neue fische
School and Pool for Digital Talent

All that Clustering

Meetup

neue fische
School and Pool for Digital Talent

Code of Conduct

We expect all participants to our events and community to abide to this code of conduct:

LadyNerds Code of Conduct (http://bit.ly/LadyNerds-CoC).

We follow the **LadyNerds Code of Conduct** because we are dedicated to providing a safe, inclusive, welcoming, and harassment-free space and experience for all members and guests, regardless of gender identity and expression, sexual orientation, disability, physical appearance, socioeconomic status, body size, ethnicity, nationality, level of experience, age, or religion (or lack thereof).

The Code of Conduct exists because of that dedication. We do not tolerate harassment in any form and we prioritise marginalised people's safety over privileged people's comfort.

neue fische
in numbers

→ 8 Bootcamp Programs
→ 5 Locations
→ 900+ Graduates
→ 85% Job success

neuefische.de

Upcoming Data Bootcamps

- 26 June: Data Practitioner Part Time - 24 weeks
- 17 July: Data Analytics - 12 weeks
- 24 July: Data Science - 12 weeks
- 21 August: Machine Learning Engineer - 4 weeks

How to get in touch:

email: studienberatung@neuefische.de

neue fische

A practical approach to clustering

with Peter McIsaac, Mia Reimer and Nico Steffen

neue fische
School and Pool for Digital Talent

Table of content

About me

Nico Steffen
Lead Coach DP

My background:

- → Naval architect & Ocean engineer
- → Used to do R&D (and DS... without knowing it)

Why I joined neue fische:

- → Love seafood
- → Wide range of topics and always new areas of application

Ask me about:

- → Life universe and everything
- → Technology, Ships
- → Dota

About me

Peter McIsaac
Senior Data Science Consultant
MID GmbH

My background:

Data science consultant and Scrum Master with previous lives in applied physics, German Studies and Python-driven digital humanities

Why I joined neue fische:

I took the Data Science boot camp in the summer of 2021 to launch my Data Science career

Ask me about:

Long-distance running, home brewing

mcisaacpm@gmail.com

About me

Mia Reimer
Coach Data Science

My background:

- → Marine Geologist
- → Attended neuefische DS Bootcamp in 2020

Why I joined neue fische:

- → Great colleagues and atmosphere
- → Always well stocked refrigerator

Ask me about:

- → Tablesoccer
- → Knitting

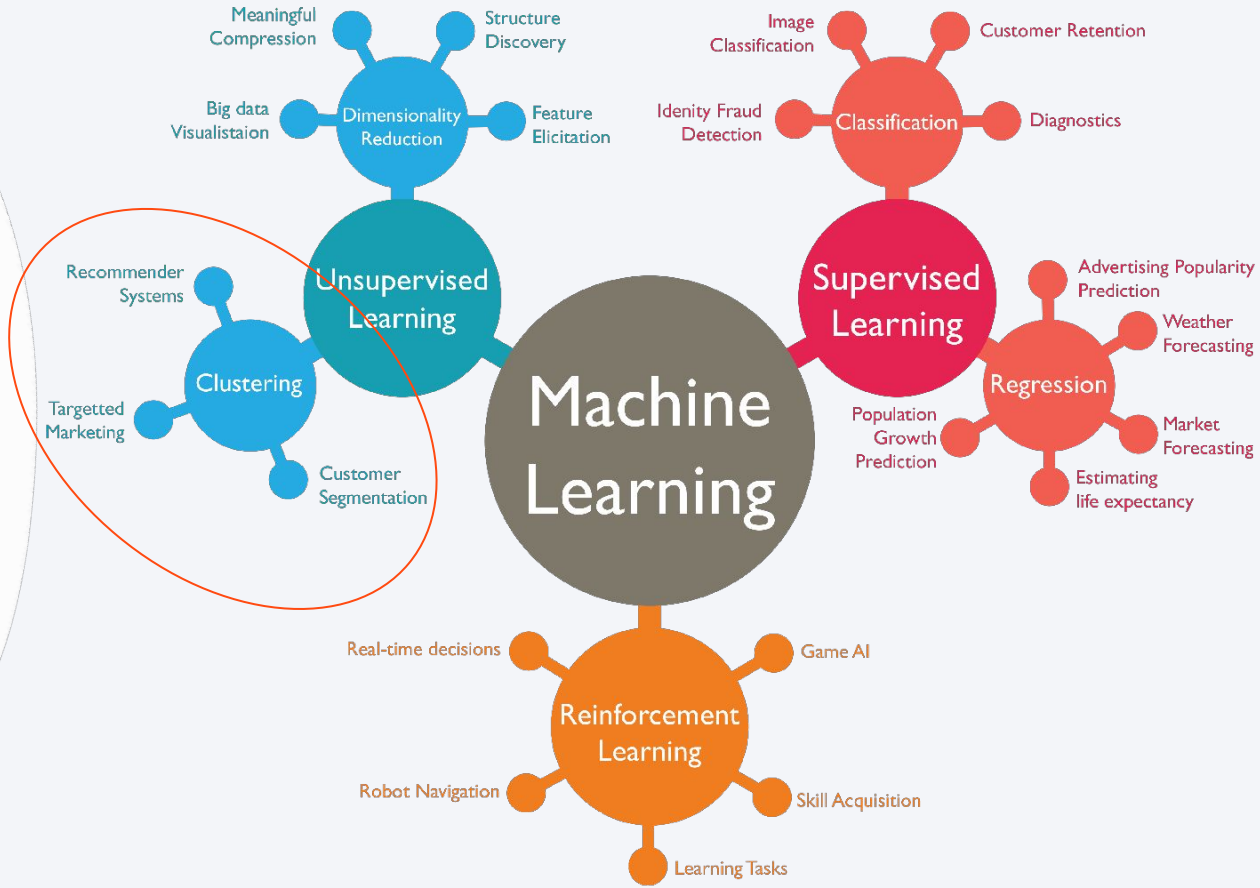
mia.reimer@neuefische.de

neue fische
School and Pool for Digital Talent

Table of content

ML Landscape

Clustering is a domain of **unsupervised learning**.



Idea behind clustering

Partition an unstructured dataset into groups (clusters)

→ observations in same cluster are similar

→ points in different clusters are different

- visualise resulting clusters (also in combination with dimensionality reduction)
- interpret clusters by determining what features define the respective cluster
- pass groups as features or outcomes to a regression / classification model

Applications for clustering:

Customer segmentation

Recommender systems

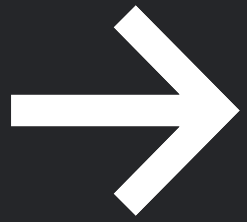
Anomaly detection

Document clustering

neue fische
School and Pool for Digital Talent

Example

K-Means



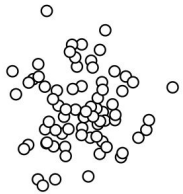
Why is K-Means so popular?

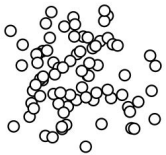
It is simple!

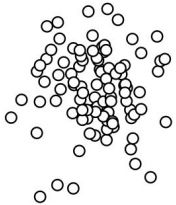
Idea:

- divide data into K clusters
- minimize the sum of the squared distances of each record to the mean of its cluster → **within cluster sum of squares**

Let’s have a look at an example.



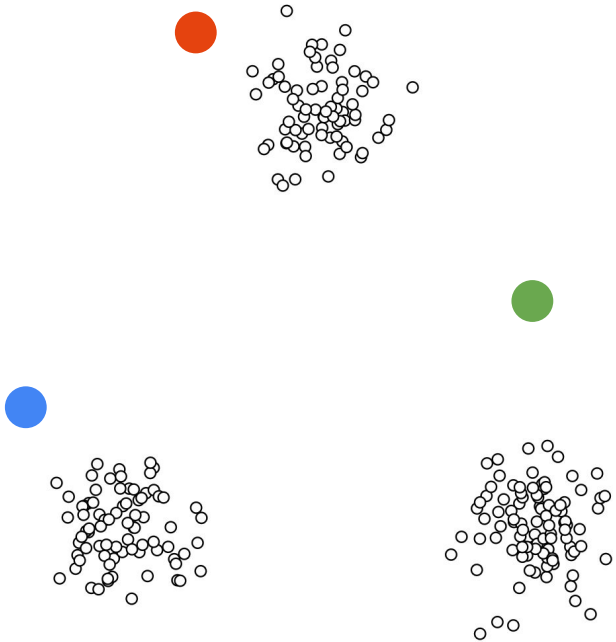




K-means algorithm with

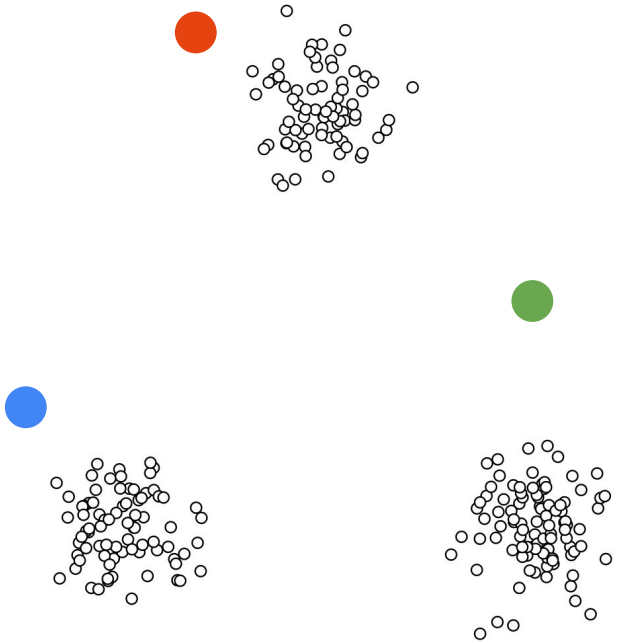
- two features x and y
- n observations
- k=3

Let’s have a look at an example.



- randomly initialize starting cluster centroids
- assign observation to closest cluster centroid

Let’s have a look at an example.

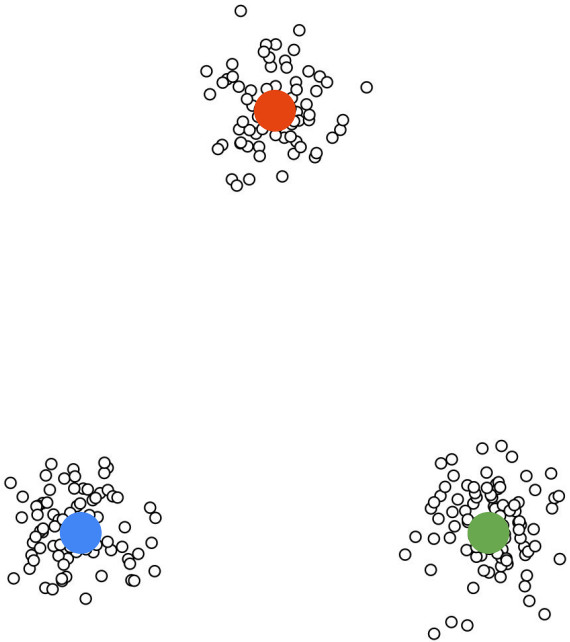


- calculate new centroid location

$$\bar{x}_k = \frac{1}{n_k} \sum_{i \in C(k)} x_i$$

$$\bar{y}_k = \frac{1}{n_k} \sum_{i \in C(k)} y_i$$

Let’s have a look at an example.



- reassign points to nearest cluster center and repeat the process

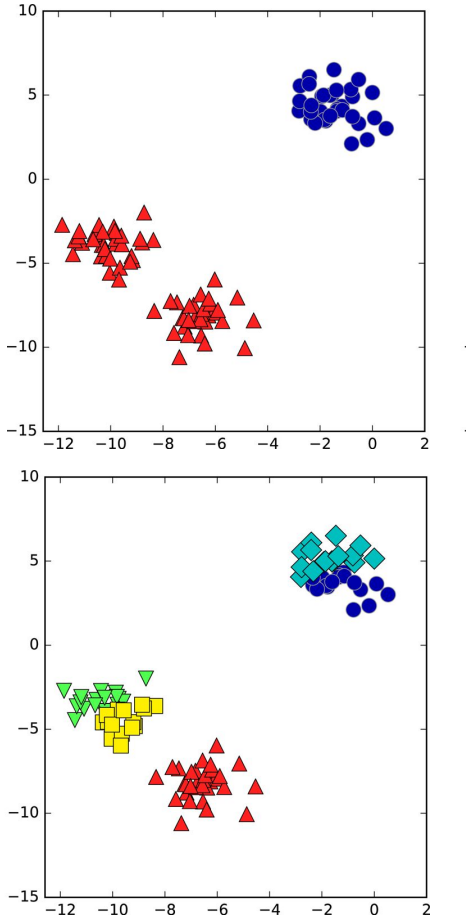
$$SS_k = \sum_{i \in C(k)} (x_i - \bar{x}_k)^2 + (y_i - \bar{y}_k)^2$$

$$\min SSS = \sum_{k=1}^{3} SS_k$$

How to decide on a value for k?

- select k based on business case
- select k based on knowledge about data or insights from EDA

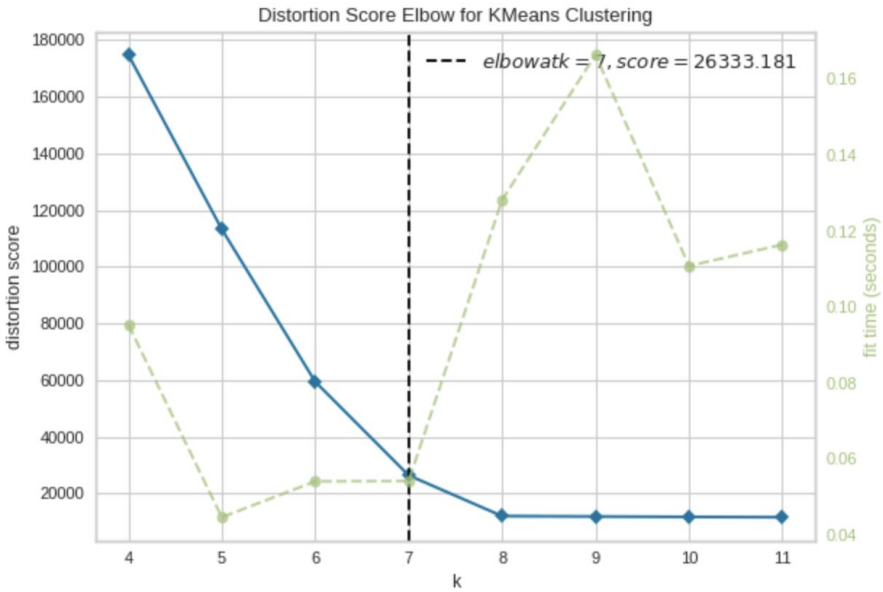
What do we do if we have no clue at all?



How to decide on a value for k?

Elbow Method:

- test different values for k
- calculate performance metric (eg. within-cluster sum of squares)
- decide for the amount of clusters where an additional cluster contributes only marginally (or point of maximum curvature)



Pros and Cons

Advantages

- straightforward and simple to understand (and explain to clients)
- good for large data and many kinds of data types
- fast calculation
- results often lend themselves to interpretation
- allows clustering of unseen data

Disadvantages

- depends on using the right number of clusters (k), which is often not known
- sensitive to outliers
- less optimal on non-linear and/or complex data
- greedy algorithm (will not always find the global optimum)

neue fische
School and Pool for Digital Talent

Hands-on Exercise

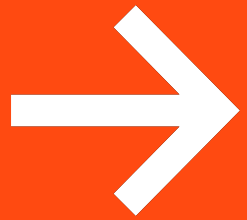
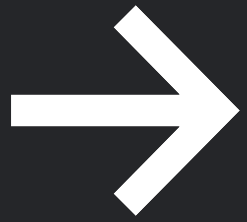


Table of content

neue fische
School and Pool for Digital Talent

neue fische
School and Pool for Digital Talent

Metrics & Evaluation

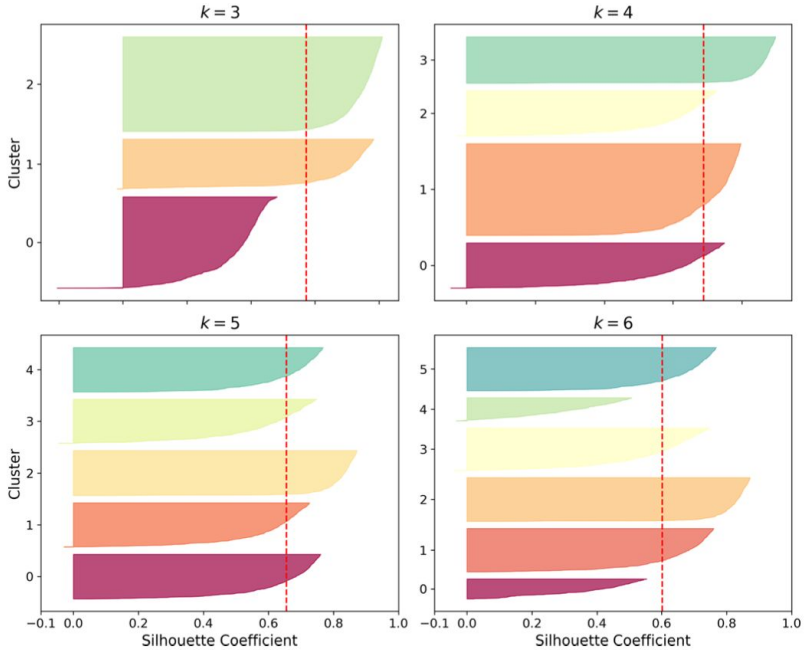


The Challenges of unlabeled data

- **“ground truth”** (i.e., # of clusters) is usually not known
- many algorithms require parameters to be estimated or manually set
- determination of meaningful results often depends on domain knowledge / context / business question
- visualization of clusters is difficult in high dimensions
- intrinsic measures such as **Silhouette Score**, **Calinski-Harabasz Index** and **Davies-Bouldin Index** are commonly used

Silhouette Score

- measures how close each point is to the points in neighboring clusters
- is calculated using the intra-cluster distance (i) and the mean nearest-cluster distance (n)
- values range from -1 to 1, with 1 indicating perfectly formed clusters and scores below 0 clusters that are overlapping or incorrectly assigned
- can be plotted



Silhouette Score - Pros and Cons

Advantages

- good interpretability
- comparisons between algorithms possible (though density-based often score better)

Disadvantages

- assumes good clusters are dense and well-separated (not all clusters might be)

Calinski-Harabasz Index

- also known as the Variance Ratio Criterion
- a measure of how similar a point is to its own cluster (cohesion) in comparison to other clusters (dispersion)
- calculated from the traces of the matrices Wk and Bk, the number of points (n) and the number of clusters (k)
- the higher the score the better the clustering

$$s = \frac{\mathrm{tr}(B_k)}{\mathrm{tr}(W_k)} \times \frac{n_E - k}{k - 1}$$

$$W_k = \sum_{q=1}^{k} \sum_{x \in C_q} (x - c_q)(x - c_q)^T$$

$$B_k = \sum_{q=1}^{k} n_q (c_q - c_E)(c_q - c_E)^T$$

Calinski-Harabasz Index

Advantages:

- fast computation

Disadvantages:

- score is unbounded
- assumes good clusters are dense and well-separated (not all clusters might be)

$$s = \frac{\mathrm{tr}(B_k)}{\mathrm{tr}(W_k)} \times \frac{n_E - k}{k - 1}$$

$$W_k = \sum_{q=1}^{k} \sum_{x \in C_q} (x - c_q)(x - c_q)^T$$

$$B_k = \sum_{q=1}^{k} n_q (c_q - c_E)(c_q - c_E)^T$$

Davies-Bouldin Index

- ratio of the within cluster scatter and the between cluster separation
- calculated as the sum of the average distance of the feature vectors S_i and S_j divided by the separation between them, $M_{i,j}$
- minimum score is 0, with low scores indicating better clustering
- can be plotted for parameters such as number of clusters or compared between algorithms

$$R_{i,j} = \frac{S_i + S_j}{M_{i,j}}$$

$$D_i \equiv max\, R_{i,\,j}$$
$$j \neq i$$

$$DB = \frac{1}{N} \sum_{i=1}^{N} D_i$$

Davies-Bouldin Index

Advantages:

- interpretability
- very fast computation

Disadvantages:

- can only use Euclidean distance (a potential problem with high dimensionality)

$$R_{i,j} = \frac{S_i + S_j}{M_{i,j}}$$

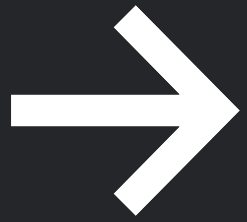
$$D_i \equiv max\, R_{i,\,j}$$

$$j \neq i$$

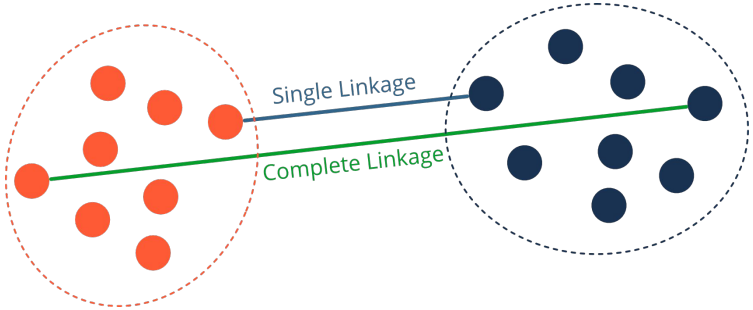
$$DB = \frac{1}{N} \sum_{i=1}^{N} D_i$$

neue fische
School and Pool for Digital Talent

Hierarchical Clustering



Agglomerative Clustering



Agglomerative clustering can find clusters not depending on random initialisation.

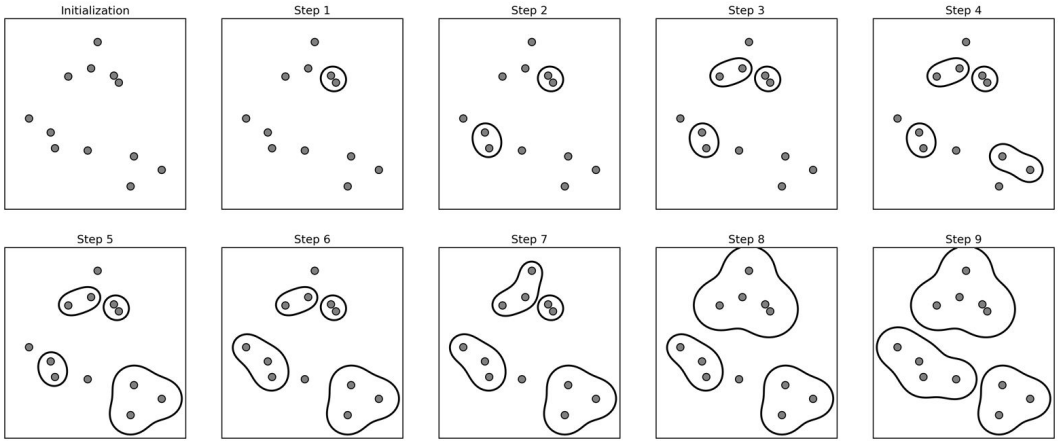
Hyperparameter:

- distance metric d: measures the distance between two points (e.g . Euclidean distance)
- dissimilarity metric D: measures the difference between two clusters based on the distances d between the members of each cluster (e.g. complete-linkage method → max. distance across all pairs of records)

Example

Start:
Each point is its own cluster.

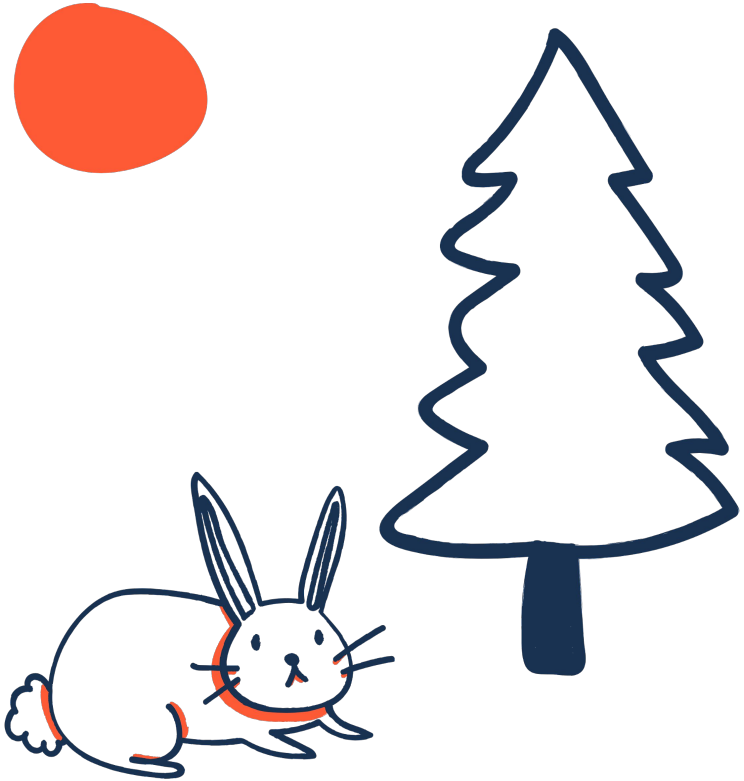
Iterate:
Merge similar clusters until a specified # of clusters is reached.



Visualising the result and finding a good number of clusters

Dendrogram:

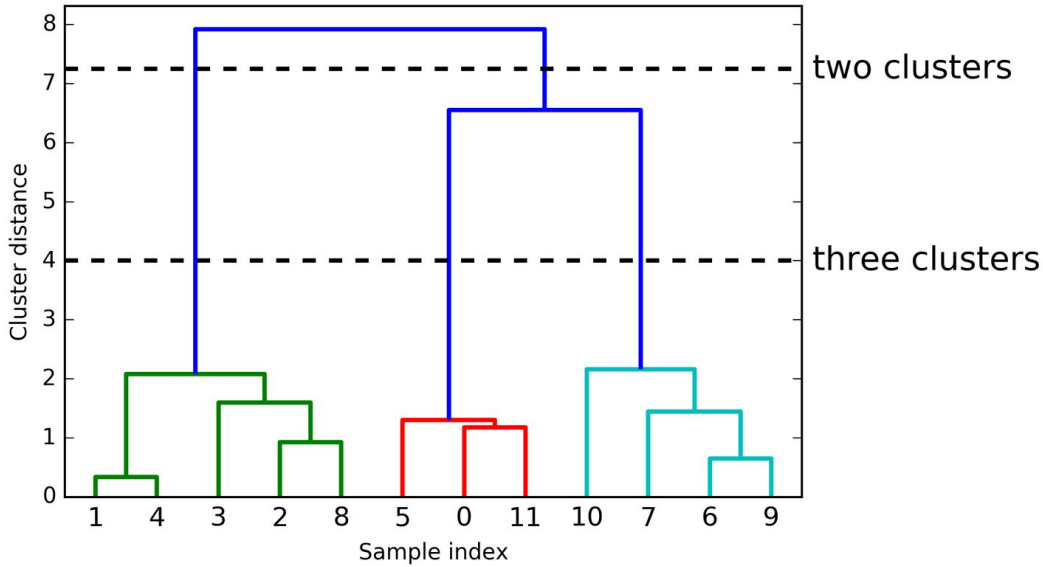
- visual representation of records and hierarchy of clusters to which they belong



Dendrogram

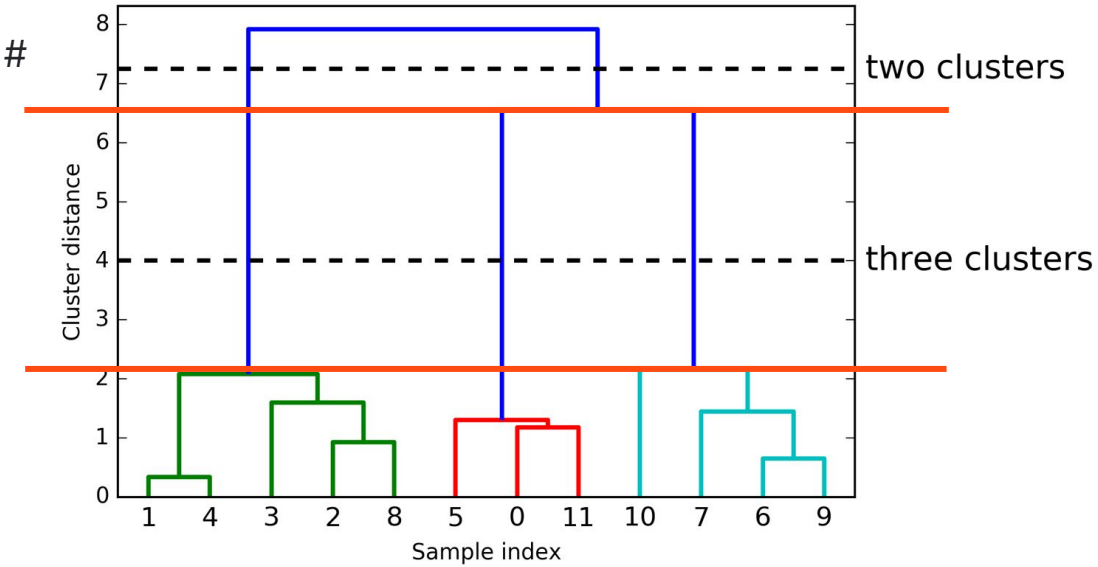
- Visualization of hierarchical clustering
- Shows order in which samples are clustered together (reading from bottom to top)

x-axis: sample numbers
y-axis: dissimilarity of clusters



How to find the best # of clusters?

→ optimal # of clusters:
largest vertical distance with same # of clusters



Pros and Cons

Advantages

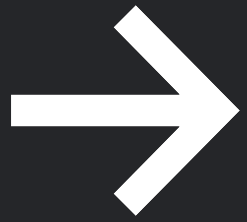
- interpretability
- easy to execute, understand and explain
- robust

Disadvantages

- not always able to capture complex structure
- does not scale well with large data sets
- cannot handle missing data
- greedy algorithm that might not be globally optimal

neue fische
School and Pool for Digital Talent

DBSCAN



DBSCAN

Density-**b**ased **s**patial **c**lustering of **a**pplications with **n**oise

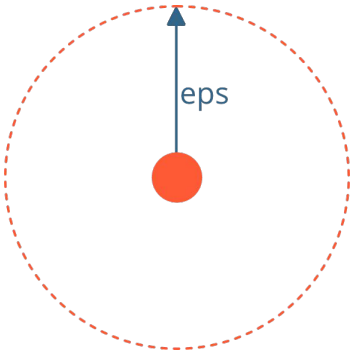
Idea:

- clusters are dense regions which are separated by areas that are relatively empty
- identifies points that don’t belong to any cluster (aka noise)

Core Notions

Hyperparameters:

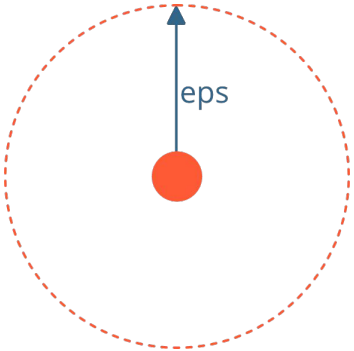
- eps (epsilon, max distance between points)
- min_samples (of a cluster)



Core Notions

Hyperparameters:

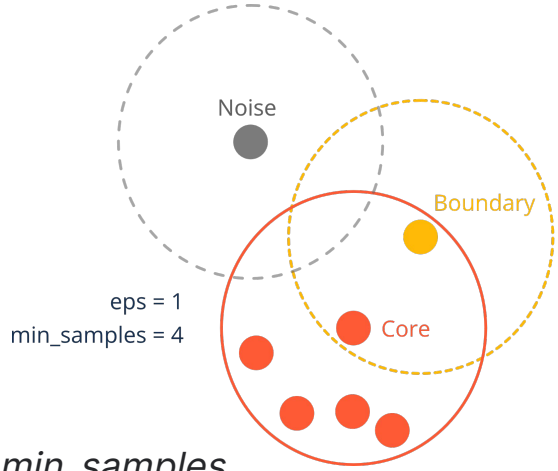
- eps (epsilon, max distance between points)
- min_samples (of a cluster)



min_samples = 5 ✓

min_samples = 6 ✗

Core Notions



Types of points:

- Core point:
  → if number of neighbouring points within distance *eps* are ≥ *min_samples*

- Boundary point:
  → if number of neighbouring points within distance *eps* are < *min_samples*
  → if point is in neighbourhood of a core point (within distance *eps*)

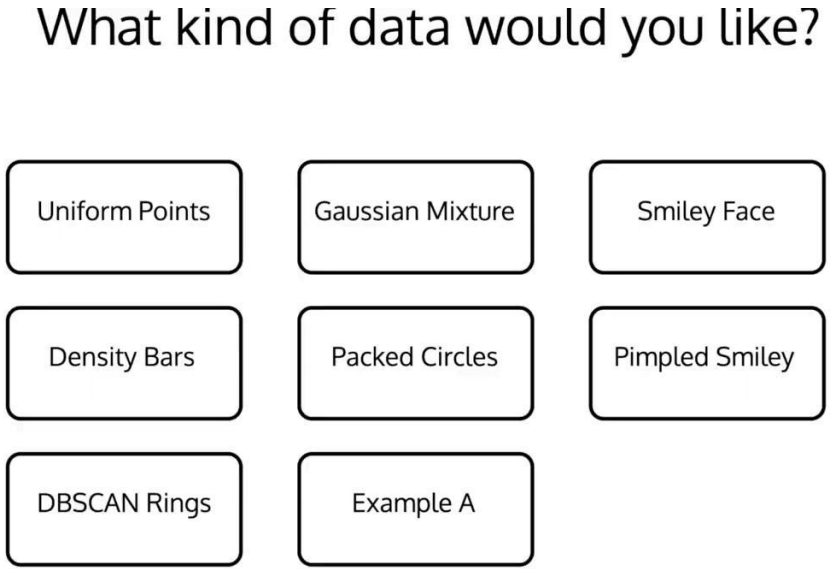
- Noise (outlier):
  → neither core nor boundary point
  → not reachable from any other point which is assigned to a cluster

DBSCAN in action

Define hyperparameters.

Start by arbitrarily choosing a starting point.

Visit every point to determine its type and create clusters.

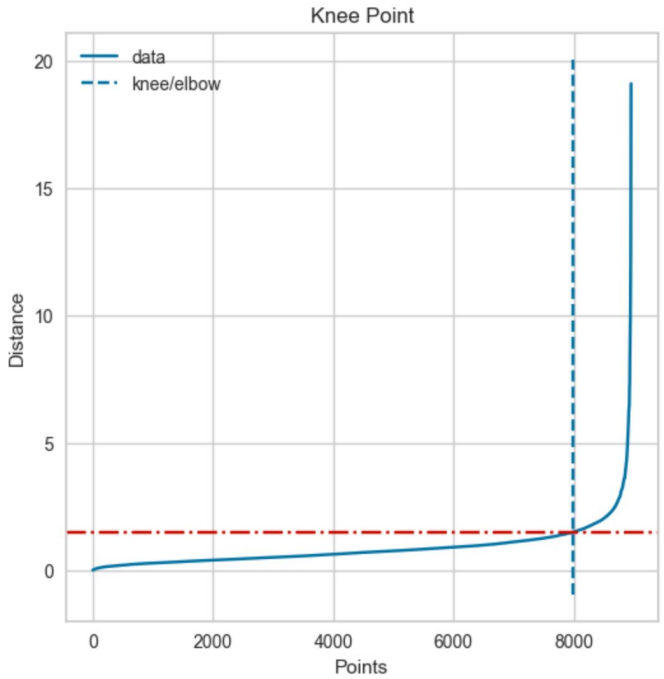


Animation by Naftali Harris

Approach to determine starting epsilon

A kind of elbow method:

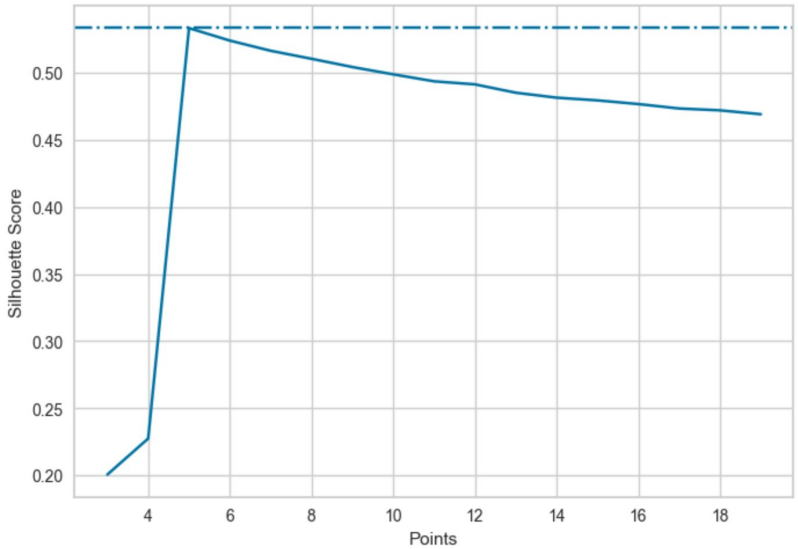
- Calculate distance of each point to its nearest neighbor (sklearn NearestNeighbor)
- the output is sorted and plotted
- the best epsilon is found at the point of maximum curvature (the elbow)
- the y-value at the elbow is an epsilon to start with



Approach to determine starting min_samples

Silhouette Scores:

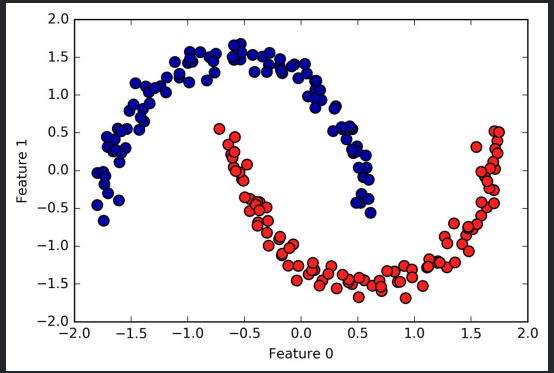
- With a starting epsilon, a (large) range of points can be fed through a for-loop calculating the Silhouette Score for the parameter.
- These can be evaluated or plotted for the number of points corresponding to the highest Silhouette Score

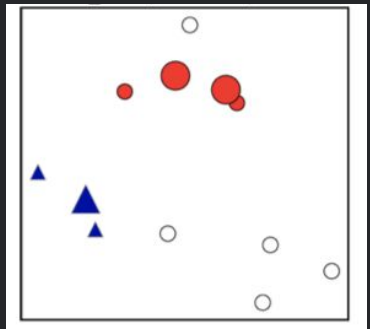


Pros

Advantages

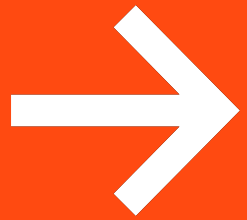
- identifies automatically # of clusters
- able to identify complex clusters
- able to identify clusters with very different sizes
- distinguish between clusters (dense data) and noise





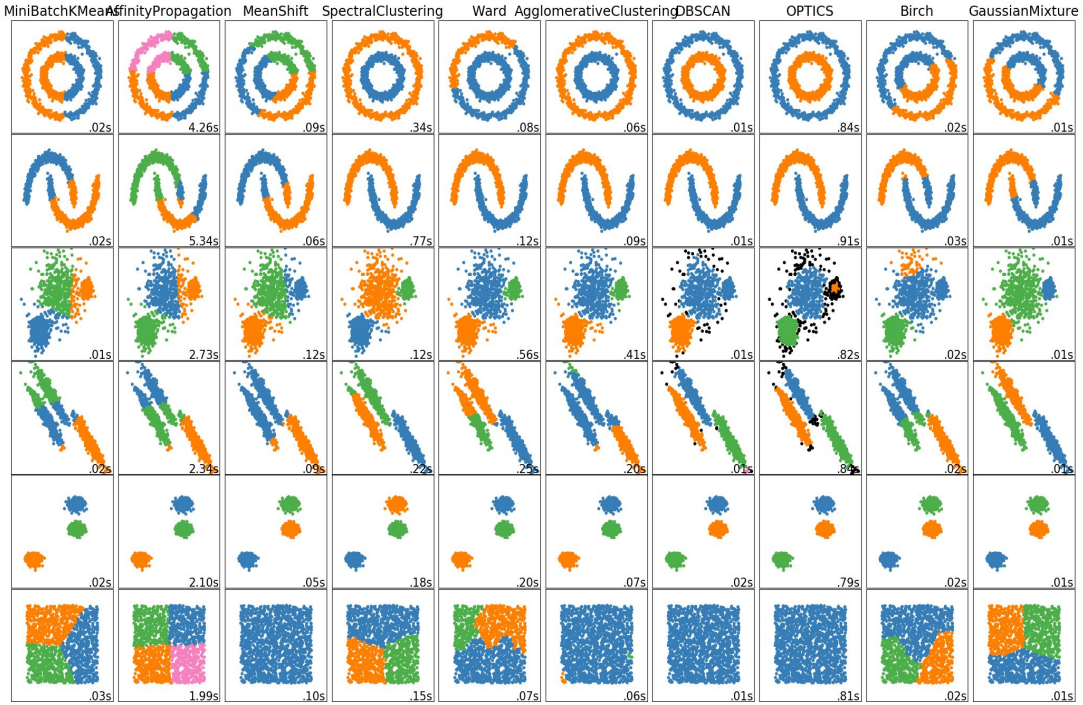
neue fische
School and Pool for Digital Talent

Hands-on Exercise

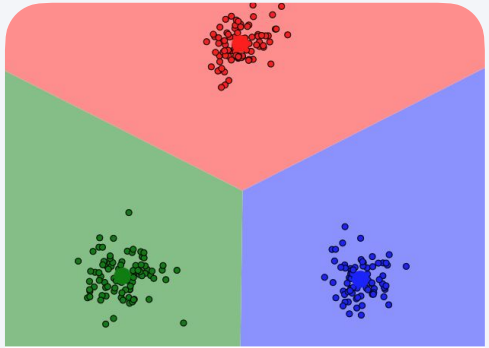


Sklearn Summary

Great [Documentation] on scikit-learn!



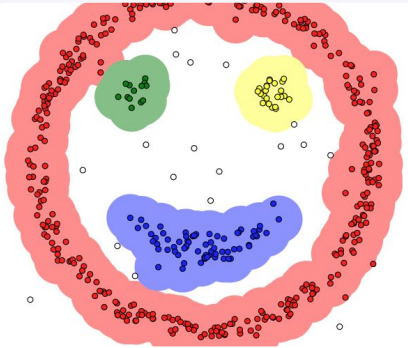
Visualisation of K-Means and DBSCAN by Naftali Harris



K-Means

If you want to see K-Means in action check out this amazing blog by Naftali Harris.

K-Means in action!



DBSCAN

There is also a fantastic animation of how DBSCAN works

DBSCAN in action!

neue fische
School and Pool for Digital Talent

Table of content

References

- https://www.scikit-yb.org/en/latest/api/cluster/elbow.html
- https://en.wikipedia.org/wiki/DBSCAN
- Practical Statistics for Data Science
- https://www.naftaliharris.com/blog/visualizing-k-means-clustering/
- https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/
- https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html?highlight=silhouette%20coefficient
- https://stories.opengov.com/sandpointid/published/DbL6VqbV4
- https://en.wikipedia.org/wiki/Davies%E2%80%93Bouldin_index