Unsupervised Learning: Clustering

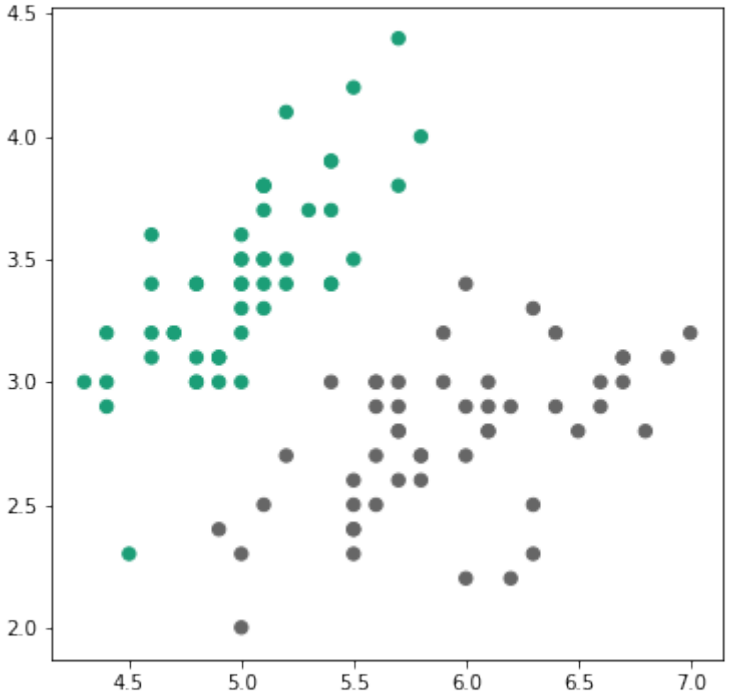
Mohd Halim Mohd Noor, PhD

Outline

- Unsupervised Learning
- K-means Clustering
- Hierarchical Clustering

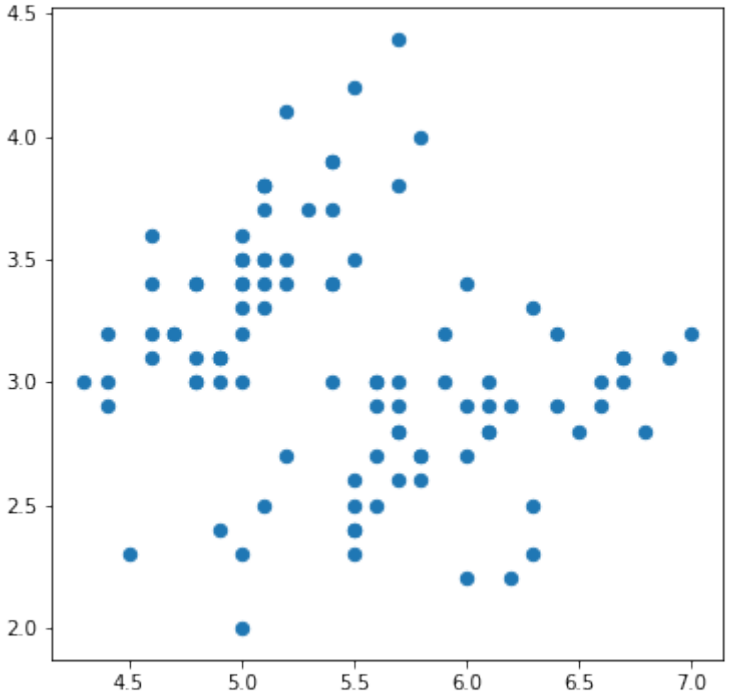
What is Unsupervised Learning?

- In supervised learning, our data (observations) consists of input and output, $(x_i, y_i), i = 1, \ldots, N$
  - Such data is called labeled data
  - y_i are the labels for the x_i



What is Unsupervised Learning?

- In unsupervised learning, we are dealing with input only $x_i, i = 1, \ldots, N$
  - This is called unlabeled data

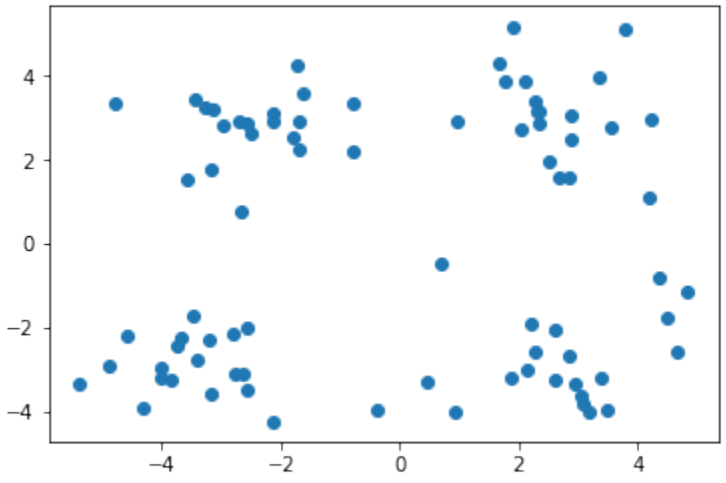


Unsupervised Learning

- A method to uncover latent structure in data (model the underlying structure of the data)
- What can we tell about the data?
  - How many groups can we make out of the data?
  - How do we effectively represent the data?
  - Grouping (clustering) customers by purchasing behaviors

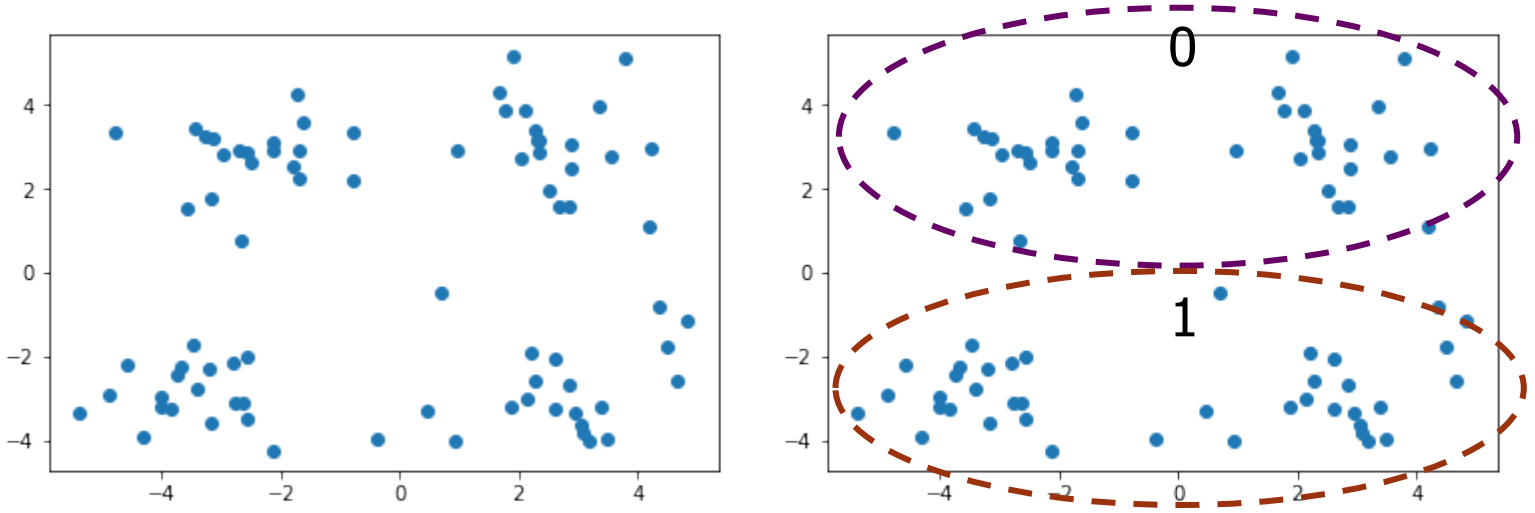
Clustering

- Automatically organizing data into groups (clusters) based on their similarity (automatic classification)
- One data point belong to only one cluster



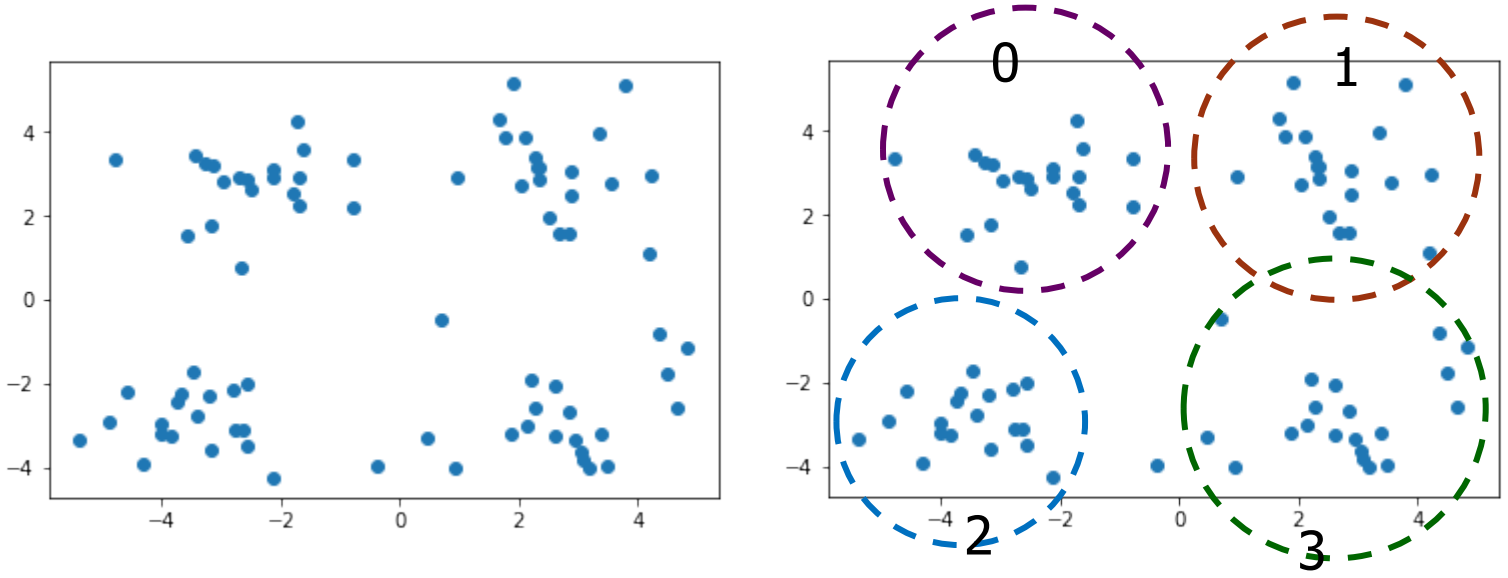
Clustering

- Automatically organizing data into groups (clusters) based on their similarity (automatic classification)
- One data point belong to only one cluster



Clustering

- Automatically organizing data into groups (clusters) based on their similarity (automatic classification)
- One data point belong to only one cluster



Clustering

- Performance is subjective and domain-specific
- Can get gibberish output

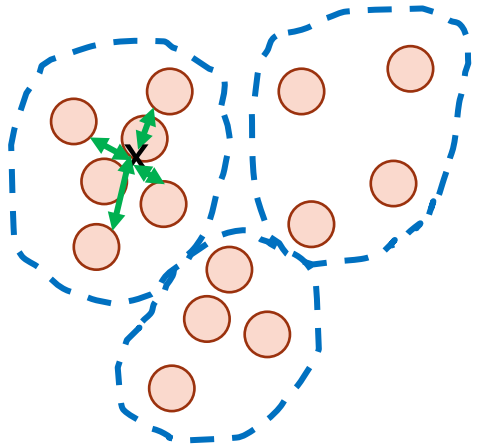
K-Means Clustering

K-means Clustering

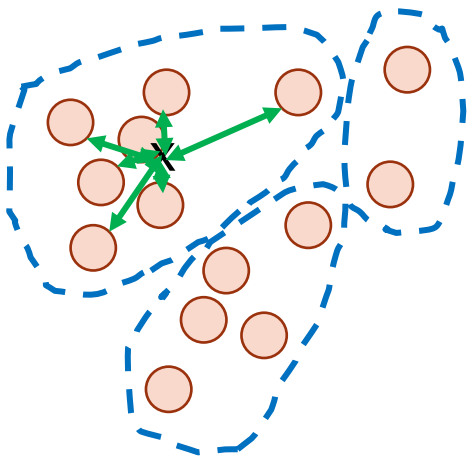
- Each cluster is represented by the center of the cluster (centroid)
- K refers to the number of centroids – must be defined

K-means Clustering

- Every data point is assigned to a cluster
  - The sum of the distance between the data points and cluster's centroid is minimum



Distance is minimum



Distance is not minimum

K-means Algorithm

0. Set K (number of centroids)

1. Initialize the centroids. The values can be randomly initialized.

2. Until the clusters stop changing, repeat:

   2a. For each data point, calculate its Euclidean distance to centroids and assign them to the nearest cluster.

   2b. Update centroids by calculating the mean of all data points belong the clusters.

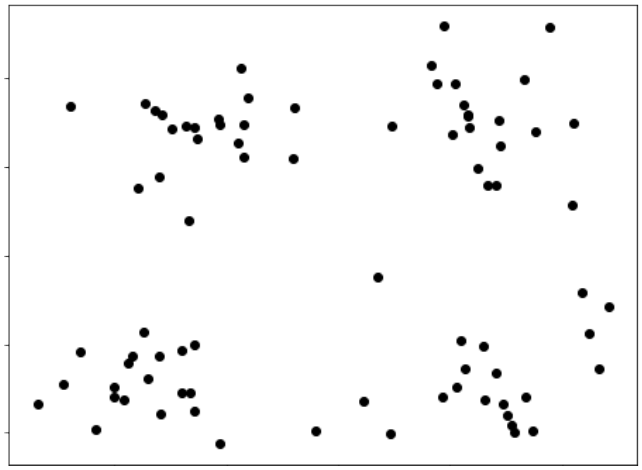
Stopping Criteria

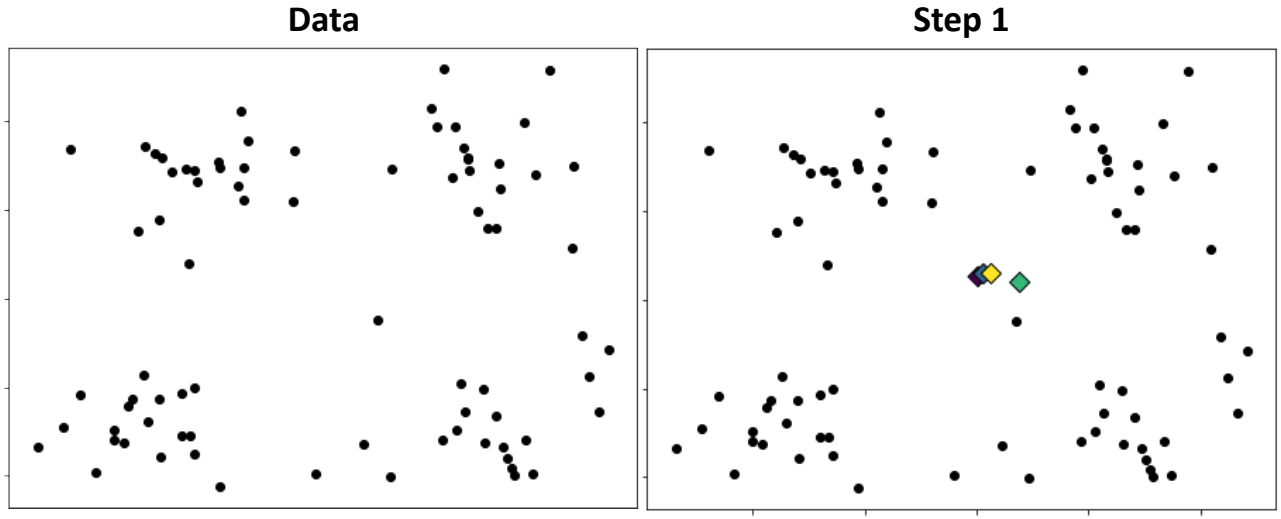
- Centroids of newly formed clusters do no change
- Maximum number of iterations has reached

UNIVERSITI SAINS MALAYSIA

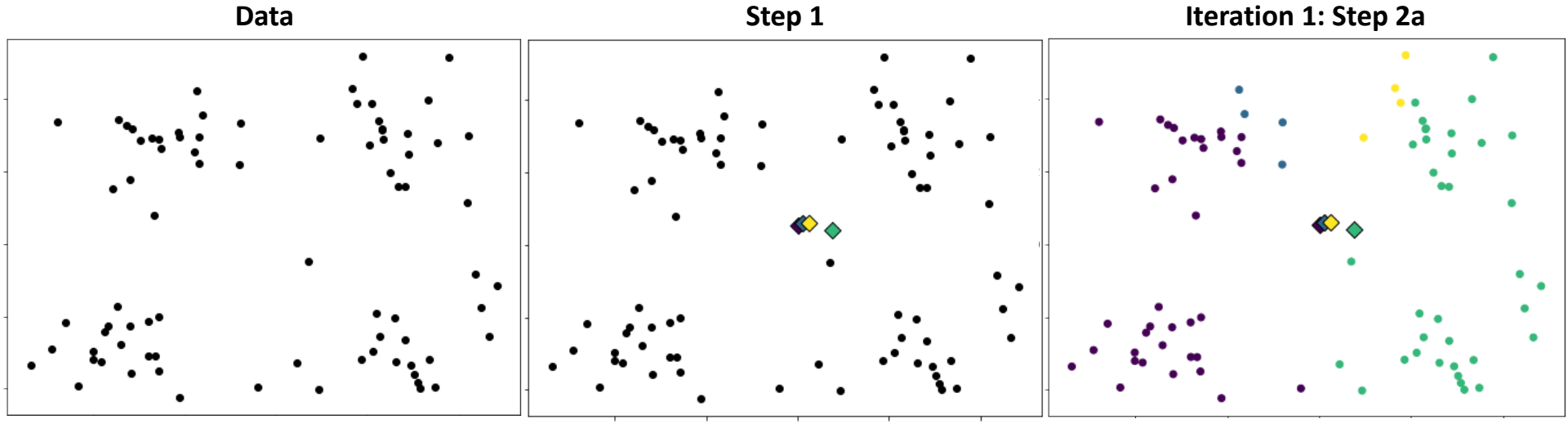
APEX™

Data





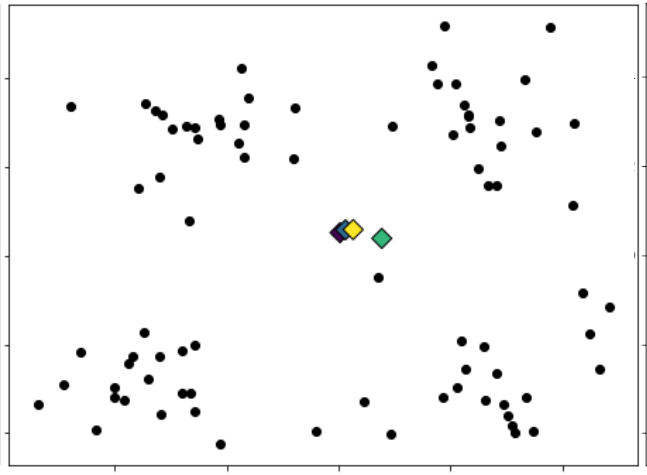
Initialize clusters (centroid), μ_k randomly



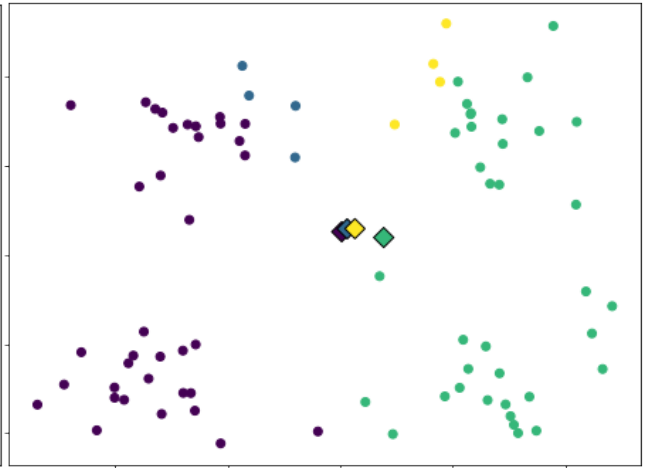
For each data point, calculate its distance to μ_k and assign them to the nearest cluster

Data

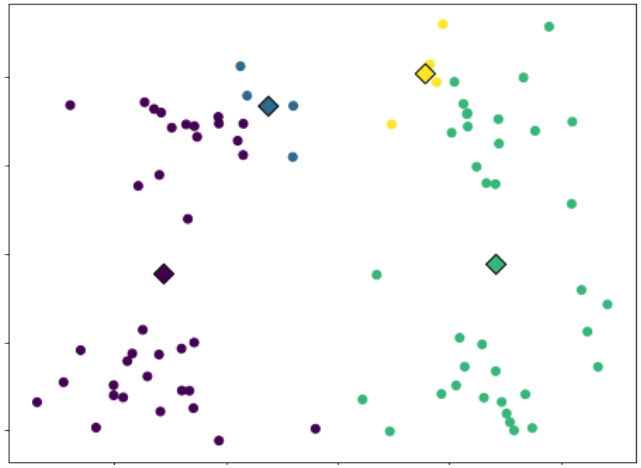
Step 1



Iteration 1: Step 2a



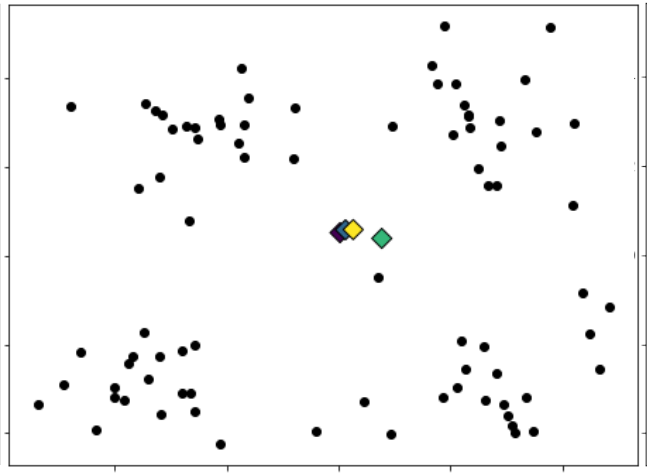
Iteration 1: Step 2b



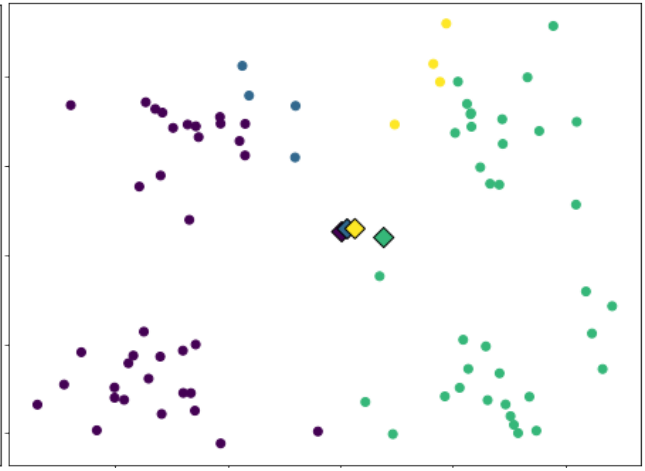
Update the clusters, μ_k by calculating the mean of all data points belong the clusters

Data

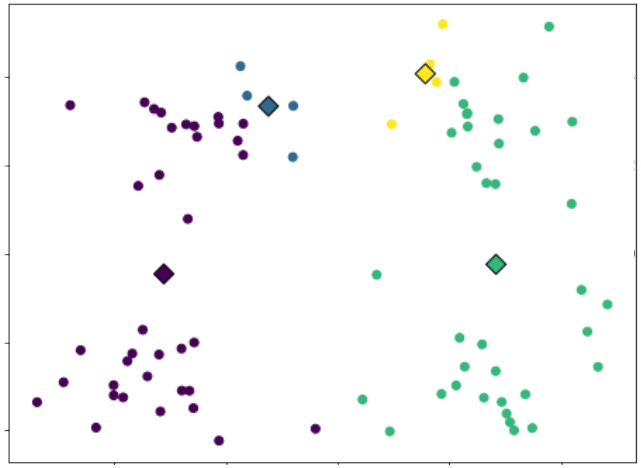
Step 1



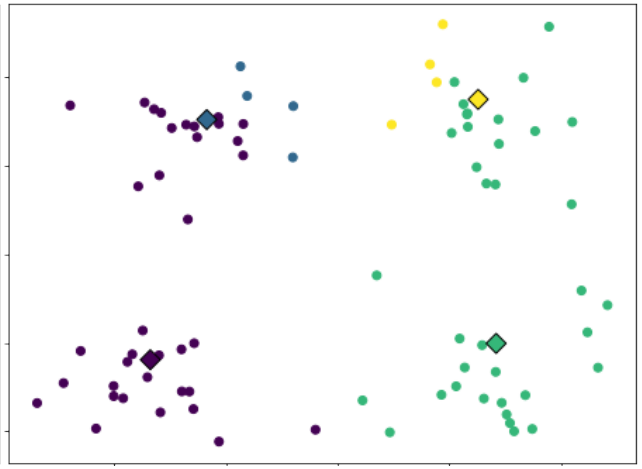
Iteration 1: Step 2a



Iteration 1: Step 2b



Iteration 2: Step 2b



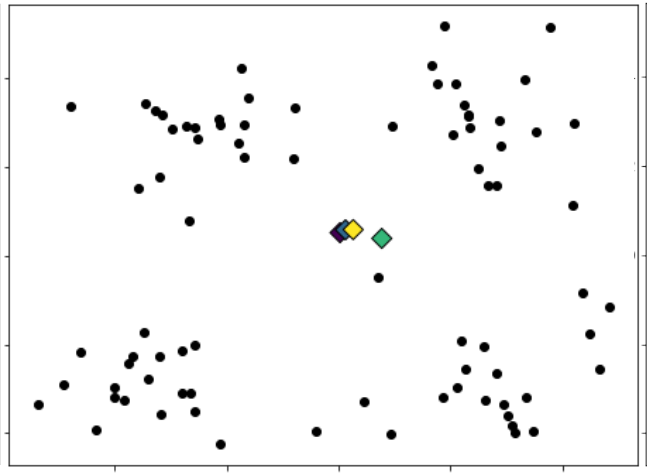
Repeat the steps until the stopping criteria is met

UNIVERSITI SAINS MALAYSIA

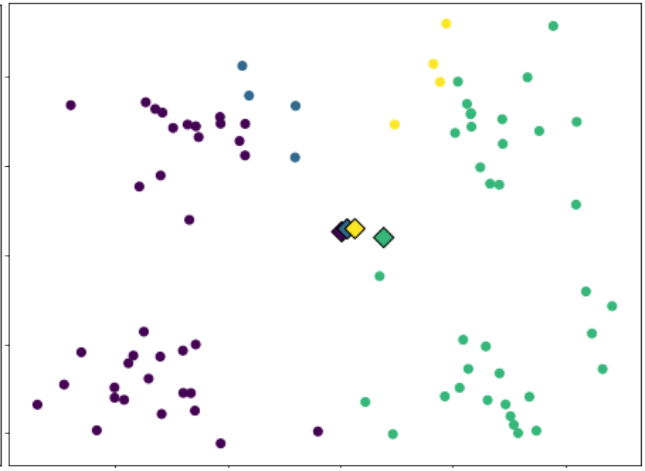
APEX™

Data

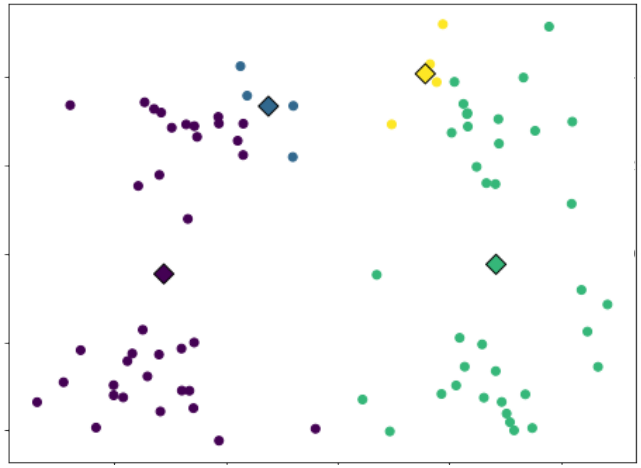
Step 1



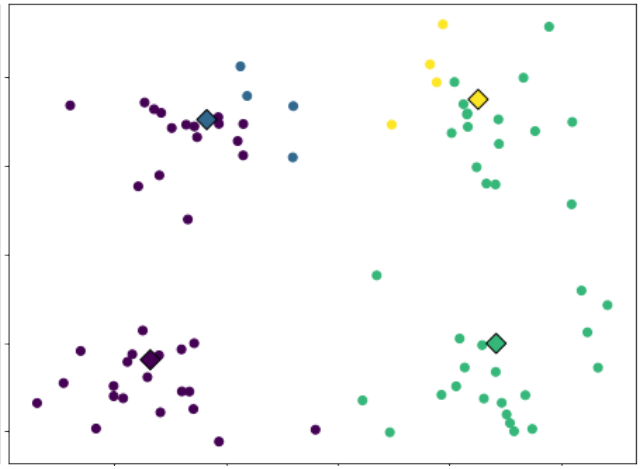
Iteration 1: Step 2a



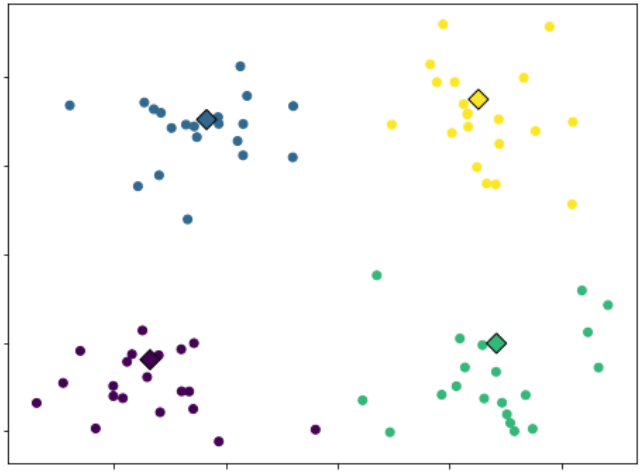
Iteration 1: Step 2b



Iteration 2: Step 2b



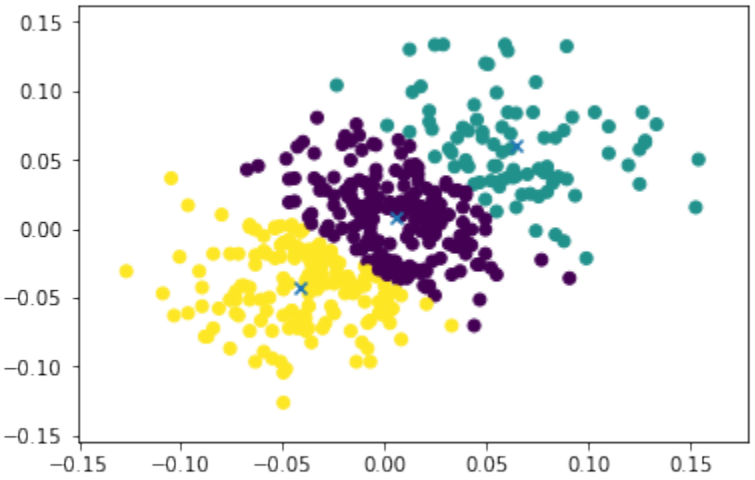
Final Result



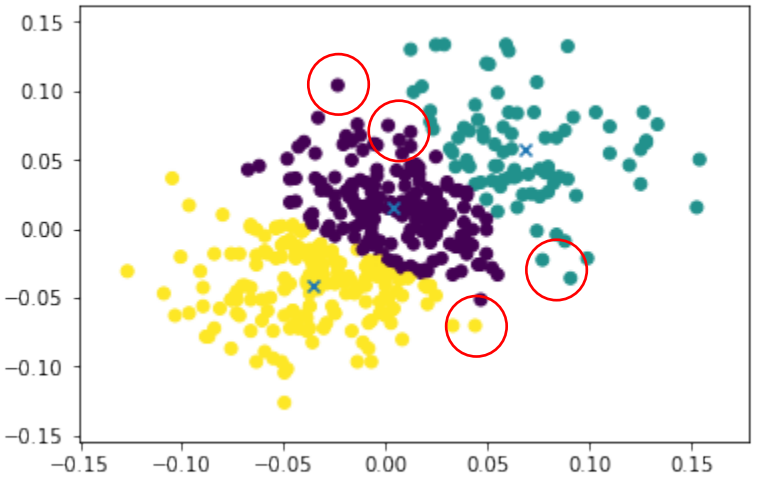
Limitation

- K-means is an objective-based approach that requires us to pre-specify the number of clusters K, the initial centroids
- The final result is somewhat random where it depends on the random initialization we started with

Limitation



Initial centroid =
[-0.12,0.15], [-0.15,0.12], [-0.10,0.11]



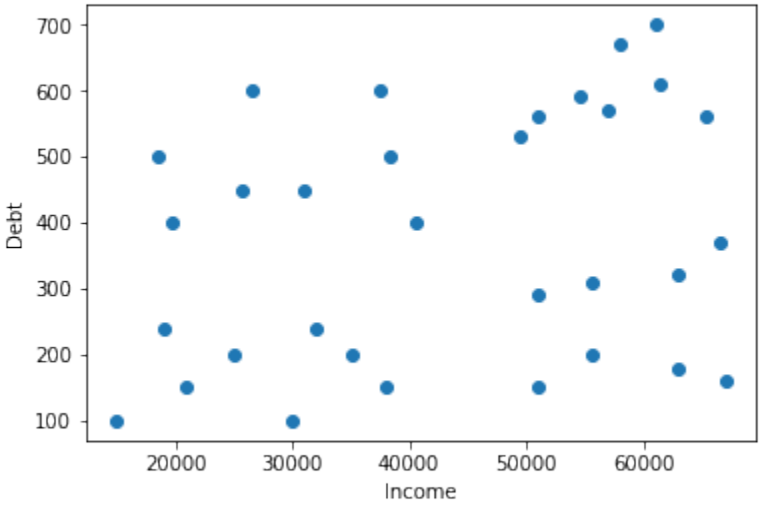
Initial centroid =
[0.12,0.10], [0.09,0.09], [0.10,0.11]

K-Means++

- A method to initialize the centroids before the K-Means clustering

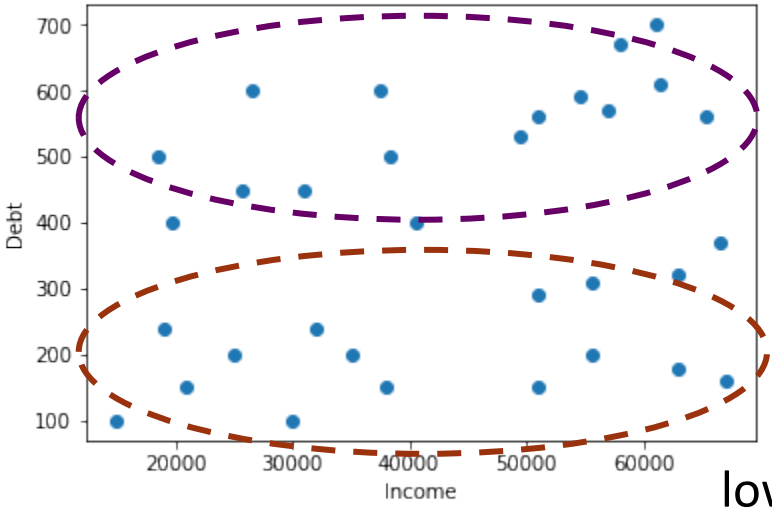
1. Randomly select the first centroid from the data points
2. Compute the distance, $D(x)$ of each point from the chosen centroid
3. Choose one new (next) data point with the probability $\frac{D(x)^2}{\sum_{x \in X} D(x)^2}$
4. We repeat step 2 and step 3 until k clusters have been chosen
5. Proceed with K-means algorithm

How many clusters?



How many clusters?

high debt



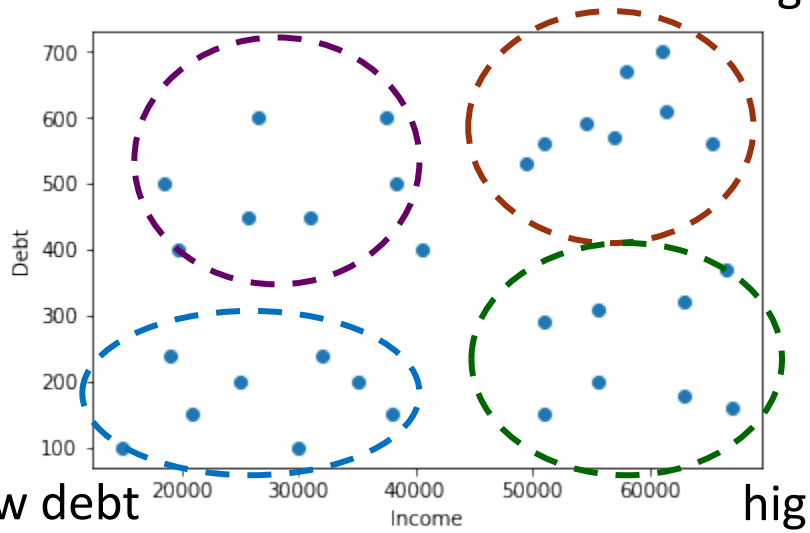
low debt

Two clusters?

How many clusters?

low income high debt

high income high debt

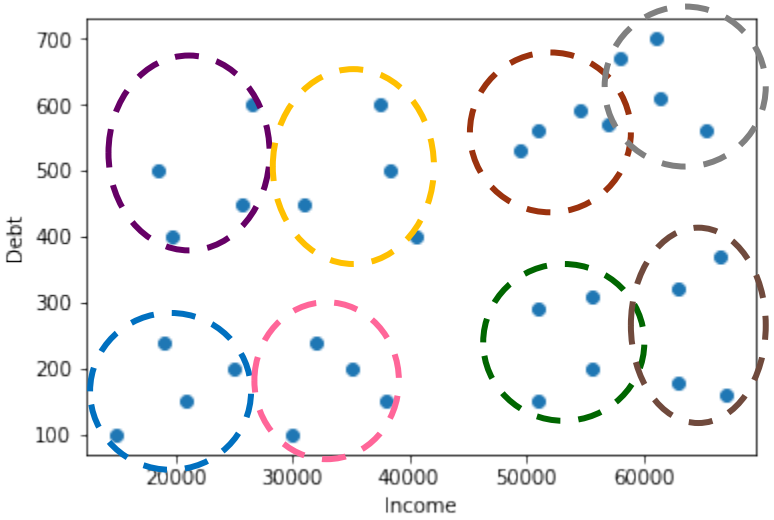


low income low debt

high income low debt

Four clusters?

How many clusters?



Eight clusters?

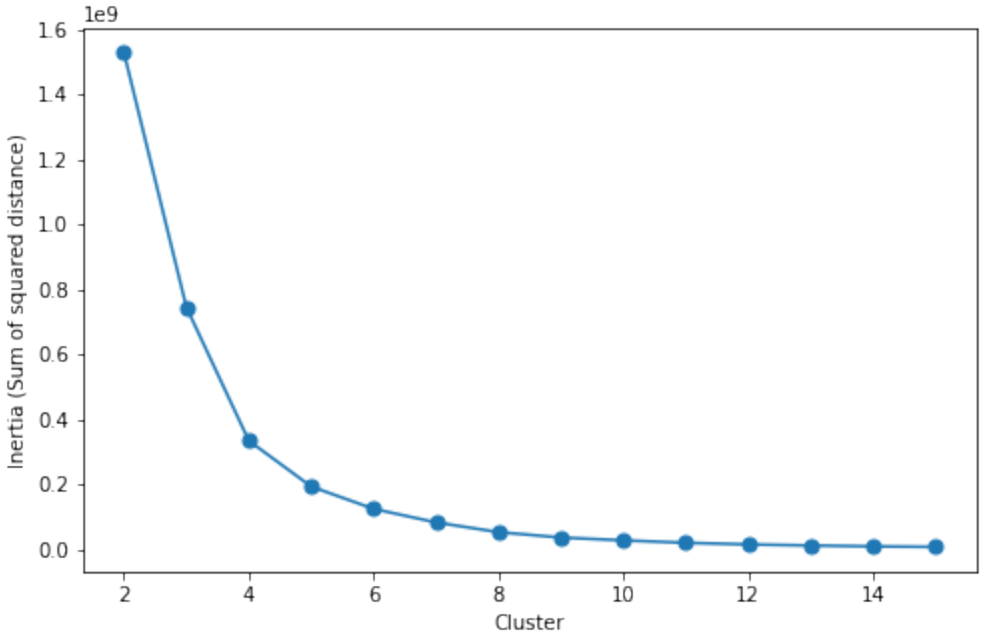
How many clusters?

- We can have any number of clusters
- What would be the maximum number of possible clusters?
- What would be the optimum number of clusters?

How many clusters?

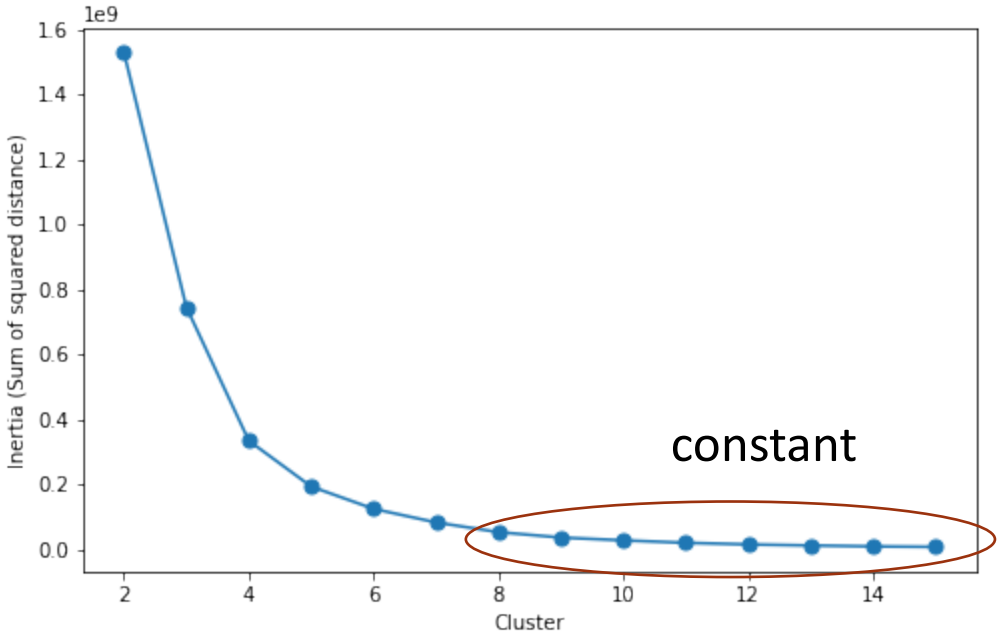
- Start with minimum number of clusters
- Calculate and record the summation of distances of the data points to their closest centroid
- Increase the number of clusters (until the maximum number of clusters)

How many clusters?



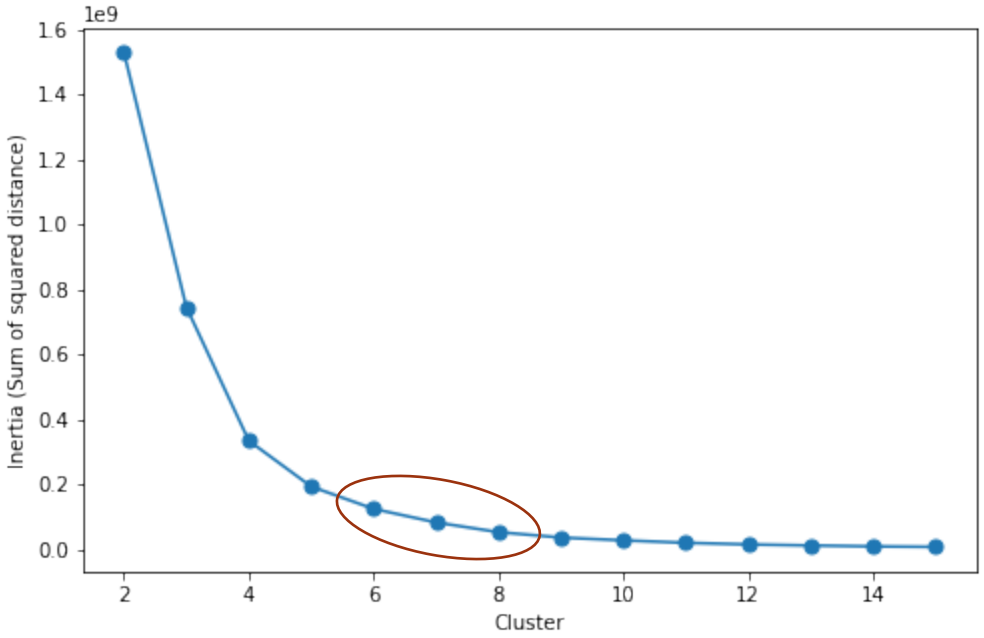
# of clusters increase from 2 to 15

How many clusters?



# of clusters increase from 2 to 15

How many clusters?



6-8 clusters are optimal

Lab/Practical Session

Lab/Practical Session

- Python packages
  - Pandas
  - Numpy
  - Scikit-learn (clustering)
  - Matplotlib & Seaborn (plotting)
- Dataset - http://bit.ly/customers_dataset
- Jupyter Notebook

- http://bit.ly/sophic-ml-workshop

Hierarchical Clustering

Hierarchical Clustering

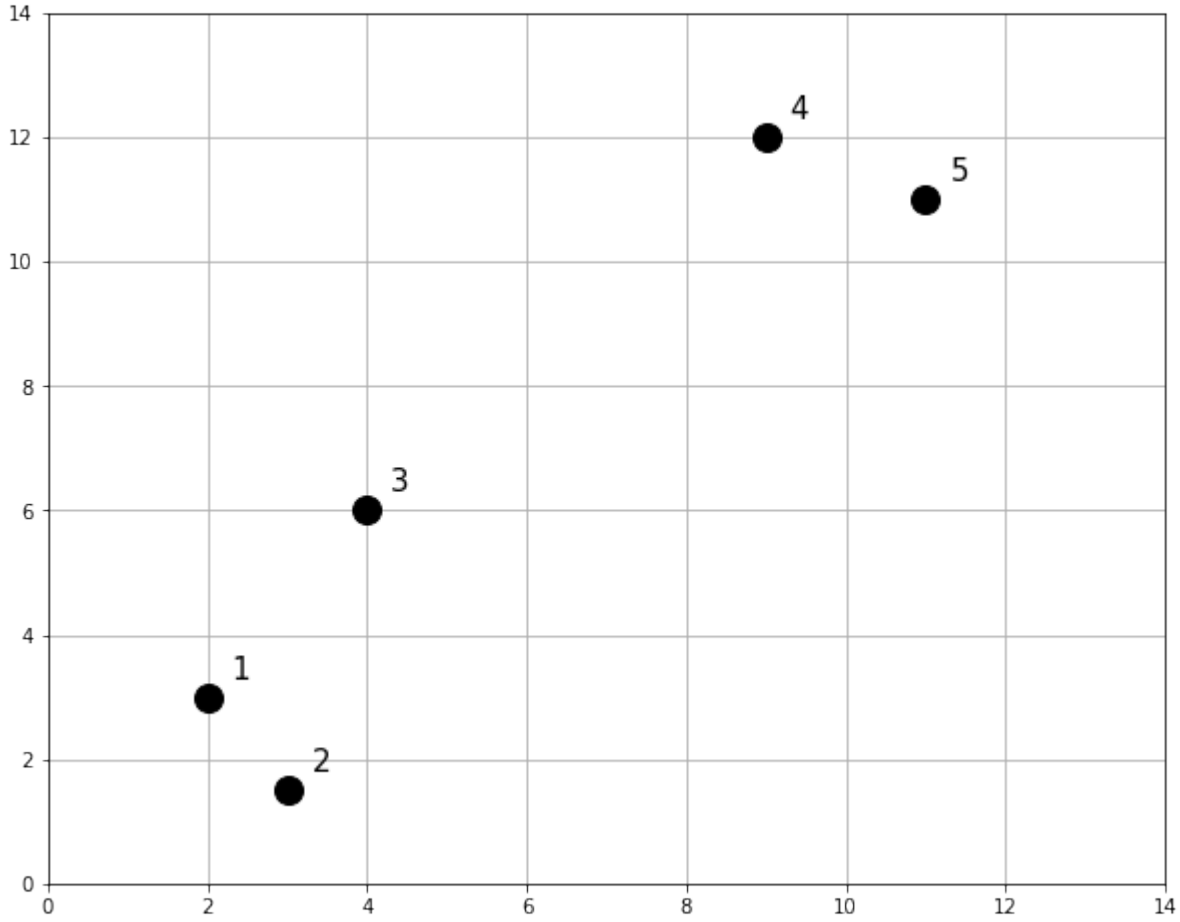
- K-means is an objective-based approach that requires us to pre-specify the number of clusters K
- The final result is somewhat random where it depends on the random initialization we started with

Hierarchical Clustering

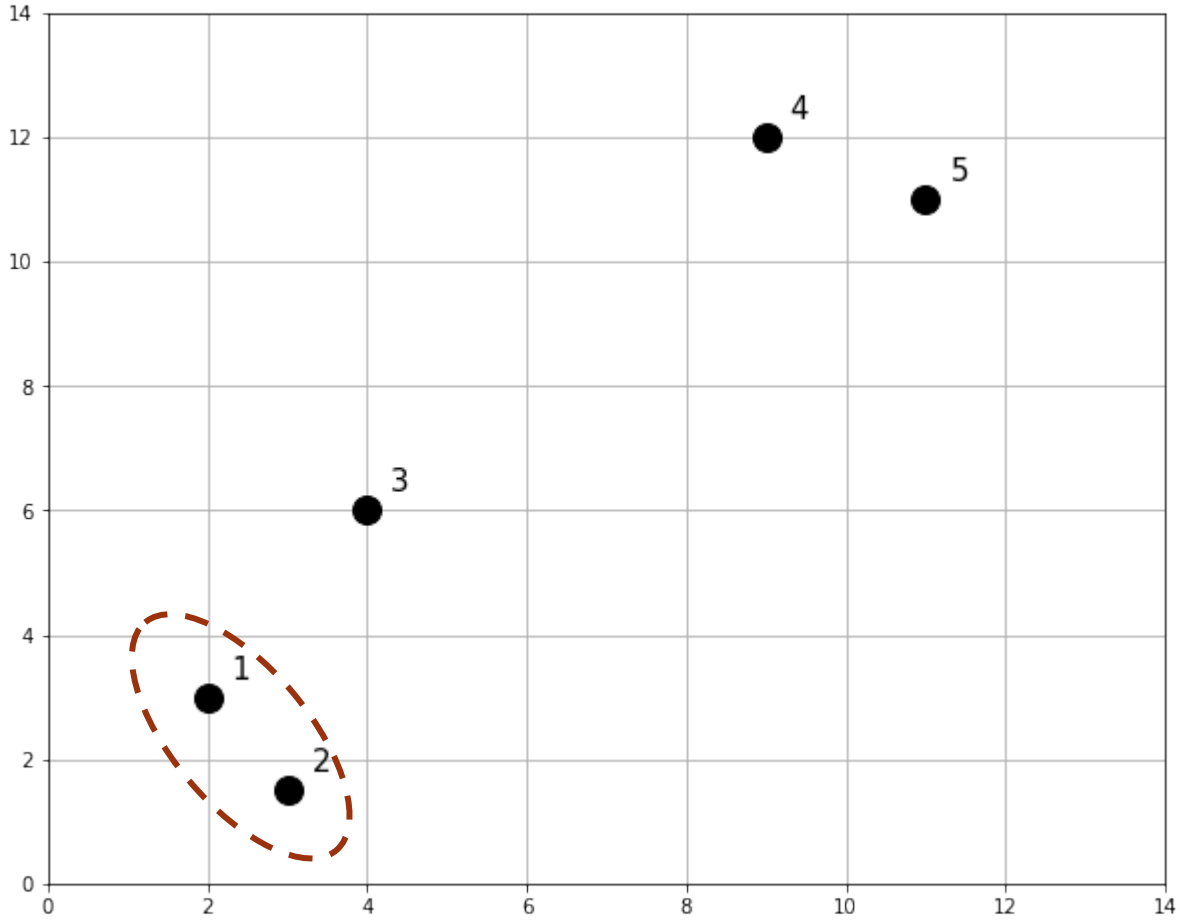
- Hierarchical clustering is an alternative approach that does not require a pre-specified choice of K which provides a deterministic result
- Bottom-up or agglomerative hierarchical clustering

Agglomerative Hierarchical Clustering

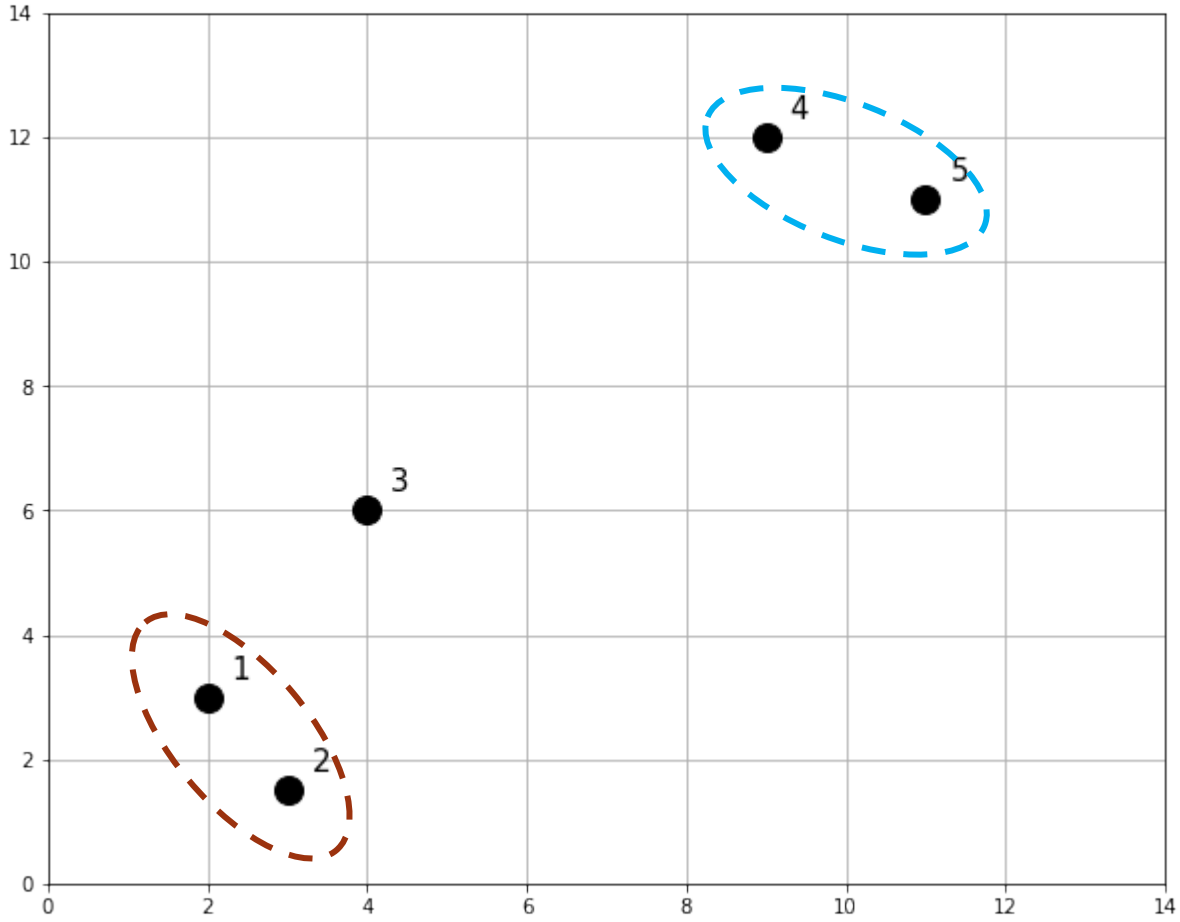
1. Start with each point as its own cluster
2. Identify two closest clusters and merge them
3. Repeat until all points are in a single cluster



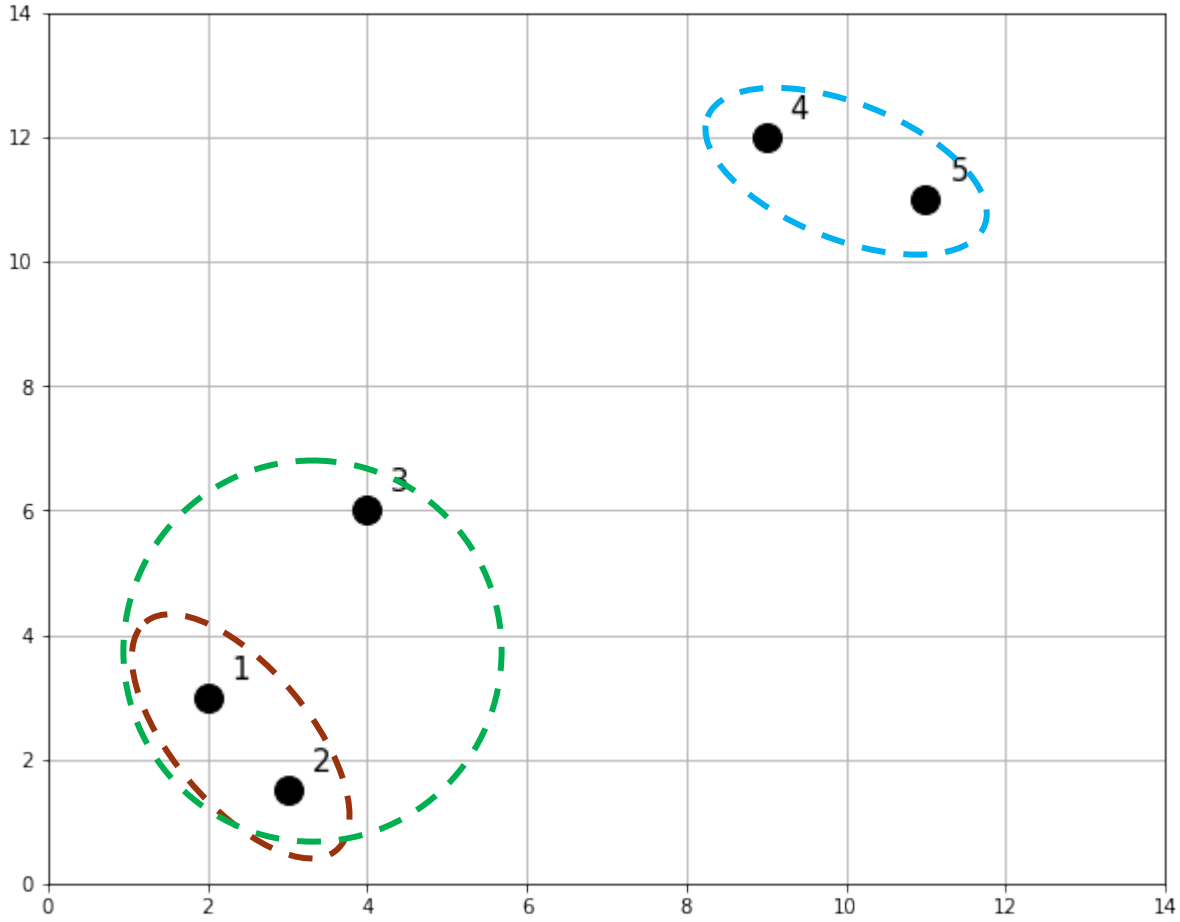
Each point starts as its own cluster



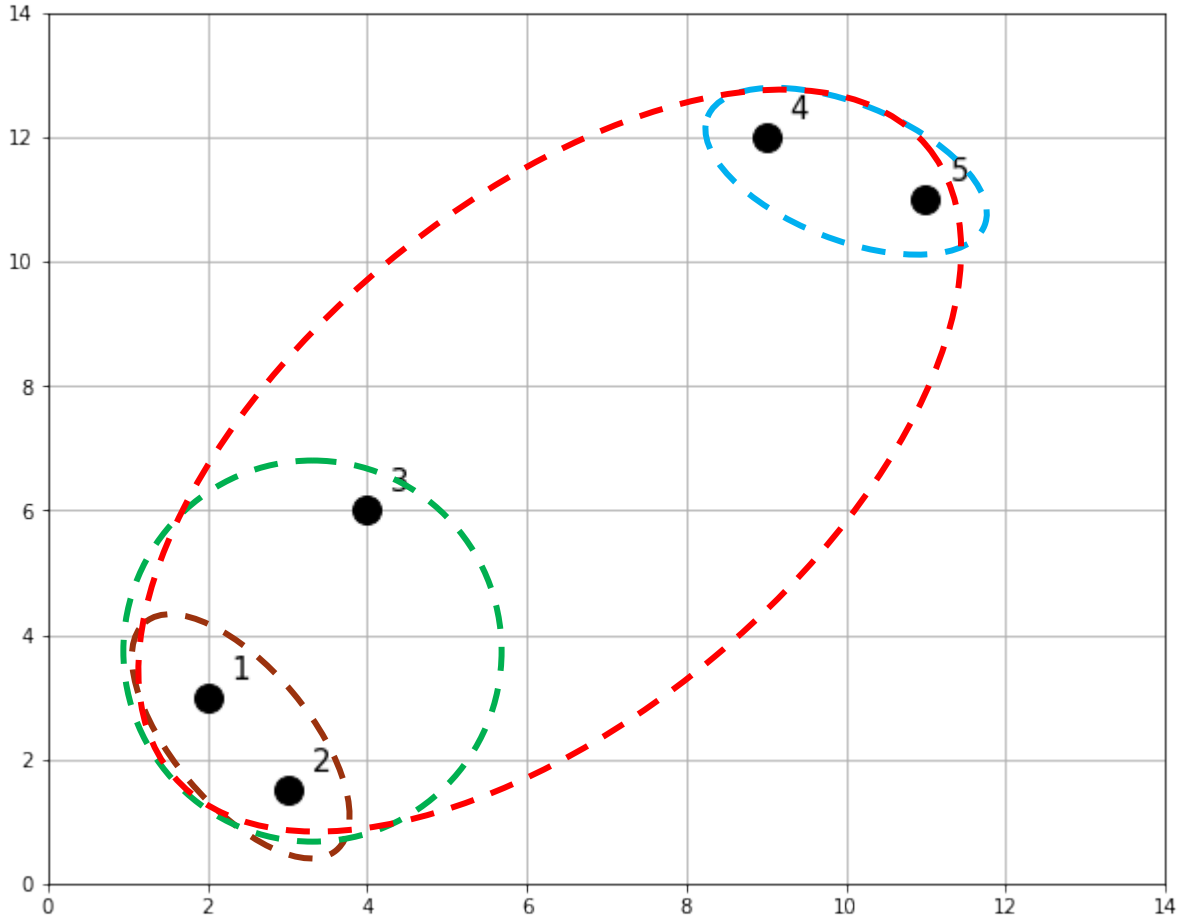
Two clusters (points) that are closest to each other are merged



The next two closest clusters are merged



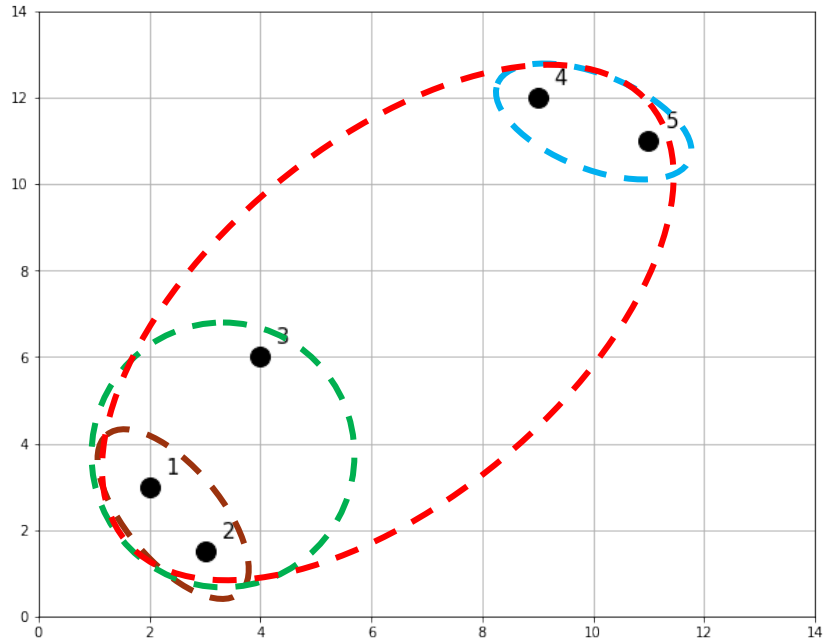
The next two closest clusters are merge

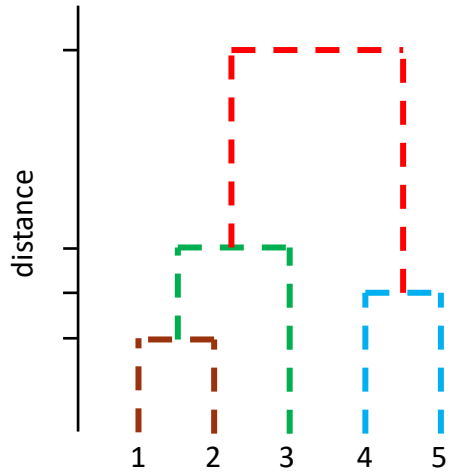


Until all the points are in a single cluster

Dendrogram

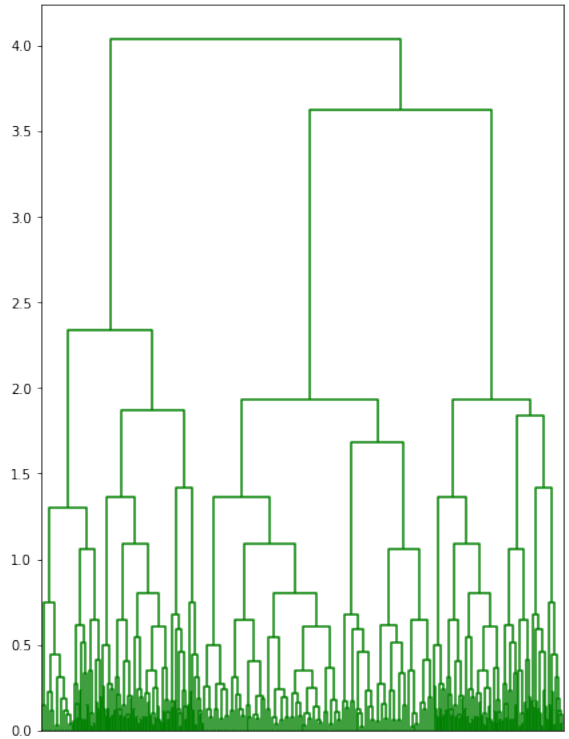
- We use dendrogram to visualize the result of the clustering





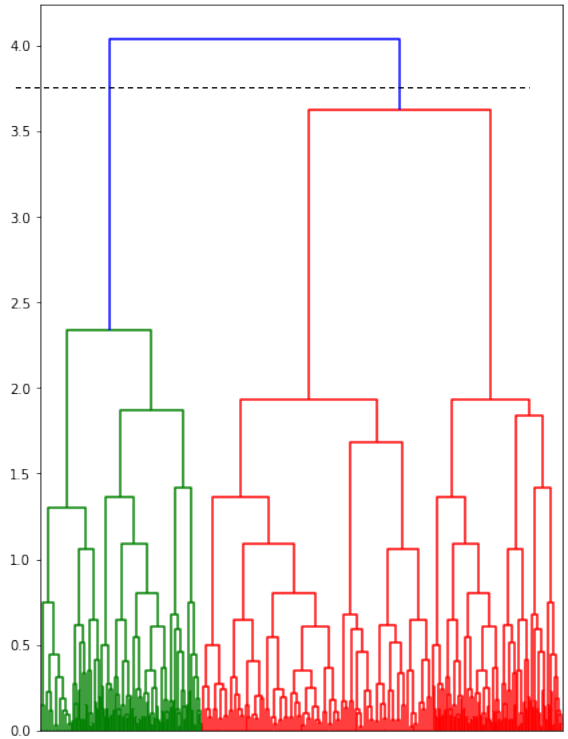
y-axis on dendrogram is the distance between the clusters that got merged at that step

Cutting dendrograms



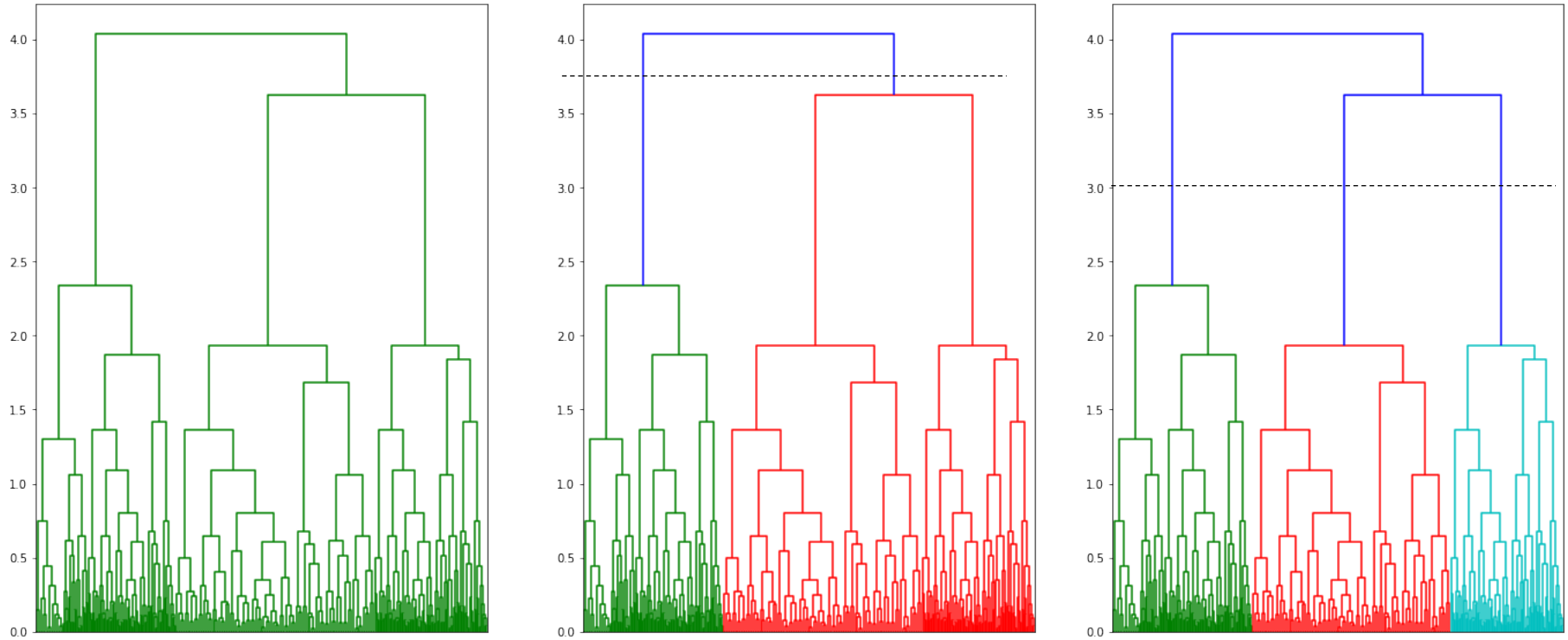
- How do we cluster the data points based on the dendrogram?

Cutting dendrograms



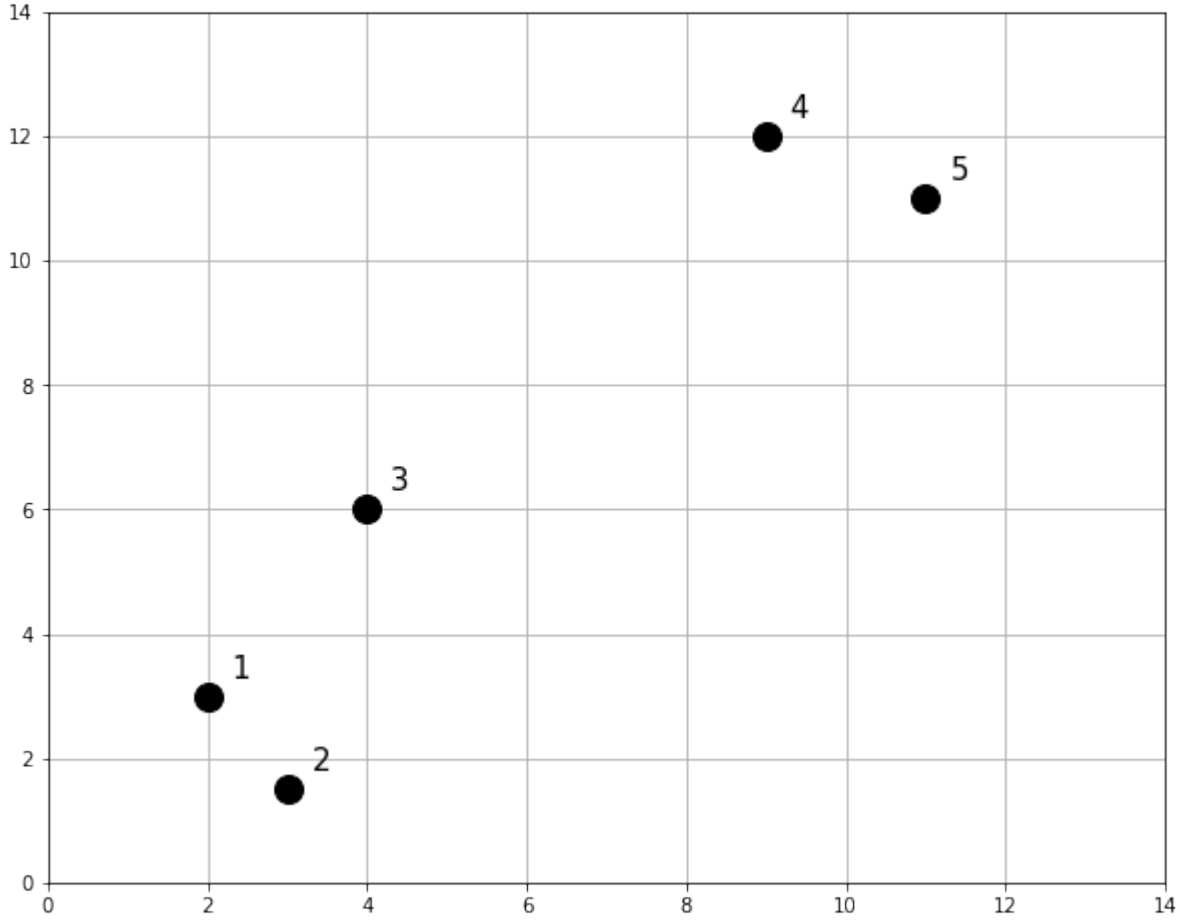
- Dendrogram cut at height 3.7, resulting in $K = 2$ clusters

Cutting dendrograms



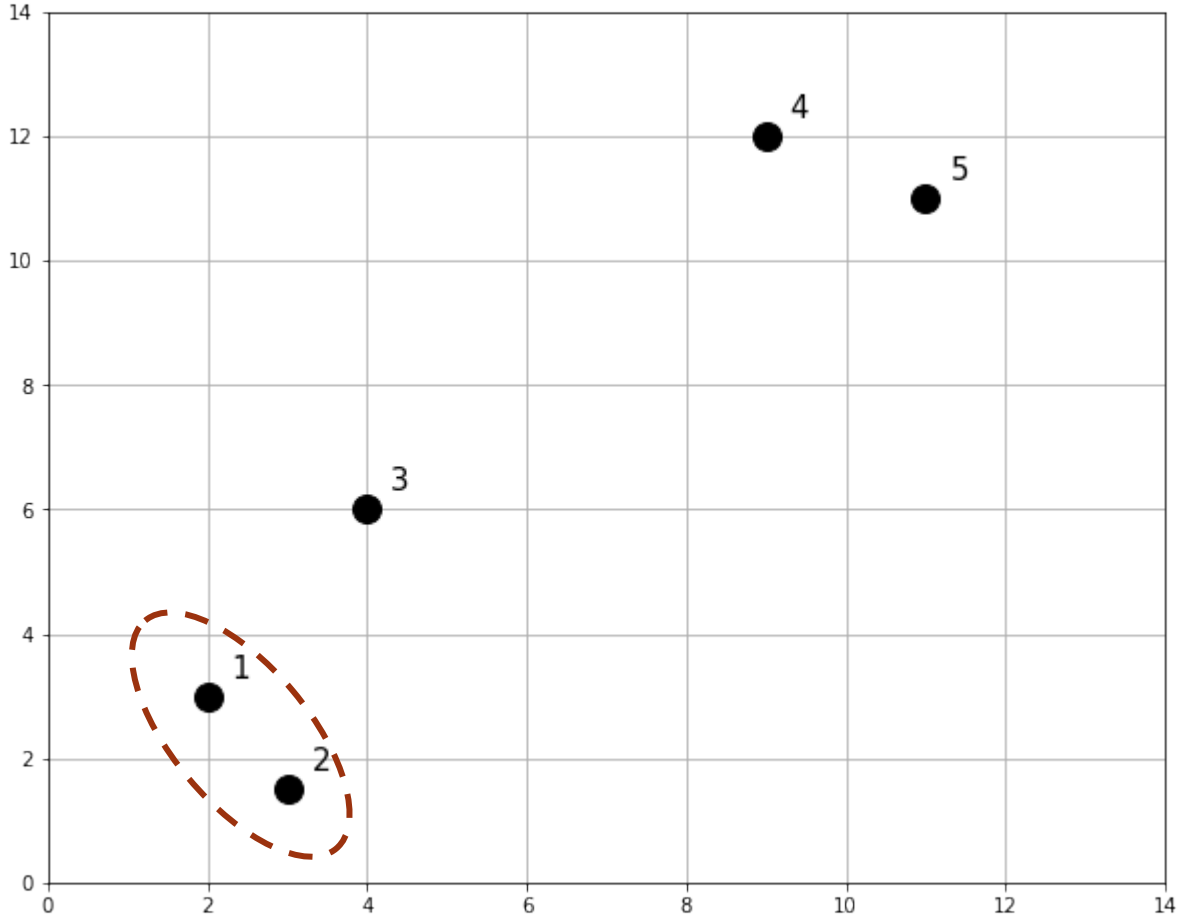
- Dendrogram cut at height 3.00, resulting in $K = 3$ clusters

Linkages



Distances between data points

Linkages



Distances between data point and cluster

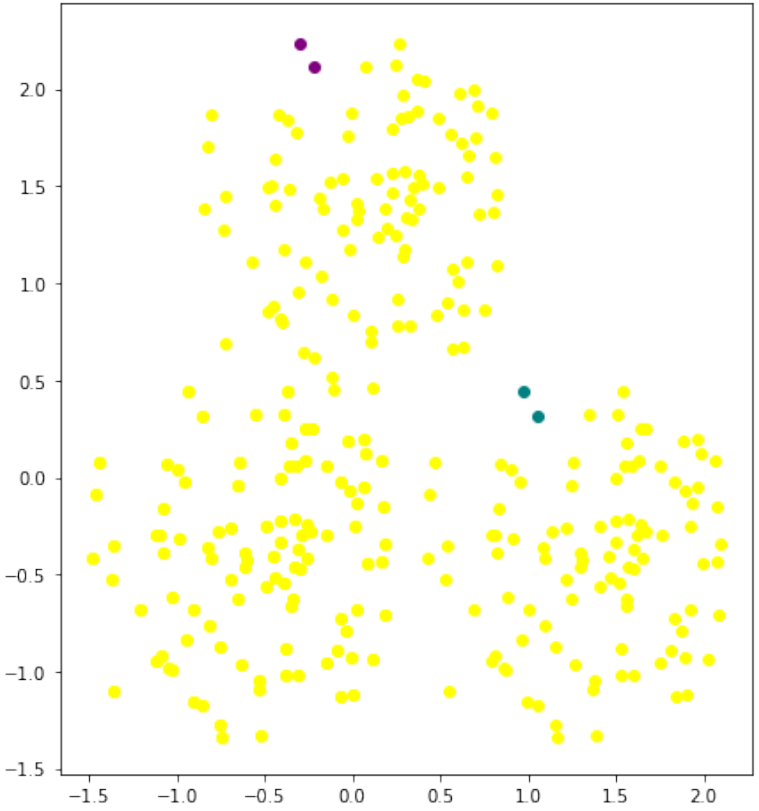
Linkages

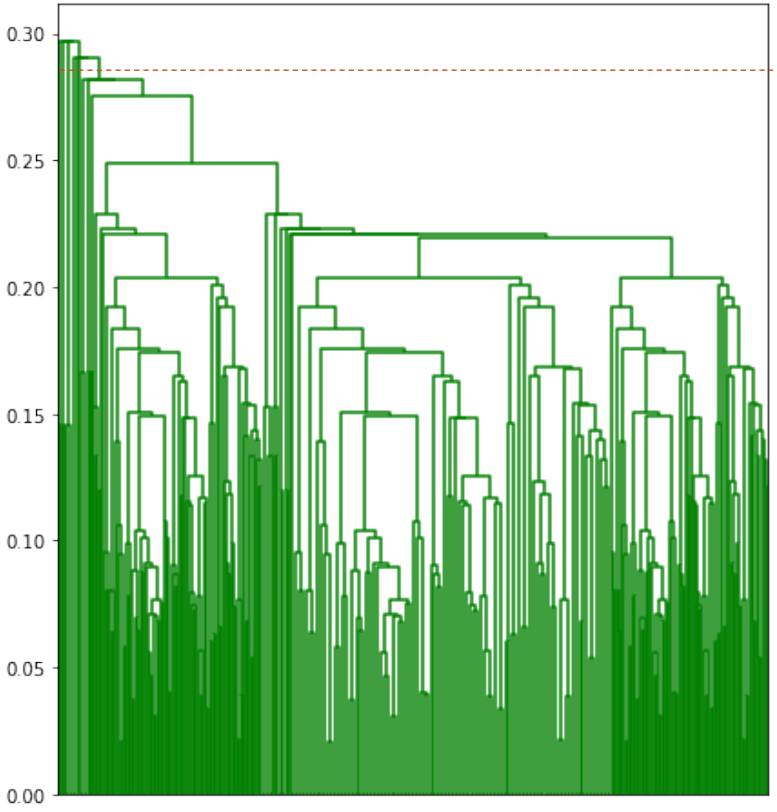
- The distance between two clusters

Linkage	*Description*
Single	Compute all distances between the data points in cluster A and the data points in cluster B, and take the smallest of the distances.
Complete	Compute all distances between the data points in cluster A and the data points in cluster B, and take the largest of the distances.
Average	Compute all distances between the data points in cluster A and the data points in cluster B, and take the smallest average of the distances.
Ward	Instead of calculating the distance directly, two clusters are picked and merged what have the minimum increase in within-cluster variance (error sum of squared) after merging.

Summary

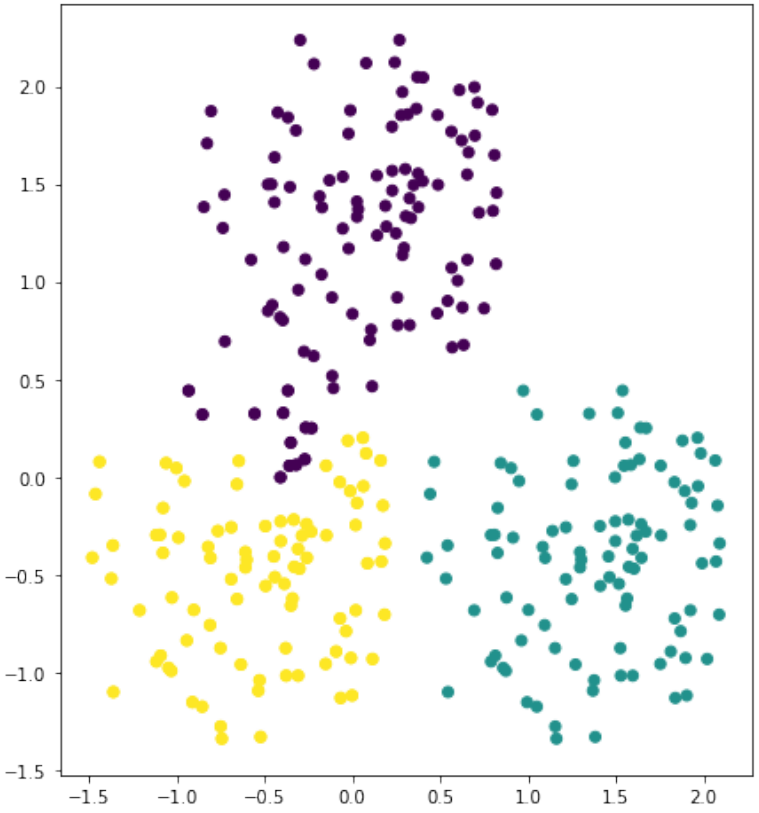
- Single linkage suffers from chaining
  - Needs one pair of data points to be close irrespective of all others
  - Clusters can be too spread out and not compact enough

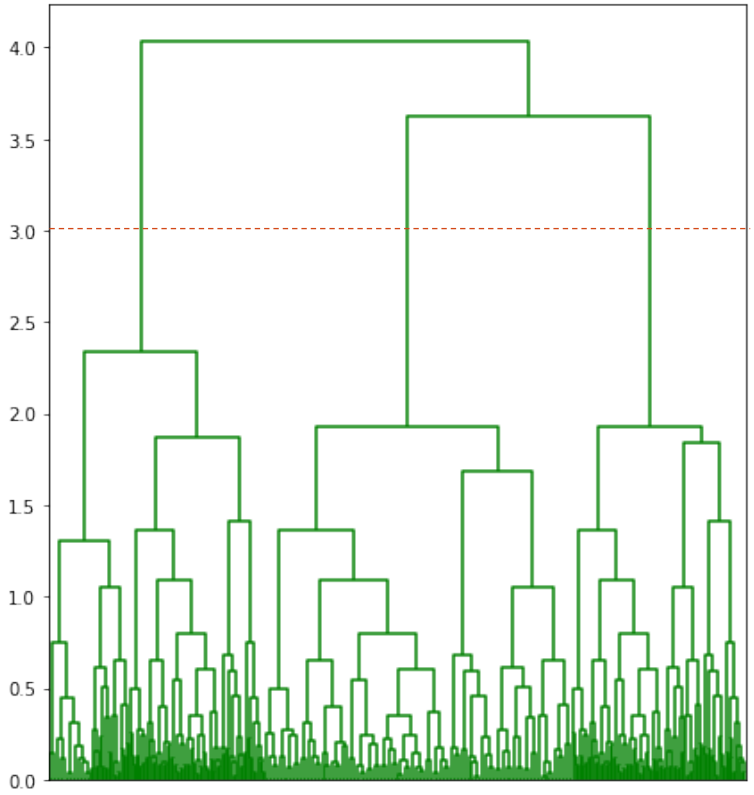




Summary

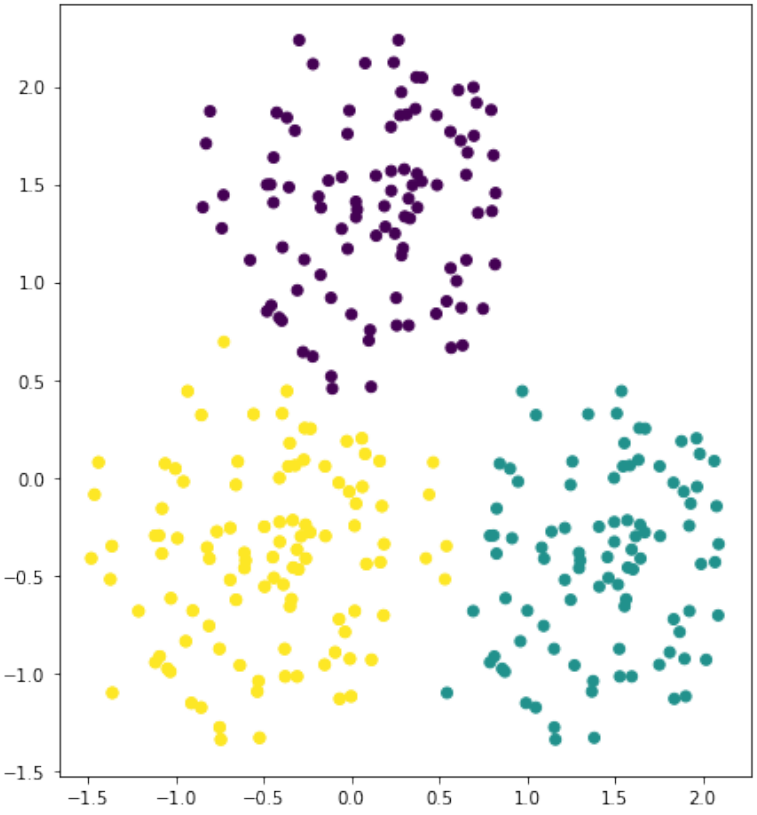
- Complete linkage suffers from crowding
  - Based on worst-case (maximum) dissimilarity between pairs
  - Data points assigned to a cluster can be much closer to members of other clusters than they are to some data points of their own cluster

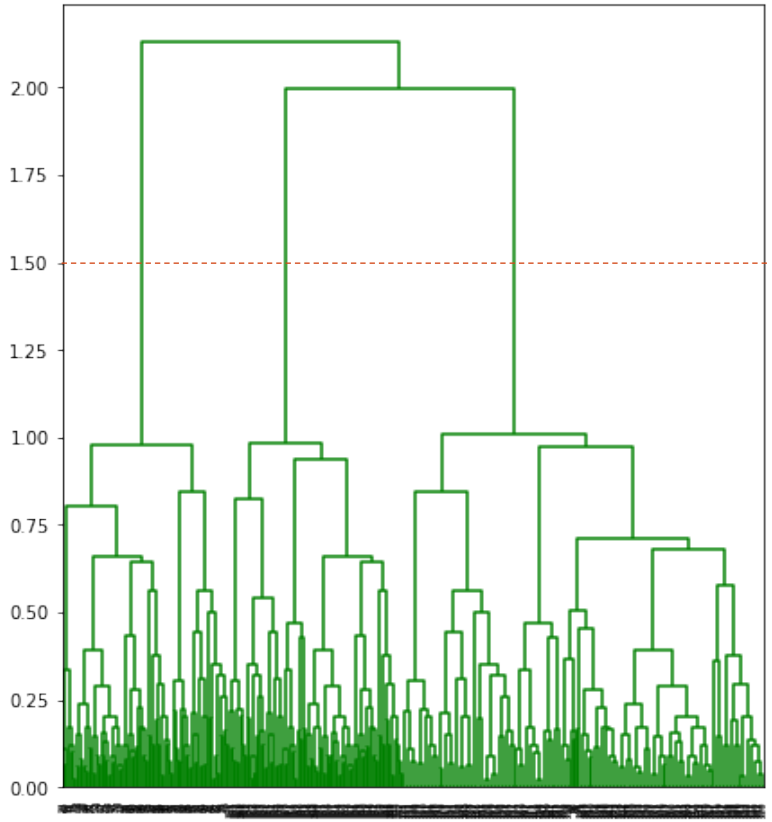




Summary

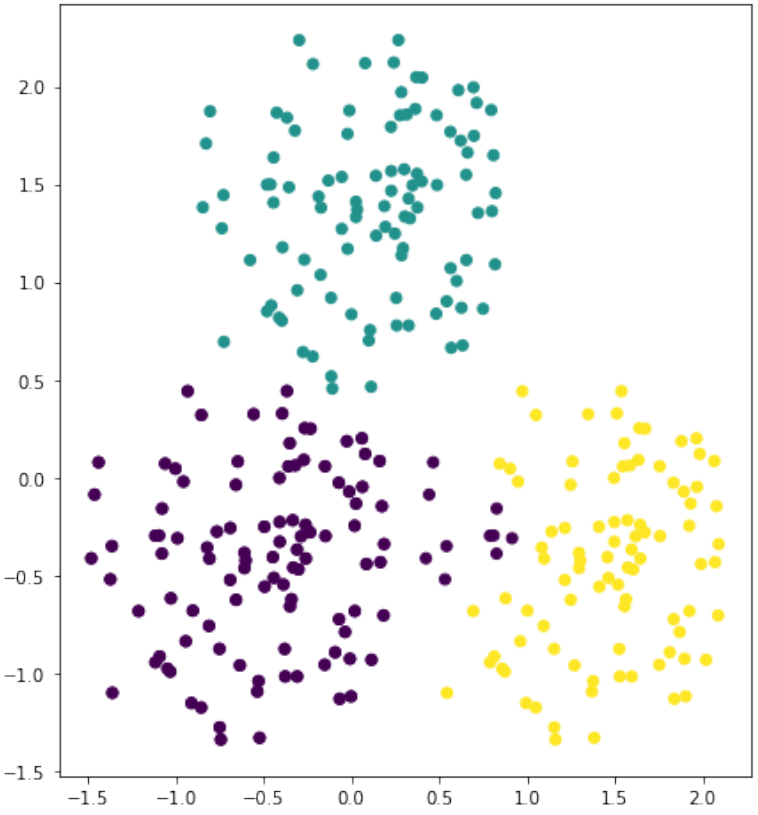
- Average linkage tries to strike a balance
  - Based on average pairwise dissimilarity
  - Clusters tend to be relatively compact and relatively far apart

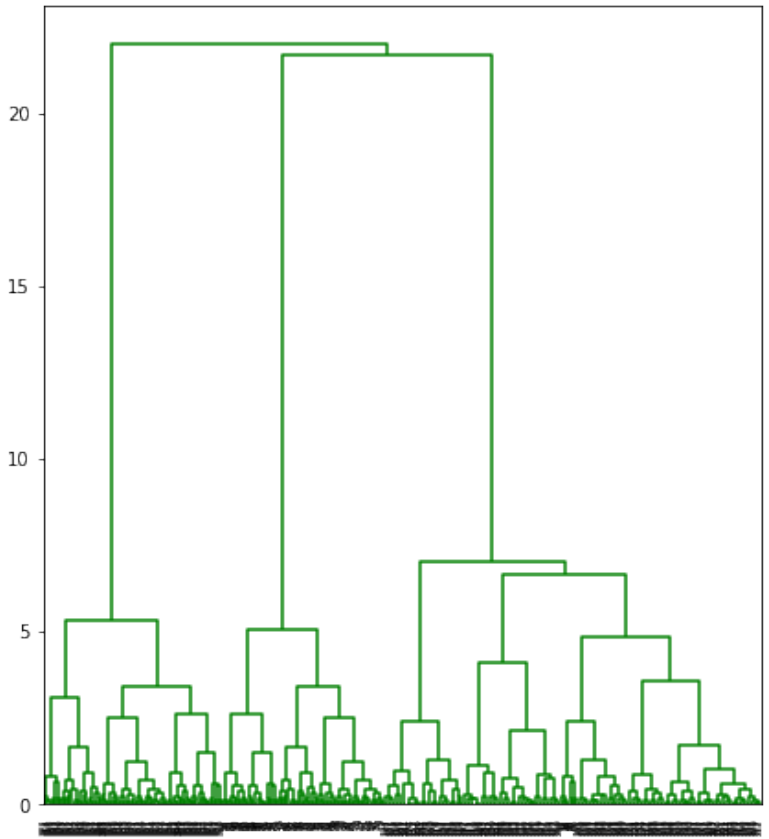




Summary

- Ward linkage tries to strike a balance
  - Based on minimizing the variance within cluster
  - Clusters tend to be relatively compact and relatively far apart





Lab/Practical Session

Lab/Practical Session

- Python packages
  - Pandas
  - Numpy
  - Scikit-learn (clustering)
  - Matplotlib & Seaborn (plotting)
- Dataset - http://bit.ly/customers_dataset
- Jupyter Notebook

- http://bit.ly/sophic-ml-workshop

End