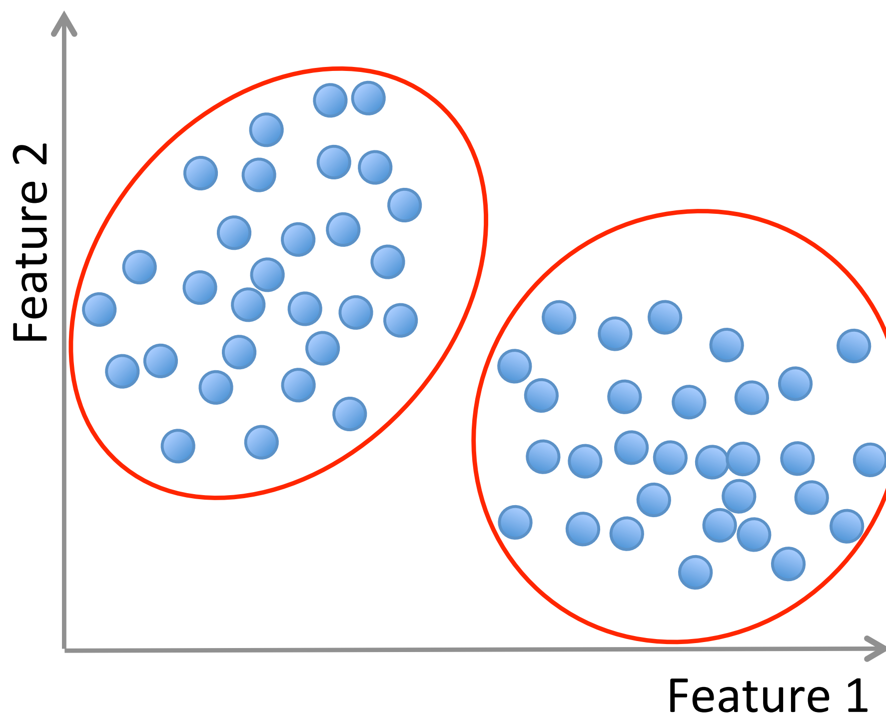
Artificial Intelligence

Machine Learning
Unsupervised learning



Unsupervised Learning

Training data: "examples" x.

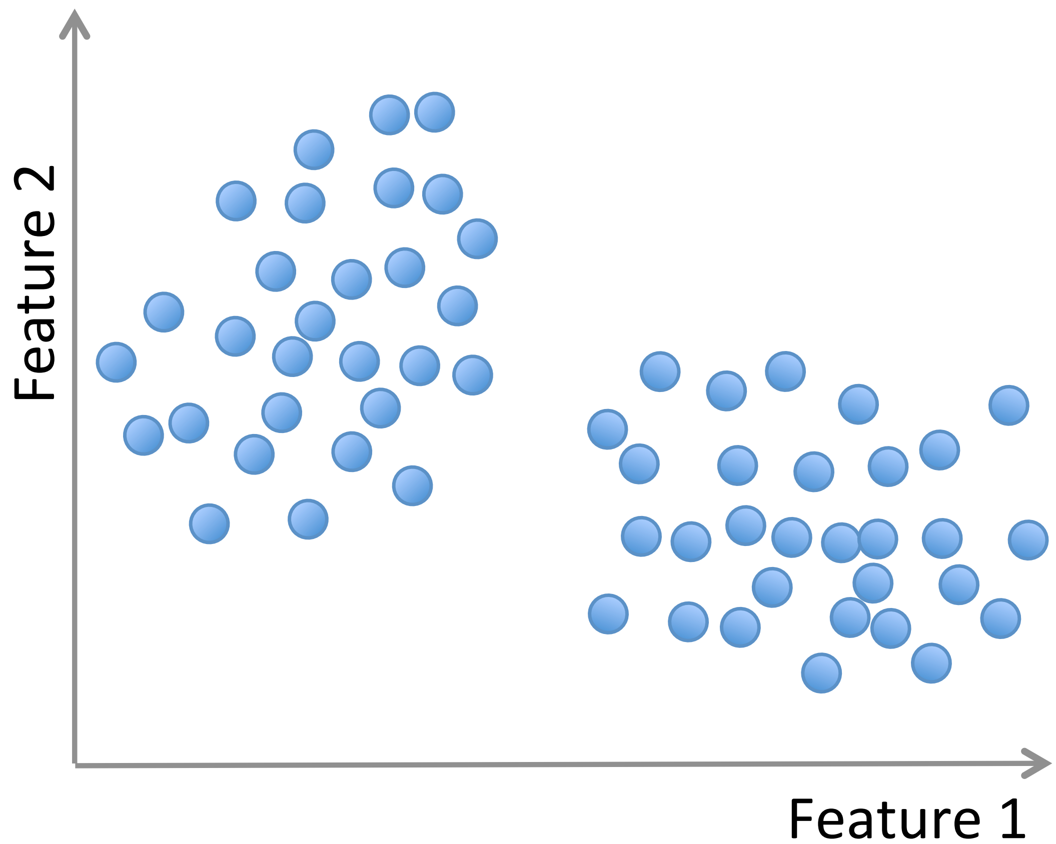
$$x_1, \ldots, x_n, \;\; x_i \in X \subset \mathbb{R}^n$$

- **Clustering/segmentation:**

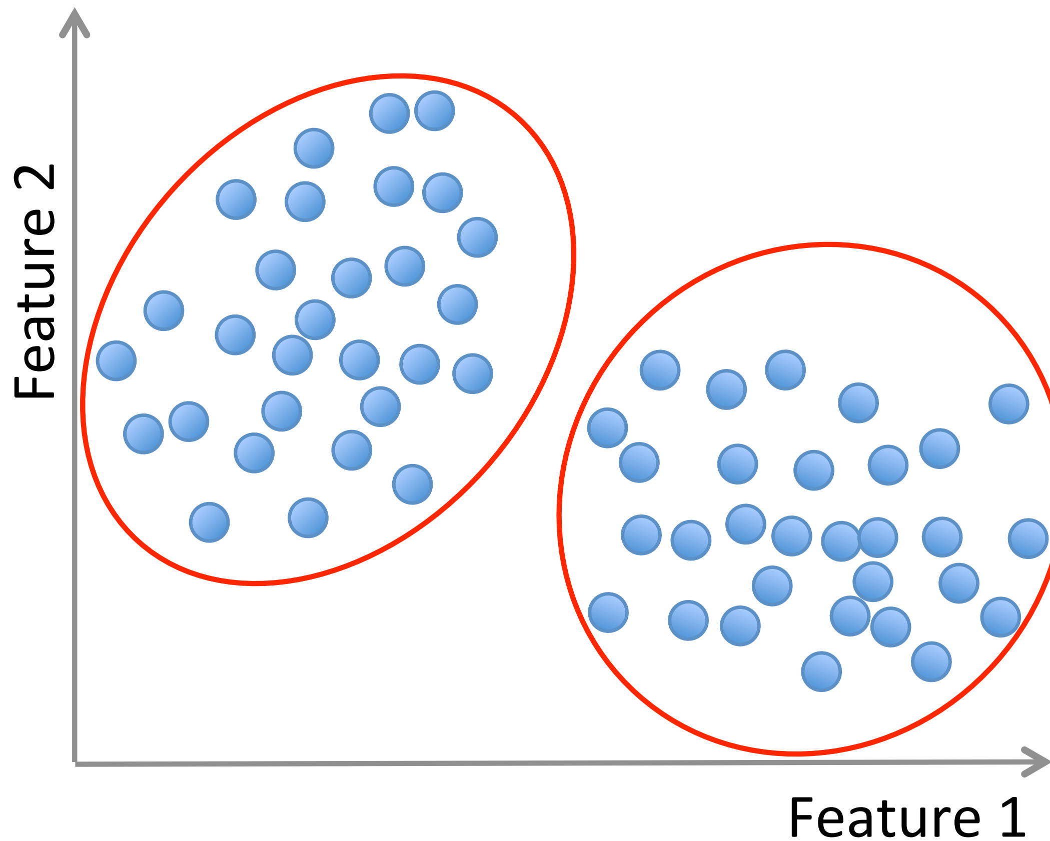
  $f : \mathbb{R}^d \longrightarrow \{C_1, \ldots C_k\}$ (set of clusters).

  Example: Find clusters in the population, fruits, species.

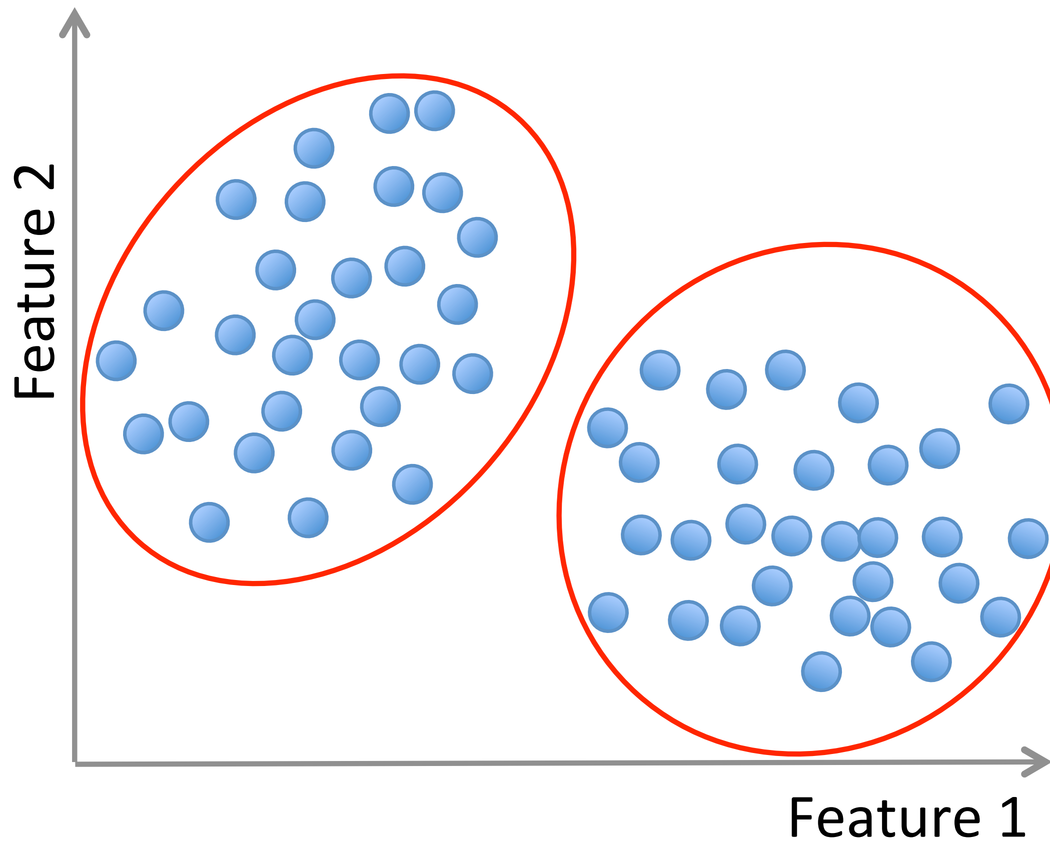
Unsupervised learning



Unsupervised learning



Unsupervised learning



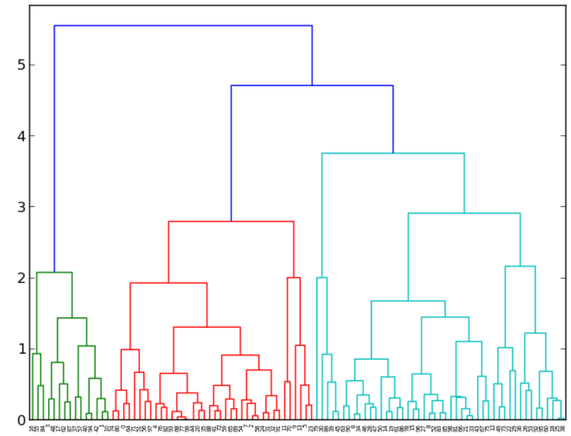
Methods: K-means, gaussian mixtures, hierarchical agglomerative clustering, spectral clustering, DBScan, etc.

Clustering examples

- Clustering of the population by their demographics.
- Clustering of geographic objects (mineral deposits, houses, etc.)
- Clustering of stars
- Audio signal separation. Example?
- Segmentation-based object categorization in Image segmentation. Example?

Clustering methods

- **Hierarchical or agglomerative clustering**: Nearby points should belong to the same cluster. Represents the clusters as dendrograms. Hierarchy of clusters.

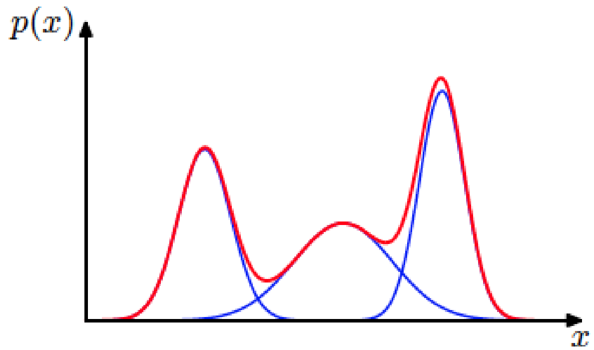


  x axis: points, y-axis: merging distance

- **Centroid-based clustering** (e.g., K-means aka Lloyd's algorithm): Each cluster is represented by a center point.

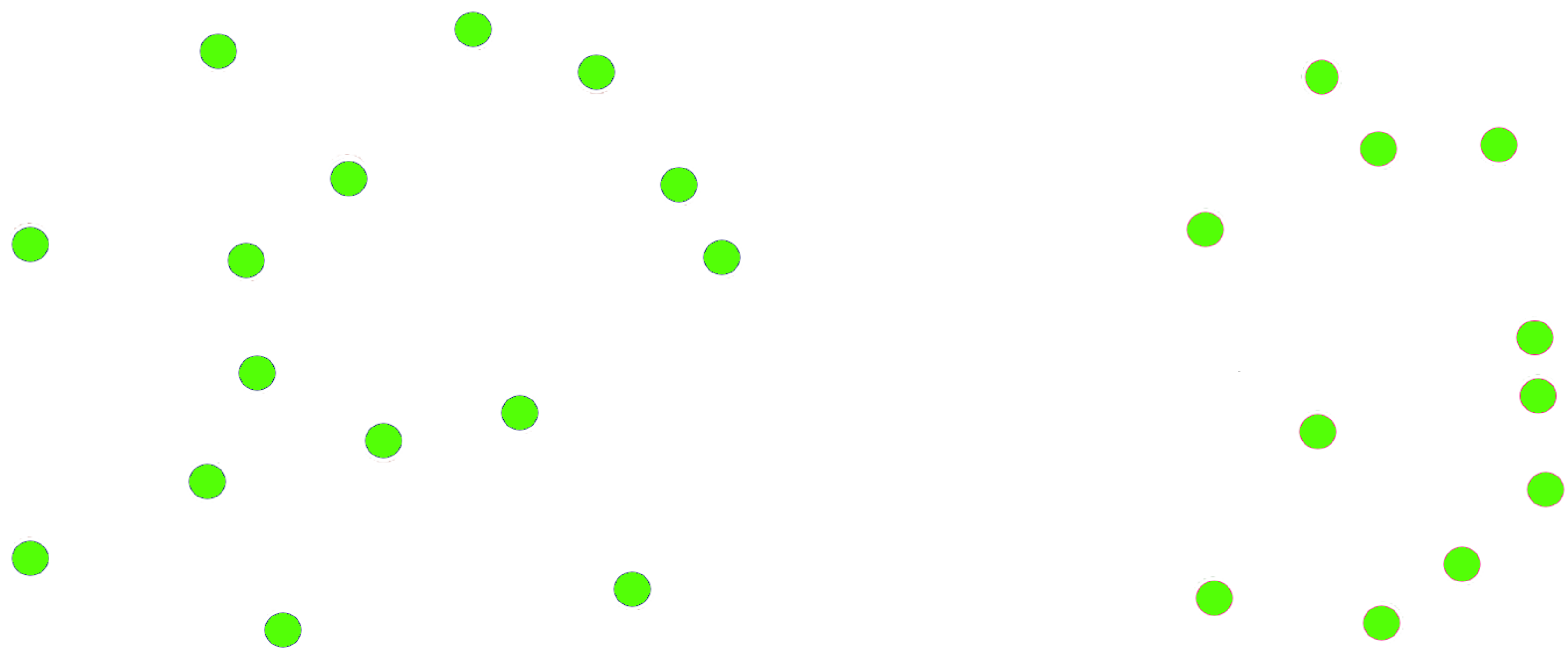
Clustering methods

- **Distribution-based clustering** (e.g, EM algorithm, Gaussian Mixture Models (GMMs)): From stats, clusters come from the same distribution, models data with a set of Gaussian distributions.

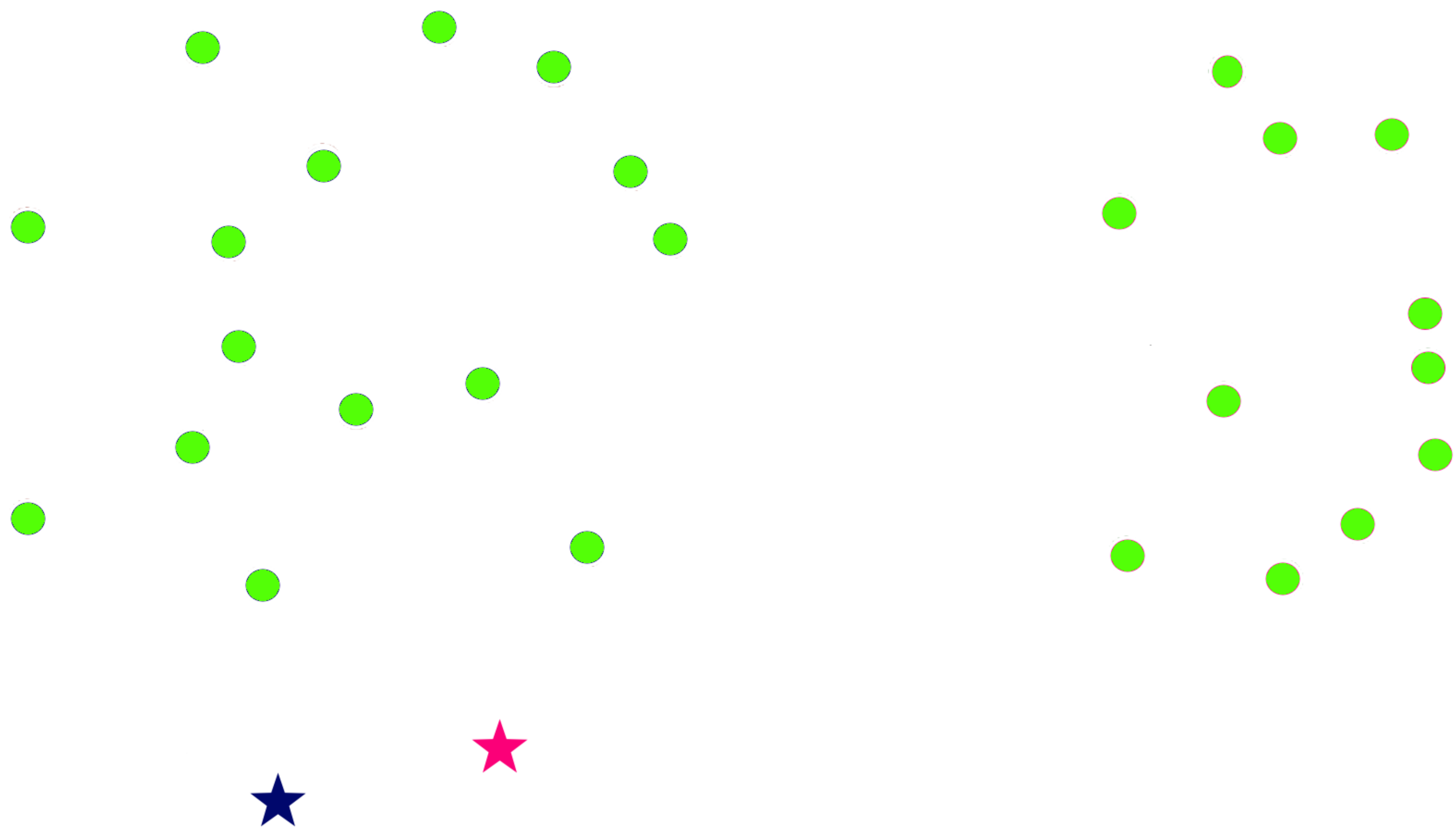


- **Density-based clustering** (E.g., DBScans, Optics, Spectral Clustering): A cluster is a high-density area of points.

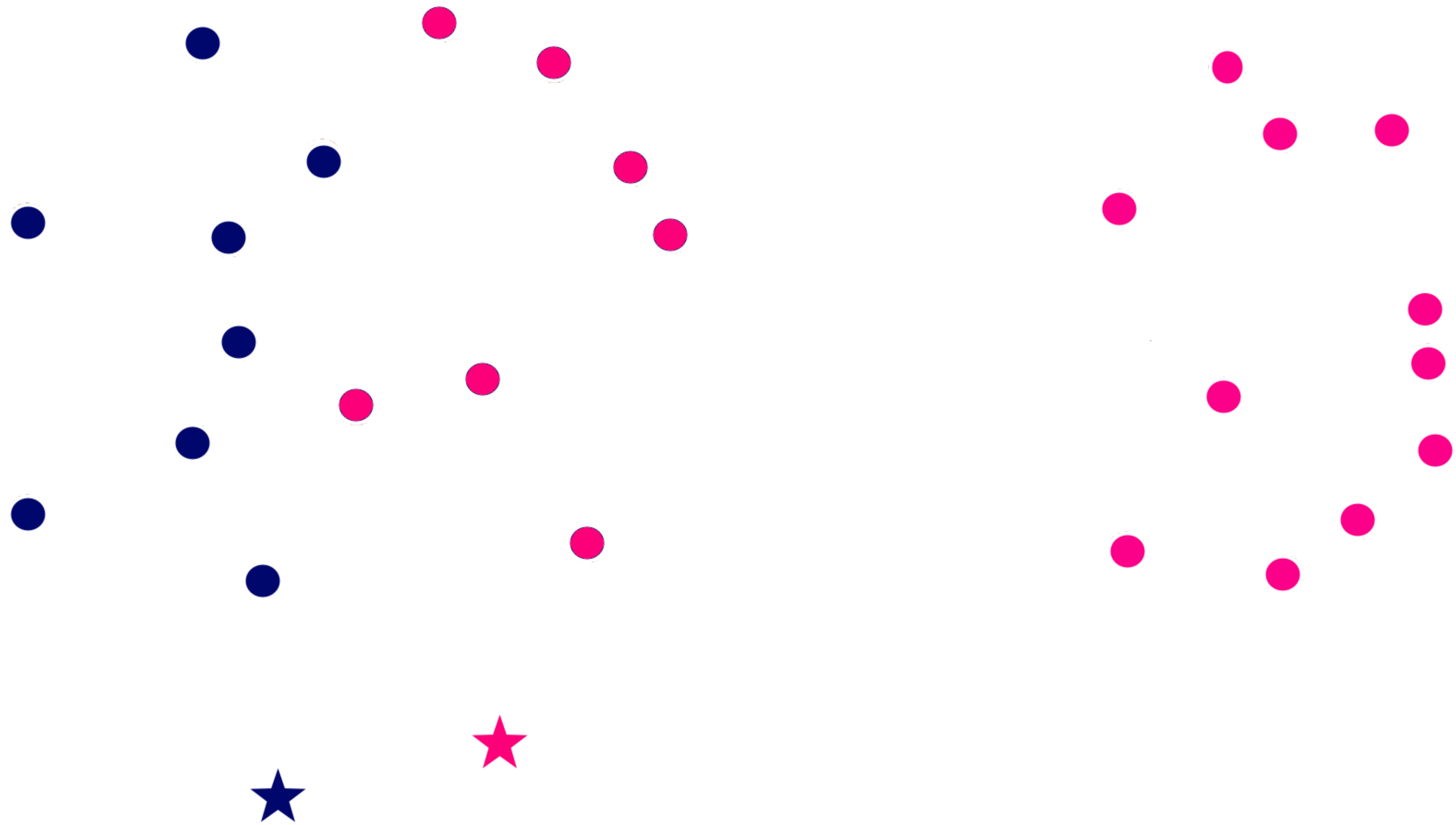
K-Means: example



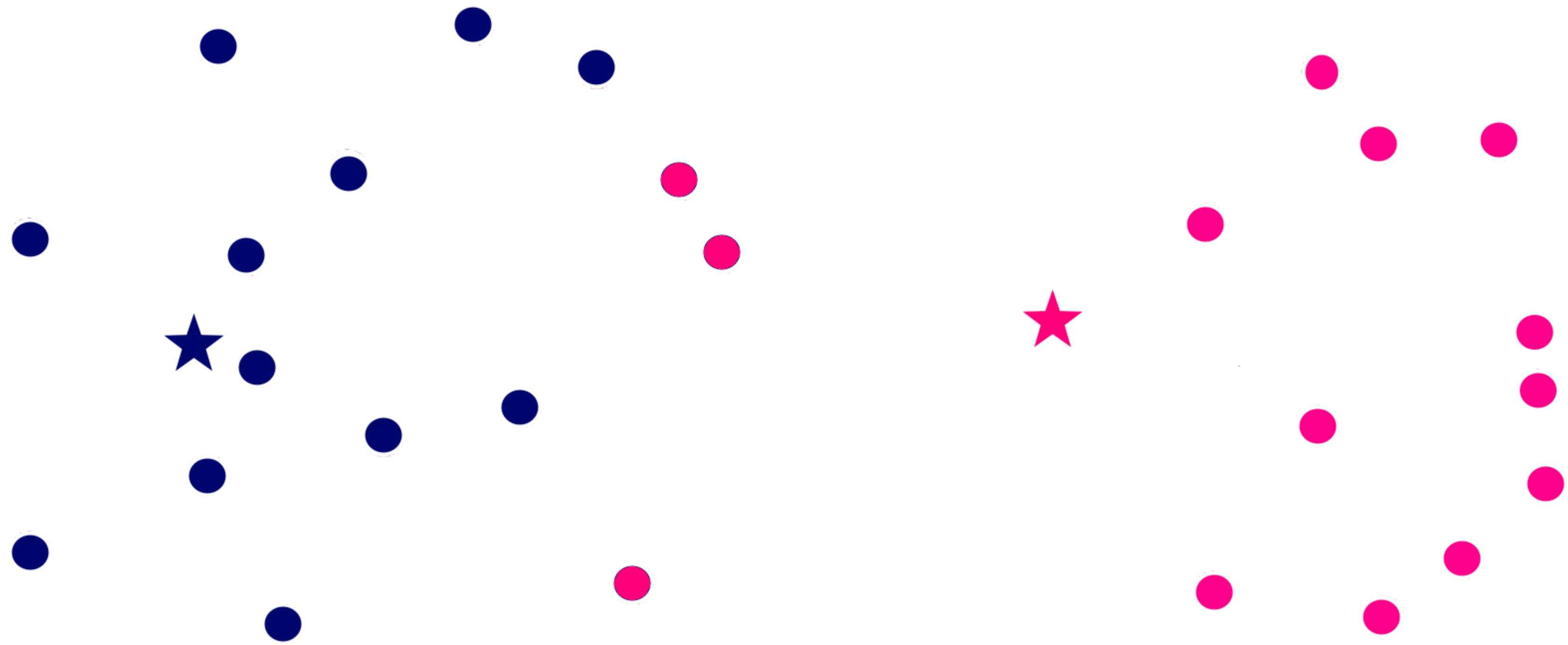
K-Means: example



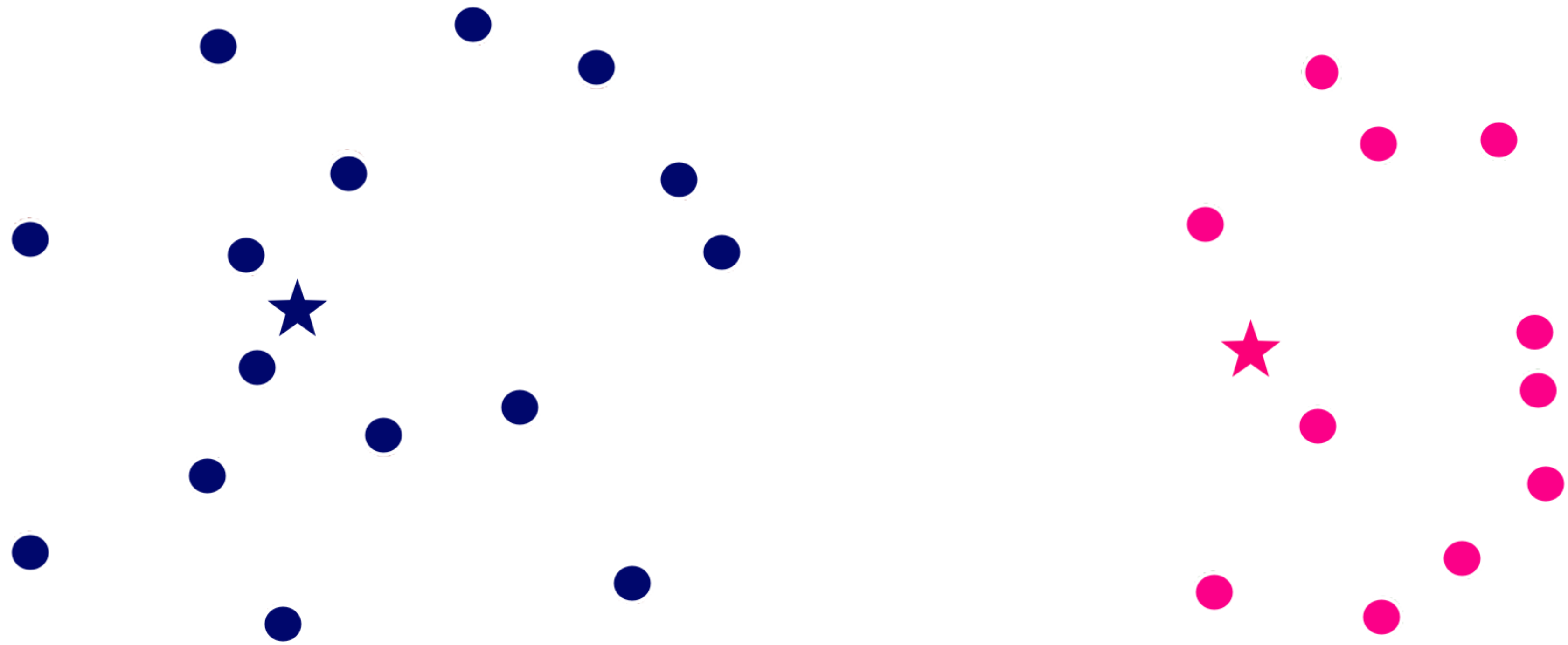
K-Means: example



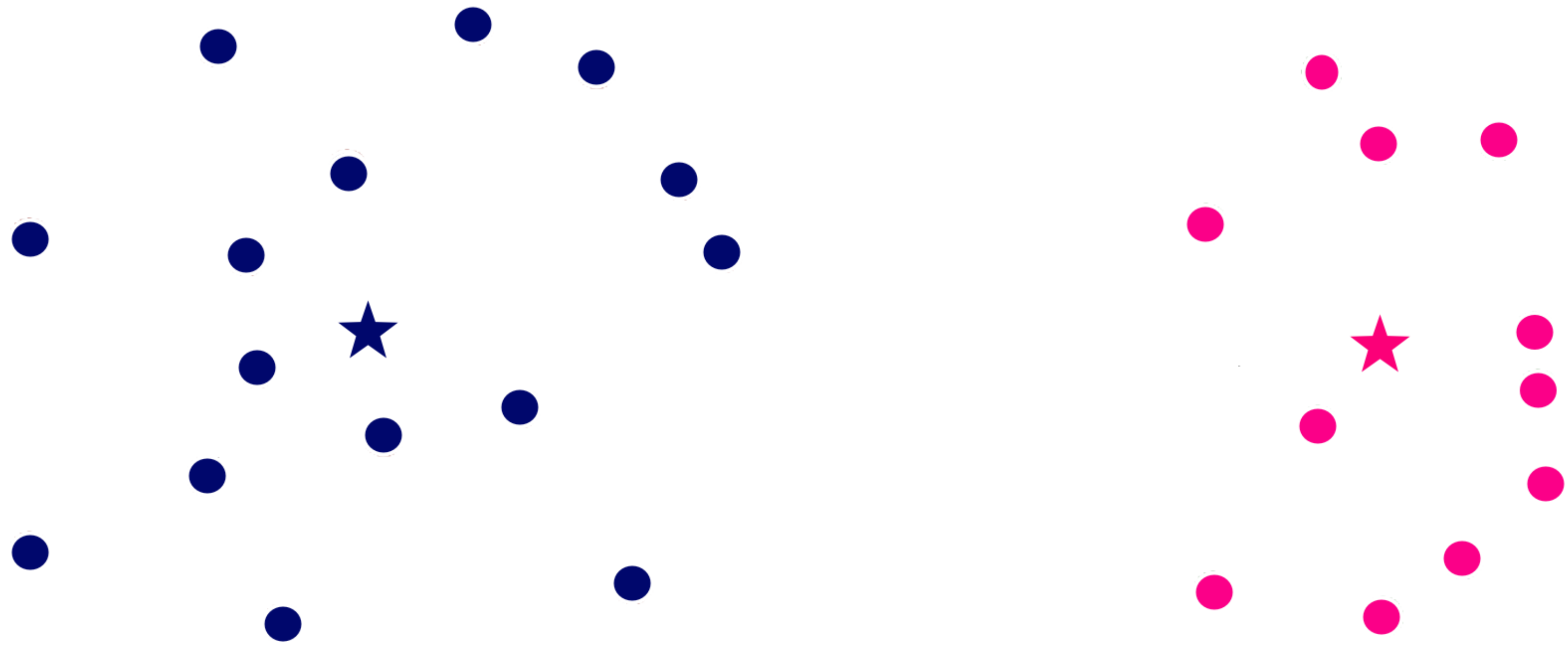
K-Means: example



K-Means: example



K-Means: example



Clustering: K-Means

- **Goal**: Assign each example $(x_1, \ldots, x_n)$ to one of the k clusters $\{\mathcal{C}_1, \ldots \mathcal{C}_k\}$.

Clustering: K-Means

- **Goal**: Assign each example $(x_1, \ldots, x_n)$ to one of the k clusters $\{\mathcal{C}_1, \ldots \mathcal{C}_k\}$.
- μ_j is the mean of all examples in the j^{th} cluster.

Clustering: K-Means

- **Goal**: Assign each example $(x_1, \ldots, x_n)$ to one of the k clusters $\{\mathcal{C}_1, \ldots \mathcal{C}_k\}$.
- μ_j is the mean of all examples in the j^{th} cluster.
- **Minimize**:

$$J = \sum_{j=1}^{k} \sum_{x_i \in \mathcal{C}_j} ||x_i - \mu_j||^2$$

Clustering: K-Means

Algorithm K-Means:

`Initialize randomly` $\mu_1, \cdots \mu_k$.

Clustering: K-Means

Algorithm K-Means:

`Initialize randomly` $\mu_1, \cdots \mu_k$.

`Repeat`

`Assign each point` x_i `to the cluster with the closest` μ_j.

Clustering: K-Means

Algorithm K-Means:

`Initialize randomly` $\mu_1, \cdots \mu_k$.

`Repeat`

`Assign each point` x_i `to the cluster with the closest` μ_j.

Clustering: K-Means

Algorithm K-Means:

Initialize randomly $\mu_1, \cdots \mu_k$.

Repeat

Assign each point x_i to the cluster with the closest μ_j.

Calculate the new mean for each cluster as follows:

$$\mu_j = \frac{1}{|\mathcal{C}_j|} \sum_{x_i \in \mathcal{C}_j} x_i$$

Until convergence*.

Clustering: K-Means

Algorithm K-Means:

Initialize randomly $\mu_1, \cdots \mu_k$.

Repeat

Assign each point x_i to the cluster with the closest μ_j.

Calculate the new mean for each cluster as follows:

$$\mu_j = \frac{1}{|\mathcal{C}_j|} \sum_{x_i \in \mathcal{C}_j} x_i$$

Until convergence*.

*Convergence: Means no change in the clusters OR maximum number of iterations reached.

K-Means: applet

https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

K-Means: pros and cons

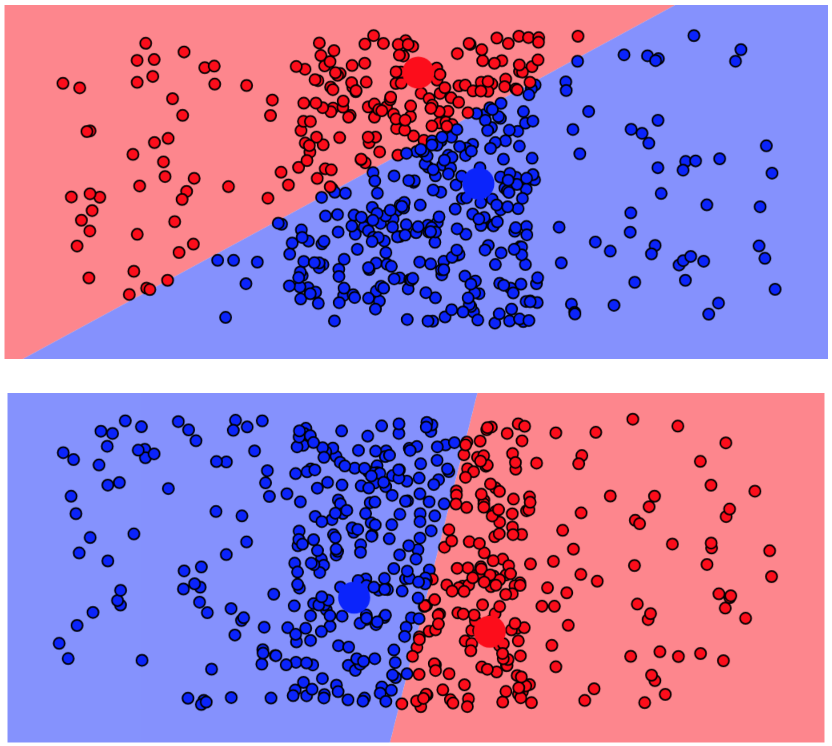
+ Easy to implement

  BUT...

- Need to know K
- Suffer from the curse of dimensionality
- Does not guarantee unique clustering depending on the initial choice of centers (Kmeans++)

  David Arthur and Sergei Vassilvitskii. k-means ++ : The Advantages of Careful Seeding.

K-Means: pros and cons



K-Means: questions

1. **How to set k to optimally cluster the data?**

2. **How to evaluate your model?**

3. **How to cluster non circular shapes?**

K-Means: question 1

How to set k to optimally cluster the data?

G-means algorithm (Hamerly and Elkan, NIPS 2003)

1. Initialize k to be a small number
2. Run k-means with those cluster centers, and store the resulting centers as C
3. Assign each point to its nearest cluster
4. Determine if the points in each cluster fit a Gaussian distribution (Anderson-Darling test).
5. For each cluster, if the points seem to be normally distributed, keep the cluster center. Otherwise, replace it with two cluster centers.
6. Repeat this algorithm from step 2. until no more cluster centers are created.

K-Means: question 2

How to evaluate your model?

- **Not trivial** (as compared to counting the number of errors in classification).
- **Manual evaluation**: human expert, often subjective, expensive judgement.
- **External evaluation**: use of ground truth of external data. E.g., mutual information, entropy, adjusted random index, etc. But if we know the classes/cluster, we don't need to cluster. Besides, labels are not always available.

K-Means: question 2

How to evaluate your model?

- **Internal evaluation**: using same data. high intra-cluster similarity (documents within a cluster are similar) and low inter-cluster similarity. E.g., Davies-Bouldin index that takes into account both the distance inside the clusters and the distance between clusters.

$$BouldinIndex = \frac{1}{K}\sum_{1}^{K} max_{j \neq i}\left(\frac{\sigma_i + \sigma_j}{d(c_i c_j)}\right)$$

  K number of clusters, c_i is the centroid of cluster i, and σ_i is average distance of all points in cluster i to c_i. Same for c_j. $d(c_i c_j)$ is the distance btw center c_i and c_j. The lower the value of the index, the wider is the separation between different clusters, and the more tightly the points within each cluster are located together.

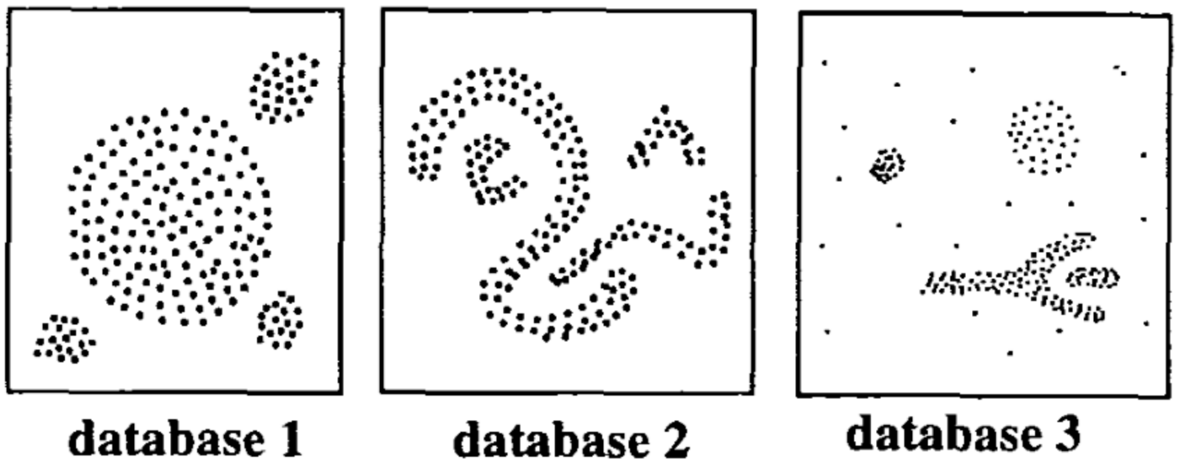
K-Means: question 3

How to cluster non circular shapes?

There are other methods: spectral clustering, DBSCAN, BIRCH, etc. that handle other shapes.

DBSCAN

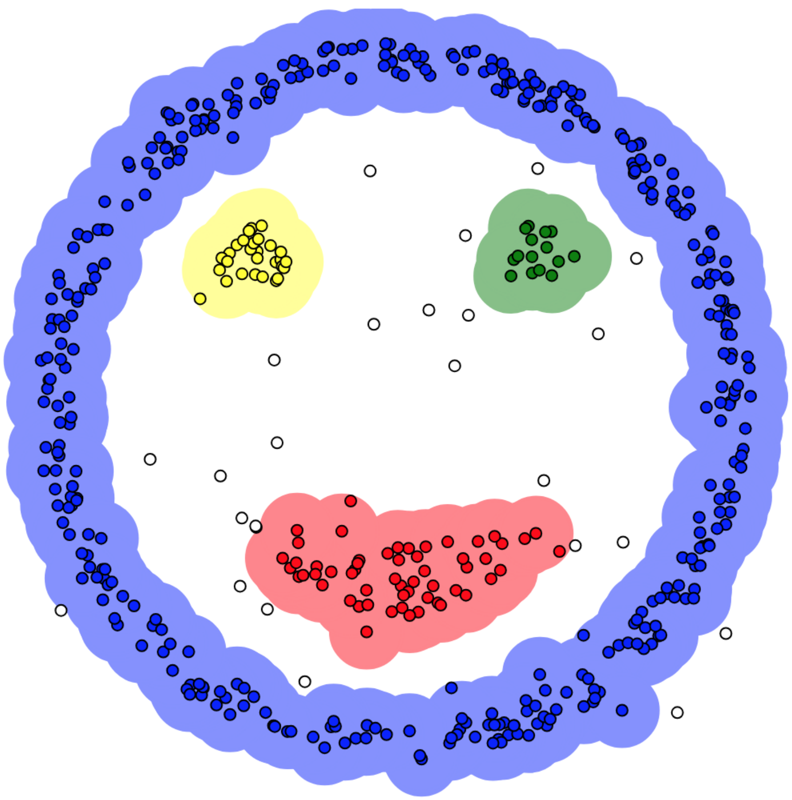
DBSCAN: Density-Based Spatial Clustering of Applications with Noise. and its variant OPTICS.



(From Ester et al. 1996)

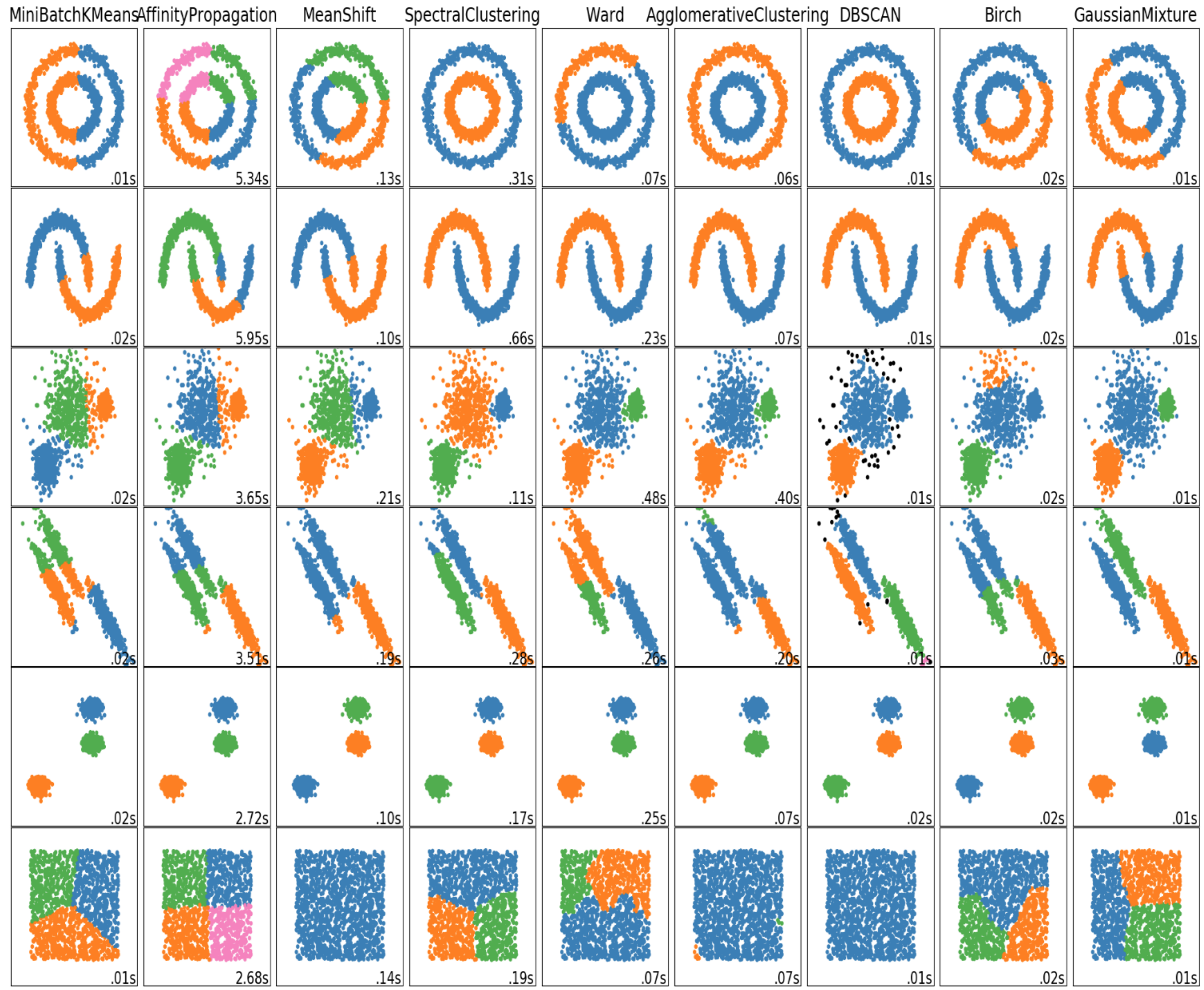
DBSCAN

DBSCAN: Density-Based Spatial Clustering of Applications with Noise. and its variant OPTICS.



https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

K-Means: questions



A comparison of the clustering algorithms in scikit-learn

`http://scikit-learn.org/stable/modules/clustering.html`

Credit and further reading

1. Greg Hamerly and Charles Elkan. Learning the k in k-means. In Neural Information Processing Systems, 2003.

2. David Arthur and Sergei Vassilvitskii. k-means $++$: The Advantages of Careful Seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, volume 8, pages 10271035, 2007.

3. Martin Ester, Hans-Peter Kriegel, Joerg Sander, Xiaowei Xu (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Institute for Computer Science, University of Munich. Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)