University of Pisa
Department of Computer Science

Spotify Tracks Dataset Analysis

Authors:

Lorenzo Bux
Giulia Diamanti
Francesco Capria

2023-2024

1 Introduction

The Spotify Tracks dataset contains 20 genres, complete with essential attributes like track names, artists, album titles, and popularity scores. Additionally, it includes detailed audio features such as danceability, energy, and loudness. Futhermore, the dataset has been splitted in two: a training set composed by 24 attributes and 15000 rows, and a test set of 24 attributes and 5000 rows.

2 Data Understanding

The training set contains a total of 15,000 rows, with each row corresponding to a track in the Spotify dataset, and comprises 24 distinct attributes. These attributes are a mix of categorical and numerical types, as detailed in Table 1 for reference. Notably, the dataset includes an attribute labeled '*processing*,' for which there is no documentation in the provided materials.

Attribute	**Description**	**Type**	**Domain**
name	Name of the track	object	Strings
duration_ms	The track length in milliseconds	int64	$\mathbb{N}$
explicit	Whether or not the track has explicit lyrics (true = yes it does; false = no it does not OR unknown)	bool	{True, False}
popularity	The popularity of a track is a value between 0 and 100, with 100 being the most popular.	int64	$\mathbb{N} \cap [0, 100]$
artists	The artists' names who performed the track. If there is more than one artist, they are separated by a ;	object	Strings
album_name	The album name in which the track appears	object	Strings
danceability	Danceability describes how suitable a track is for dancing. A value of 0.0 is least danceable and 1.0 is most danceable	float64	$\mathbb{R} \cap [0.0, 1.0]$
energy	Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity.	float64	$\mathbb{R} \cap [0.0, 1.0]$
key	The key the track is in. Integers map to pitches using standard Pitch Class notation.	int64	$\mathbb{N} \cap [0, 11]$
loudness	The overall loudness of a track in decibels (dB)	float64	$\mathbb{R}$
mode	Mode indicates the modality (major or minor) of a track. Major is represented by 1 and minor is 0	float64	{0, 1}
speechiness	Speechiness detects the presence of spoken words in a track.	float64	$\mathbb{R} \cap [0.0, 0.939]$
acousticness	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic	float64	$\mathbb{R} \cap [0.0, 1.0]$
instrumentalness	Predicts whether a track contains no vocals. The closer the instrumentalness value is to 1.0	float64	$\mathbb{R} \cap [0.0, 1.0]$

liveness	Detects the presence of an audience in the recording	float64	$\mathbb{R} \cap [0.0, 0.994]$
valence	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track	float64	$\mathbb{R} \cap [0.0, 1.0]$
tempo	The overall estimated tempo of a track in beats per minute (BPM)	float64	$\mathbb{R} \cap [0.0, 220.525]$
features_duration_ms	The duration of the track in milliseconds	int64	$\mathbb{N}$
time_signature	An estimated time signature	float64	$\mathbb{R}$
n_beats	The total number of time intervals of beats throughout the track	float64	$\mathbb{R}$
n_bars	The total number of time intervals of the bars throughout the track	float64	$\mathbb{R}$
popularity_confidence	The confidence, from 0.0 to 1.0, of the popularity of the song	float64	$\mathbb{R} \cap [0.0, 1.0]$
processing	Information is not provided	float64	$\mathbb{R}$
genre	The genre in which the track belongs	object	Strings

Table 1: Description of dataset attributes.

2.1 Correlations

In the process of refining our dataset for effective data mining, we conducted a thorough analysis to identify correlations and irrelevant attributes. This step is crucial since irrelevant or redundant attributes can negatively impact classification accuracy and the quality of clusters (in next paragraph we will present how we decided to handle these attributes).
During our analysis, we discovered several significant correlations:

- acousticness, energy, and loudness: A positive correlation between *loudness* and *energy* was found, indicating louder tracks are often more energetic. There's also a negative correlation between *acousticness* and both *energy* and *loudness*, suggesting that more acoustic tracks are generally quieter and less intense.

- duration (ms) and number of beats/bars: We observed a strong correlation, confirming that longer tracks typically have more beats and bars, consistent across different music genres.

- number of beats/bars and tempo: A high correlation between these factors suggests a steady ratio, with tempo aligning closely with the number of beats and bars, affecting the track's rhythm.

- processing, key, and mode: An unexpected correlation was found between *processing* and *key*, and indirectly with *mode*, implying that processing might be influenced by these elements and could be an indicator of a track's harmonic content.

Additionally, a positive correlation between *valence* and *danceability* suggests happier tracks tend to be more danceable. An inverse correlation between *liveness* and *danceability* was also noted, indicating studio tracks may be more dance-oriented than live recordings.

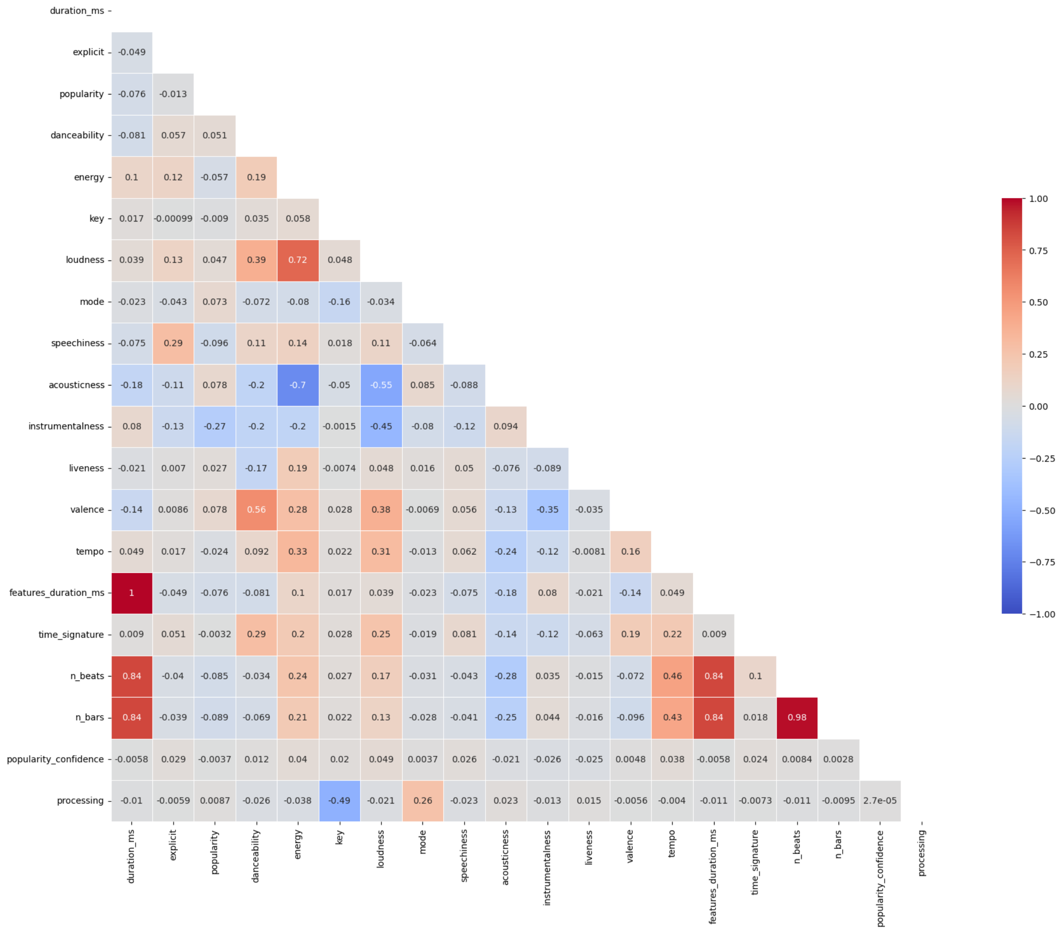

Figure 1: Heatmap of the correlation matrix

2.2 Looking for potential outliers

In music data analysis, it's crucial to identify atypical data points that deviate from common patterns. These points, often highlighted in box plots, can reveal unique events or anomalies and significantly impact on statistical interpretations.

- *duration_ms*: tracks with extremely short or long durations may indicate special editions or interludes.
- *loudness*: variations in loudness reflect differences in production or recording styles.
- *liveness*: high liveness values may suggest live recordings.
- *speechiness*: increased speechiness is typical in some genres (e.g. j-dance has the highest average speechiness).
- *tempo*: some tracks have very high or very low value of tempo.
- *n_beats*: some tracks have very high or very low value of tempo, these could represent tracks with unique structural elements.

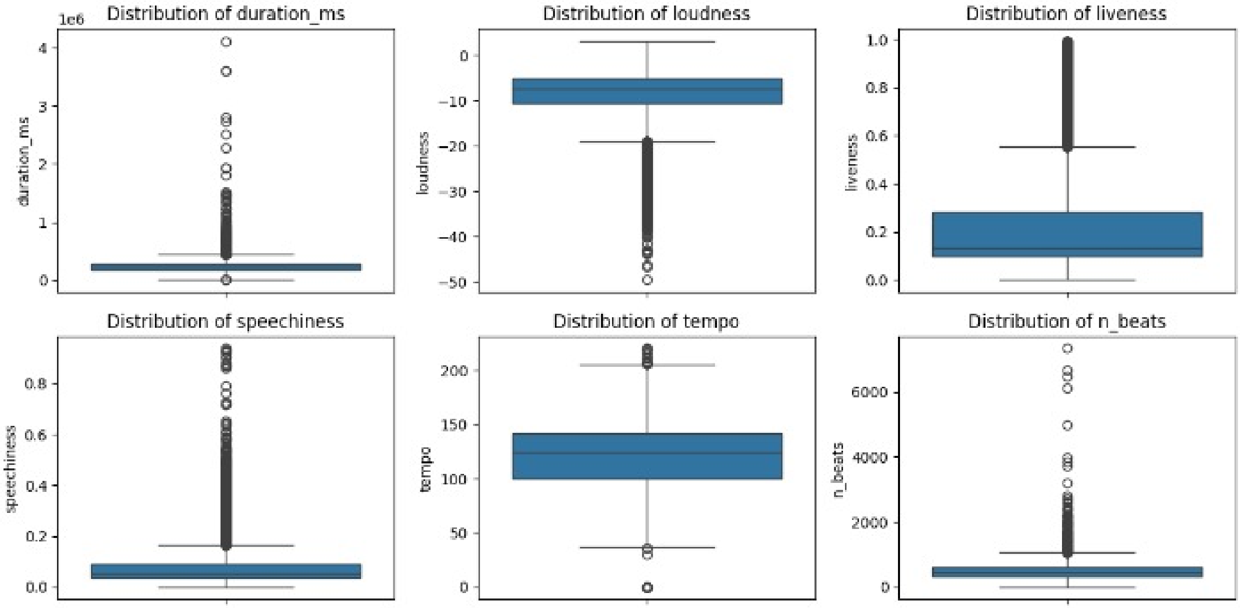

Figure 2: some boxplots of the attributes to identify outliers

These boxplots (Figure 2) we searched for potential outliers and they revealed several points with values above or below quartiles, however since these are not isolated points we cannot consider them as outliers and therefore they will not be removed from the dataset for the time being. These points with particular values for certain attributes could be characteristic of a specific genre. For example we noticed that in our dataset there are 84 tracks of the *sleep* genre, so it is probably a typical feature of tracks of this genre that may contain sounds of background noises that do not have a high (if not zero precisely) BPM value.

3 Data preparation

3.1 Filling in missing values

In our dataset, we've found some missing data in different attributes, which we need to address as part of our data preparation phase:

- *mode*: 4,450 missing values
- *time_signature*: 2,062 missing values
- *popularity_confidence*: 12,783 missing values

Our approach to handling missing values focused on preserving the dataset's original characteristics. We utilized imputation methods that mirrored the data's inherent patterns and distributions, thus maintaining the original correlations and avoiding any bias in subsequent analyses.

- Imputing *time_signature*: For the 2,062 missing *time_signature* entries, we imputed using the mode (predominantly 4.0), minimizing the impact on the dataset's overall characteristics. This ensured that the general trends and relationships within the data remained consistent.
- Missing *mode* values: With 4,450 missing entries in the *mode* attribute, we used a pivot table to create a mode frequency matrix based on *key* and *processing*, which are closely correlated with *mode*. This strategy, supported by a heatmap visualization (Figure 3), allowed us to impute missing *mode* values accurately by assigning the most frequent *mode* for each *key* and *processing* combination. This method preserved the data's integrity and original structure more effectively than simple mode imputation.

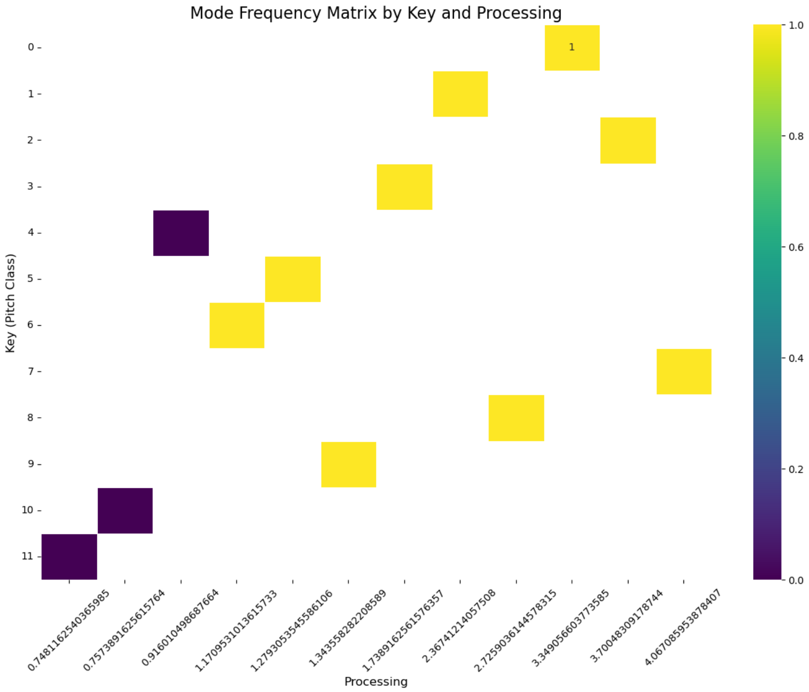

Figure 3: The pivot table, represented as a heatmap, that is used to fill in the missing values of *mode*.

3.2 Irrelevant features

Following our correlation analysis and comprehensive dataset review, we streamlined the dataset for our data mining objectives by removing certain attributes:

- *popularity_confidence*: removed due to numerous missing values and redundancy with the *popularity* score, adding no unique information.
- *processing*: the correlation between *processing* and *key* suggested a derived relationship, in particular for every *processing* value corresponds a unique value of *key*. This redundancy led to the removal of the *processing* attribute, as *key* sufficiently represents the information needed for our analysis.
- *features_duration_ms*: was delated due to its near-perfect correlation with *duration_ms* (Figure 4), which sufficiently represents track length.

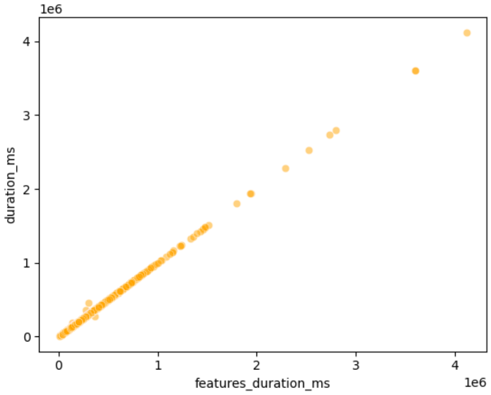

Figure 4: The scatter plot of feature_duration_ms vs. duration_ms

- *n_bars*: we discovered a relationship between *time_signature* and the attributes *n_beats* and *n_bars*, characterized by the equation *time_signature = n_beats/n_bars*. Given this relationship, we have decided to remove *n_bars* from our dataset to minimize redundancy while preserving essential rhythmic information. Maintaining *n_beats* is crucial as it offers distinct insight into the track's rhythmic complexity, which is not fully encapsulated by *time_signature* alone. This decision allows us to maintain a focus on the most informative aspects of our dataset's rhythmic structure.

4 Clustering

This section provides an overview of the behavior of different clustering algorithms applied to our data. We compare three main families of clustering: Centroid-based, Density-based, and Agglomerative clustering, each offering unique insights into the inherent groupings within our dataset.

For preparing the dataset, we focused on enhancing its suitability for clustering analysis. This involved removing text-based type attributes, which are less relevant for our algorithms, and categorical attributes. At the end of the preparation for the clustering phase, we obtained a dataset with 12 attributes: *duration_ms, popularity, danceability, energy, loudness, speechiness, acousticness, instrumentalness, liveness, valence, tempo, n beats*. At this point, the data was normalized using the MinMaxScaler, ensuring it is optimally formatted for effective clustering.

4.1 Hierarchical clustering

In our approach we adopted firstly an agglomerative hierarchical clustering approach. This method incrementally builds clusters, revealing the structure of the dataset.

We explored various distance metrics (Euclidean, Manhattan, Cosine) and linkage methods (complete, single, Ward, group average) to avoid overlooking any clustering patterns.

By analyzing dendrograms from various metric-linkage combinations, we aimed to find a balance in cluster specificity. We found that using the Ward method with Euclidean distance yielded the most distinct clustering, suggesting 2, 3, or 4 as potential cluster numbers. The Complete linkage method suggested 8 clusters, while the Single linkage method with Euclidean distance indicated 2 clusters when we cut at the level 1.0. The Euclidean distance metric paired with Ward linkage appeared most effective, optimizing cluster cohesion and separation. This combination minimized variance within clusters, resulting in more meaningful groupings.

These insights guided our application of K-Means clustering with the optimal cluster numbers identified from the dendrogram (particularly 2, 3, and 4), so we employed K-Means with a data-informed approach.

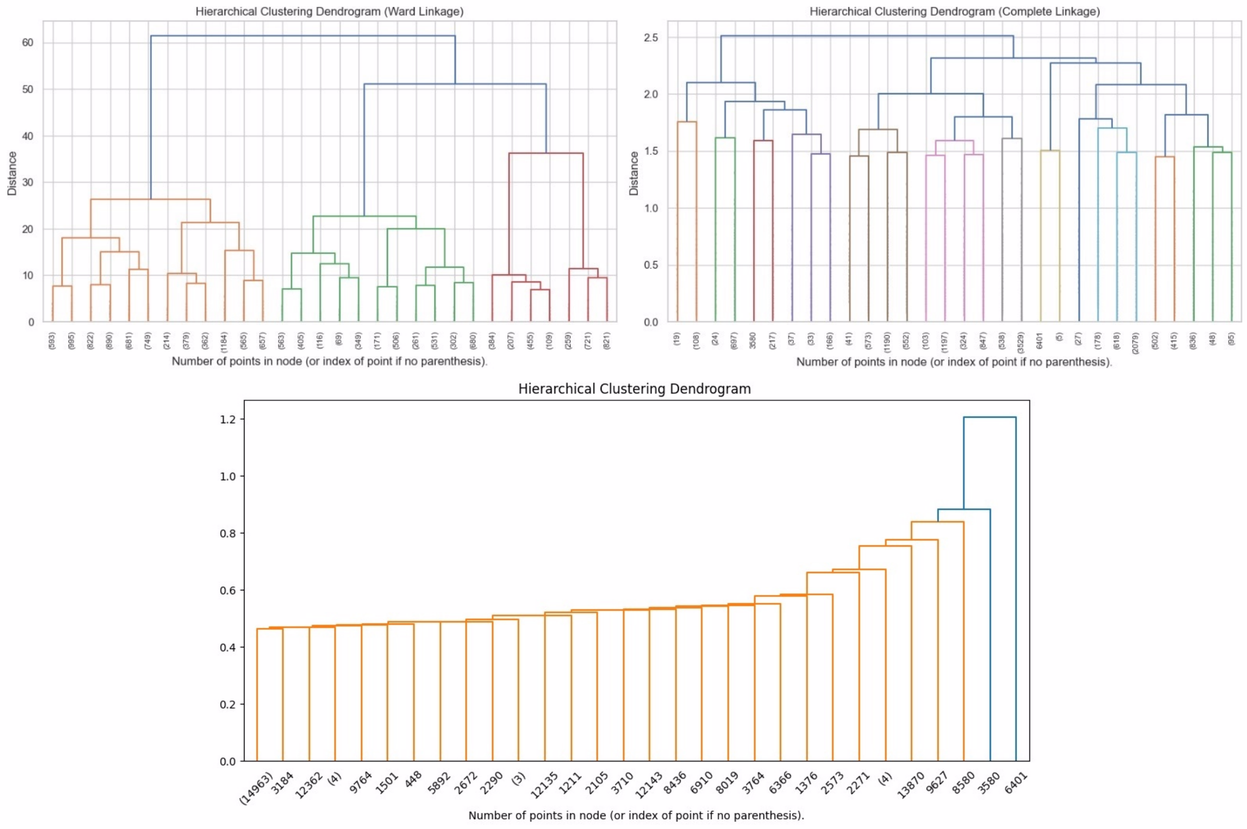

Figure 5: comparing Hierarchical clustering methods dendrograms (Ward, Complete Linkage, Single Linkage)

4.2 Density-based clustering: DBSCAN

As we continued our exploration of clustering techniques for our music dataset, we passed to the Density-Based Clustering.

With DBSCAN clustering we start with the crucial task of determining the optimal radius (Eps) for defining dense regions in the dataset. This step was essential, as the choice of epsilon significantly influences the formation of clusters in DBSCAN. We utilized the k-nearest neighbor graph (Figure 6). This graph plots the distance of each point to its k-th nearest neighbor, providing a visual means to identify the 'knee' or the point of maximum curvature. The knee in this graph served as an indicator for the optimal epsilon, guiding us to a radius that effectively balances between distinguishing meaningful clusters and minimizing noise.

We started by calculating the average distance of the k nearest points to find the Eps. For the calculation of the min_samples, we chose the value with the highest possible Silhouette Score (Figure 7).

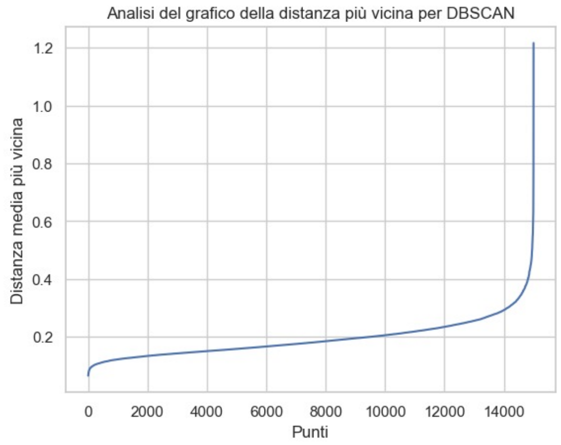

Figure 6: DBSCAN MinScore

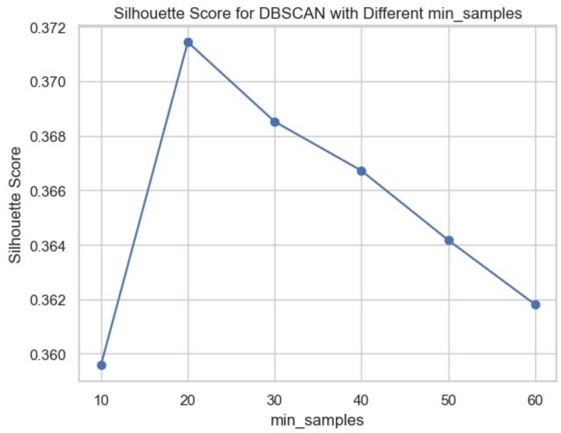

Figure 7: Silhouette DBSCAN

As we navigated through the DBSCAN process, several observations became apparent:
with Single linkage in Hierarchical clustering, we have results similar to DBSCAN. These two techniques result in a single cluster, and the rest is detected as outliers due to:

- Density Dependence: both algorithms are density-oriented. DBSCAN identifies clusters based on the density of points, while single linkage considers the distance between the nearest points. In cases where data is distributed in regions of similar density, both algorithms can detect these regions as clusters.
- Outlier Detection: both DBSCAN and single linkage clustering can identify isolated points as noise or outliers. DBSCAN does this through the concept of low-density points, while single linkage can form clusters of points that are close to each other, isolated from the rest of the dataset.

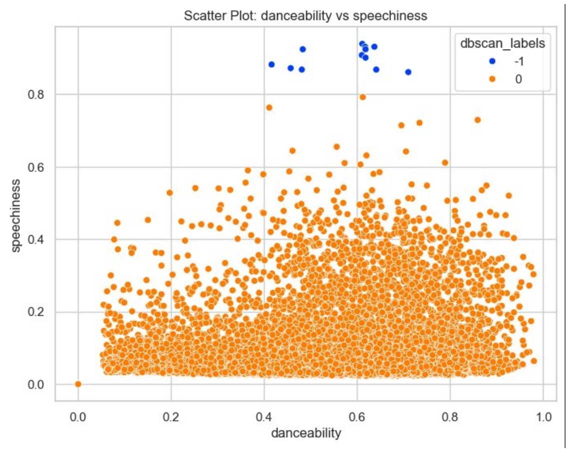

Figure 8: DBSCAN: *speechiness* and *danceability*

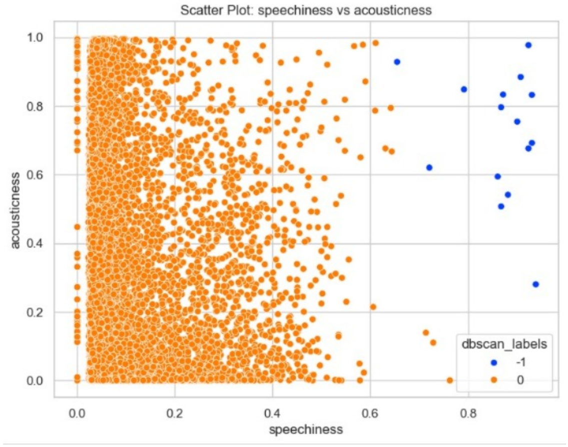

Figure 9: DBSCAN: *acousticness* and *speechiness*

In figures 9 and 10 there is an example of two pairs of attributes plotted, confirming the fact that only 2 clusters have been identified and the blue clusters are very small compared to the orange big one, so we can consider them as noise points.
From this, we deduced that these methods are not suitable for our dataset. Therefore, we tried using the k-means technique.

4.3 Centroid-based clustering: K-Means

For k-means clustering, we initially chose to analyze the silhouette and SSE values, also considering the number of clusters obtained from hierarchical agglomerative clustering (k= 2, 3, 4, 8). We started plotting the results for k-means algorithm for SSE and Silhouette scores as we can see in Figure 10.

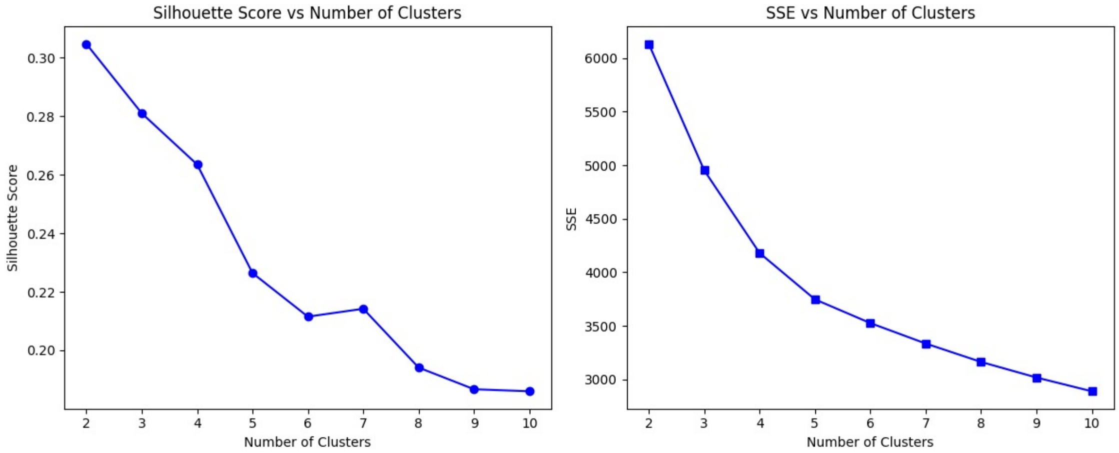

Figure 10: SSE and Silhouette scores for K-Means clustering

We can see from the sum of squared errors (SSE) graph (on the right) that the optimal number of clusters appears to be between 4 and 5. Upon further examination of the results on the silhouette score graph, we observe a significant drop between k=4 and k=5 of the silhouette score. Therefore, we can assert that a good value for the number of clusters could be 4. However, we have made several attempts, testing various other possible cluster numbers as well (in particular: k=2,3 and 4, values that gave us good silhouette scores).

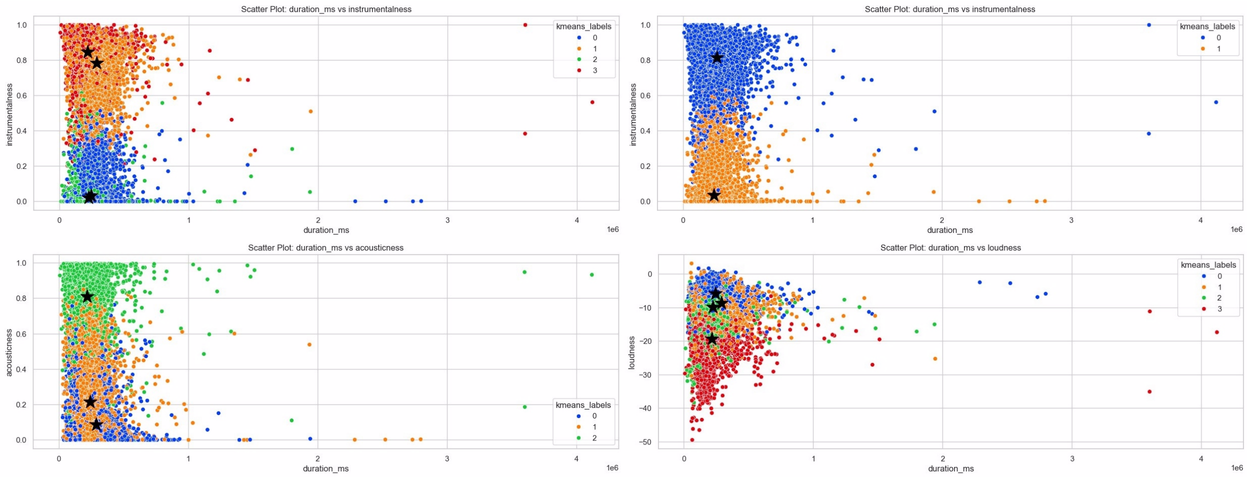

Figure 11: k-means plotted attributes : k=4 on the top left and on the bottom right sides, k=2 on the top right and k=3 on the bottom left

The exclusive nature of K-Means, assigning each data point to only one cluster, failed to encapsulate the complex and overlapping relationships among the musical attributes.
Furthermore, due to the presence of 12 attributes in our clustering dataset, makes difficult the visualization of the clustering with only two attributes.

4.4 Final discussion on clustering

In our cluster analysis, the centroids derived from applying k-means with k=4 have provided valuable insights into the unique characteristics of each cluster. The plot distinctly illustrates how these clusters diverge in specific attributes, offering an understanding of the dataset's inherent structure.

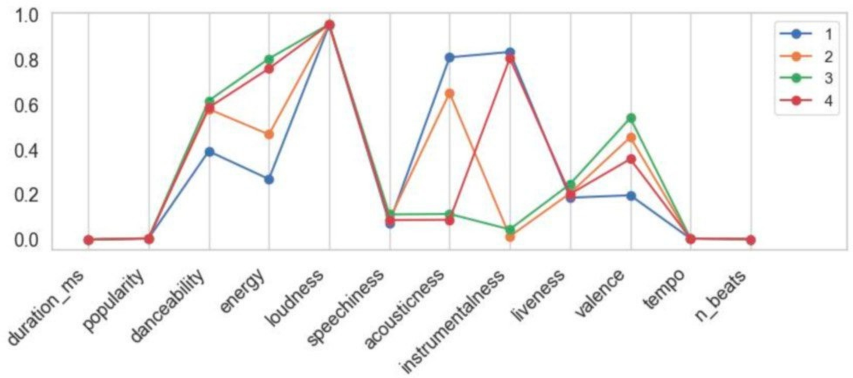

Figure 12: clusters centroids for attribute

- danceability: this attribute reveals significant differences among the clusters. Cluster 1 stands out with its notably lower danceability scores, suggesting a collection of tracks less suited for dancing. In contrast, clusters 2, 3, and 4 show considerably higher values, indicating these tracks are more rhythmically inclined and likely more danceable.

- energy: an interesting variation is observed in the energy attribute across the clusters. Cluster 1 is characterized by its low energy levels, possibly indicating slower or more mellow tracks. Cluster 2 occupies an intermediate position, signifying a moderate level of intensity and dynamism. Clusters 3 and 4, on the other hand, are marked by high energy levels, likely indicative of tracks that are more vibrant and intense.

- acousticness: the acousticness attribute exhibits a clear distinction between the clusters. Clusters 1 and 2 are marked by higher acousticness, suggesting these tracks feature more acoustic sounds and possibly less electronic manipulation. Conversely, clusters 3 and 4 display lower acousticness, hinting at a prevalence of electronic or heavily produced tracks in these groups.

- valence: pertaining to the valence attribute, cluster 3 emerges as the most distinctive, boasting the highest valence scores. This could suggest that tracks in this cluster have a more positive or euphoric tone. Meanwhile, cluster 1 is on the opposite spectrum, with the lowest valence scores, indicating tracks that might be more somber.

These findings underscore the diversity and complexity of the musical tracks in our dataset. Each cluster encapsulates a unique combination of musical attributes, reflecting varied musical styles and moods.