unsupervised learning algorithm. Cluster Analysis helps us describe, analyze, and gain insight into the data.

Advantages

Relatively simple to implement.

Scales a large data set.

Faster than hierarchical, because of order of time complexity is linear with the number of data

Works well with spherical clusters

Disadvantages

Difficult to predict K-values

Different initial partitions can result in different final clusters

If does not work well with clusters of different size and density

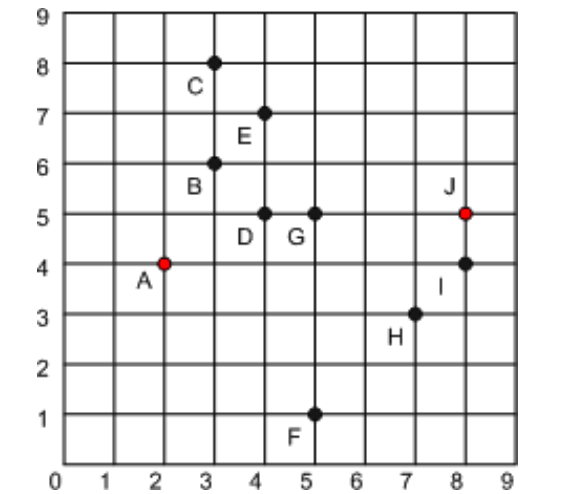
Need normalization

How does it work?

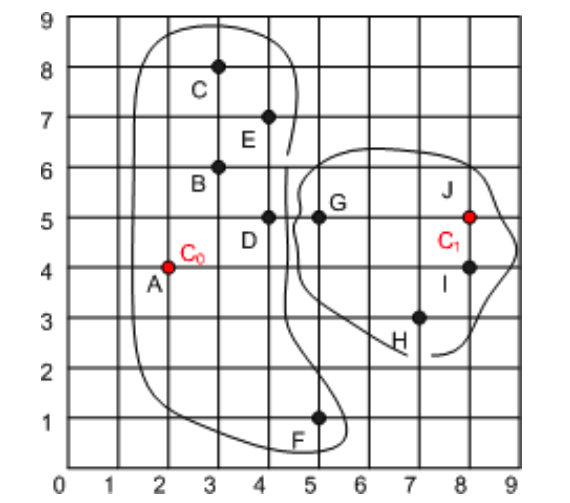
{A: (2, 4), B: (3, 6), C: (3, 8), D: (4, 5), E: (4, 7), F: (5, 1), G: (5, 5), H: (7, 3), I: (8, 4), J: (8, 5)}

Graph of above dataset.

Randomly pick number of cluster and centroids



Then we find the euclidean distance between each point.



C\_0 and C\_1 was randomly chosen centroids.

Euclidean distance C\_0 to F is [(5-2)^2+(1-4)^2]^0.5 = 4.24

Euclidean distance C\_1 to F is [(5-8)^2+(1-5)^2]^0.5 = 5

Since F is closer to C\_0 it will be assign to C\_0

Therefore

C\_0 = {A,B,C,D,E,F}

C\_1 = {G,H,I,J}

Then next the new centroid is calculated C\_0 = (avg\_x, avg\_y)

X\_0 = (2+3+3+4+$+5)/6=3.5 , y\_0 = (4+6+8+5+7+1)/6=5.17 etc

C\_0 = (3.5, 5.17), c\_1 = (7,4.5)

And we find distance from each point to centroids and assign the point to closer cluster and keep repeating the above process until there is no change in cluster membership.

C\_0 = {A,B,C,D,E,G}

C\_1 = {F,H,I,J}

Python Sklearn Implementation

Import libraries

Load data

Separate futures and target

(Split data into train and test for some cases to find accuracy, Also we may need to normalize input for some ML algorithm)

Define the model

Fit the Model

Predict