Clustering - Group 4

Final presentation - July 12th 2023

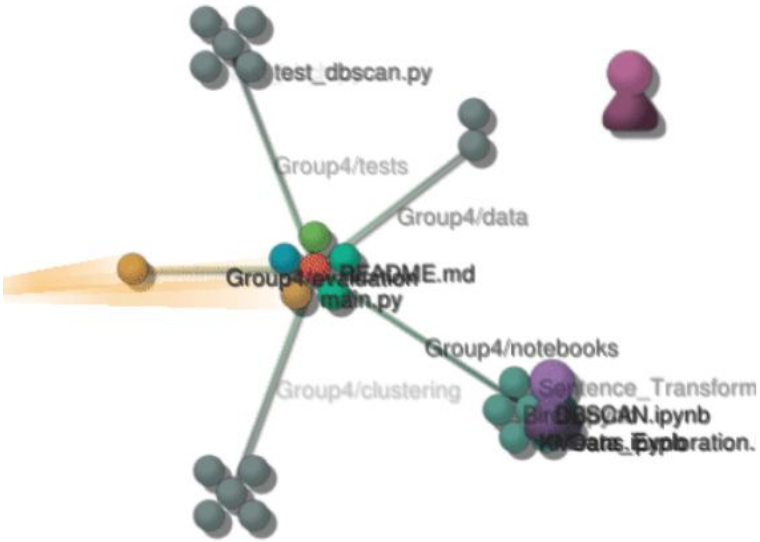
Lab Course “Introduction to Computing in Data Science”
at University of Leipzig

Summer Term 2023

Project approach

- Workflow
  - Create Issue for each task
  - Push changes to feature branch
  - Pull Requests

    Kanban board and Discord
  - Presentations on Google Slides
- Phases
  - Implementation of clustering algorithms
  - Data preparation and exploration
  - Evaluation of algorithms
  - Visualization of results



Animation created with https://gource.io/

Project approach

Phase 1: Implementation of clustering algorithms

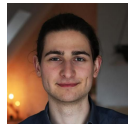
- Test-driven development approach
- Verify results with sklearn implementation

```python
from clustering.dbscan import dbscan

import pytest


@pytest.mark.parametrize("n_centers", [1, 2, 4, 8])
@pytest.mark.parametrize("n_features", [2, 3, 5, 10])
@pytest.mark.parametrize("epsilon", [0.5, 1, 2, 3])
def test_dbscan_compare_results(n_centers: int, n_features: int, epsilon: float):
    # Generate dataset with `n_centers` blobs (circular clusters)
    X, _ = make_blobs(n_samples=n_centers * 70, centers=n_centers, n_features=n_features)

    # Run own implementation and scikit-learn version with the same parameters and compare the results
    _, computed_labels = dbscan(X, epsilon=epsilon, min_points=2 * n_features)
    _, expected_labels = sklearn_dbscan(X, eps=epsilon, min_samples=2 * n_features)
    assert_array_equal(computed_labels, expected_labels)
```

Contributions:

Sebastian
(Affinity Propagation)

Marcel
(BIRCH)



Florian
(DBSCAN)

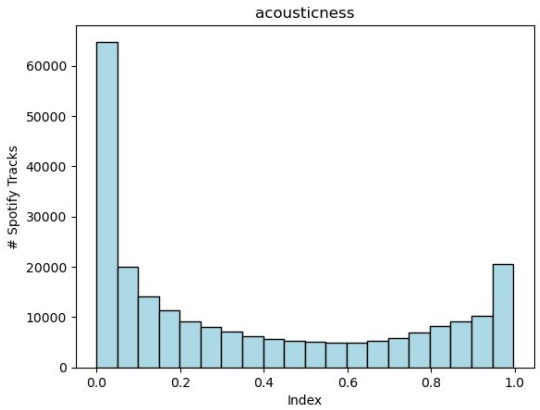
Project approach

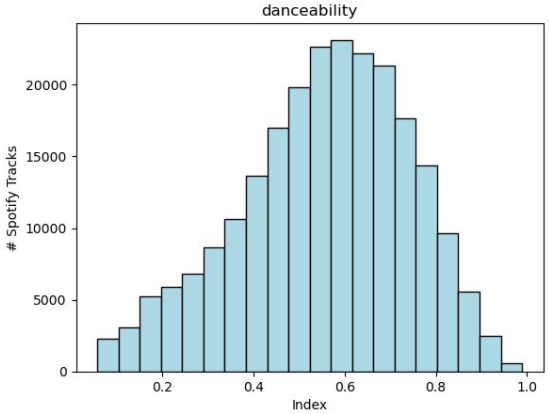
Phase 2: Data preparation and exploration

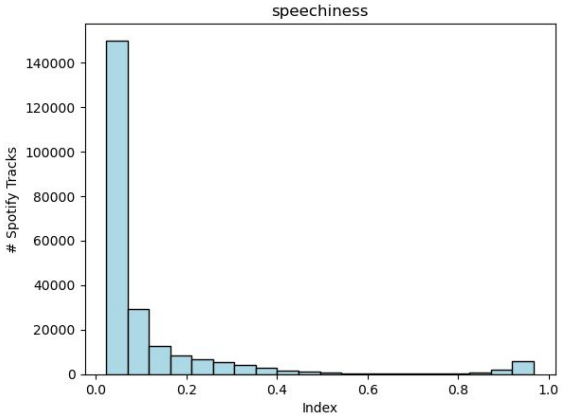
- Load and preprocess dataset
- Visualize attributes in Jupyter notebook

Contributions:

Clara







Project approach

Phase 3: Evaluation of clustering algorithms

- Implement evaluation metrics
- Find optimal parameters for each algorithm
- Evaluate clustering results

Phase 4: Visualization of results

- More on that later

Contributions:

Sebastian
(Affinity Propagation)

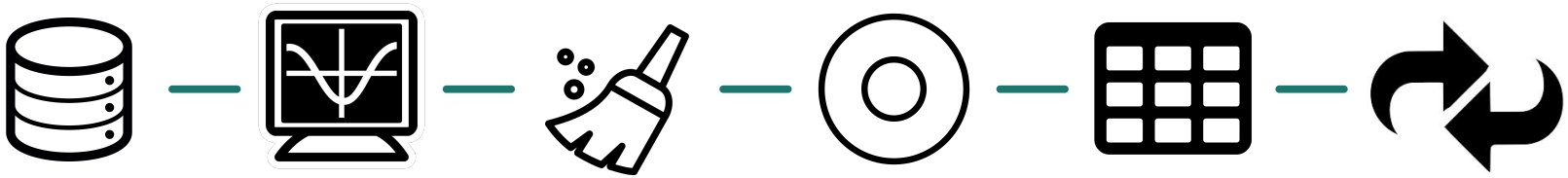
Marcel
(BIRCH)

Florian
(DBSCAN)

Ralf
(k-Means, Eval. metrics)

Clara
(Visualizations)

Dataset Preparation



Spotify Tracks DB

kaggle.com

Data Discovery

18 features

26 music genres

232,725 tracks

~8,000 tracks per genre

Data Cleaning

remove duplicates

merge same category (‘Children’s Music’)

exclude ‘A Capella’ genre (119 samples)

Sampling

500 tracks per genre

total: 12,500 samples

Feature Selection

10 features:
acousticness, danceability, duration, energy, …

excluded:
track_id, track_name, artist_name, ..

Transformation

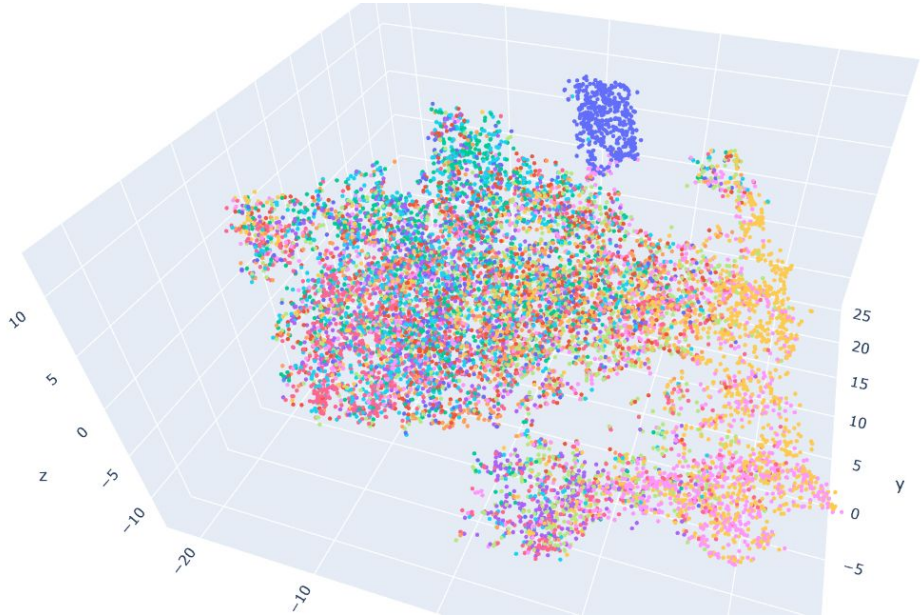
Standardization of X with MinMaxScaler

Encoding of genre category to numeric label with LabelEncoder

Visualizations of Ground Truth

Dimensional Reduction to 3D by applying

t-Distributed Stochastic Neighbor Embedding (t-SNE)

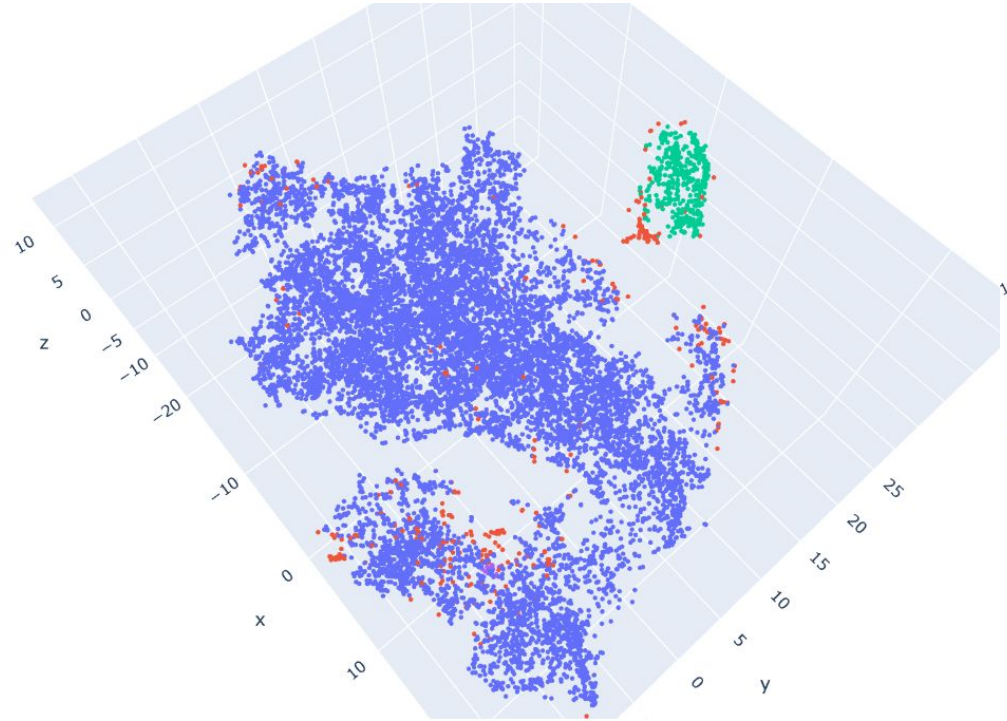


<u>Outcome:</u>

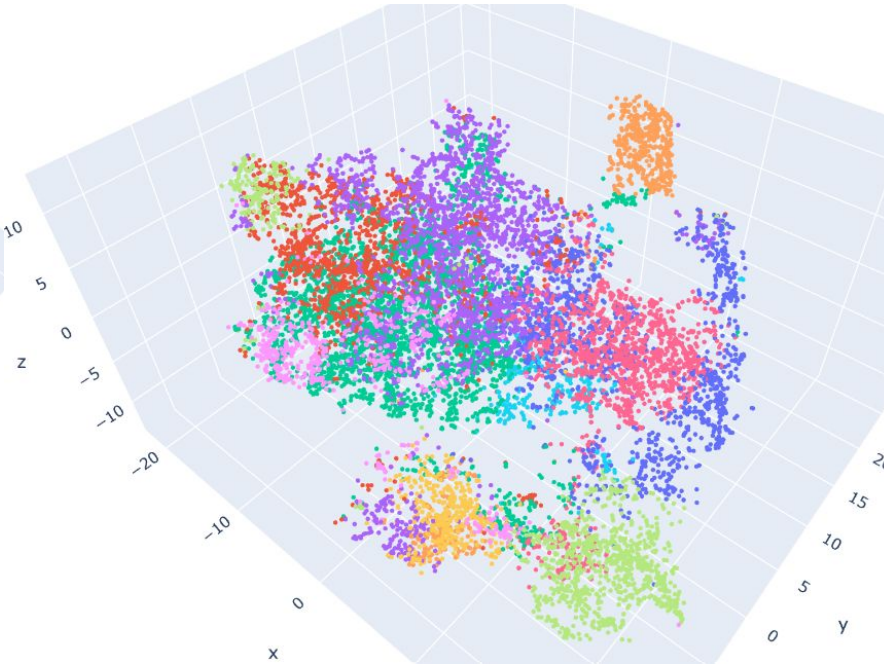
Data points show no clear discernible clusters in 3D visualization -
except for genre ‘Comedy’ (blue)

Visualizations of Clustering Algorithms

DBSCAN

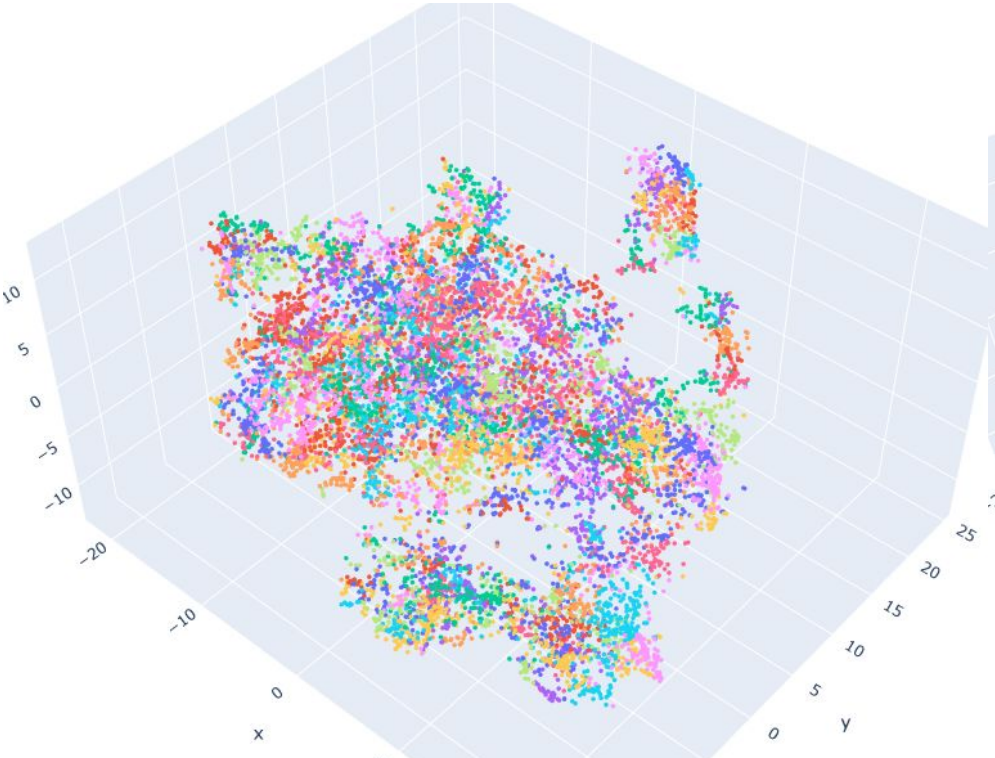


BIRCH

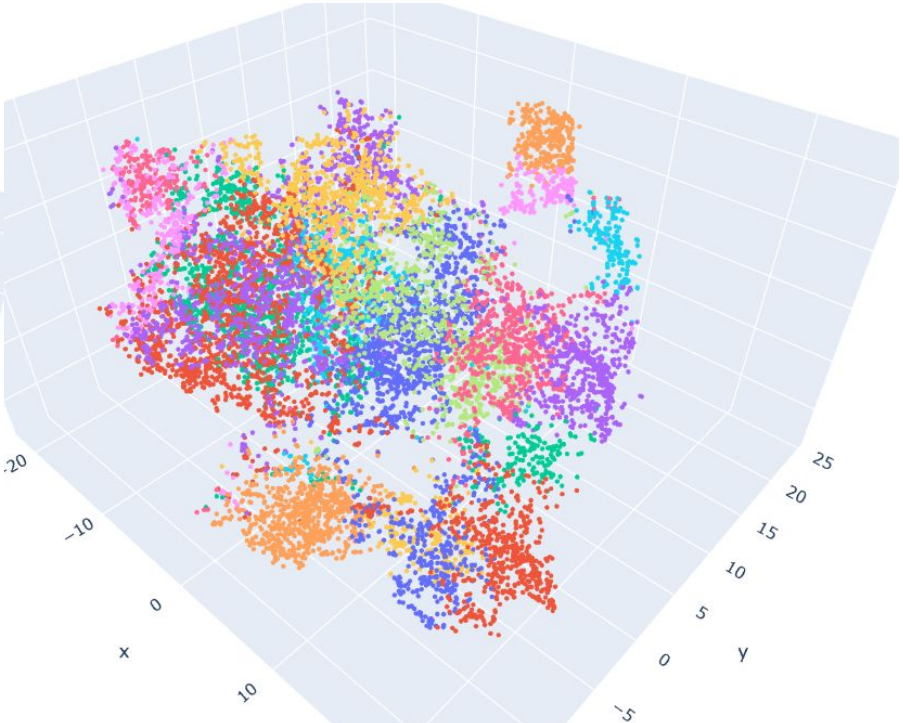


Visualizations of Clustering Algorithms

Affinity Propagation



KMeans



Clustering Evaluation

- on Spotify song dataset
- 200 sample songs per genre x 25 genres (n = 200*25 = 5000)
- 10 numeric features (d=10)
- compare with ground truth from genres
- adjusted RAND metric (-1 to 1) takes into account chance

- k-Means(n_clusters=25, init="auto")
- DBSCAN(min_samples=20, eps=.324)
- BIRCH(n_clusters=25, branching_factor=50, threshold=.25)
- AffinityPropagation(damping=.7, convergence_iter=20, max_iter=200)

Algorithm	Lib	RAND	Adj. RAND	Mutual Information	Adj Mut Info	Norm Mut Info	Homogeneity	Completeness	V-Measure	Fowlkes-Mallows
K-Means	SK	0,92	0,10	0,83	0,25	0,26	0,26	0,27	0,26	0,14
MiniBatch ...	SK	0,93	0,11	0,82	0,25	0,26	0,26	0,26	0,26	0,15
Bisecting ...	SK	0,92	0,11	0,79	0,24	0,25	0,25	0,26	0,25	0,15
DBSCAN	SK, G4	0,34	0,01	0,17	0,09	0,09	0,05	0,28	0,09	0,18
BIRCH	SK	0,90	0,08	0,74	0,23	0,25	0,23	0,26	0,25	0,14
	G4	0,90	0,10	0,80	0,25	0,26	0,25	0,28	0,26	0,15
Affinity Propagation	SK, G4	0,96	0,04	1,23	0,22	0,30	0,38	0,25	0,30	0,07

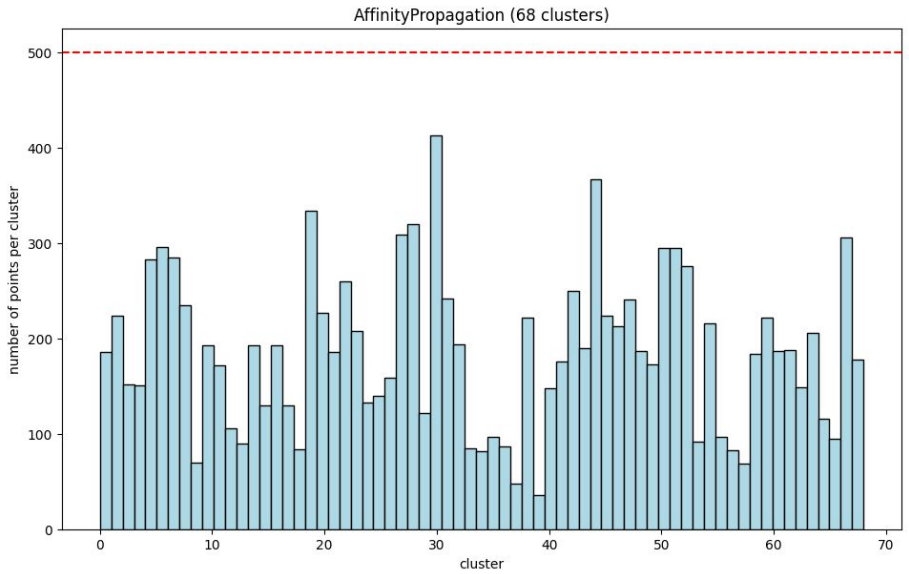
Runtime for this example

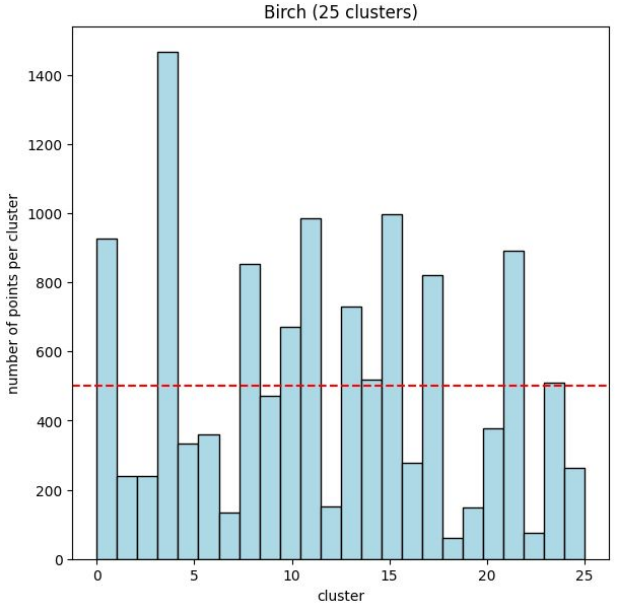
- same scenario as last slide
- with scikit-learn-intelex for Intel extensions (acceleration on CPU)
- CPU: Intel i5-8265U (Notebook 2018), RAM: 8 GB, OS: Linux

Algorithm	Lib	Runtime (ms)	Factor
K-Means	SK	18	1
MiniBatch ...	SK	103	6
Bisecting ...	SK	89	5
DBSCAN	SK	28	2
	G4	1.202	65
BIRCH	SK	265	14
	G4	269	15
Affinity	SK	30.786	1.671
Propagation	G4	56.417	3.062

K-Means in more detail

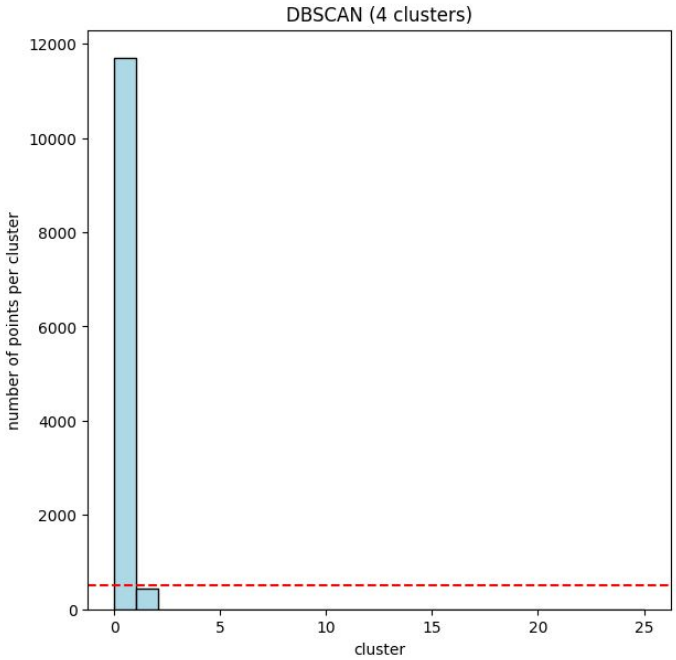
- Why did k-Means perform better than the other algorithms?
  - Fixed number of clusters
  - Ground Truth: 500 points per cluster with 25 clusters

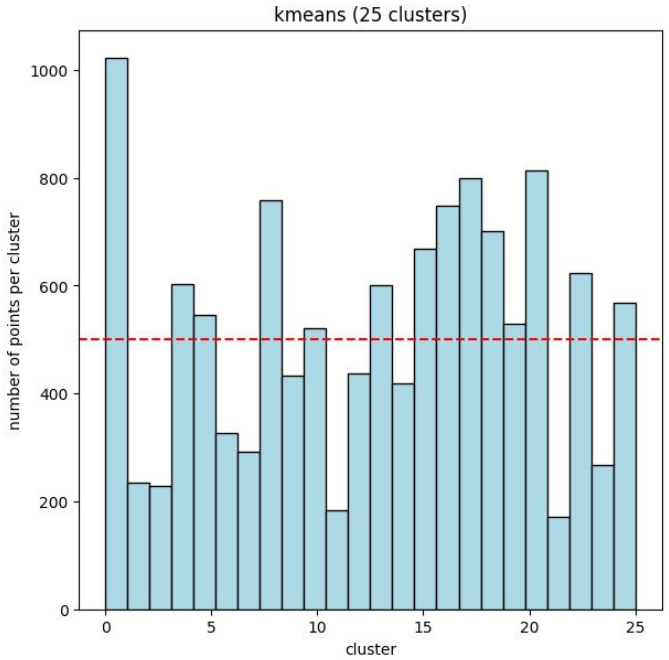




K-Means in more detail

- Help the clustering with giving it a part of the solution (number of clusters)

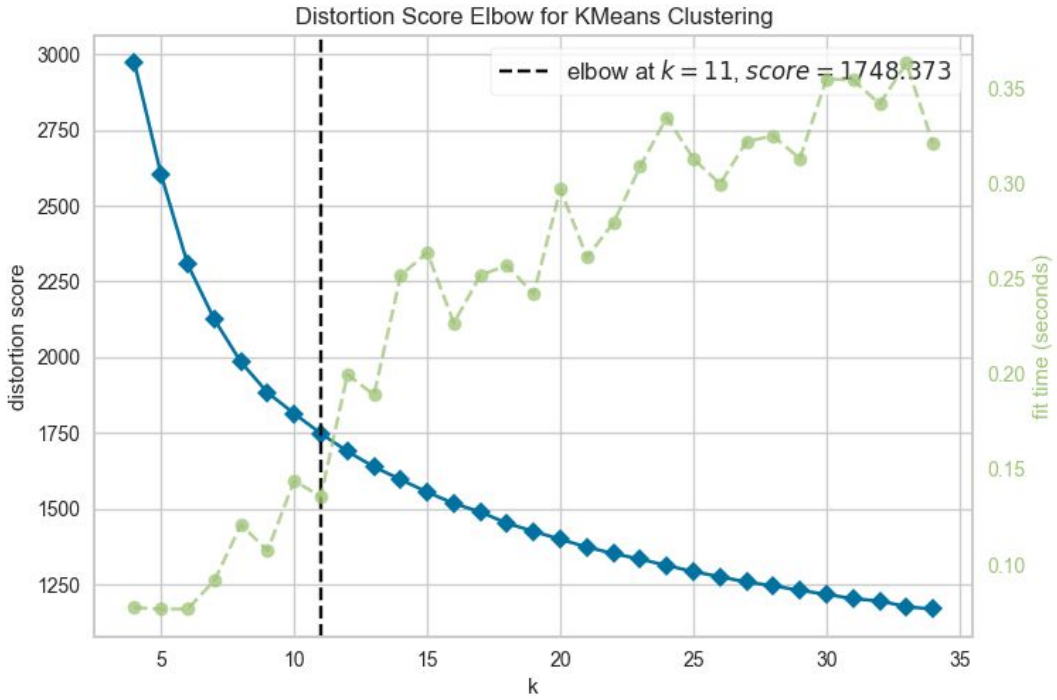




K-Means in more detail

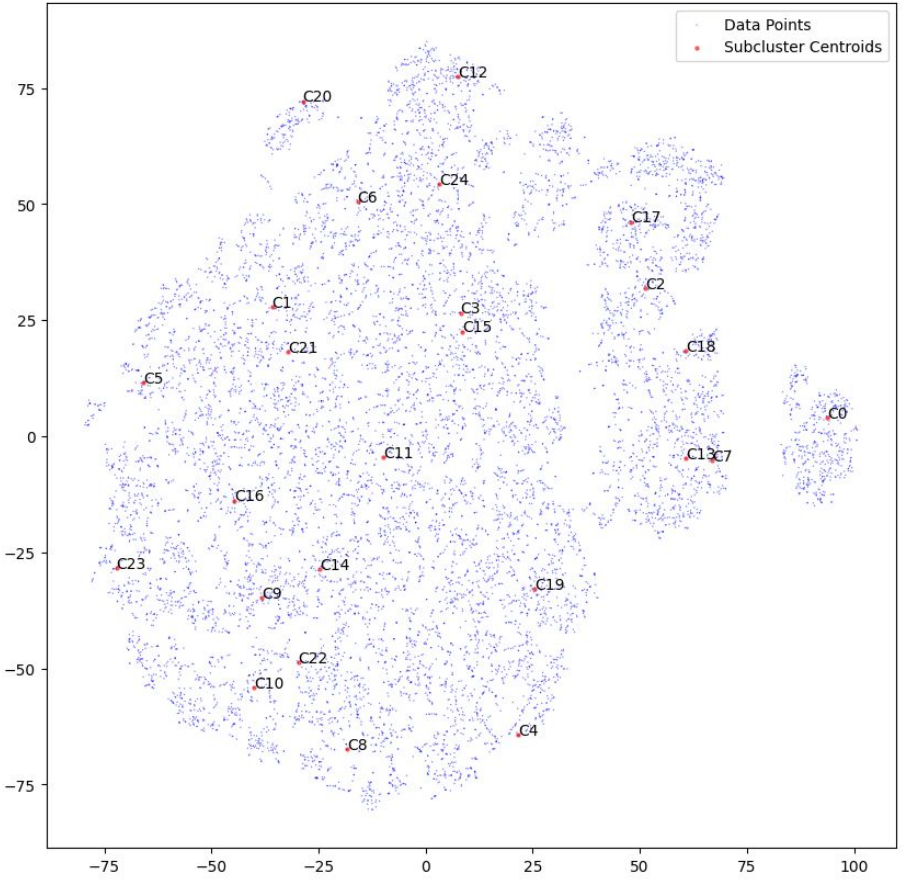
- Elbow Method
- This is a classically used heuristic
- Finds the optimal number of clusters at the “elbow” of the graph

- Adj. Rand. Index of K-means with 11 cluster: 0.097071



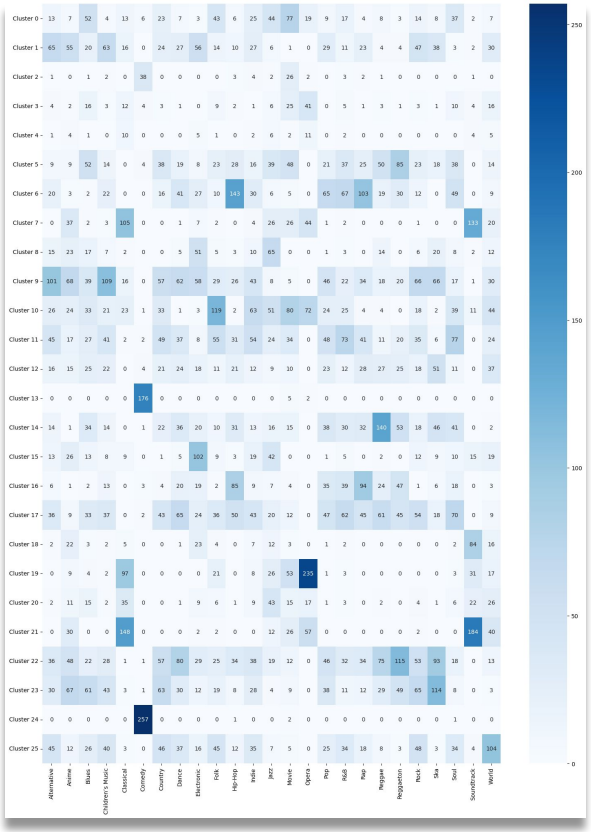
K-Means in more detail

- K-Means starts with random centroids, which are seemingly even distributed over the dataset
- sklearn uses a probability based method: **greedy k-Means++**,
- K-Means++:
  - First center placed uniformly at random
  - Every center after is chosen with the probability of $\frac{D(x')^2}{\sum_{x \in X} D(x)^2}$
  - D(x): shortest distance of x to the closest center



k-Means on attribute level identifies genre similarities

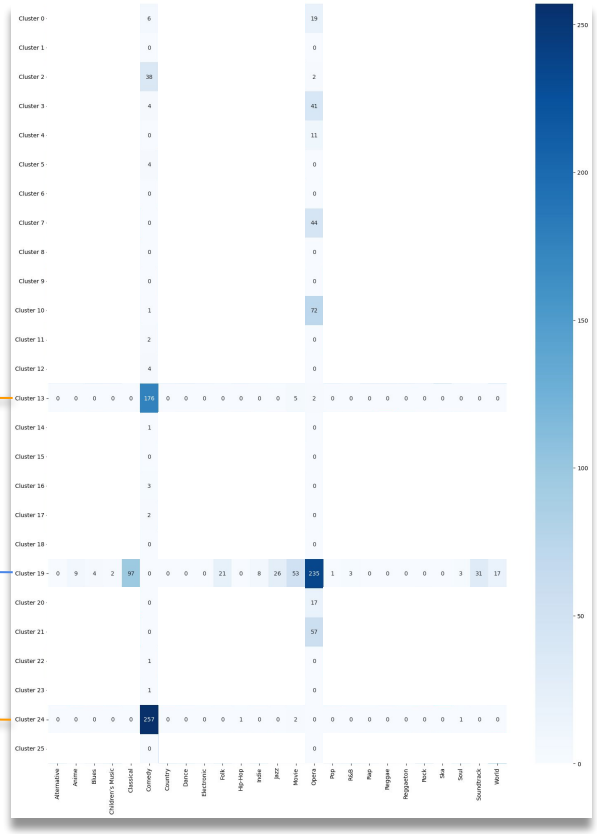
The contingency matrix on the right shows the cluster/genre overlap for k-Means on the song attributes



k-Means on attribute level identifies genre similarities

The contingency matrix on the right shows the cluster/genre overlap for k-Means on the song attributes

- Strong clusters for
  - Comedy
  - Opera



k-Means on attribute level identifies genre similarities

The contingency matrix on the right shows the cluster/genre overlap for k-Means on the song attributes

- Strong clusters for
  - Comedy
  - Opera

- Similarities between
  - Hip-Hop and Rap
  - Classical and Soundtrack

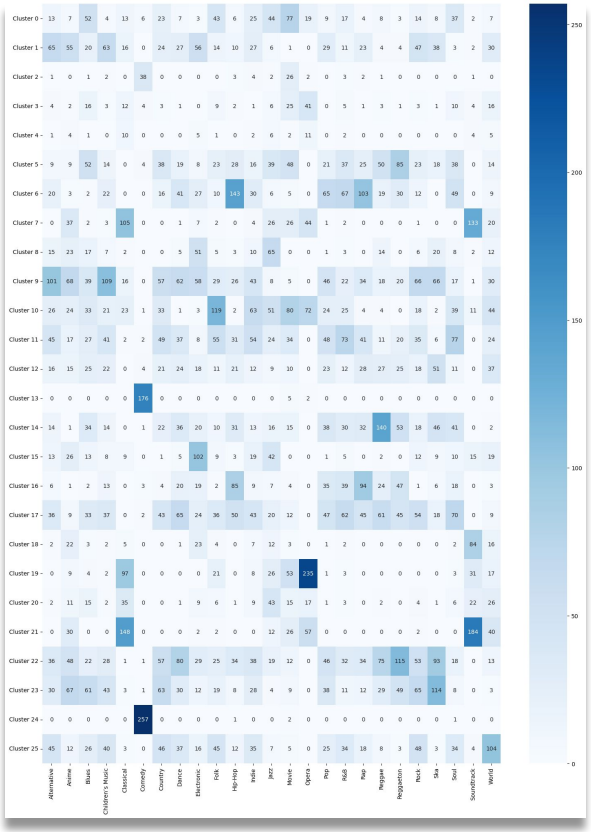
k-Means on attribute level identifies genre similarities

The contingency matrix on the right shows the cluster/genre overlap for k-Means on the song attributes

- Strong clusters for
  - Comedy
  - Opera

- Similarities between
  - Hip-Hop and Rap
  - Classical and Soundtrack

Adj. Rand. Index: **0.0991538**



k-Means on title embeddings identifies few strong clusters

The contingency matrix becomes more clear cut when clustering on embeddings of the song titles

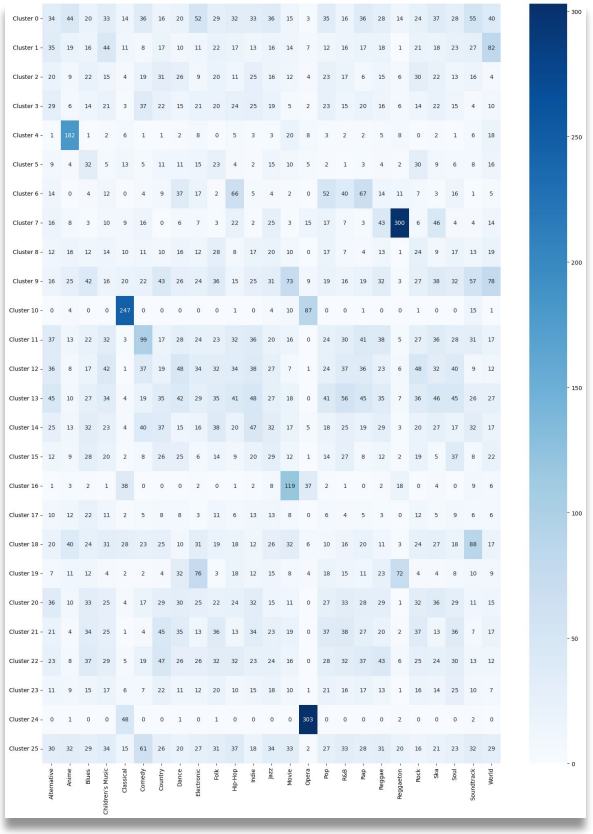
“Colors of The Wind”

Title of the song

Hugging Face

Sentence transformer “all-mpnet-base-v2”

768-dim embedding vector

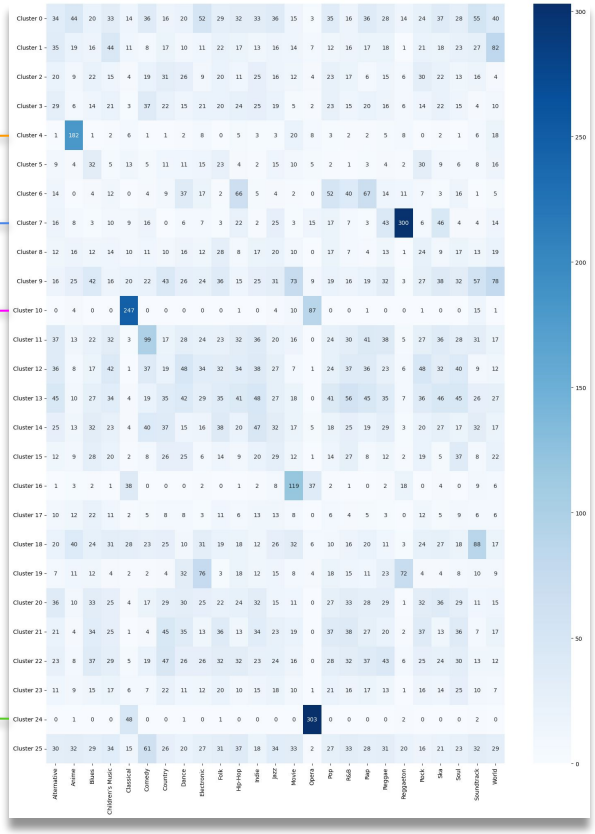


k-Means on title embeddings identifies few strong clusters

The contingency matrix becomes more clear cut when clustering on embeddings of the song titles.

The following genres had distinct clusters

- Anime
- Reggaeton
- Classical
- Opera

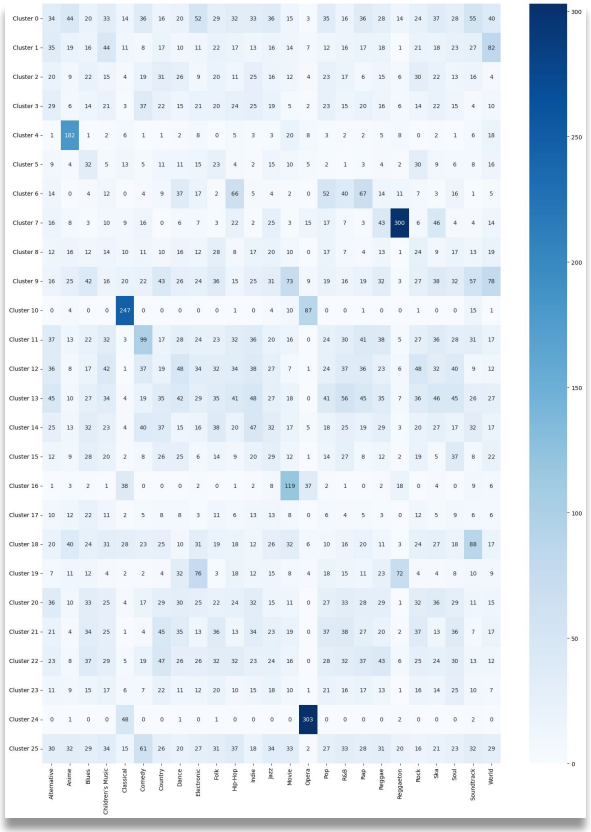


k-Means on title embeddings identifies few strong clusters

The contingency matrix becomes more clear cut when clustering on embeddings of the song titles.

The following genres had distinct clusters

- Anime → Japanese
- Reggaeton → Spanish
- Classical → *Words like “Op.”, “Act”, “No.”*
- Opera → Italian

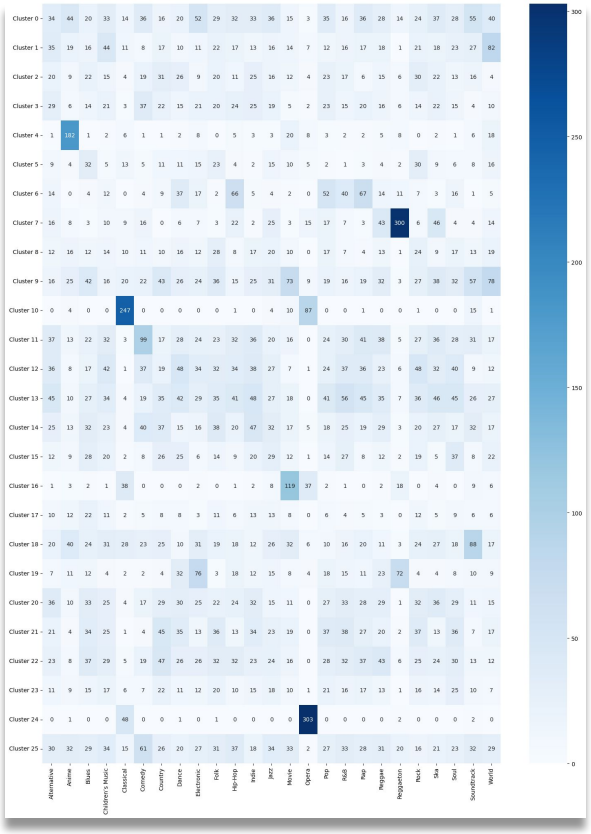


k-Means on title embeddings identifies few strong clusters

The contingency matrix becomes more clear cut when clustering on embeddings of the song titles.

The following genres had distinct clusters

- Anime → Japanese
- Reggaeton → Spanish
- Classical → *Words like “Op.”, “Act”, “No.”*
- Opera → Italian



Adj. Rand. Index of: **0.0574186**

Our findings can be summarized in three conclusions

1. **Genres cannot not be reduced to a standard profile of song attributes**
    a. “Comedy” was the only clearly separable genre
    (High speechiness, liveness, and acousticness)

Our findings can be summarized in three conclusions

1. **Genres cannot not be reduced to a standard profile of song attributes**
    a. “Comedy” was the only clearly separable genre
    (High speechiness, liveness, and acousticness)

2. **More sophisticated approaches do not necessarily produce better results**
    a. Specifics of the dataset benefited k-Means
    b. Some algorithms have high complexity (Affinity Propagation)

Our findings can be summarized in three conclusions

1. **Genres cannot not be reduced to a standard profile of song attributes**
   a. “Comedy” was the only clearly separable genre
      (High speechiness, liveness, and acousticness)

2. **More sophisticated approaches do not necessarily produce better results**
   a. Specifics of the dataset benefited k-Means
   b. Some algorithms have high complexity (Affinity Propagation)

3. **High level metrics only give a first impression**
   a. Choice of metrics can favor certain algorithms / outcomes
   b. Lower scoring results can provide valuable insights