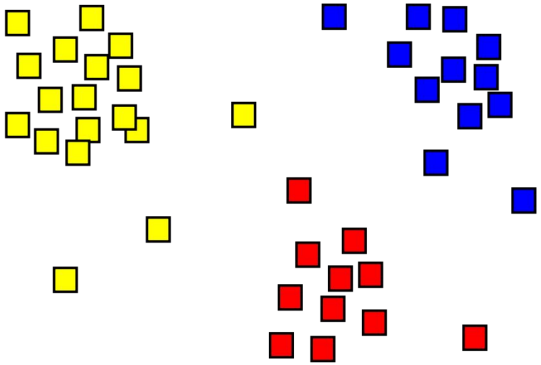
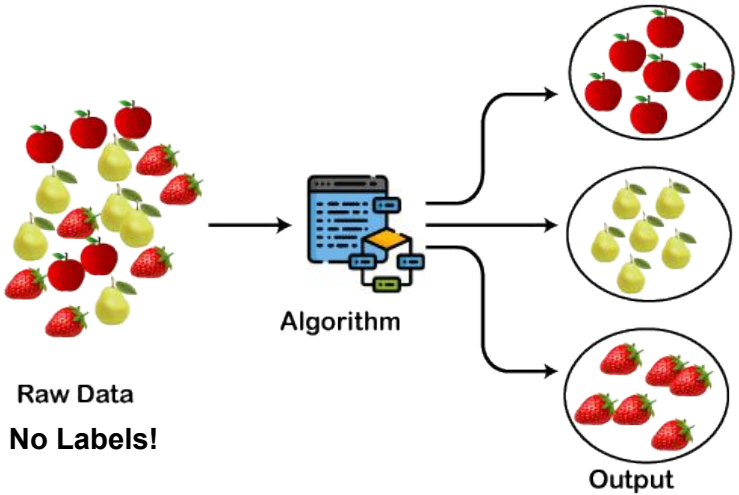
Group 4: Clustering

Presentation 1 on June 9th 2023
Lab Course “Introduction to Computing in Data Science”
Summer Term 2023, University of Leipzig





[1] Image source: https://en.wikipedia.org/wiki/Cluster_analysis#/media/File:Cluster-2.svg
[2] Image source: https://www.javatpoint.com/clustering-in-machine-learning

Introduction to Clustering

- identify structures in datasets, that are called “clusters”
  - points in the *same cluster* should be *close or similar* to each other
  - *different clusters* should be *separable or different* from each other
- unsupervised learning
  - results can be hard to evaluate and “subjective”

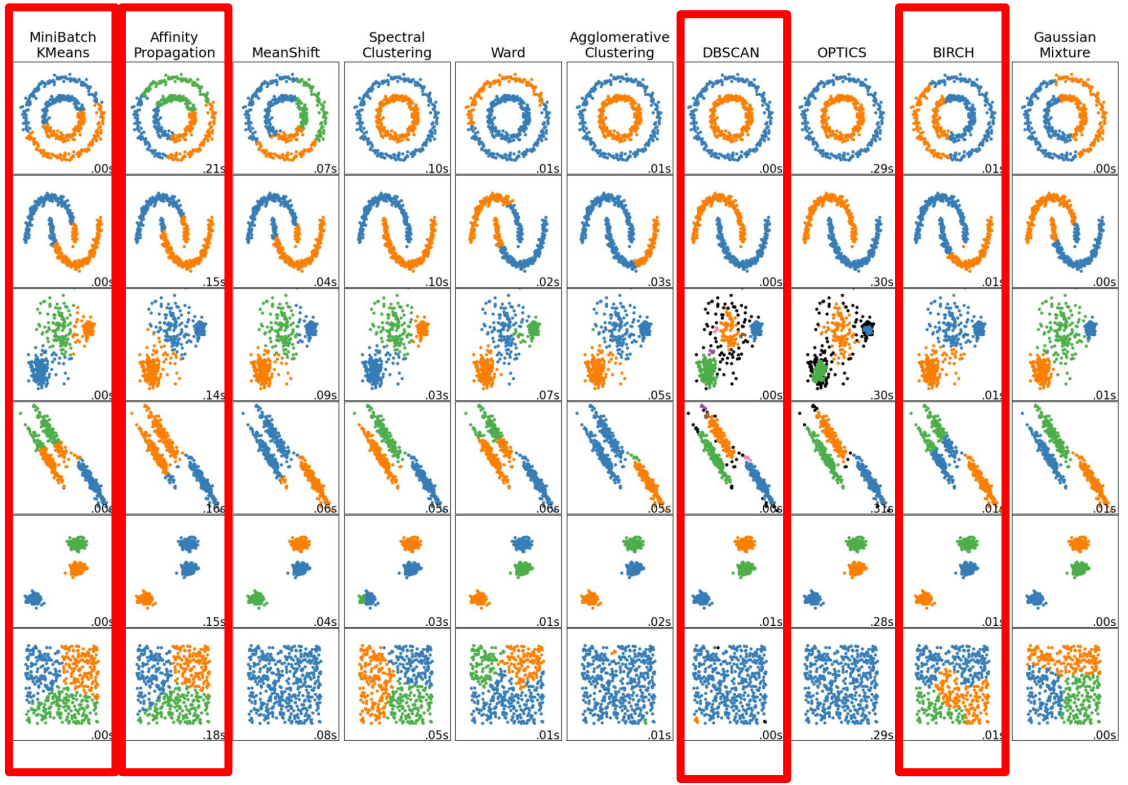
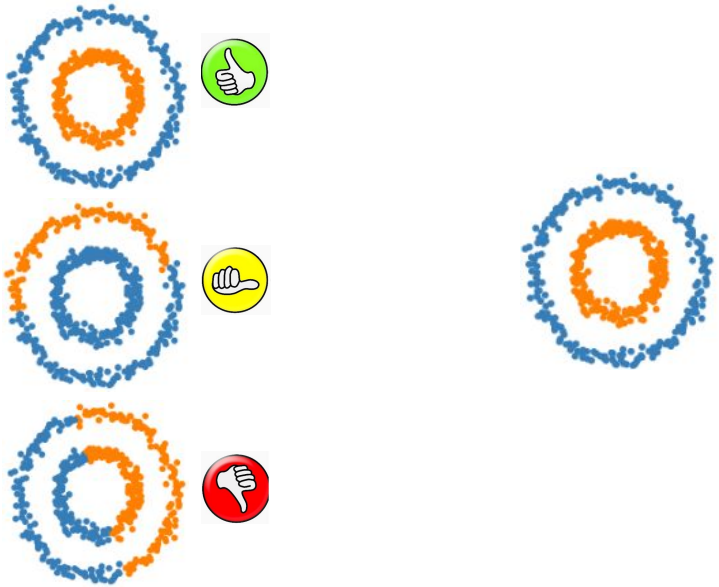


Image from: https://scikit-learn.org/stable/modules/clustering.html

Clustering Metrics and Evaluation

Ground-truth IS available:
Extrinsic measures: Compare Ground truth

Clustering Result **vs.** **Ground Truth**



Ground-truth NOT available:
Intrinsic measures
minimize intra-cluster distance AND
maximize inter-cluster distance

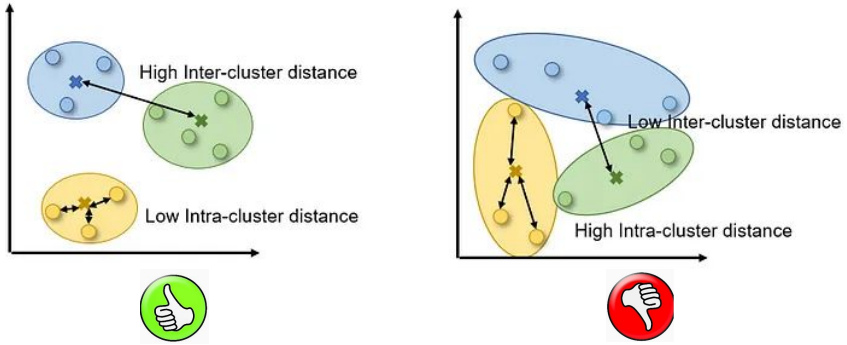
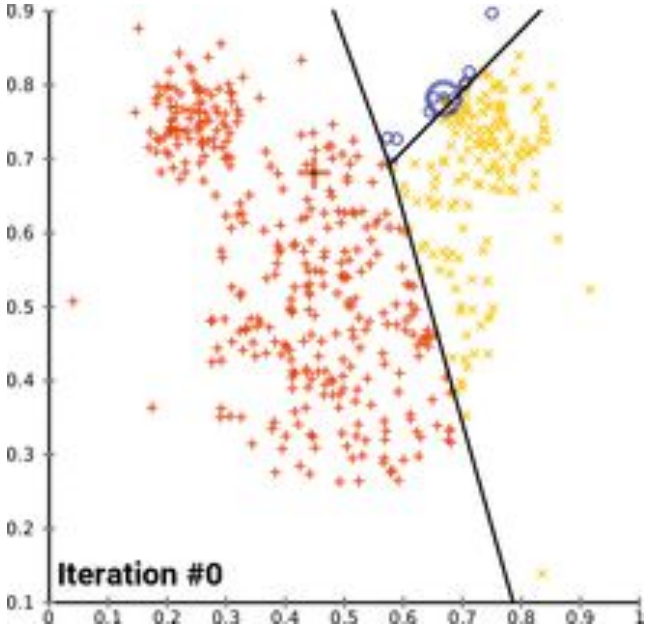


Image from https://medium.com/@jodancker/a-brief-introduction-to-cluster-validation-ca4215295b06

K-Means Algorithm

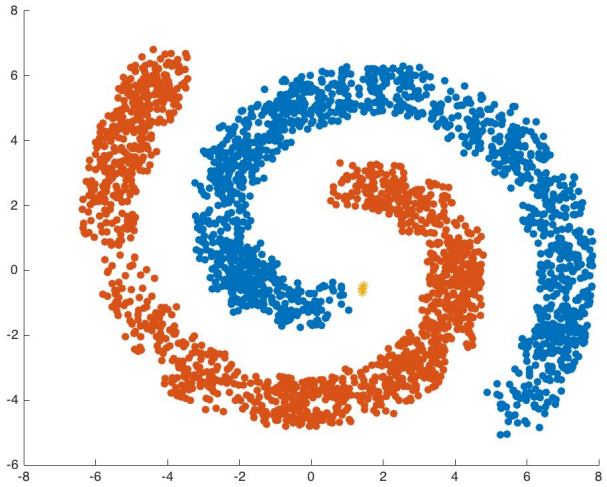
- Proposed by different authors in varying forms (Lloyd 1957/1982, Forgy 1965, McQueen 1967)
- Implemented in scikit-learn in multiple ways [1] (K-Means, MiniBatch K-Means, Bisecting K-Means)
- Parameters:
  - ***K: (given number of clusters)***
  - *optional: initial centroids*
  - *[more implementation-dependent parameters]*
- Best results with flat geometry and clusters of similar diameter
- usually scalable and fast



[1] https://scikit-learn.org/stable/modules/clustering.html#k-means
Animation from: https://de.wikipedia.org/wiki/K-Means-Algorithmus#/media/Datei:K-means_convergence.gif

Density Based Spatial Clustering of Applications with Noise (DBSCAN)

- Proposed by Ester et al. in 1996 [1]
- Implemented in scikit-learn [2]
- Distance-based clustering
- Parameters: *Eps* and *MinPts*
- Independent of cluster shapes and number of clusters
- Points that are not part of a cluster are considered as Noise (Outlier Detection)

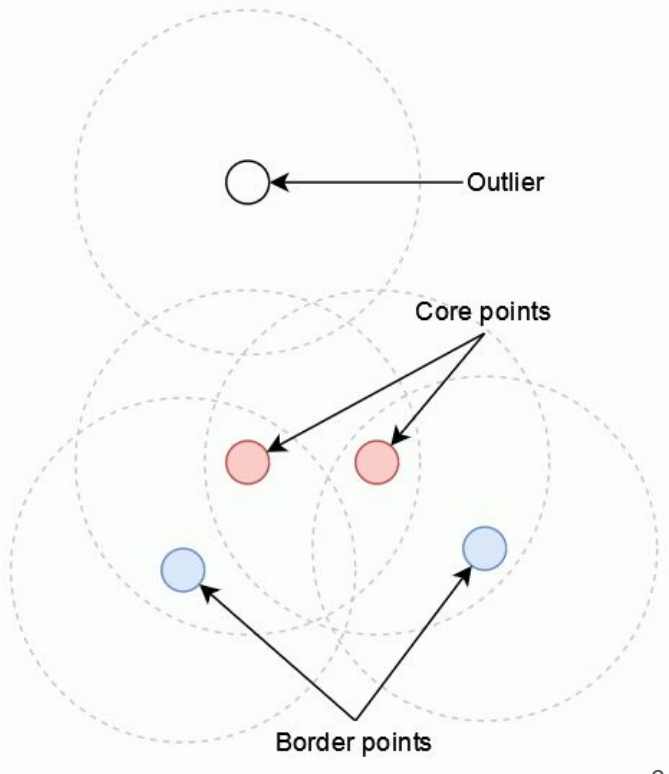


[1] M. Ester, H. Kriegel, J. Sander and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.
[2] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

Density Based Spatial Clustering of Applications with Noise (DBSCAN)

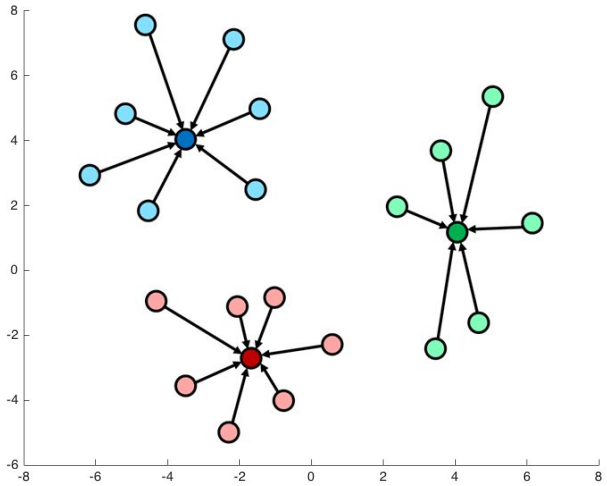
Concept:

- A point *p* is a core point if there are at least *MinPts* other points **directly density-reachable** from *p*, i. e., within a distance of *Eps*
- Points *p* and *q* are **density-reachable** if there is a path of directly density-reachable points from *p* to *q*
- Points *p* and *q* are **density-connected** if there is a core point *o* from which both *p* and *q* are density-reachable
- Cluster consists of all points that are either density-reachable or density-connected to each other



Affinity Propagation

- Proposed by Frey & Dueck in 2007 [1]
- Implemented in scikit-learn [2]
- Exemplar clustering
- Parameters: *Damping factor, preferences*



Advantages

- Number of clusters chosen from data
- Deterministic algorithm
- No need to pre-assign exemplars

Disadvantages

- Quadratic complexity with respect to number of points: $O(|P|^2 * T)$
  (T = Iterations until convergence)

[1] B. J. Frey, D. Dueck. Clustering by passing messages between data points. In: *Science* 2007;315(5814):972-976. .
[2] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AffinityPropagation.html

Affinity Propagation

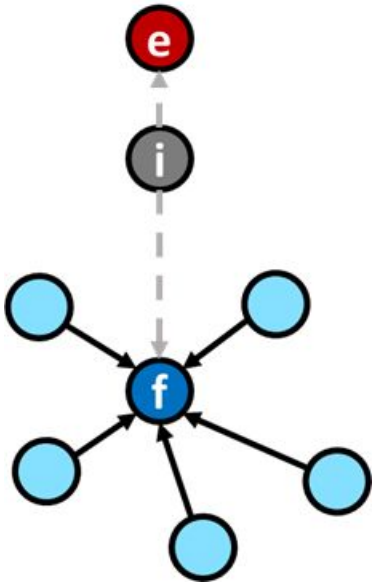
Exemplars are chosen based on iterative messaging until convergence

- **Responsibility** of point *e* to be exemplar of point *i*
  - ↑ High similarity of *e* and *i*
  - ↓ High availability of another potential exemplar with high similarity

  On the right, e has a low responsibility, because f has high availability

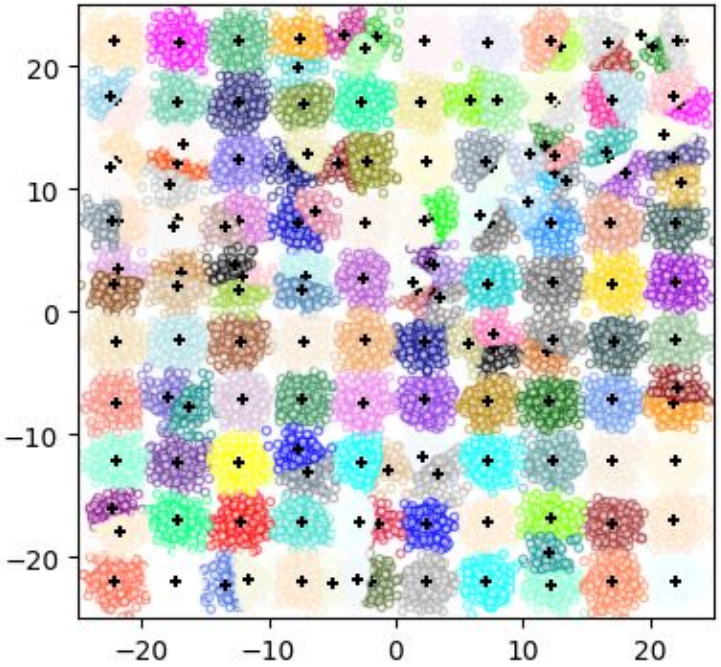
- **Availability** of point *e* to be the exemplar of point *i*
  - ↑ High responsibility of *e* to itself and/or other points
  - ↓ Low responsibility of *e* to itself and/or other points

  On the right, e has a low availability, because it has a low responsibility to other points



BIRCH

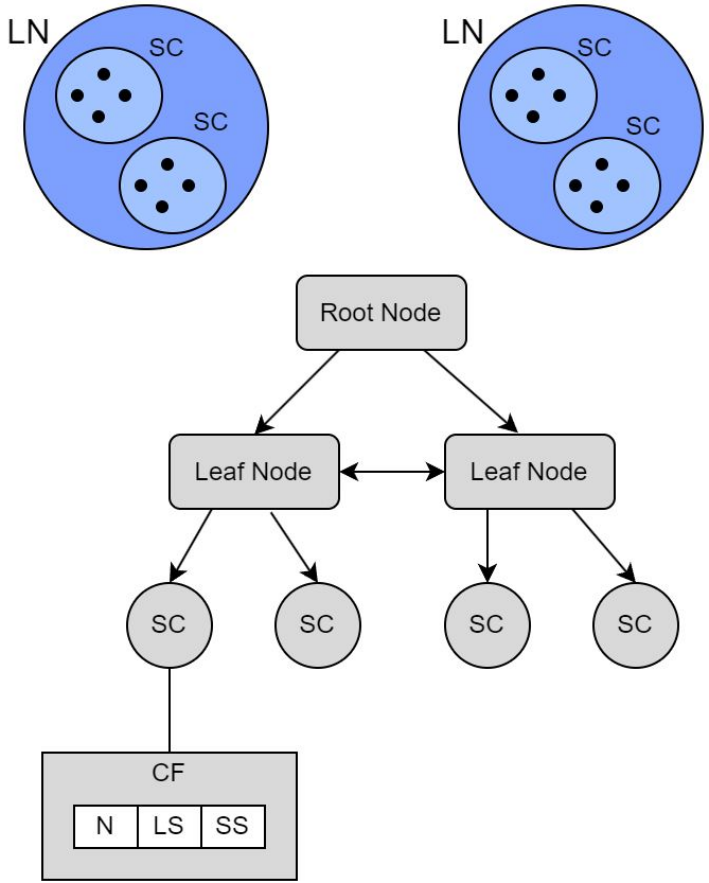
- BIRCH = **B**alanced **I**terative **R**educing and **C**lustering using **H**ierarchies
- Proposed by Zhang et al. in 1996 [1]
- Implemented in scikit-learn [2]
- Euclidean distance between points
- Saves cluster information in a tree structure
- Parameters: *branching_factor (B), threshold (T)* and *n_clusters*
- Outlier Detection



[1] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. 1996. BIRCH: an efficient data clustering method for very large databases. SIGMOD Rec. 25, 2 (June 1996), 103–114. https://doi.org/10.1145/235968.233324
[2] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.Birch.html

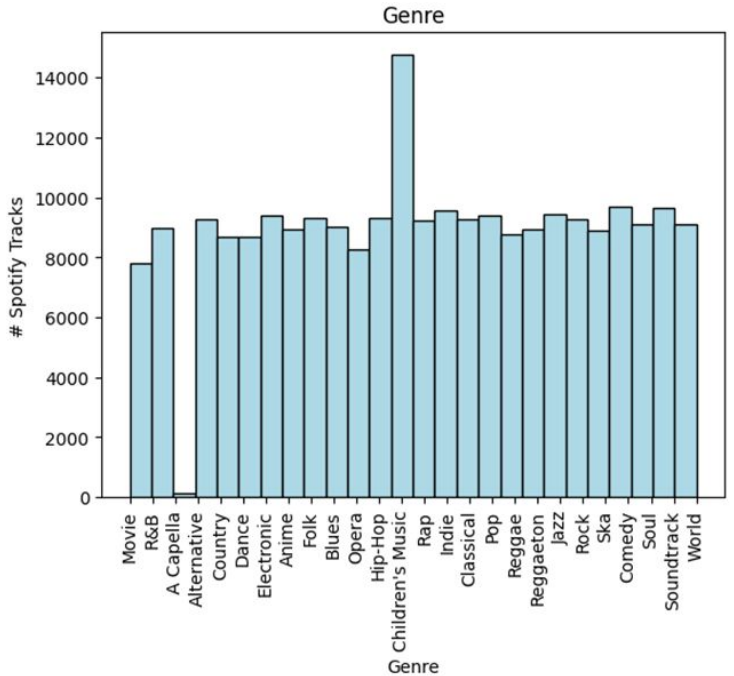
BIRCH

- CF (Clustering Feature):
  - ***N*** = Number of Data Points
  - ***LS*** = Linear Sum of the *N* data points
  - ***SS*** = Square Sum of the *N* data points
- CF Tree:
  - Each **non-leaf node** contains at most *B* entries
  - Each **leaf node** contains at most *L* entries
  - **non-leaf node** entries have a CF triple and a *child* pointer to their corresponding child node
  - **Leaf nodes** have *prev* and *next* pointer to link all leaf nodes together
  - **Leaf node entries** are the subclusters which consist of CF
  - All **entries** in a **leaf node** must satisfy the threshold requirement *T*, which is a Radius



Dataset: Spotify Tracks

- Datasource: Spotify Tracks DB provided by kaggle [1]
- DB size: 232,725 tracks
- 26 different genre
- ~8,000 tracks per genre
- 18 Feature: genre, track, artist
  - **confidence measures**: acousticness, instrumentalness, ..
  - **perceptual measures**: energy, loudness, ..
  - **descriptors**: duration, tempo, ..



[1] https://www.kaggle.com/datasets/zaheenhamidani/ultimate-spotify-tracks-db

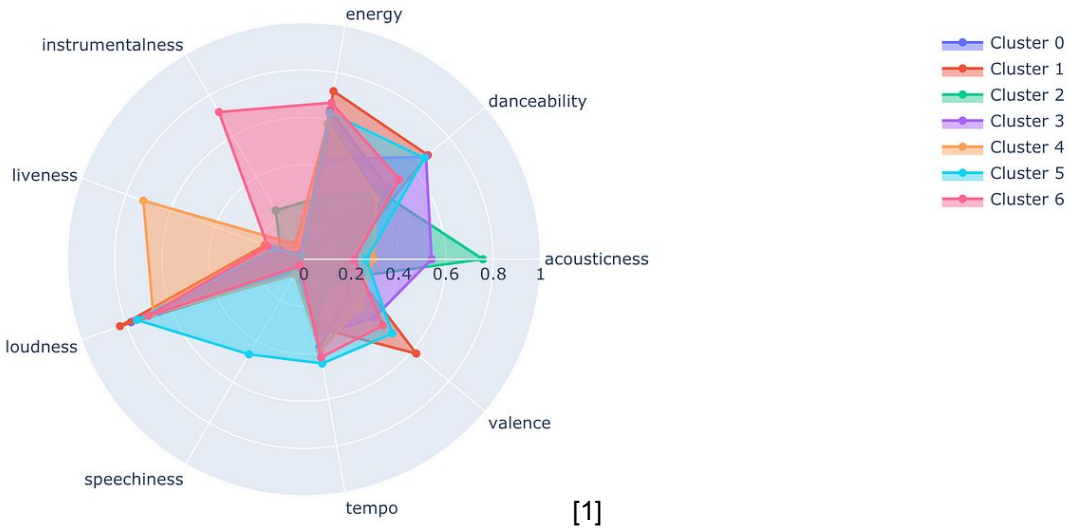
Clustering of our Data

Goal: Clustering music tracks by similar audio features

Main questions:

How well do our clusters match the music genre given by the database?

Which algorithm performs the best on our dataset?



[1]

[1] https://towardsdatascience.com/k-means-clustering-and-pca-to-categorize-music-by-similar-audio-features-df09c93e8b64

Workflow with Github

Work is organized using built-in features of GitHub:

Branches and Pull Requests

Group’s main branch is protected

To merge changes into the main branch, the Pull Request must be approved by the team leader

Issues

Tasks are tracked as Issues

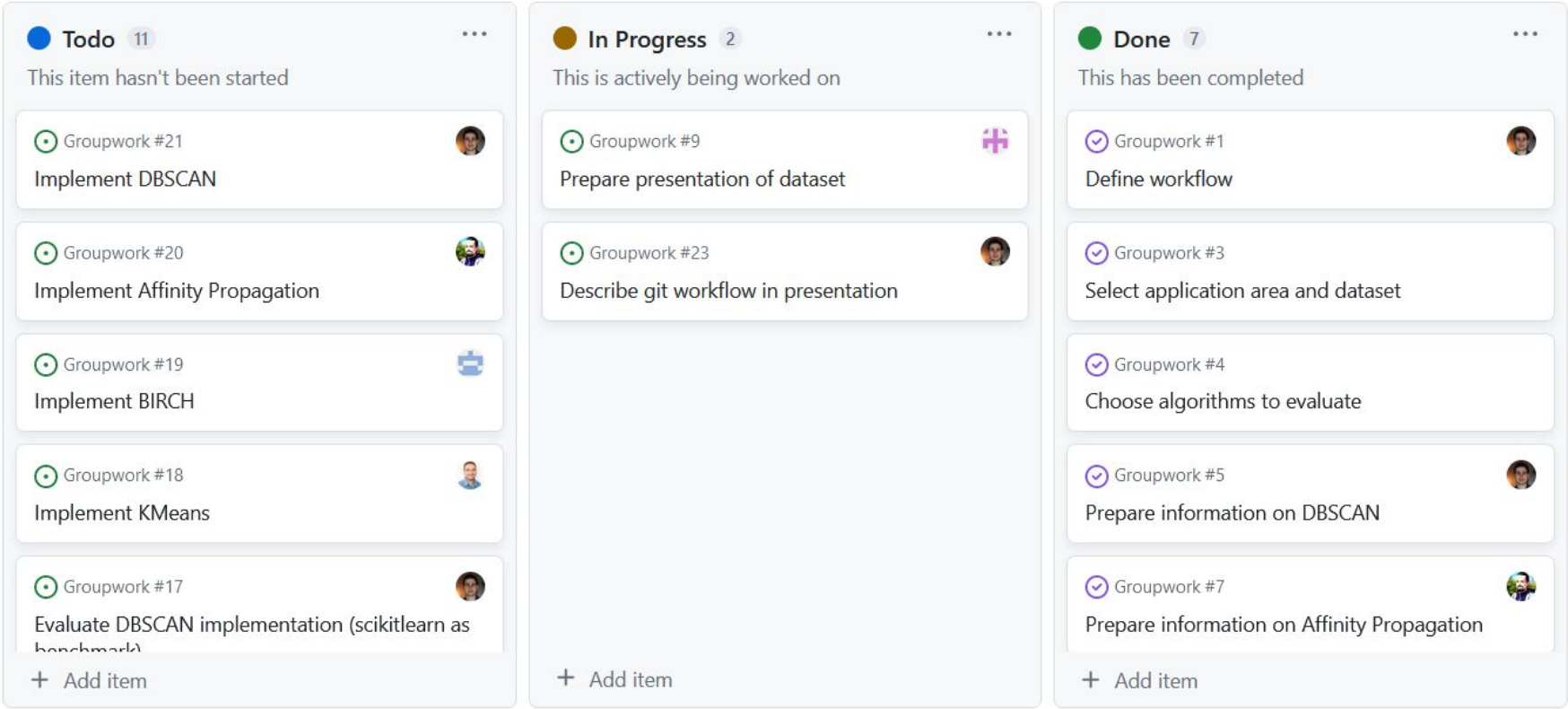
Issues should include a short description and acceptance criteria

Issue template is available

Projects

Kanban board is used to organize tasks and track the project’s progress
(see next slide)

Kanban board



Backup

Clustering Metrics and their Properties

	Measure	Range (Basis for comparison)	Interpretable	Adjusted for Chance	No assumption on cluster structure
Rand Index	Extrinsic	[-1, 1] for ARI	Yes	ARI	Yes
Mutual Information	Extrinsic	Upper bound 1		AMI	
V-measure	Extrinsic	[0, 1]	Yes	No	Yes
Fowlkes-Mallows Scores	Extrinsic	Upper bound 1		Yes	Yes
Silhouette Coefficient	Intrinsic	[-1, 1]	Yes		No, better on density-based algorithms
Calinski-Harabasz Index	Intrinsic				No, better on density-based algorithms
Davies-Bouldin Index	Intrinsic		Yes		No, better on density-based algorithms

Table from https://towardsdatascience.com/7-evaluation-metrics-for-clustering-algorithms-bdc537ff54d2

Affinity Propagation - Relevant equations

$i, j \in P$ Points to be assigned to an exemplar

$e, f \in P$ Potential exemplars

$s(i,e)$ Similarity of points *i* and *e*. Example: Negative squared Euclidean distance

$$S = \left(s(i,e)\right)_{i,e=1}^{|P|} \in \mathbb{R}^{|P|\times|P|}, \qquad s(i,e) = -\left(||i-e||\right)^2$$

Responsibility (Initialized with 0's)

$$r(i,e) \leftarrow s(i,e) - \max_{f \neq e}\{a(i,f) + s(i,f)\}$$

$$R = \left(r(i,e)\right)_{i,e=1}^{|P|} \in \mathbb{R}^{|P|\times|P|}$$

Availability (Initialized with 0's)

$$a(i,e) \leftarrow \min\left\{0, r(e,e) + \sum_{j \neq i,e} \max\{0, r(j,e)\}\right\}$$

$$a(e,e) \leftarrow \sum_{i \neq e} \max\{0, r(i,e)\}$$

$$A = \left(a(i,e)\right)_{i,e=1}^{|P|} \in \mathbb{R}^{|P|\times|P|}$$