Clustering

A Brief Introduction

What is Clustering?

Cluster *analysis* or clustering is the task of **grouping a set of objects** in such a way that objects in the same group (called a cluster) are more similar (**in some sense**) to each other than to those in other groups (clusters).

Taken from [wikipedia]

What is Clustering?

Cluster *analysis* or clustering is the task of **grouping a set of objects** in such a way that objects in the same group (called a cluster) are more similar (**in some sense**) to each other than to those in other groups (clusters).

Taken from [wikipedia]

Objects -> people, animals, products, transactions, pixels in an image, audio signals, etc.

What is Clustering?

Cluster *analysis* or clustering is the task of **grouping a set of objects** in such a way that objects in the same group (called a cluster) are more similar (**in some sense**) to each other than to those in other groups (clusters).

Taken from [wikipedia]

Objects -> people, animals, products, transactions, pixels in an image, audio signals, etc.

Whatever you are analysing

What is Clustering?

Cluster *analysis* or clustering is the task of **grouping a set of objects** in such a way that objects in the same group (called a cluster) are more similar (**in some sense**) to each other than to those in other groups (clusters).

Taken from wikipedia

Objects -> people, animals, products, transactions, pixels in an image, audio signals, etc.

Whatever you are analysing

Some sense -> with respect to some features and some similarity measure.

What is Clustering?

Cluster *analysis* or clustering is the task of **grouping a set of objects** in such a way that objects in the same group (called a cluster) are more similar (**in some sense**) to each other than to those in other groups (clusters).

Taken from wikipedia

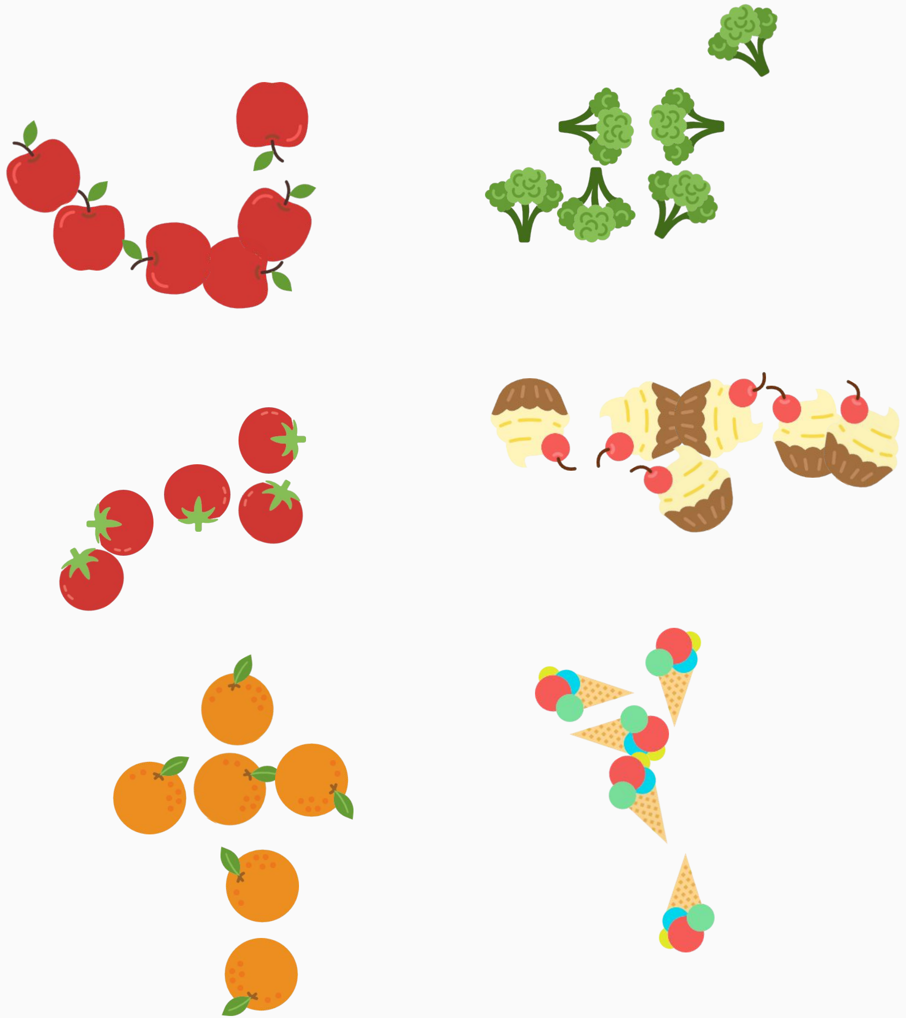
Objects -> people, animals, products, transactions, pixels in an image, audio signals, etc.

Whatever you are analysing

Some sense -> with respect to some features and some similarity measure.

Two people can do a good job with totally different results (ambiguity)

Let’s analyse food ...



Let’s analyse food *with respect to its taste*

Let’s analyse food *with respect to its type*

Let's analyse food *with respect to the stores where they are sold*

Applications

Recommendation Engines

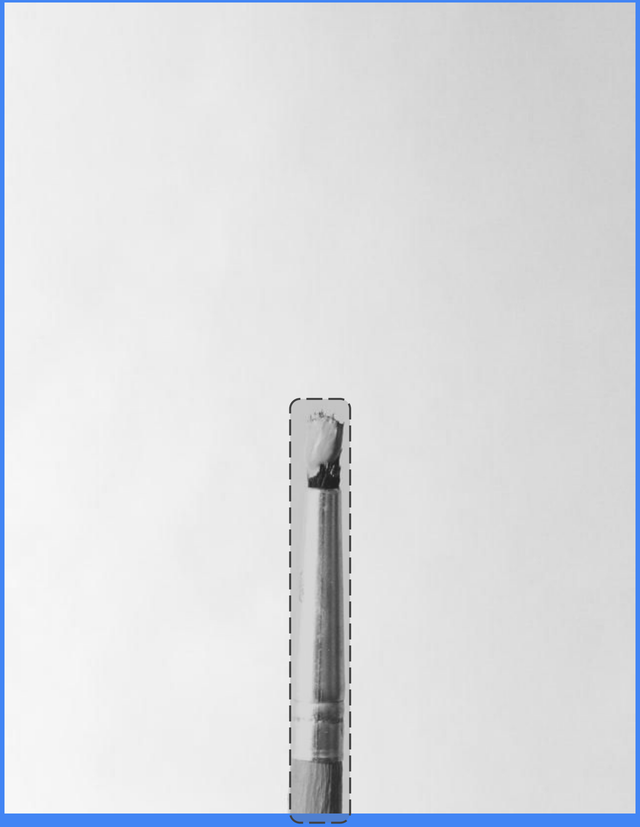
Group customers by some features and then recommend similar products to similar customers with respect to those features

Customer Segmentation

Let’s group our customers by some features and then summarize some demographics about those customers

Image Segmentation

Create groups of pixels with
respect to coherent objects



Speech segmentation

Imagine you have multiple voices in a recording and you wish to isolate them

Anything ...

Really, just imagine things you initially have combined and you wish to split in parts

“Clustering is different from classification”

- God

Clustering is different from classification

Clustering			
Classification			

Adapted from [Modern Multivariate Statistical Techniques]

Clustering is different from classification

	Prior information	
Clustering		
Classification		

Adapted from [Modern Multivariate Statistical Techniques]

Clustering is different from classification

	Prior information	
Clustering	*Number of classes* and *membership* of objects is **unknown**	
Classification		

Adapted from [Modern Multivariate Statistical Techniques]

Clustering is different from classification

	Prior information	
Clustering	*Number of classes* and *membership* of objects is **unknown**	
Classification	*Number of classes* and *membership* of objects is **known**	

Adapted from [Modern Multivariate Statistical Techniques]()

Clustering is different from classification

	Prior information	Objective
Clustering	*Number of classes* and *membership* of objects is **unknown**	
Classification	*Number of classes* and *membership* of objects is **known**	

Adapted from [Modern Multivariate Statistical Techniques]

Clustering is different from classification

	Prior information	Objective
Clustering	*Number of classes* and *membership* of objects is **unknown**	**Exploratory** data analysis
Classification	*Number of classes* and *membership* of objects is **known**	

Adapted from Modern Multivariate Statistical Techniques

Clustering is different from classification

	Prior information	Objective
Clustering	*Number of classes* and *membership* of objects is **unknown**	**Exploratory** data analysis
Classification	*Number of classes* and *membership* of objects is **known**	Classify **new items**

Adapted from [Modern Multivariate Statistical Techniques]

Clustering is different from classification

However….nothing should stop you from using clustering algorithms in any innovative way it make sense for a particular application. For instance, it could be used as automatic preprocessing or post-processing in certain computer vision applications.

Algorithms

Types of algorithms

Connectivity-based

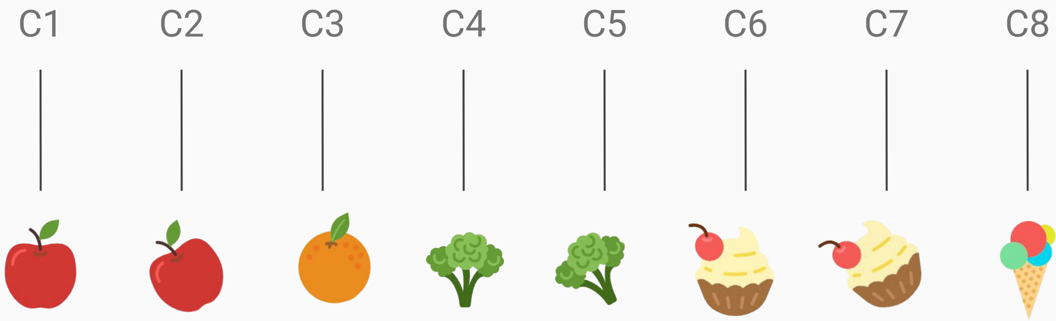
Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that are more dissimilar should belong to the same cluster*. Also, you can iteratively form new clusters out of old ones or create new a division of a previous cluster.



Types of algorithms

Connectivity-based

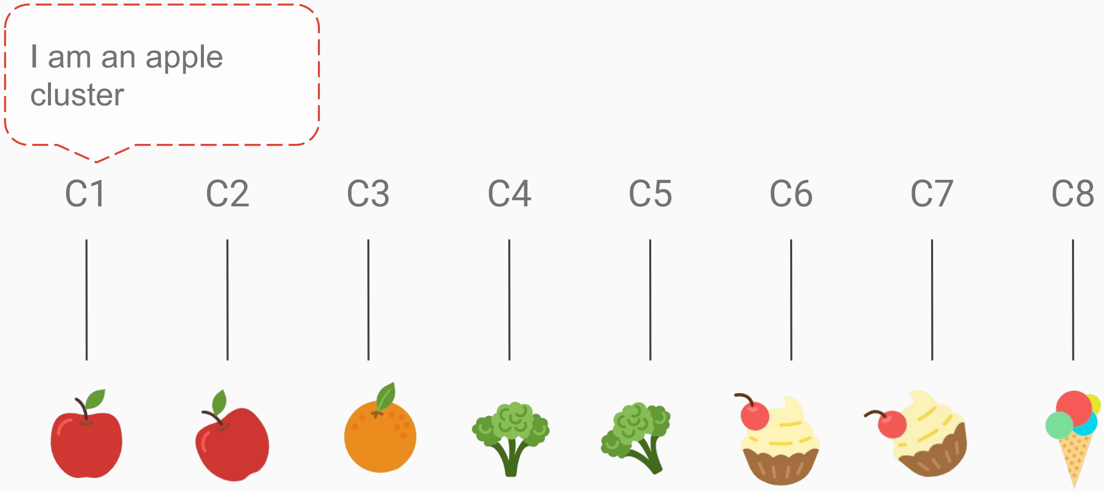
Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that are more dissimilar should belong to the same cluster*. Also, you can iteratively form new clusters out of old ones or create new a division of a previous cluster.



Types of algorithms

Connectivity-based

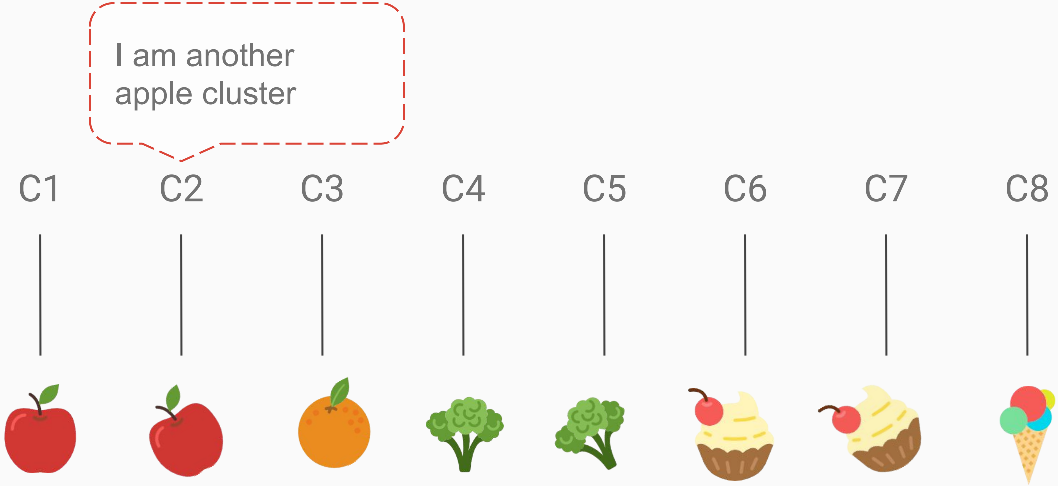
Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that are more dissimilar should belong to the same cluster*. Also, you can iteratively form new clusters out of old ones or create new a division of a previous cluster.



Types of algorithms

Connectivity-based

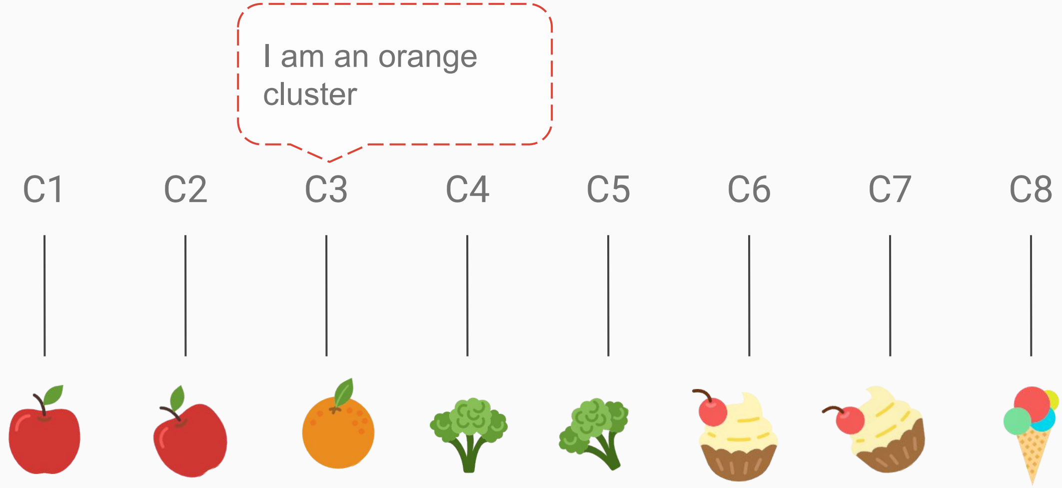
Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that are more dissimilar should belong to the same cluster*. Also, you can iteratively form new clusters out of old ones or create new a division of a previous cluster.



Types of algorithms

Connectivity-based

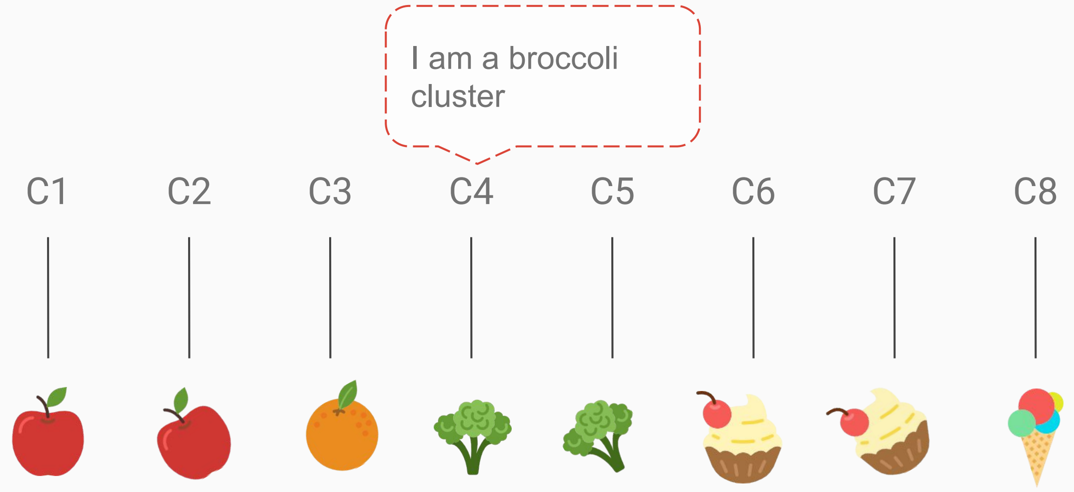
Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that are more dissimilar should belong to the same cluster*. Also, you can iteratively form new clusters out of old ones or create new a division of a previous cluster.



Types of algorithms

Connectivity-based

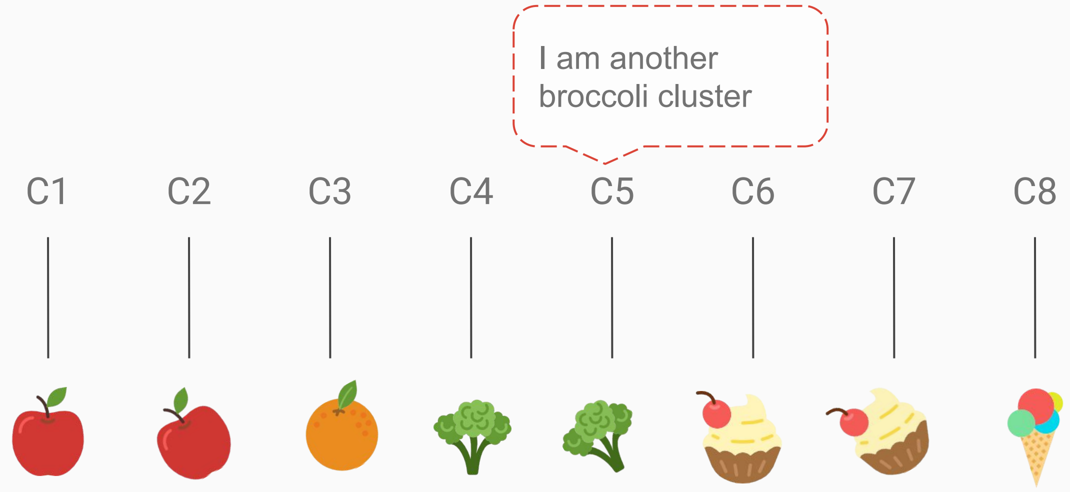
Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that are more dissimilar should belong to the same cluster*. Also, you can iteratively form new clusters out of old ones or create new a division of a previous cluster.



Types of algorithms

Connectivity-based

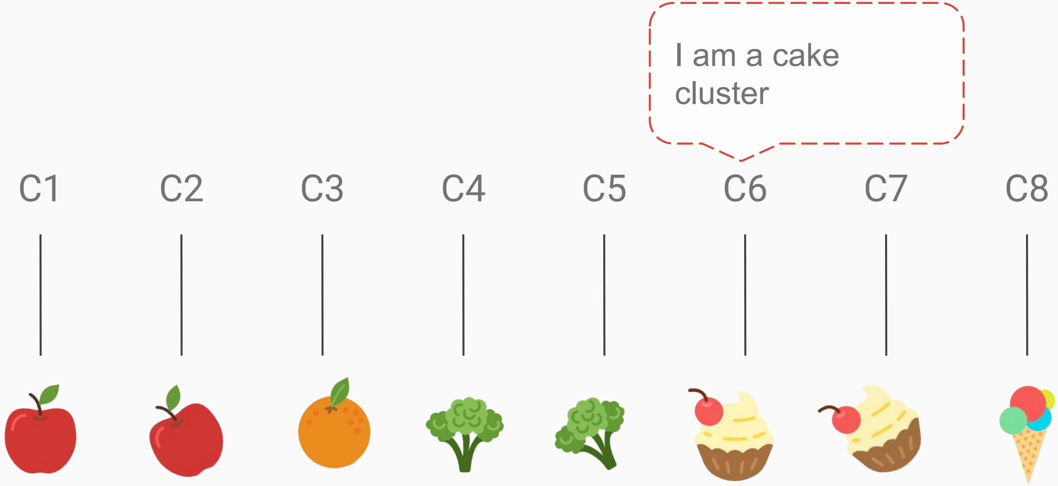
Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that are more dissimilar should belong to the same cluster*. Also, you can iteratively form new clusters out of old ones or create new a division of a previous cluster.



Types of algorithms

Connectivity-based

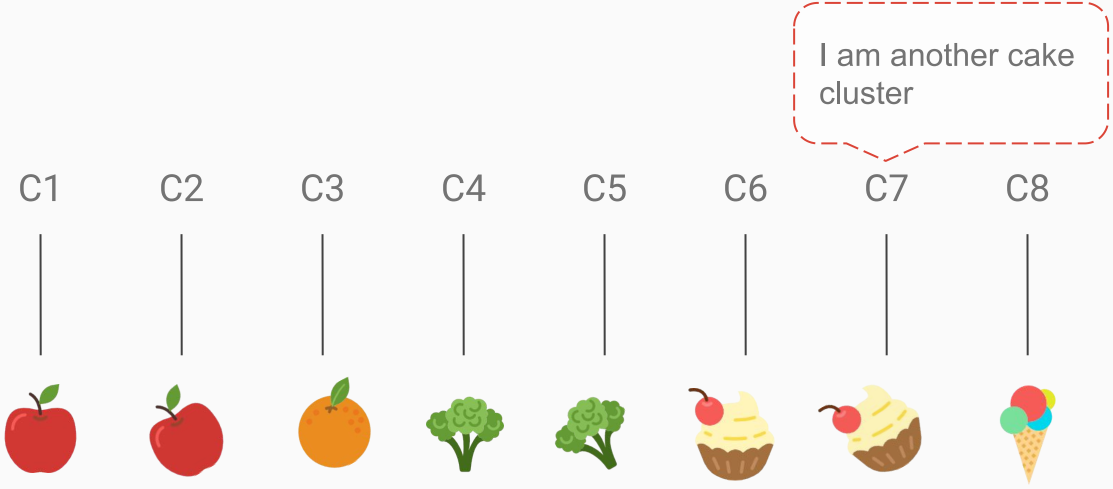
Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that are more dissimilar should belong to the same cluster*. Also, you can iteratively form new clusters out of old ones or create new a division of a previous cluster.



Types of algorithms

Connectivity-based

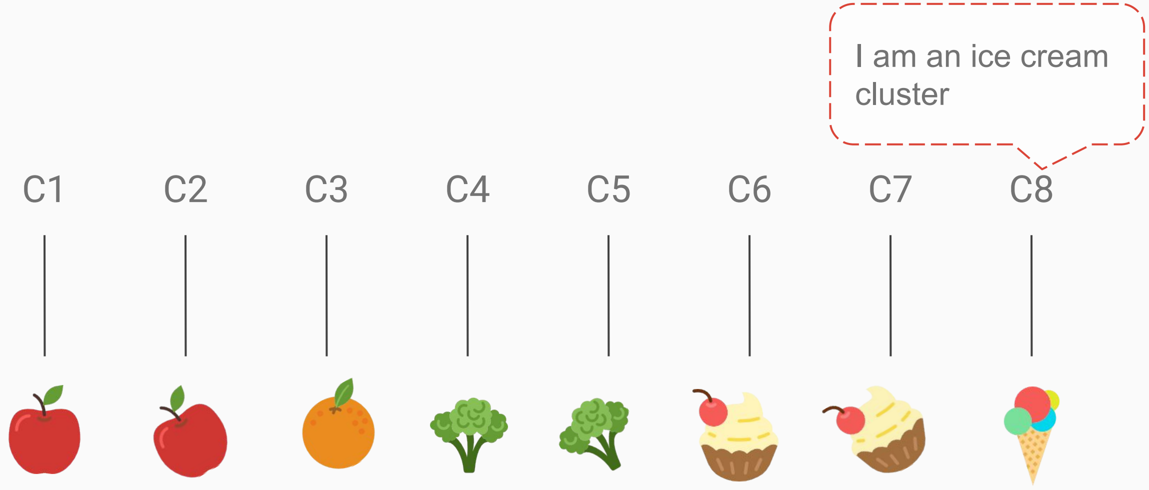
Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that are more dissimilar should belong to the same cluster*. Also, you can iteratively form new clusters out of old ones or create new a division of a previous cluster.



Types of algorithms

Connectivity-based

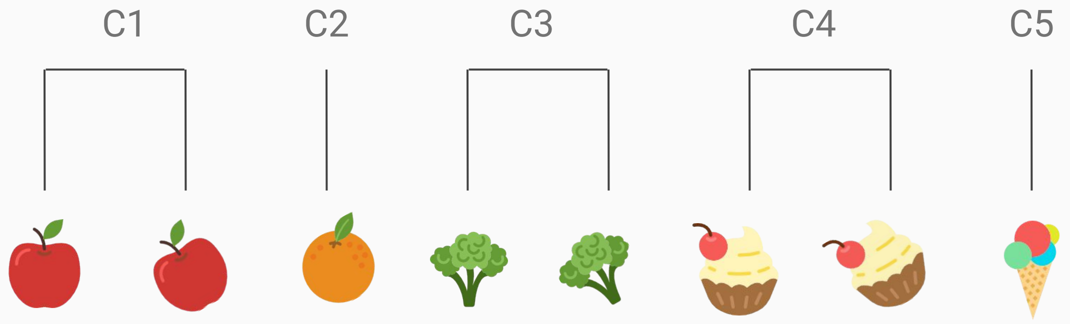
Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that are more dissimilar should belong to the same cluster*. Also, you can iteratively form new clusters out of old ones or create new a division of a previous cluster.



Types of algorithms

Connectivity-based

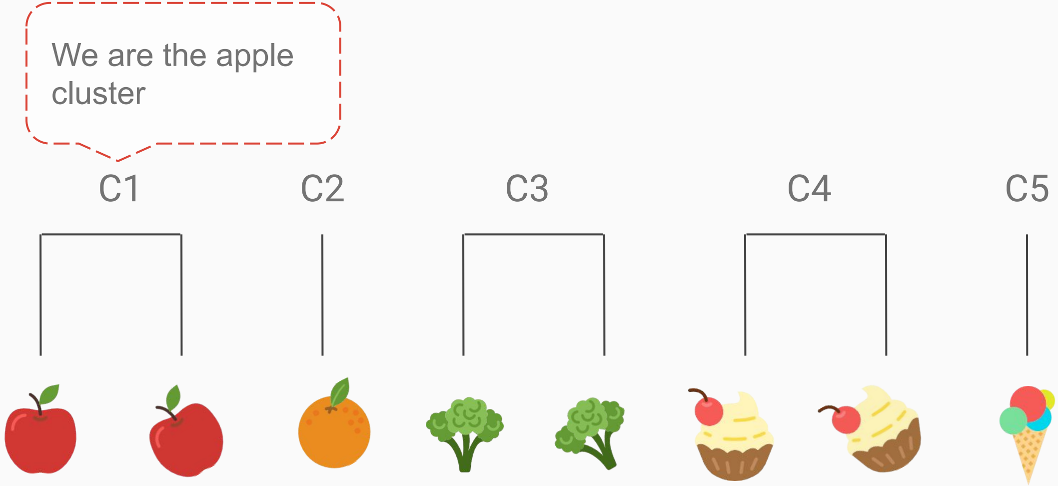
Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that are more dissimilar should belong to the same cluster*. Also, you can iteratively form new clusters out of old ones or create new a division of a previous cluster.



Types of algorithms

Connectivity-based

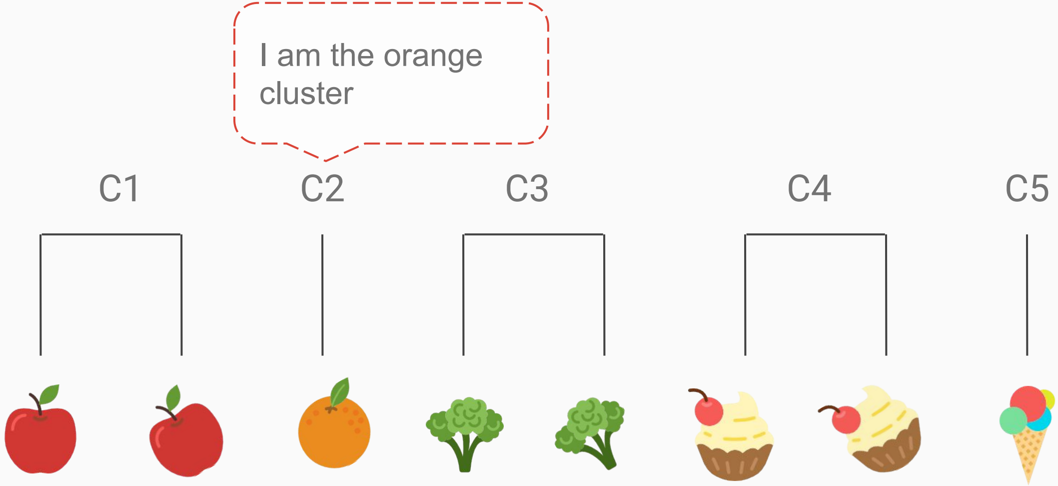
Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that are more dissimilar should belong to the same cluster*. Also, you can iteratively form new clusters out of old ones or create new a division of a previous cluster.



Types of algorithms

Connectivity-based

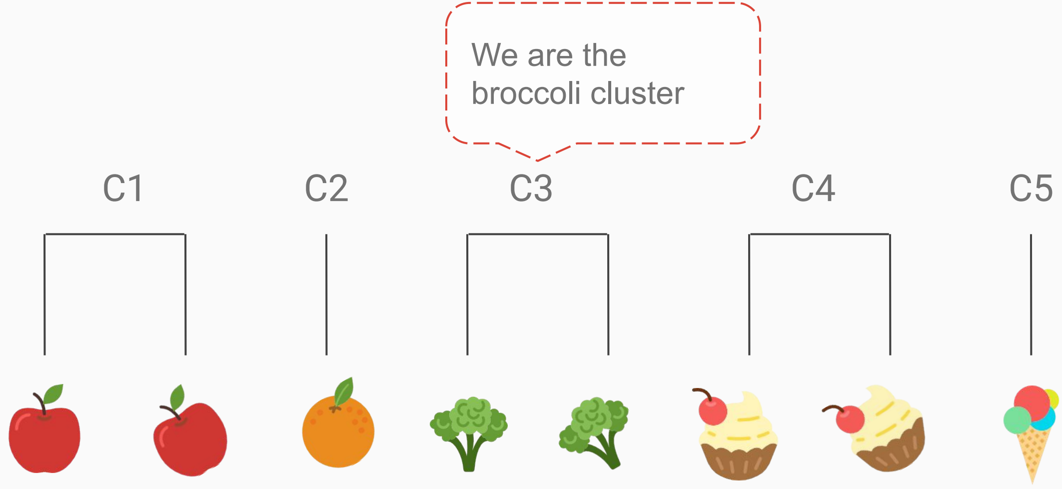
Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that are more dissimilar should belong to the same cluster*. Also, you can iteratively form new clusters out of old ones or create new a division of a previous cluster.



Types of algorithms

Connectivity-based

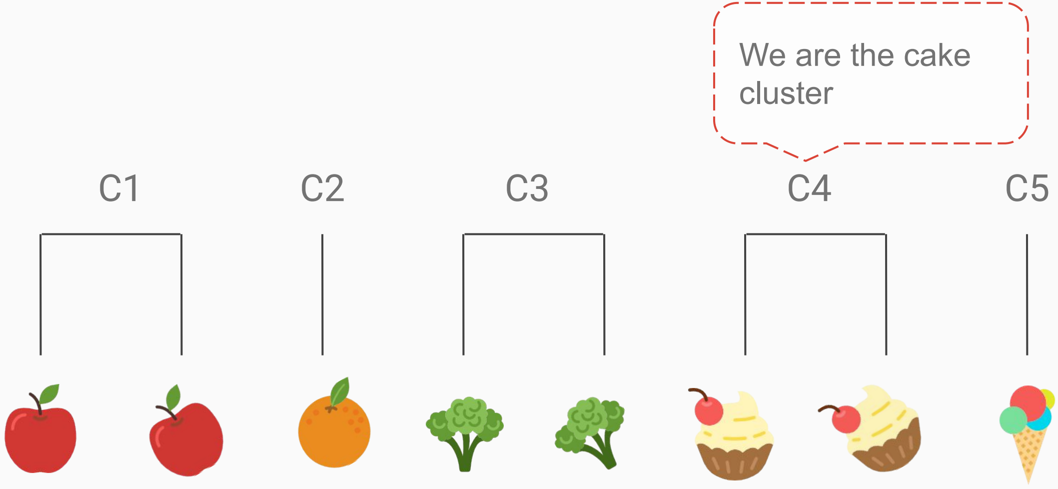
Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that are more dissimilar should belong to the same cluster*. Also, you can iteratively form new clusters out of old ones or create new a division of a previous cluster.



Types of algorithms

Connectivity-based

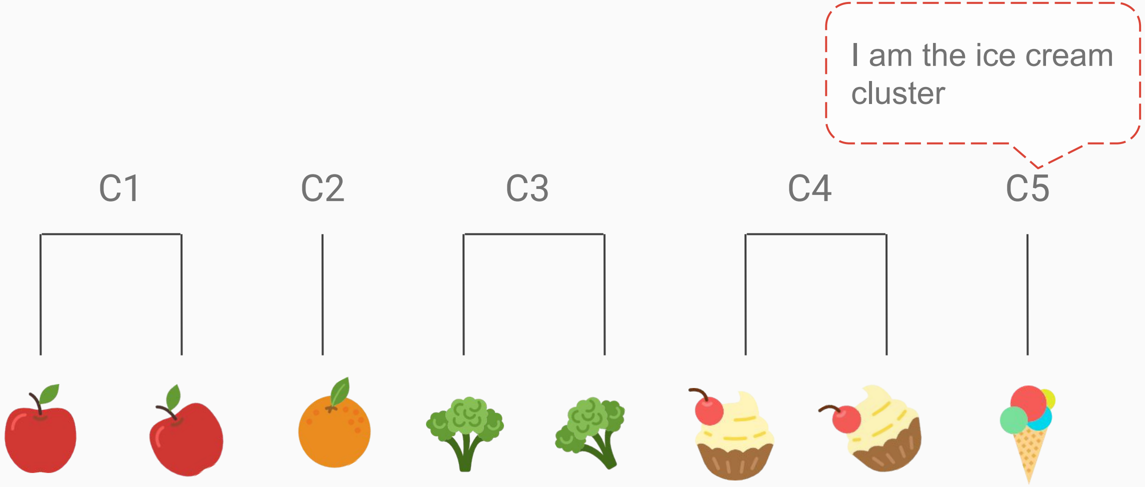
Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that are more dissimilar should belong to the same cluster*. Also, you can iteratively form new clusters out of old ones or create new a division of a previous cluster.



Types of algorithms

Connectivity-based

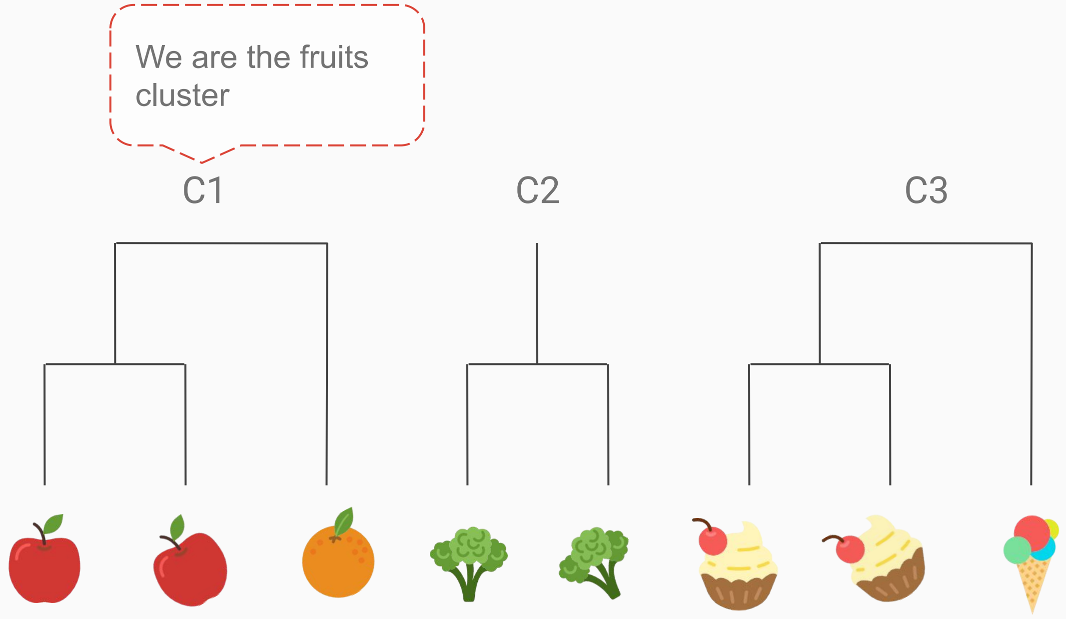
Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that are more dissimilar should belong to the same cluster*. Also, you can iteratively form new clusters out of old ones or create new a division of a previous cluster.



Types of algorithms

Connectivity-based

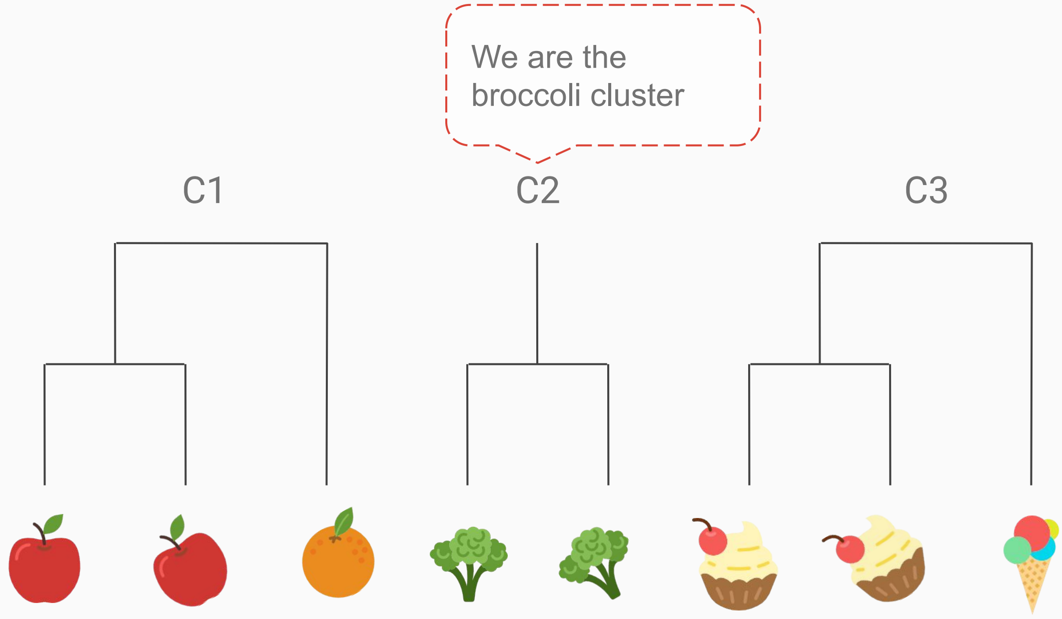
Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that are more dissimilar should belong to the same cluster*. Also, you can iteratively form new clusters out of old ones or create new a division of a previous cluster.



Types of algorithms

Connectivity-based

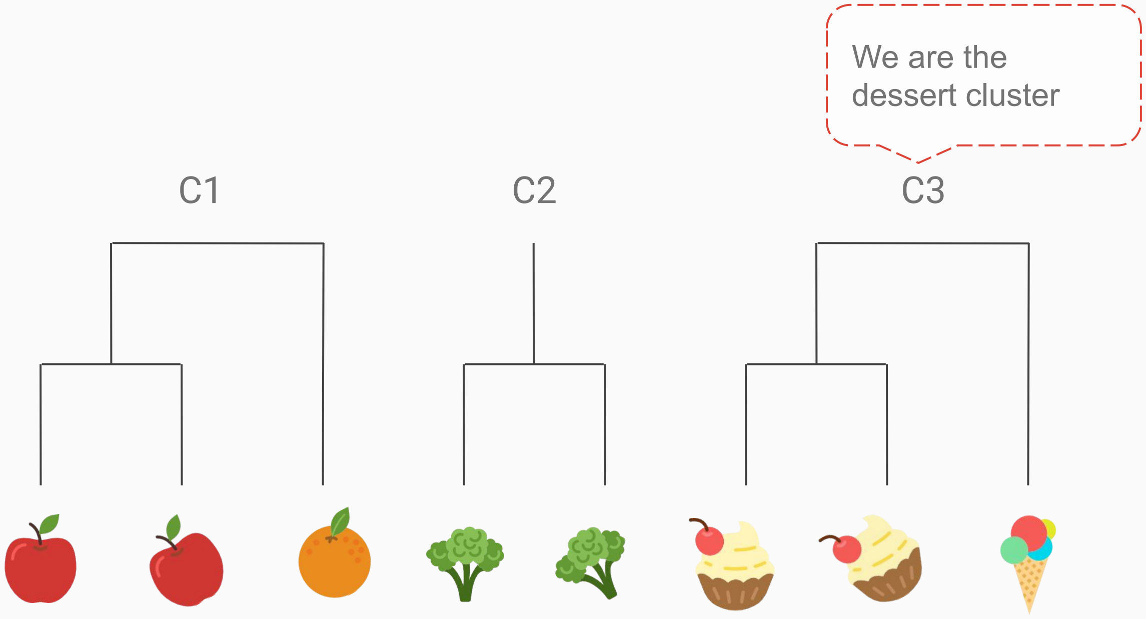
Centroid-based

Distribution-based

Density-based

...

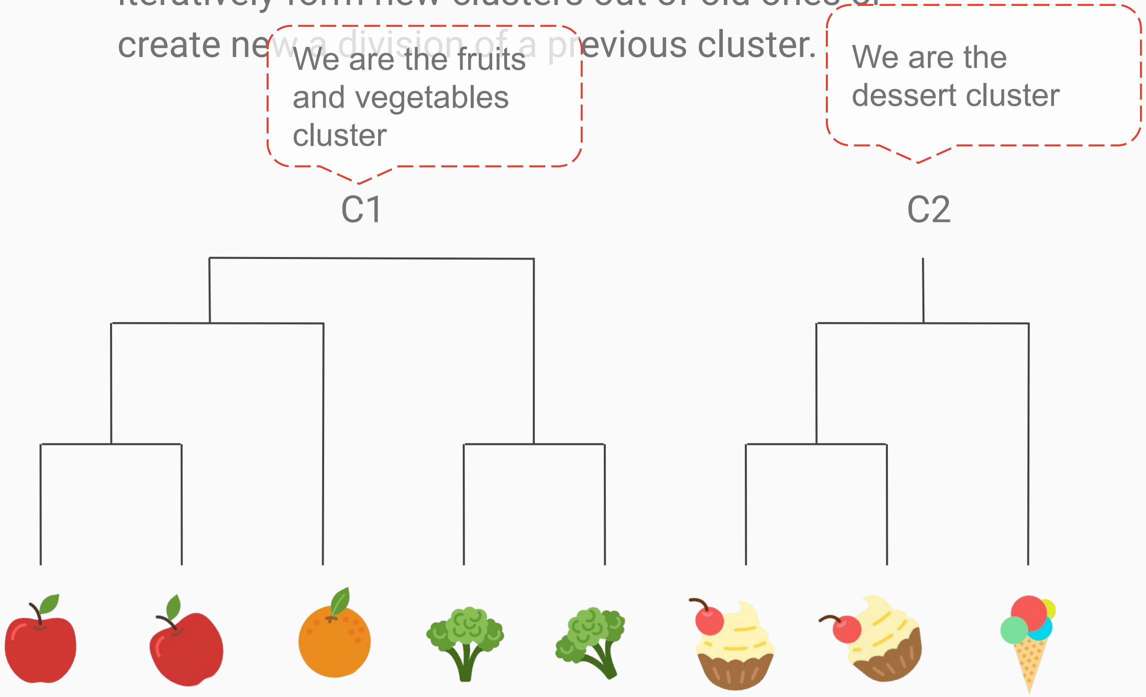
Idea -> *objects that are more dissimilar should belong to the same cluster*. Also, you can iteratively form new clusters out of old ones or create new a division of a previous cluster.



Types of algorithms

- **Connectivity-based**
- Centroid-based
- Distribution-based
- Density-based
- ...

Idea -> *objects that are more dissimilar should belong to the same cluster*. Also, you can iteratively form new clusters out of old ones or create new a division of a previous cluster.



Types of algorithms

Connectivity-based

Centroid-based

Distribution-based

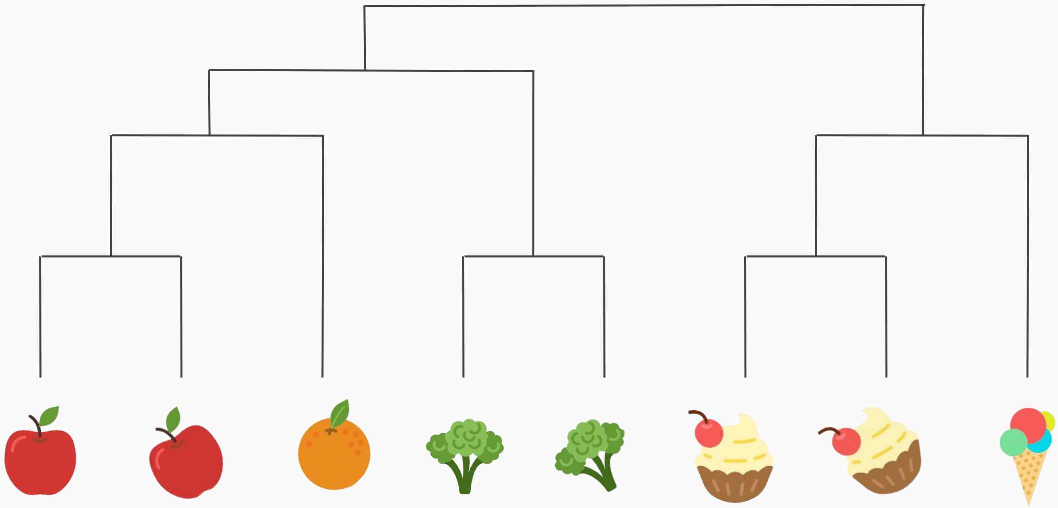
Density-based

...

Idea -> *objects that are more dissimilar should belong to the same cluster.* Also, you can iteratively form new clusters out of old ones or create new a division of a previous cluster.

We are the food cluster

C1



Types of algorithms

Connectivity-based

Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that are more dissimilar should belong to the same cluster*. Also, you can iteratively form new clusters out of old ones or create new a division of a previous cluster.

Advantage -> They create a dendrogram, which is a *useful representation of how objects can be grouped to form clusters.*

Disadvantage -> They are computationally expensive, since they compare objects and then clusters. *Use them if you have a small dataset.*

Popular example -> Agglomerative nesting

Types of algorithms

Connectivity-based

Centroid-based

Distribution-based

Density-based

...

TABLE 12.1. *Algorithm for agglomerative hierarchical clustering.*

1. Input: Items $\mathcal{L} = \{\mathbf{x}_i, i = 1, 2, \ldots, n\}$, n = initial number of clusters, each cluster of which contains one item.
2. Compute $\mathbf{D} = (d_{ij})$, the $(n \times n)$-matrix of dissimilarities between the n clusters, where $d_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$, $i, j = 1, 2, \ldots, n$.
3. Find the smallest dissimilarity, say, d_{IJ}, in $\mathbf{D} = \mathbf{D}^{(1)}$. Merge clusters I and J to form a new cluster IJ.
4. Compute dissimilarities, $d_{IJ,K}$, between the new cluster IJ and all other clusters $K \neq IJ$. These dissimilarities depend upon which linkage method is used. For all clusters $K \neq I, J$, we have the following linkage options:

   Single linkage: $d_{IJ,K} = \min\{d_{I,K}, d_{J,K}\}$.

   Complete linkage: $d_{IJ,K} = \max\{d_{I,K}, d_{J,K}\}$.

   Average linkage: $d_{IJ,K} = \sum_{i \in IJ} \sum_{k \in K} d_{ik}/(N_{IJ}N_K)$,

   where N_{IJ} and N_K are the numbers of items in clusters IJ and K, respectively.
5. Form a new $((n-1) \times (n-1))$-matrix, $\mathbf{D}^{(2)}$, by deleting rows and columns I and J and adding a new row and column IJ with dissimilarities computed from step 4.
6. Repeat steps 3, 4, and 5 a total of $n-1$ times. At the ith step, $\mathbf{D}^{(i)}$ is a symmetric $((n-i+1) \times (n-i+1))$-matrix, $i = 1, 2, \ldots, n$. At the last step $(i = n)$, $\mathbf{D}^{(n)} = 0$, and all items are merged together into a single cluster.
7. Output: List of which clusters are merged at each step, the value (or *height*) of the dissimilarity of each merge, and a dendrogram to summarize the clustering procedure.

Taken from [Modern Multivariate Statistical Techniques]

Types of algorithms

Connectivity-based

Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that belong to a cluster are located nearby the centroid of their cluster and far from the centroid of other clusters.*

Types of algorithms

Connectivity-based

Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that belong to a cluster are located nearby the centroid of their cluster and far from the centroid of other clusters.*

Types of algorithms

Connectivity-based

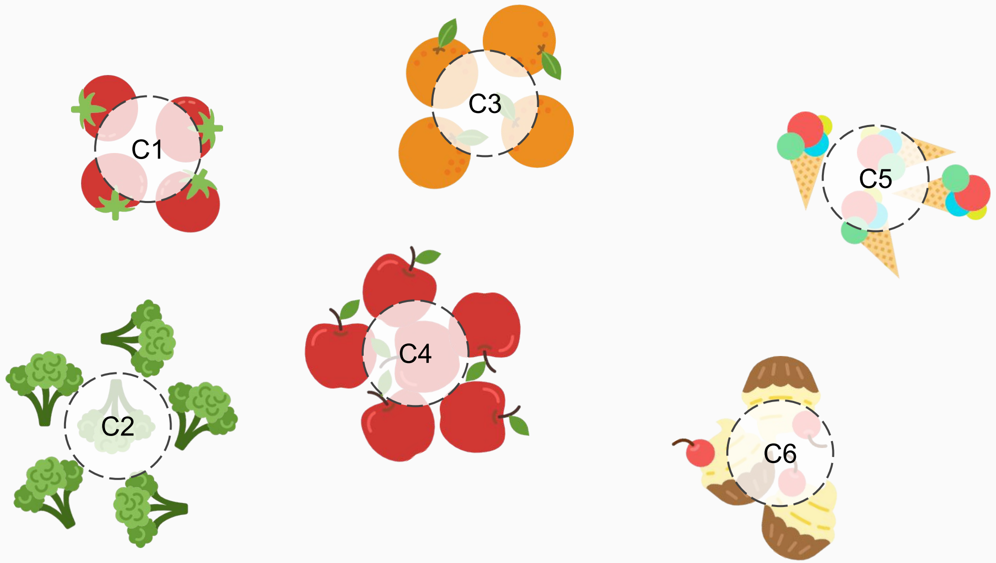
Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that belong to a cluster are located nearby the centroid of their cluster and far from the centroid of other clusters.*



Types of algorithms

Connectivity-based

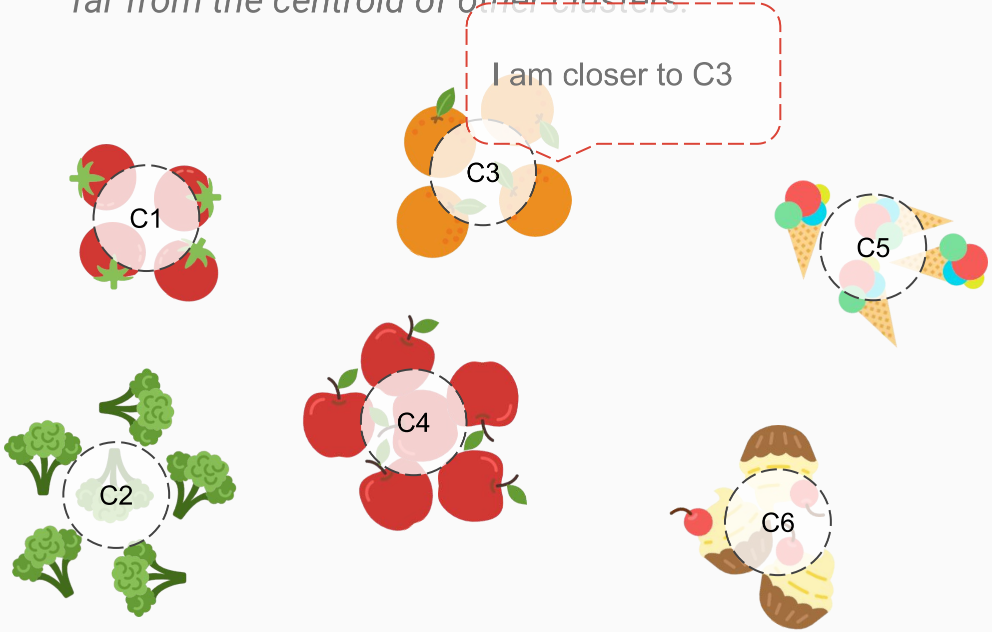
Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that belong to a cluster are located nearby the centroid of their cluster and far from the centroid of other clusters.*



Types of algorithms

Connectivity-based

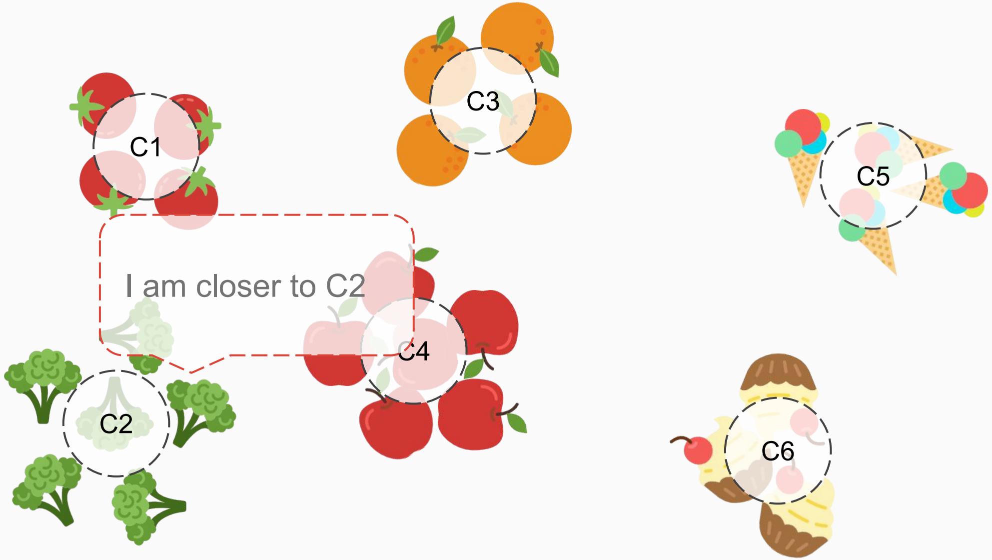
Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that belong to a cluster are located nearby the centroid of their cluster and far from the centroid of other clusters.*



Types of algorithms

Connectivity-based

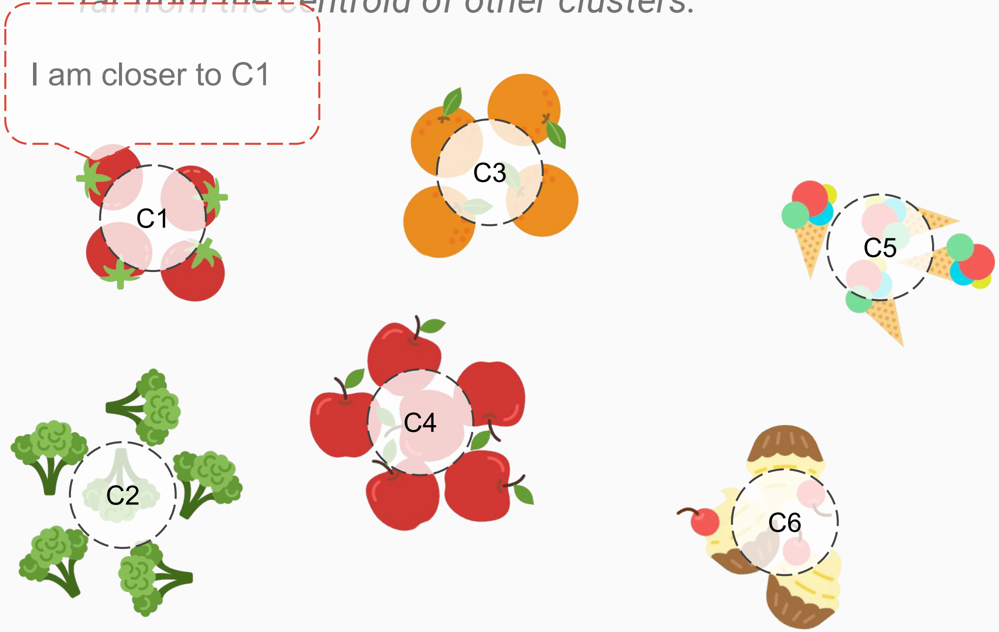
Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that belong to a cluster are located nearby the centroid of their cluster and far from the centroid of other clusters.*



Types of algorithms

Connectivity-based

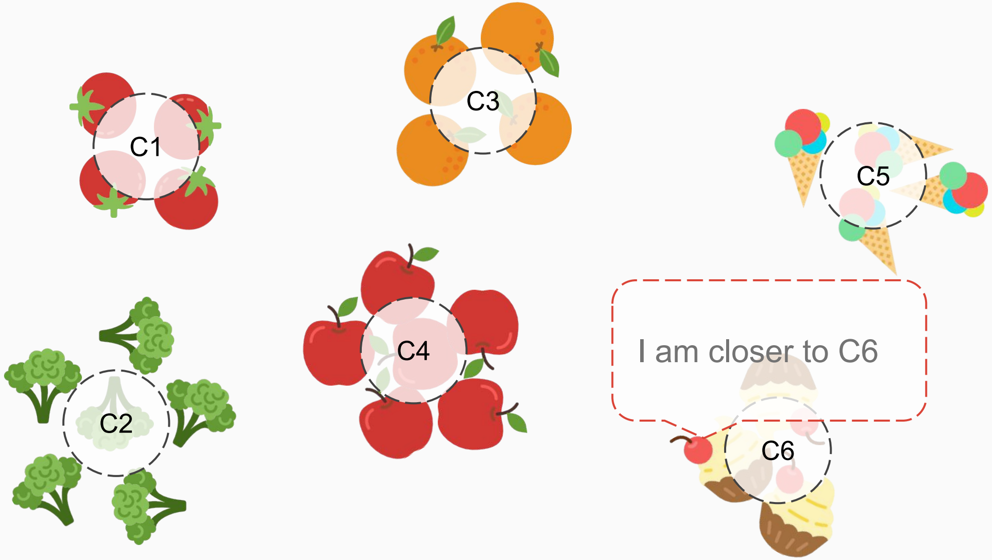
Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that belong to a cluster are located nearby the centroid of their cluster and far from the centroid of other clusters.*



Types of algorithms

Connectivity-based

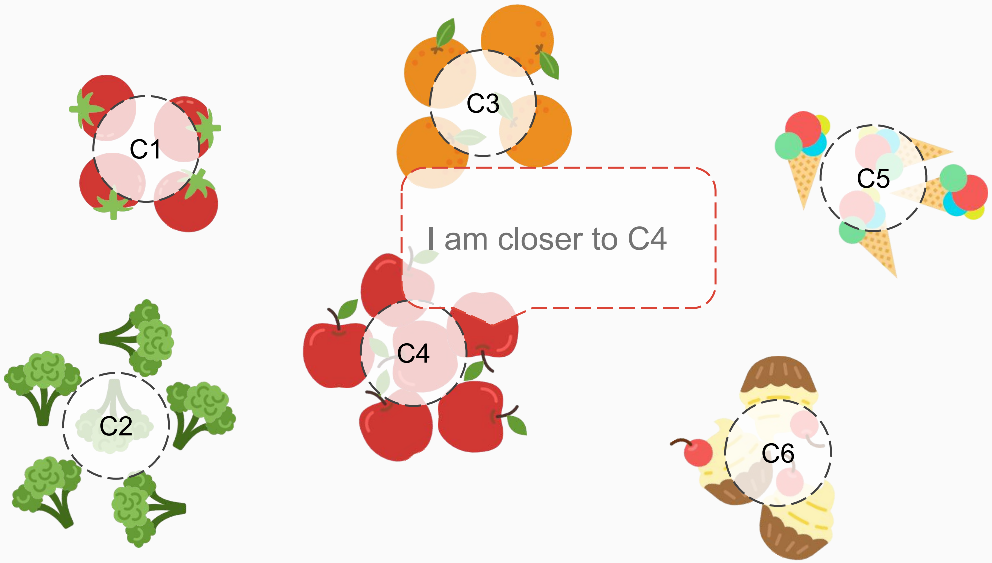
Centroid-based

Distribution-based

Density-based

...

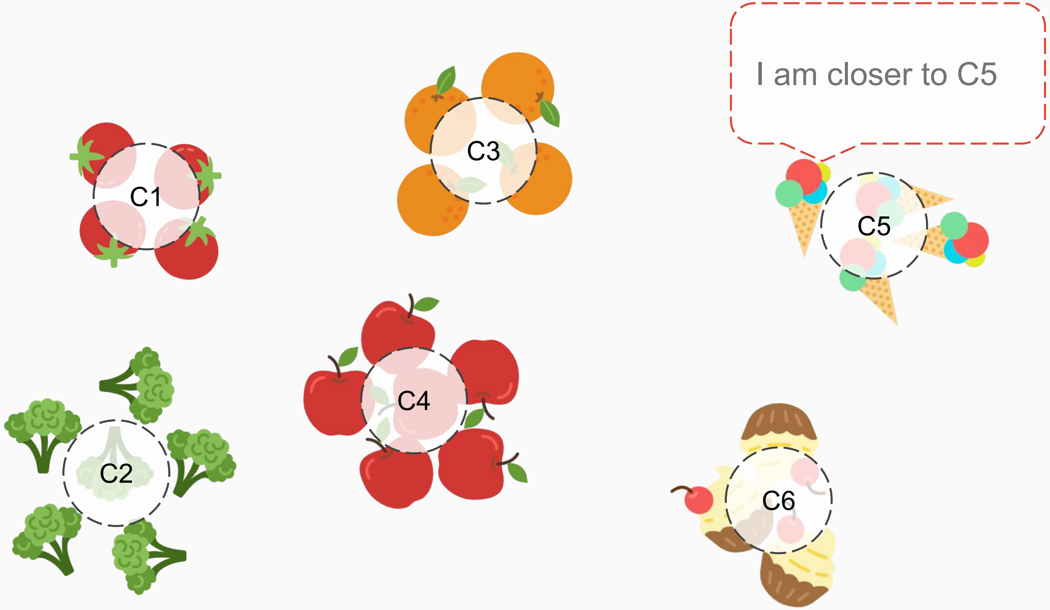
Idea -> *objects that belong to a cluster are located nearby the centroid of their cluster and far from the centroid of other clusters.*



Types of algorithms

- Connectivity-based
- **Centroid-based**
- Distribution-based
- Density-based
- ...

Idea -> *objects that belong to a cluster are located nearby the centroid of their cluster and far from the centroid of other clusters.*



Types of algorithms

Connectivity-based

Centroid-based

Distribution-based

Density-based

...

Idea -> *objects that belong to a cluster are located nearby the centroid of their cluster and far from the centroid of other clusters.*

Advantage -> Computationally efficient (for local optima).

Disadvantage -> *assumes a shape* for the clusters.

Popular example -> K-means

Types of algorithms

Connectivity-based

Centroid-based

Distribution-based

Density-based

...

TABLE 12.2. *Algorithm for K-means clustering.*

1. Input: Items $\mathcal{L} = \{\mathbf{x}_i, i = 1, 2, \ldots, n\}$, K = number of clusters.
2. Do one of the following:
   - Form an initial random assignment of the items into K clusters and, for cluster k, compute its current centroid, $\bar{\mathbf{x}}_k$, $k = 1, 2, \ldots, K$.
   - Pre-specify K cluster centroids, $\bar{\mathbf{x}}_k$, $k = 1, 2, \ldots, K$.
3. Compute the squared-Euclidean distance of each item to its current cluster centroid:
$$\text{ESS} = \sum_{k=1}^{K} \sum_{c(i)=k} (\mathbf{x}_i - \bar{\mathbf{x}}_k)^\tau (\mathbf{x}_i - \bar{\mathbf{x}}_k),$$
   where $\bar{\mathbf{x}}_k$ is the kth cluster centroid and $c(i)$ is the cluster containing $\mathbf{x}_i$.
4. Reassign each item to its nearest cluster centroid so that ESS is reduced in magnitude. Update the cluster centroids after each reassignment.
5. Repeat steps 3 and 4 until no further reassignment of items takes place.

Taken from [Modern Multivariate Statistical Techniques]

Types of algorithms

Connectivity-based

Centroid-based

Distribution-based

Density-based

...

Idea -> Each object in your dataset comes from a probability distribution. Now imagine your dataset is built out of many of such probability distributions, where *each cluster corresponds to one of those probability distributions.*

Types of algorithms

Connectivity-based

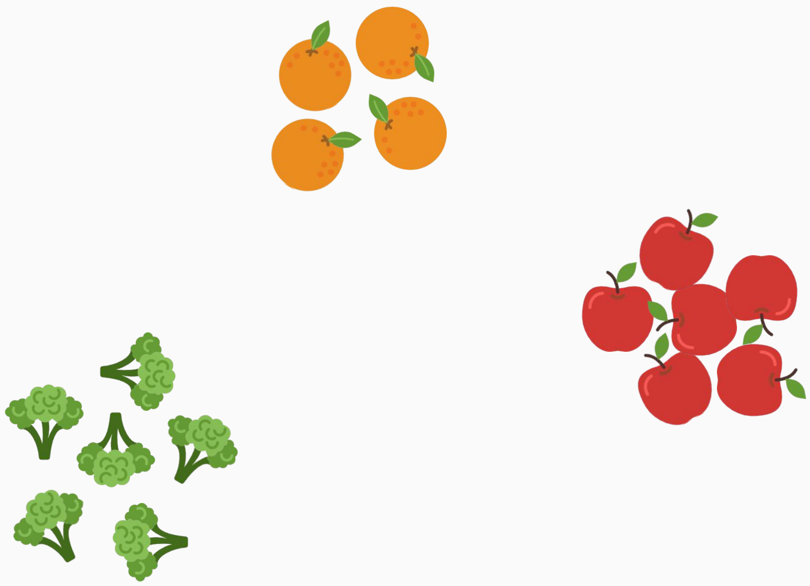
Centroid-based

Distribution-based

Density-based

...

Idea -> Each object in your dataset comes from a probability distribution. Now imagine your dataset is built out of many of such probability distributions, where *each cluster corresponds to one of those probability distributions.*



Types of algorithms

Connectivity-based

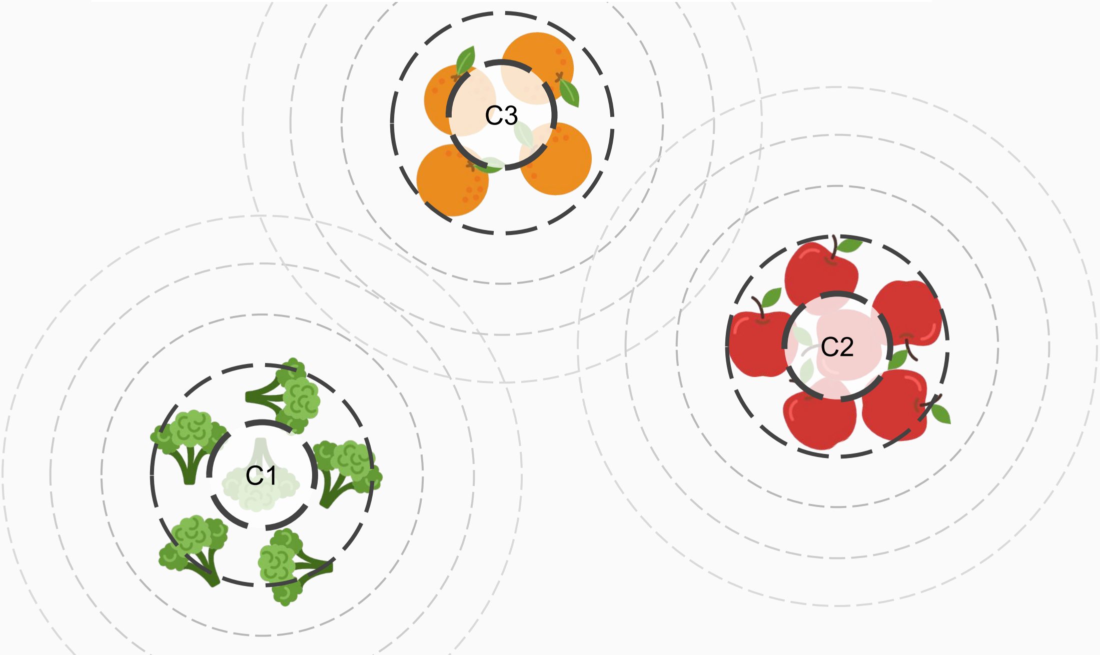
Centroid-based

Distribution-based

Density-based

...

Idea -> Each object in your dataset comes from a probability distribution. Now imagine your dataset is built out of many of such probability distributions, where *each cluster corresponds to one of those probability distributions.*



Types of algorithms

Connectivity-based

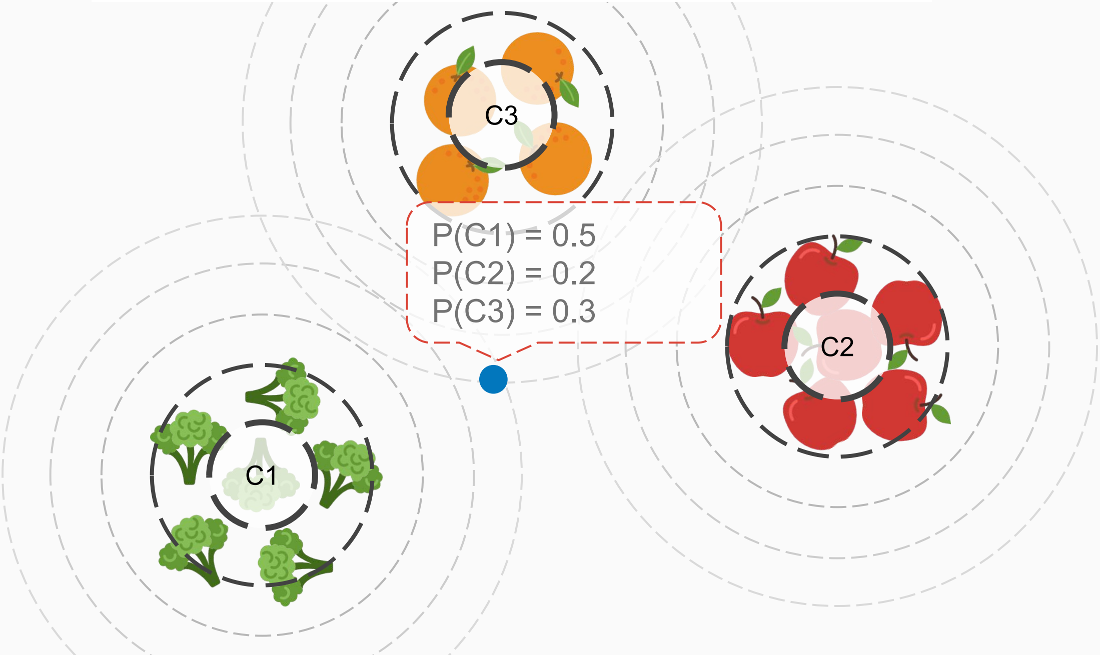
Centroid-based

Distribution-based

Density-based

...

Idea -> Each object in your dataset comes from a probability distribution. Now imagine your dataset is built out of many of such probability distributions, where *each cluster corresponds to one of those probability distributions.*



Types of algorithms

Connectivity-based

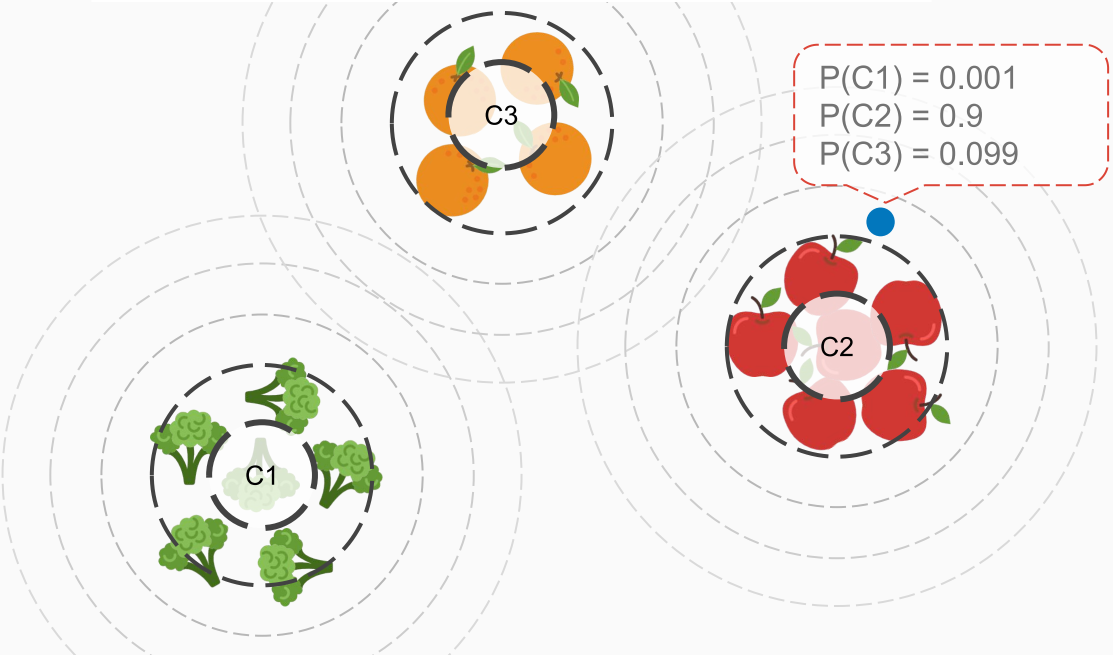
Centroid-based

Distribution-based

Density-based

...

Idea -> Each object in your dataset comes from a probability distribution. Now imagine your dataset is built out of many of such probability distributions, where *each cluster corresponds to one of those probability distributions.*



Types of algorithms

Connectivity-based

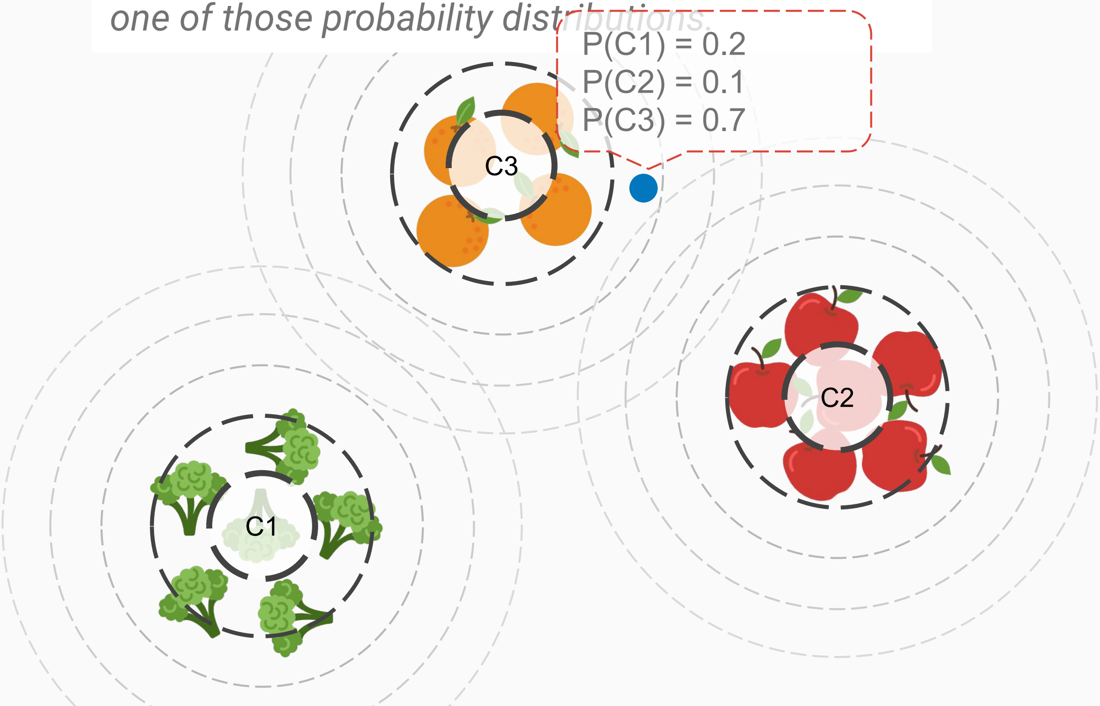
Centroid-based

Distribution-based

Density-based

...

Idea -> Each object in your dataset comes from a probability distribution. Now imagine your dataset is built out of many of such probability distributions, where *each cluster corresponds to one of those probability distributions.*



Types of algorithms

Connectivity-based

Centroid-based

Distribution-based

Density-based

...

Idea -> Each object in your dataset comes from a probability distribution. Now imagine your dataset is built out of many of such probability distributions, where *each cluster corresponds to one of those probability distributions*.

Advantage -> They can give you the probability of belonging to each cluster.

Disadvantage -> Assumes a probability distribution for each cluster.

Popular example -> Gaussian Mixture Models

Types of algorithms

Connectivity-based

Centroid-based

Distribution-based

Density-based

...

Idea -> Clusters are regions in the feature space that have higher density of objects.

Types of algorithms

Connectivity-based

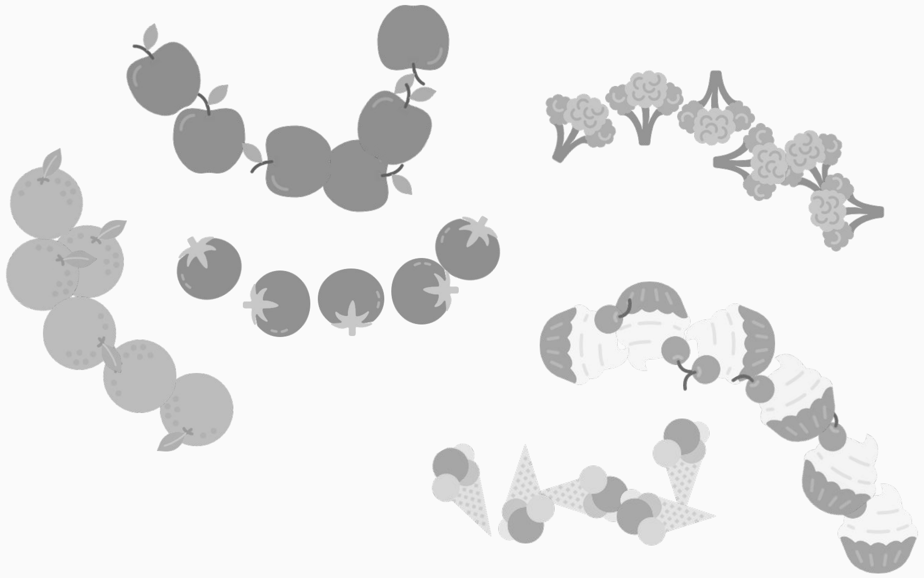
Centroid-based

Distribution-based

Density-based

...

Idea -> Clusters are regions in the feature space that have higher density of objects.

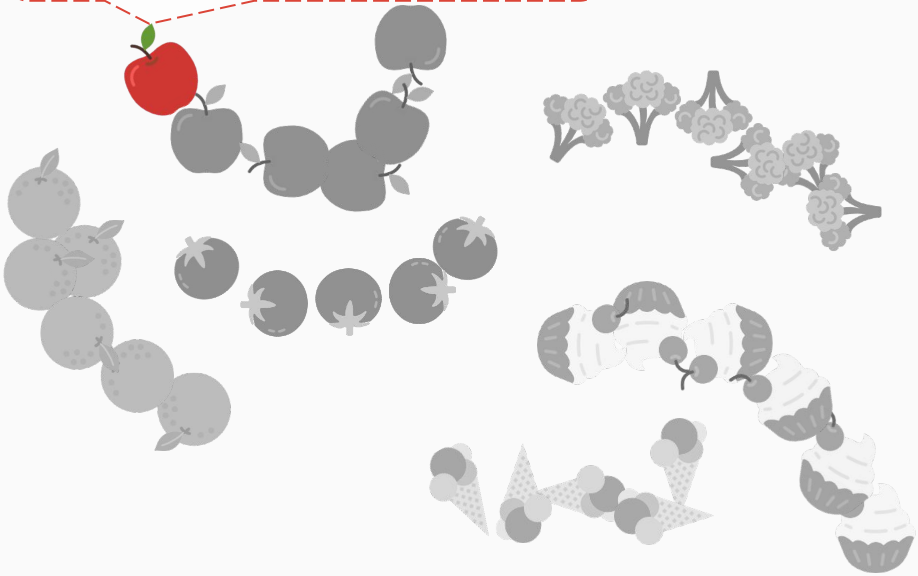


Types of algorithms

- Connectivity-based
- Centroid-based
- Distribution-based
- **Density-based**
- ...

Idea -> Clusters are regions in the feature space that have higher density of objects.

Let’s say I am my own cluster. Now I will bring the closest object to the same cluster



Types of algorithms

Connectivity-based

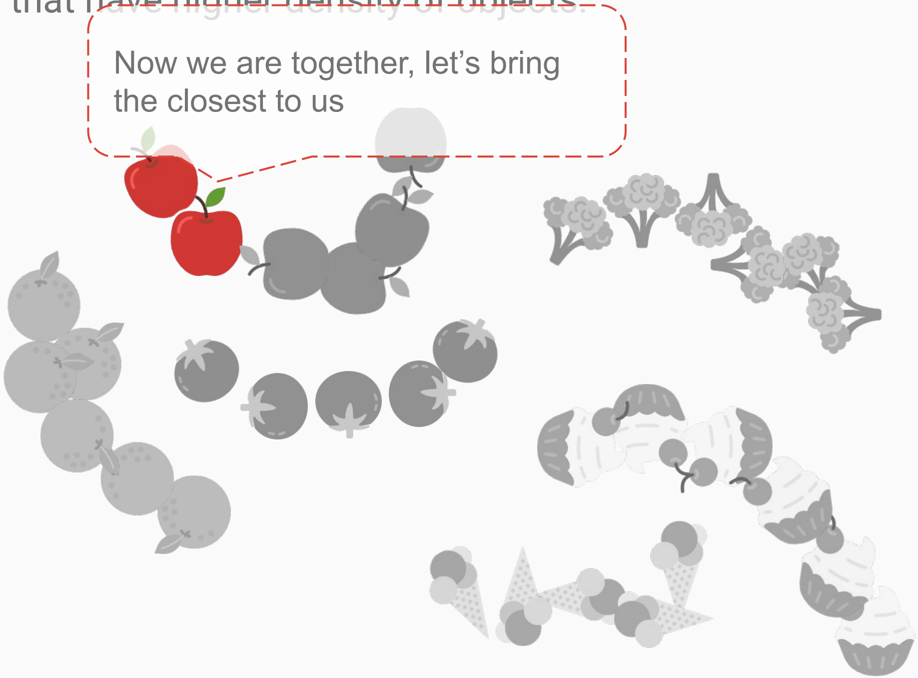
Centroid-based

Distribution-based

Density-based

...

Idea -> Clusters are regions in the feature space that have higher density of objects.



Types of algorithms

Connectivity-based

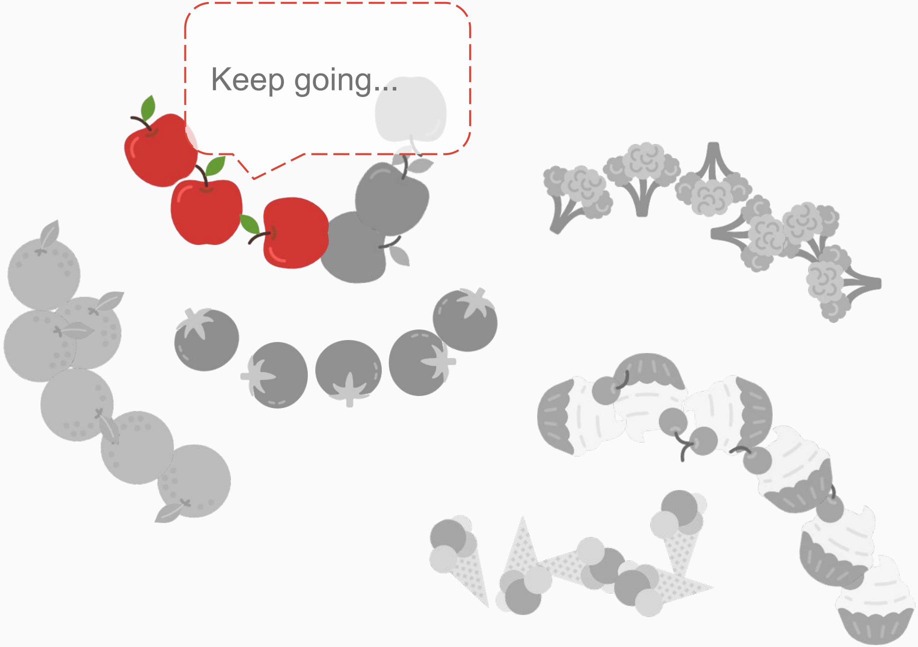
Centroid-based

Distribution-based

Density-based

...

Idea -> Clusters are regions in the feature space that have higher density of objects.



Types of algorithms

Connectivity-based

Centroid-based

Distribution-based

Density-based

...

Idea -> Clusters are regions in the feature space that have higher density of objects.

Types of algorithms

Connectivity-based

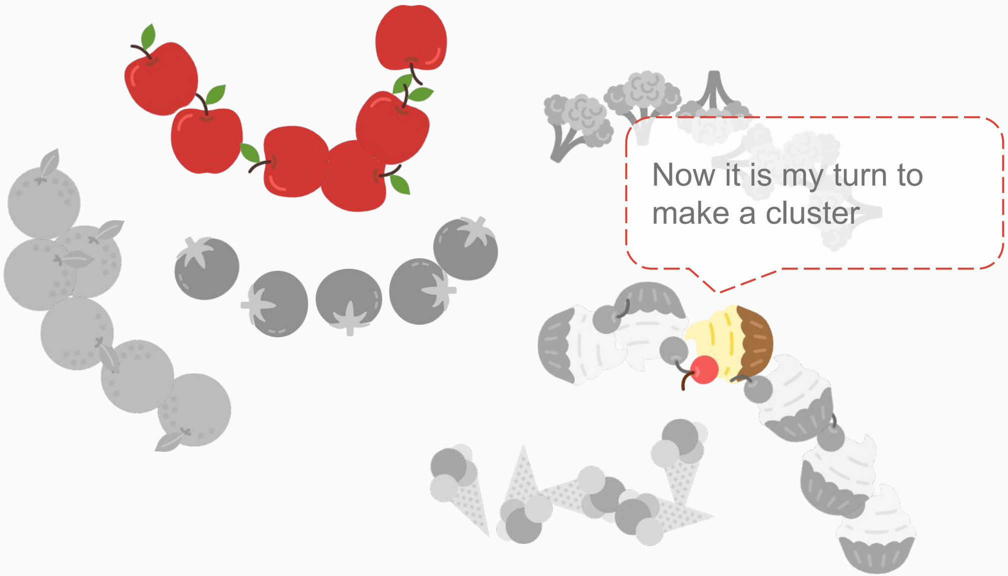
Centroid-based

Distribution-based

Density-based

...

Idea -> Clusters are regions in the feature space that have higher density of objects.



Types of algorithms

Connectivity-based

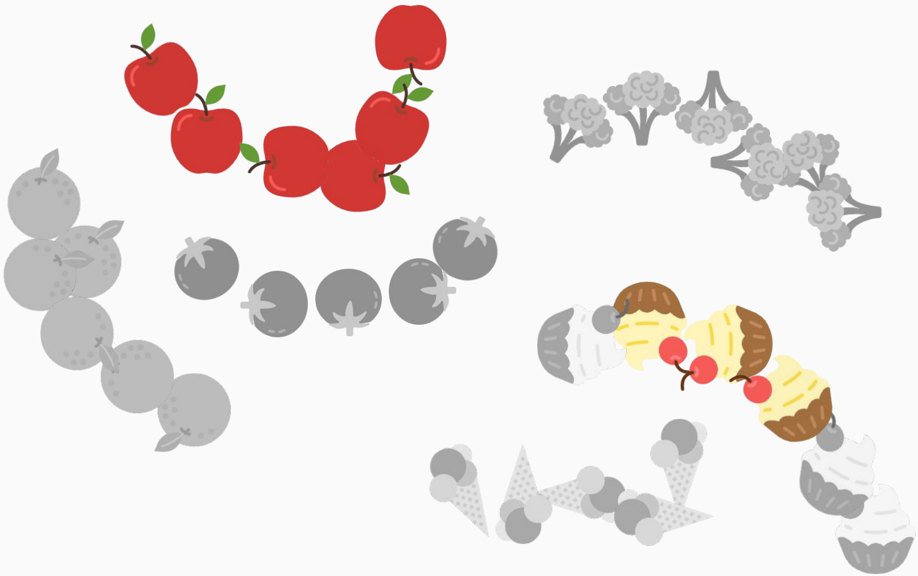
Centroid-based

Distribution-based

Density-based

...

Idea -> Clusters are regions in the feature space that have higher density of objects.



Types of algorithms

Connectivity-based

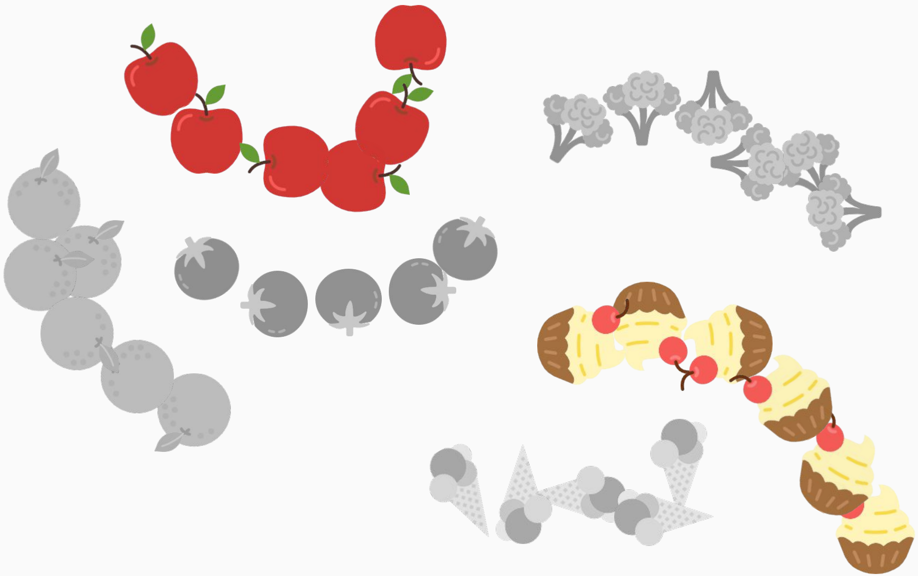
Centroid-based

Distribution-based

Density-based

...

Idea -> Clusters are regions in the feature space that have higher density of objects.



Types of algorithms

Connectivity-based

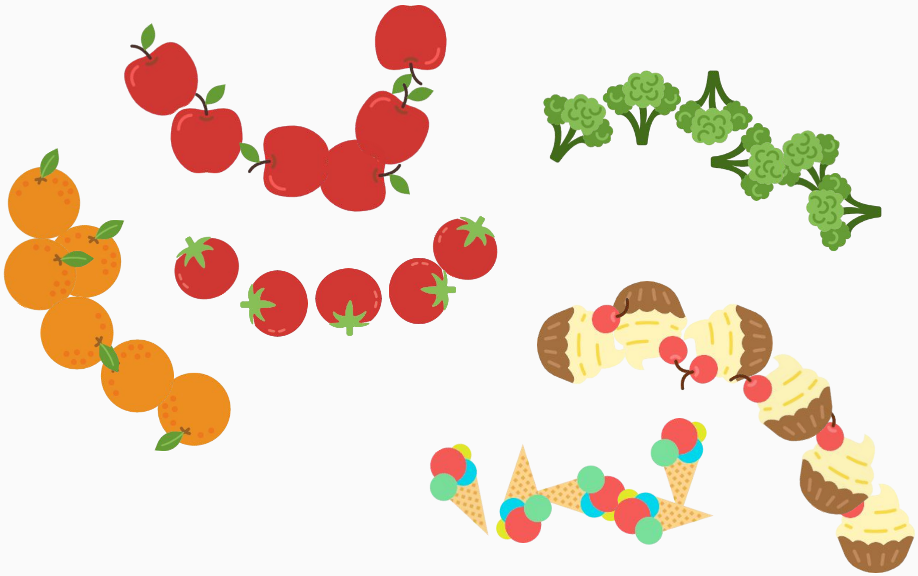
Centroid-based

Distribution-based

Density-based

...

Idea -> Clusters are regions in the feature space that have higher density of objects.



Types of algorithms

Connectivity-based

Centroid-based

Distribution-based

Density-based

...

Idea -> Clusters are regions in the feature space that have higher density of objects.

Advantage -> Clusters can be of any shape.

Disadvantage -> There needs to be a gap or density drop at the border between clusters.

Popular example -> DBSCAN

Types of algorithms

Connectivity-based

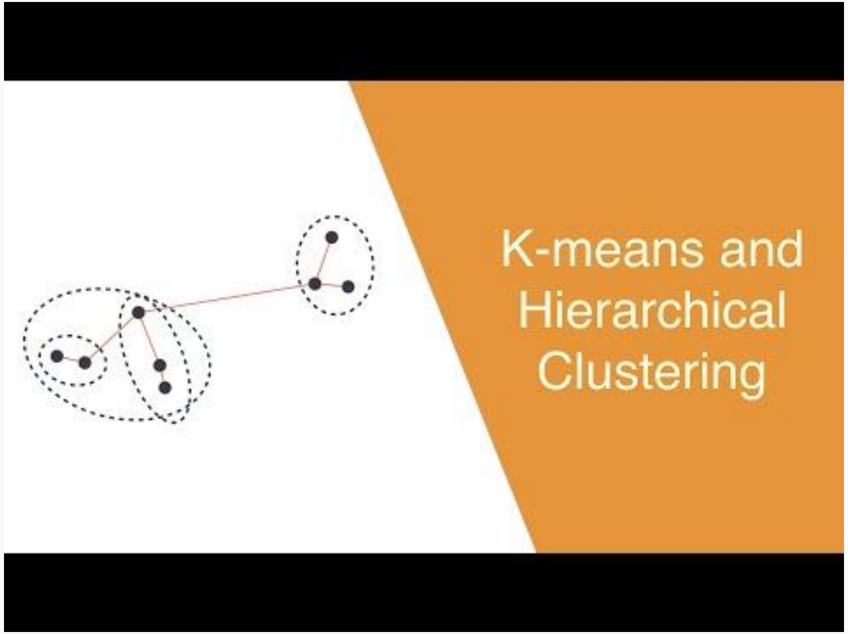
Centroid-based

Distribution-based

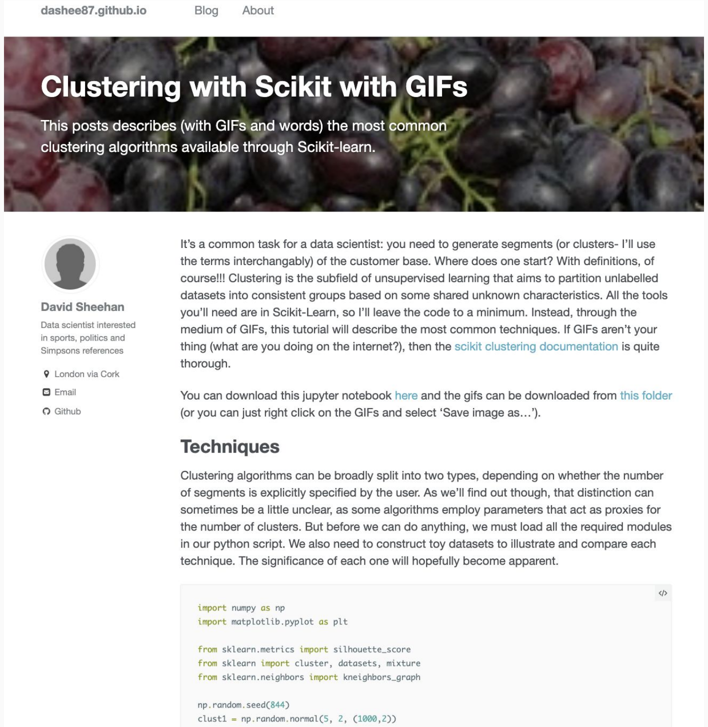
Density-based

...

Human creativity can go amazingly far, keep learning!



This is an amazing explanation of two of the most popular clustering algorithms. I am glad I found it, because it means I do not have to make it. **Please go to** https://www.youtube.com/watch?v=QXOkPvFM6NU



dashee87.github.io Blog About

Clustering with Scikit with GIFs

This posts describes (with GIFs and words) the most common clustering algorithms available through Scikit-learn.

David Sheehan

Data scientist interested in sports, politics and Simpsons references

London via Cork

Email

Github

It's a common task for a data scientist: you need to generate segments (or clusters- I'll use the terms interchangably) of the customer base. Where does one start? With definitions, of course!!! Clustering is the subfield of unsupervised learning that aims to partition unlabelled datasets into consistent groups based on some shared unknown characteristics. All the tools you'll need are in Scikit-Learn, so I'll leave the code to a minimum. Instead, through the medium of GIFs, this tutorial will describe the most common techniques. If GIFs aren't your thing (what are you doing on the internet?), then the scikit clustering documentation is quite thorough.

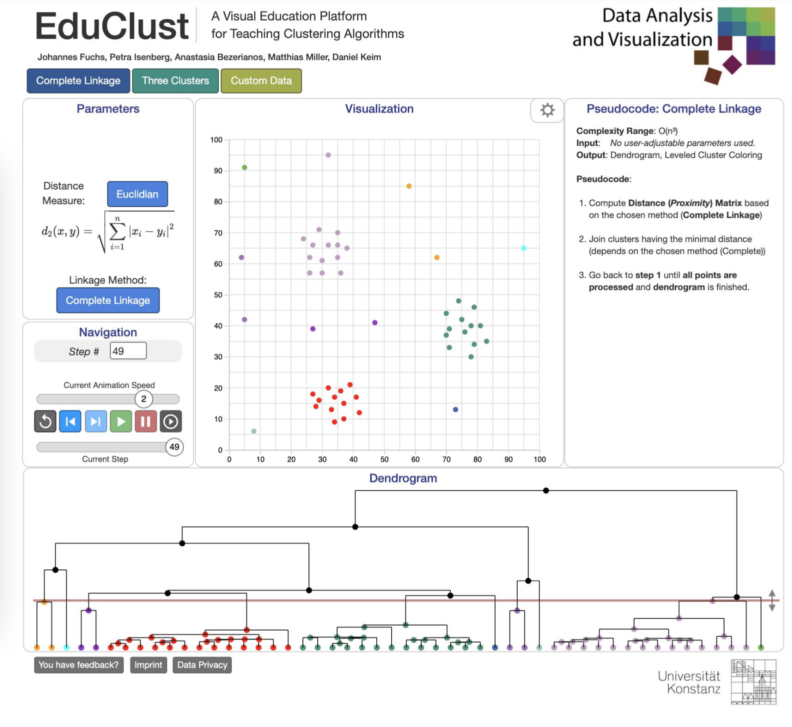
You can download this jupyter notebook here and the gifs can be downloaded from this folder (or you can just right click on the GIFs and select 'Save image as…').

Techniques

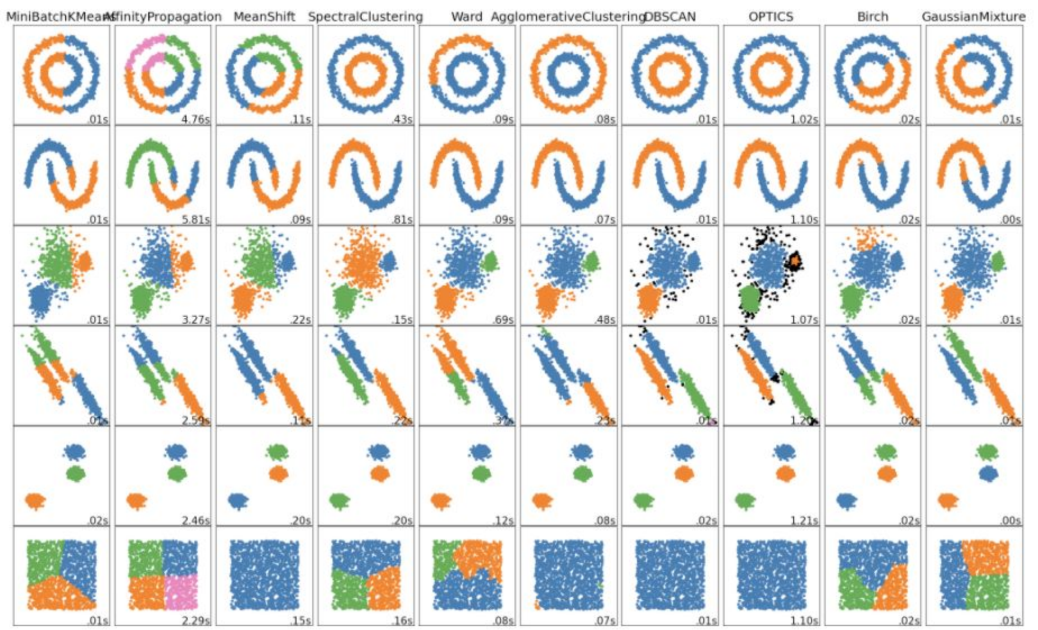
Clustering algorithms can be broadly split into two types, depending on whether the number of segments is explicitly specified by the user. As we'll find out though, that distinction can sometimes be a little unclear, as some algorithms employ parameters that act as proxies for the number of clusters. But before we can do anything, we must load all the required modules in our python script. We also need to construct toy datasets to illustrate and compare each technique. The significance of each one will hopefully become apparent.

```
import numpy as np
import matplotlib.pyplot as plt

from sklearn.metrics import silhouette_score
from sklearn import cluster, datasets, mixture
from sklearn.neighbors import kneighbors_graph

np.random.seed(844)
clust1 = np.random.normal(5, 2, (1000,2))
```

This is another good source. **Please go to**
https://dashee87.github.io/data%20science/general/Clustering-with-Scikit-with-GIFs/



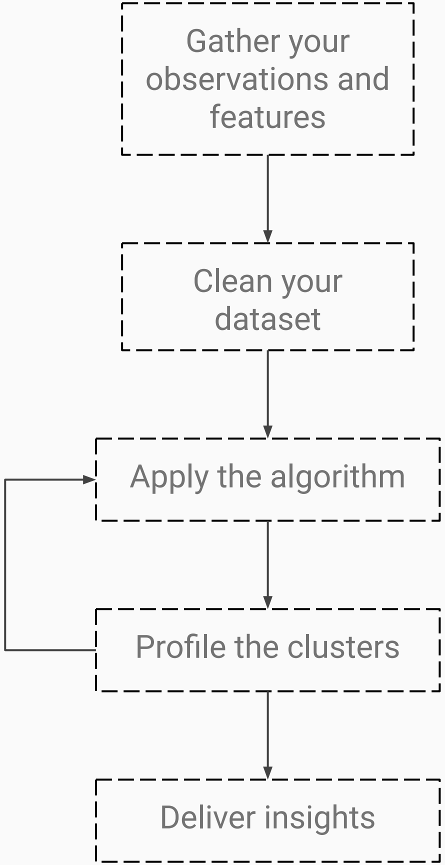
This is possibly the best tool to understand different clustering algorithms. **Please go to https://educlust.dbvis.de/**



No algorithm is perfect, look at the clusters they produce with different synthetic datasets (imagine real life, which is often harder). Taken from: https://scikit-learn.org/stable/modules/clustering.html

Practice

How should I apply it to real life?


Gather your observations and features
Clean your dataset
Apply the algorithm
Profile the clusters
Deliver insights


12

Gather your observations and features

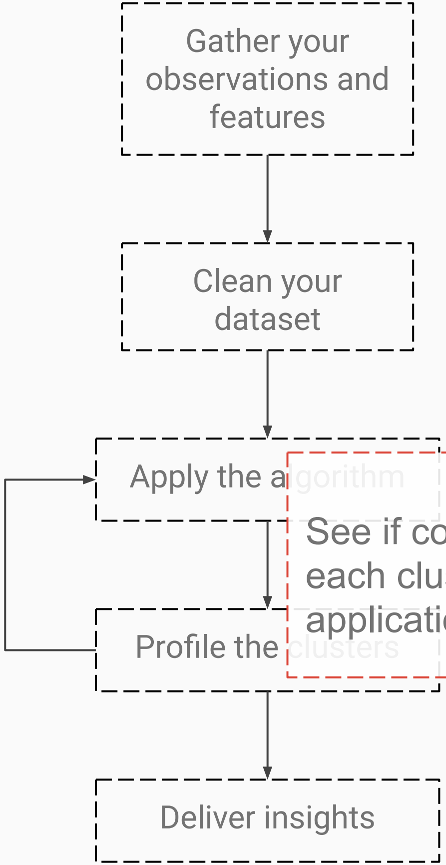
Clean your dataset

Properly treating outliers becomes more critical than in other algorithms

Apply the algorithm

Profile the clusters

Deliver insights



See if common statistics for each cluster *make sense* in your application

Gather your observations and features

Clean your dataset

Apply the algorithm

Profile the clusters

Deliver insights

You can automate this if you have already validated that the clusters generated by an algorithm make sense for your application

How do I know the clusters are right?

Profile the clusters and manually check if they give you useful knowledge about whatever you are analysing.

Use *internal* metrics like Davies-Bouldin index, Dunn index or Silhouette coefficient.

Use *external* metrics, which requires some classification labels.

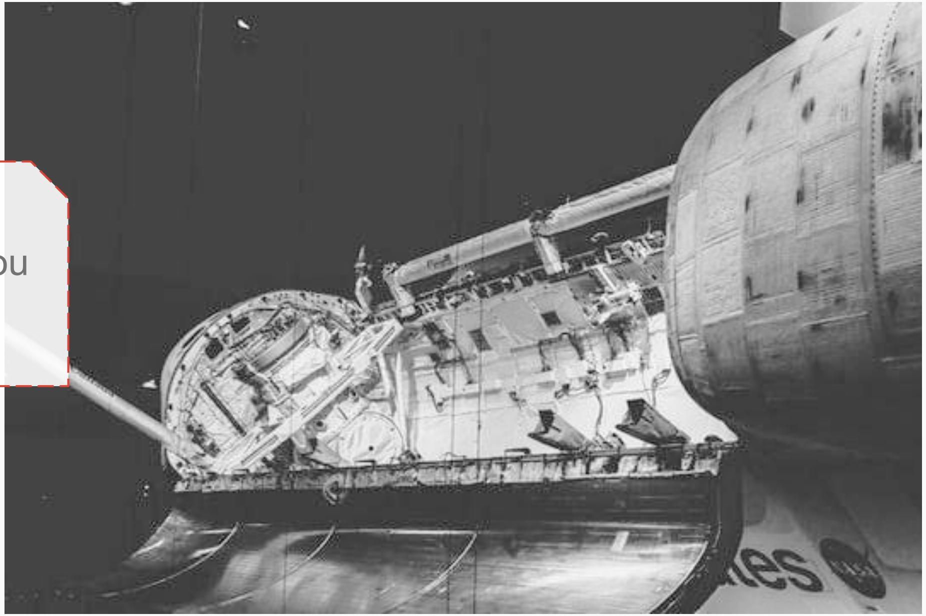
How do I know the clusters are right?

Profile the clusters and manually check if they give you useful knowledge about whatever you are analysing.

Use *internal* metrics like Davies Bouldin index, Dunn index or Silhouette coefficient.

Some of them tend to favor some algorithms, make sure you still do manual analysis of the clusters

Use *external* metrics, which requires some classification labels.



How do I know the clusters are right?

Profile the clusters and manually check if they give you useful knowledge about whatever you are analysing.

Use *internal* metrics like Davies-Bouldin index, Dunn index or Silhouette coefficient.

Use *external* metrics, which requires some classification labels.

If *all* your data has labels, think carefully why you need clustering at all, you may be good with a classification algorithm

How do I know the clusters are right?

Profile the clusters and manually check if they give you useful knowledge about whatever you are analysing.

Use *internal* metrics like Davies-Bouldin index, Dunn index or Silhouette coefficient.

Use *external* metrics, which requires some classification labels.

Clustering is a tool for exploratory analysis, use it to explore and deliver insights.

Clustering is a tool for exploratory analysis, use it to explore and deliver insights.

You can use it for automated processes in properly controlled environments and validated use cases.