Clustering

Clustering

- Explore the structure of the dataset by splitting the data points into different groups via well-studied clustering algorithms

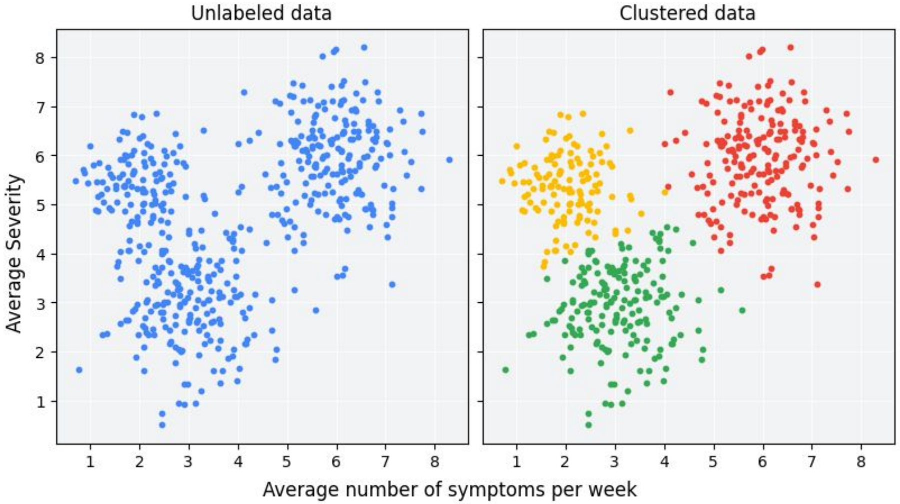


Image source: https://developers.google.com/machine-learning/clustering/overview

Clustering

- When some labels of a classification task are not available, clustering methods can sometimes be used to fill in the rest of the labels

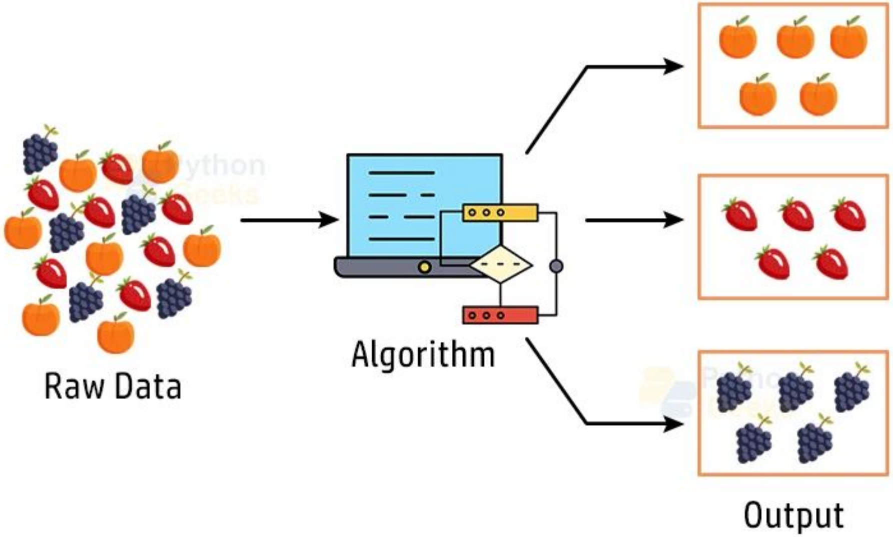


Image source: https://pythongeeks.org/clustering-in-machine-learning/

Clustering

- It can also be used to identify and visualize interesting groups of data
- If they are outliers in your dataset that form a cluster, you know there is probably a common source/reason for the outliers

Clustering

- Approximate idea: group geometrically close points into the same group, and geometrically distant points into different groups

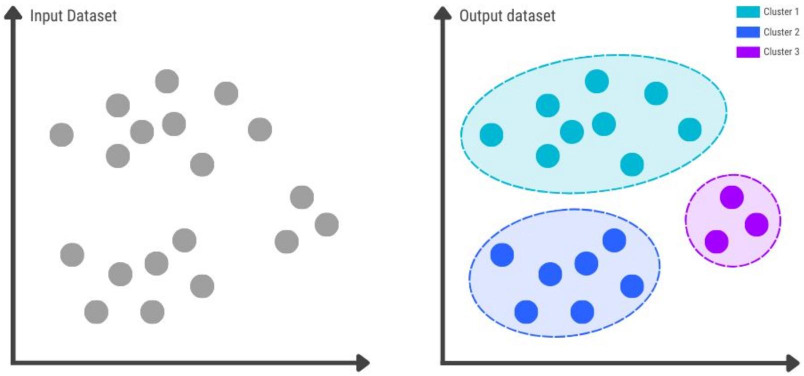


Image source: https://www.pi.exchange/blog/clustering-in-machine-learning

Clustering Algorithms

- Centroid-based: K-Means
- Density-based: DBSCAN family
- Gaussian mixture models (GMMs); they are useful for clustering data into visible clusters with complex shapes or even those that overlap

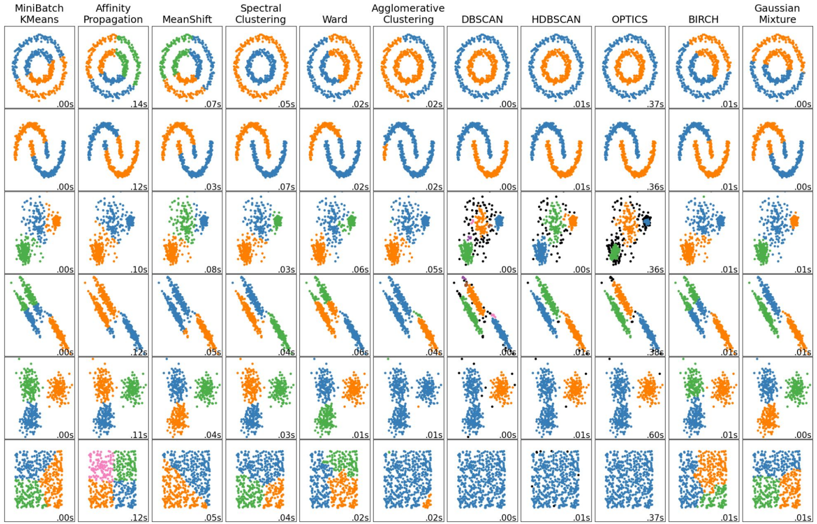


Image source: https://scikit-learn.org/stable/modules/clustering.html

K-Means Clustering

- Specify K random initial points as the initial "centroids" of the K clusters (that is where the "K" in the "K-Means" comes from)

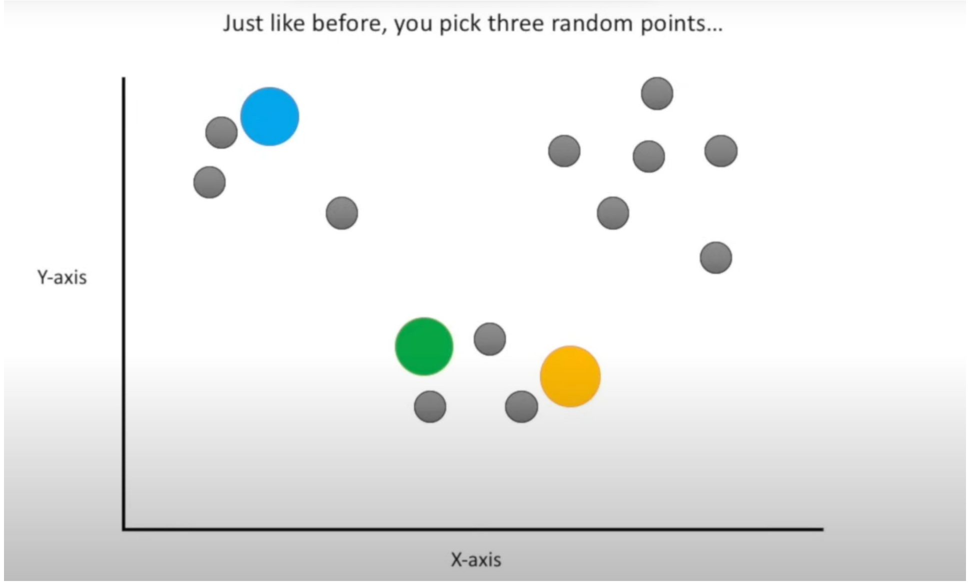
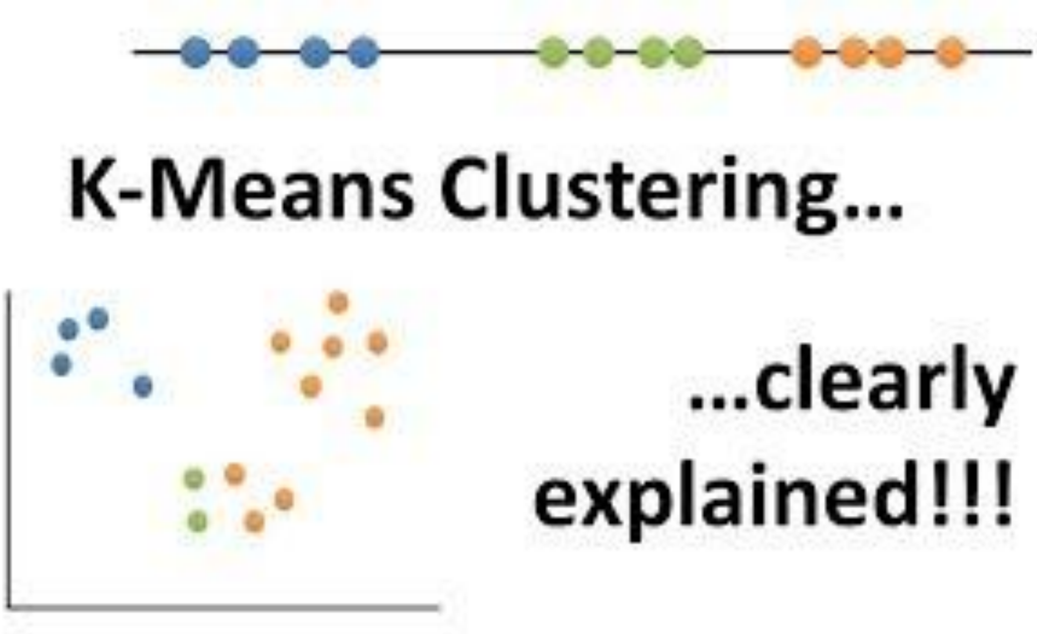


Image source: https://www.youtube.com/watch?v=4b5d3muPQmA

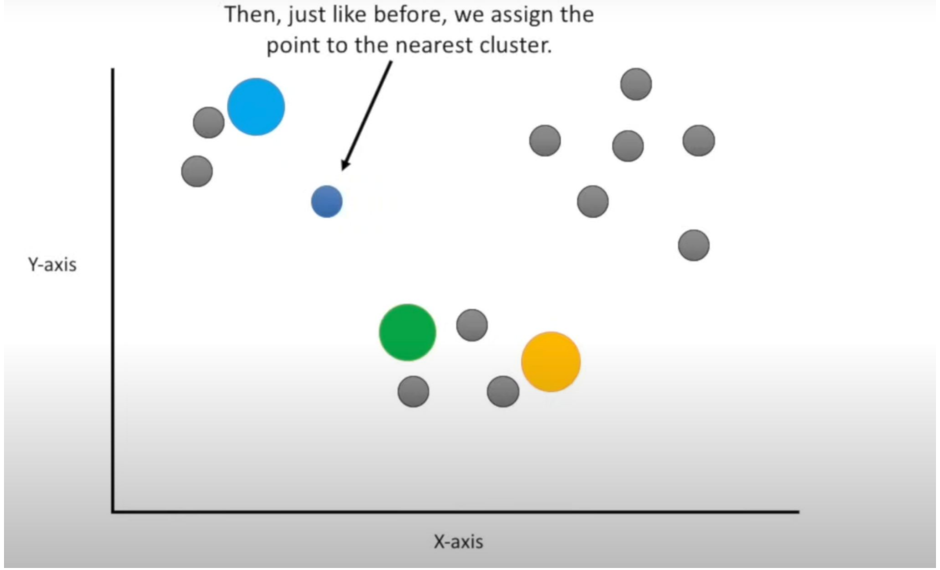
K-Means Clustering

Video: https://www.youtube.com/watch?v=4b5d3muPQmA



K-Means Clustering

- Associate each remaining data point with the centroid that is the least distant from that point (that is the initial clustering)

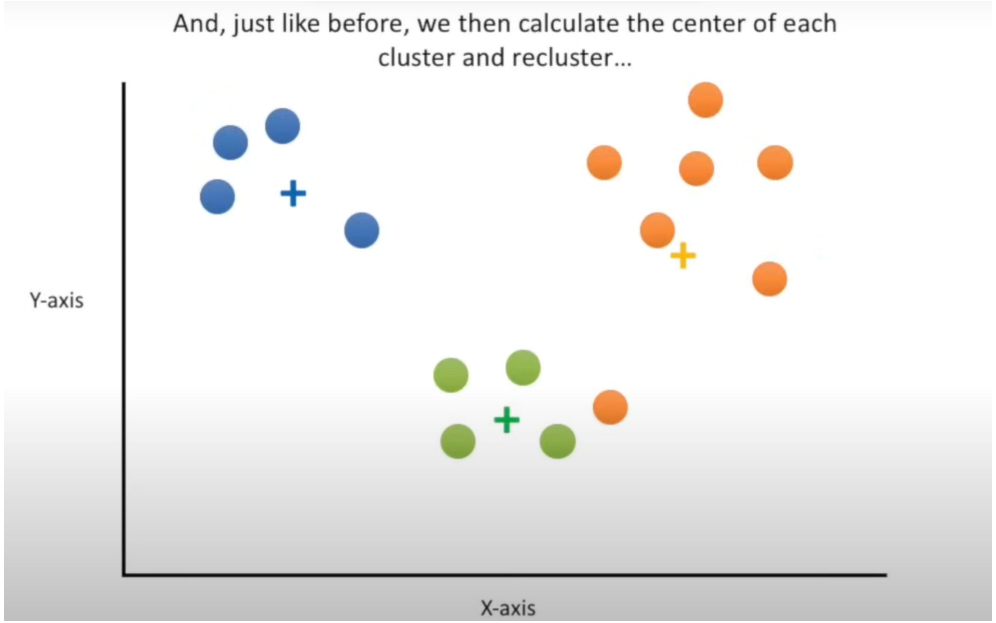


Then, just like before, we assign the point to the nearest cluster.
Y-axis
X-axis

Image source: https://www.youtube.com/watch?v=4b5d3muPQmA

K-Means Clustering

- Recalculate the centroids based on the initial clustering, and then ignore all clustering labels from the data to cluster the data again based on the new centroids



And, just like before, we then calculate the center of each cluster and recluster...

Image source: https://www.youtube.com/watch?v=4b5d3muPQmA

K-Means Clustering

- Iterative refinement: Repeat the process until it converges (i.e., nothing changes throughout cycles of reclustering) or it reaches the maximum step limit

K-Means Clustering

- Advantages: allows specifying the number of classes, which can be useful when you know what you need to classify data into
- Advantages: can classify unseen points as members of one of the clusters

K-Means Clustering

- Disadvantages: it cannot effectively model nonlinear dependencies, and may not work well for highly asymmetric data
- It relies on calculating mean values of points so it can be sensitive to outliers

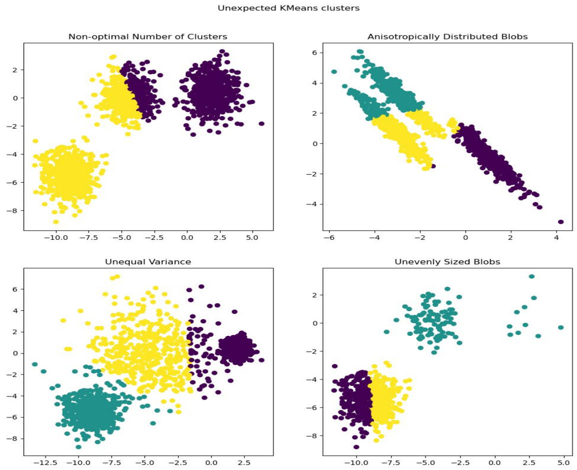
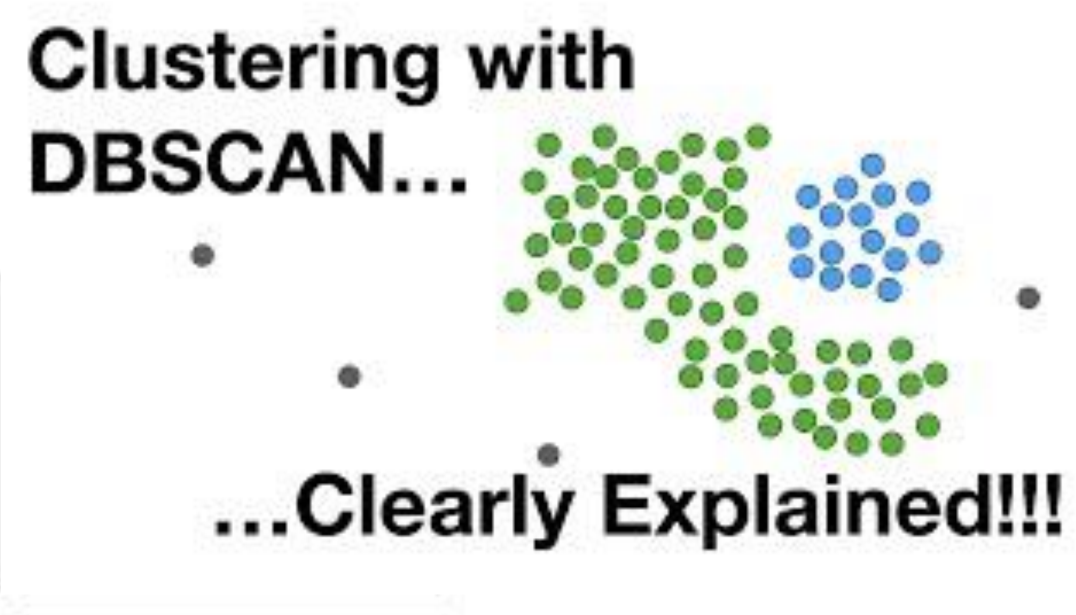


Image source: https://scikit-learn.org/stable/modules/clustering.html

DBSCAN

Video: https://www.youtube.com/watch?v=RDZUdRSDOok



DBSCAN

- Specify: eps (how close must a point be to extend the cluster) and min_samples (how many points we need to make a cluster)

DBSCAN

- Randomly select a point; if a point has (min_samples - 1) neighbors within a distance of eps, it is selected as a "core point" and a cluster is formed
- Expand the cluster based on the distance threshold eps; if a point has enough close neighbors to start a cluster itself, it can add other points; otherwise it cannot add other points
- Repeat until no points can be added
- Find another core point from the remaining data, and do the cluster expansion process again. Repeat until no new clusters can be formed.
- Dense regions tend to get clustered together

(add youtube video screenshots every step of the way)

DBSCAN

- Advantages: can model nonlinear boundaries between classes
- Disadvantages: requires good understanding of the scale of the data and careful hyperparameter tuning

Try It Out

- Experience K-Means and DBSCAN clustering in this notebook: https://colab.research.google.com/drive/196UkbTkRnz2X7-CXSpHyCcFp3r9xHZio#scrollTo=6zf-9_wtTzVW

Useful Tip

- If you struggle to achieve the right separation, try to separate the dataset into more clusters
- And then group some of these clusters together to achieve the separation you want
- This has appeared in the Malaysia AI Olympiad (eye of feature engineering question) and is a decent-scoring solution in the IOAI (IOAI 2025 Q5) that doesn’t use pseudo-labels to train a classifier.