Università di Pisa

Data Mining Project 2023/2024

Exploration of the "Spotify Tracks Dataset"

Silvio Calderaro (625577), Salvatore Ergoli (660527), Manuele Messere (622479)

Contents

1 Introduction

The purpose of this report is to provide an in-depth examination of the *Spotify Tracks* (STD) dataset. Specifically, the analysis first focused on data exploration and preparation to identify trends in features, relationships, and potential data quality issues. Next, attention was focused on applying clustering techniques to categorize the music tracks into homogeneous groups. This phase aims to provide a detailed view of the dynamics and relationships within the data. Then, a classification task was performed, which further deepened the understanding of the inherent dynamics of the dataset, allowing for a more accurate categorization of the music tracks. Finally, the analysis was completed by applying pattern mining techniques, which focused on identifying recurring patterns or significant structures within the analyzed data.

2 Data Understanding

The Spotify Tracks dataset consists of 15,000 instances and includes 24 different features. In this section, a comprehensive examination of data exploration is undertaken, and the approach to dealing with missing data within the dataset is explained.

2.1 Data Semantics

Variables can be divided into two main categories: categorical (*qualitative*) and numeric (*quantitative*). Categorical variables are further subdivided into *nominal*: name, artists, album name, genre and explicit, represented *binary*. As for quantitative variables, they include *ratio*: duration_ms, energy, danceability, speechiness, loudness, acousticness, instrumentalness, liveness, tempo, features_duration_ms, number_beats, number_bars, processing, popularity confidence, time_signature; mode representative *binary* feature and *discrete*, like variable key. Descriptions of the features are given in the next table 1, excluding the more intuitively understood nominal variables.

Feature Name	**Feature Type**	**Description**
mode	Numeric, Binary	Represents the tonality of the track, which can be ”minor” (represented as 0) or ”major” (represented as 1). This attribute can influence the emotional and harmonic aspects of the song.
speechiness	Numeric, Ratio	Measures the presence of spoken words in the track on a scale from 0.0 to 1.0, indicating how predominant vocal parts are in the song.
energy	Numeric, Ratio	Represents the overall energy of a musical track and is expressed on a scale from 0.0 to 1.0. This is useful for classifying songs based on their level of intensity.
danceability	Numeric, Ratio	Measures how a song is suitable for dancing on a scale from 0.0 to 1.0.
loudness	Numeric, Ratio	It measures the sound intensity (volume) expressed in dB
acousticness	Numeric, Ratio	Exhibits on a scale from 0.0 to 1.0, indicating how much the music relies on acoustic sounds.
instrumentalness	Numeric, Ratio	Predicts whether a song contains vocal parts, with values closer to 1.0 indicating a predominantly instrumental track, while lower values indicate a greater presence of vocals.
liveness	Numeric, Ratio	Represents the presence of an audience during the recording on a scale from 0.0 to 1.0 and provides information about the live performance aspect of the song.
tempo	Numeric, Ratio	It estimated tempo of the track in beats per minute (BPM)
duration_ms	Numeric, Ratio	It variable measures the length of the track in milliseconds and it is a metric for understanding how long the song lasts.
features_duration_ms	Numeric, Discrete	Represents the duration of the track in milliseconds, which may differ from ”duration_ms” in some circumstances, such as extended versions or remixes of a song.

n_beats	Numeric, Ratio	Denotes the total number of beat intervals during the track and provides information about the rhythmic structure.
n_bars	Numeric, Ratio	Total number of bar intervals during the track and provides details about the song's structure.
popularity_confidence	Numeric, Ratio	Indicates the confidence in the popularity of the song on a scale from 0.0 to 1.0 and indicates how much one can trust the assigned popularity value of the track.
time_signature	Numeric, Ratio	Depicts the time signature of the track, such as "4/4" or "3/4," and is crucial for understanding the rhythmic structure of the song.
key	Numeric, Discrete	Represents the key of the track as integers, mapped according to the standard notation of pitch classes.
valence	Numeric, Ratio	Describes the musical positivity of the track on a scale from 0.0 to 1.0, indicating how positively the music conveys emotions. This feature is useful for evaluating the emotional aspect of a song and can help categorize music based on its overall mood.
popularity	Numeric, Discrete	It shows on a scale from 0 to 100 how much Spotify's users appreciated the track.

Table 1: Description of dataset features.

The features presented in the table can be organized into thematic groupings to facilitate the identification of relevant semantic features in the dataset. For example, metrics describing *musical aspects* include: acousticness, instrumentalness, speechiness, liveness, key, mode, and genre. The *acoustic-physical dimension* is described by the properties of energy, loudness, and time signature. *Rhythmic characteristics* are represented by n_beats, n_bars, and tempo. Similarly, *temporal features* are expressed by time_signature, duration_ms, and features_duration_ms. Finally, features that reflect *the degree of song appreciation* include danceability, popularity, and valence.

2.2 Distribution of the variables and statistics

In this section, the distributions of some important characteristics are analysed in detail, both individually and in relation to each other, with the aim of identifying relevant trends. The main statistical indices have been calculated for all the characteristics. Without including all the values, some reflections on the most interesting results obtained are suggested.

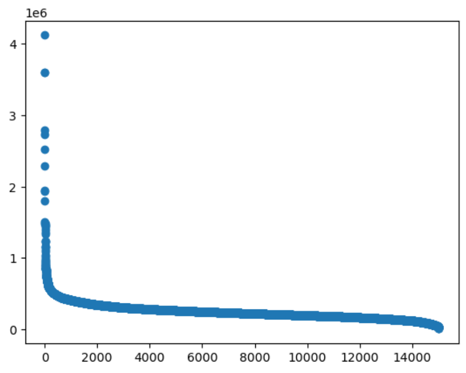


Figure 1: Zipfian's distribution of Duration_ms

Duration_ms
This variable has a characteristic zipfian distribution. In Figure 1, the rank, which represents the position of the song in descending frequency order, is plotted on the x-axis, while the duration of the song is plotted on the y-axis. It can be seen that the duration of a song is inversely proportional to its rank.
The normal distribution of the data for this variable, obtained with a pair plot, shows a pronounced leftward shift in the curve, confirmed by a high skewness value of 7.73.

Popularity

The graph shows the frequency distribution of popularity. Popularity is measured on a scale from 0 to 100, where 0 is minimum popularity and 100 is maximum popularity. The graph shows that most items have relatively low popularity. The distribution is heavily skewed to the left of the graph, in fact there are a lot of unpopular items, with the highest peak occurring in the first hist. The distribution then falls off immediately. In the last part of the graph, there is an imbalance in which, beyond the value of 0.6, there is an increase in the frequency of items in relation to the central part of the distribution. The statistical indices for the mean and median are not very different, at 25 and 27 respectively. The value of the standard deviation, on the other hand, is significantly high: 18.

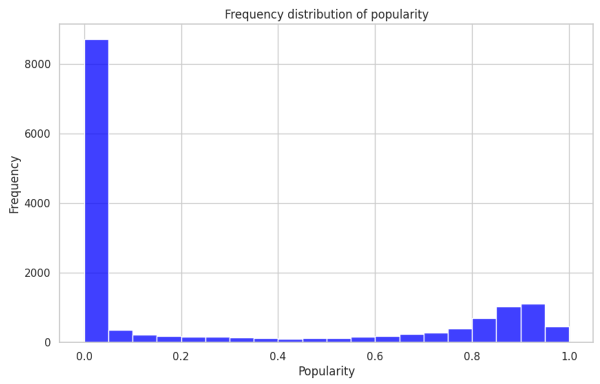


Figure 2: Distribution of Popularity

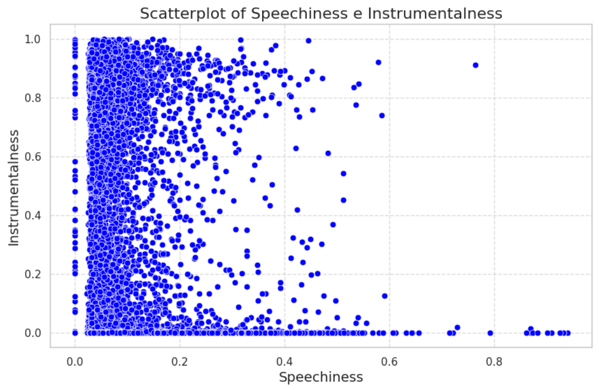


Figure 3: Distribution of Speechiness and Instrumentalness.

Speechiness and Instrumentalness

The plot presented here shows the distribution of the instrumentalness' variables, which refers to the instrumental part of the songs and speechiness, which refers to the vocal part. It is interesting to compare these features because they analyse similar and complementary musical aspects. The graph shows that some of the songs are in the lower left part of the graph, which means that they have a low level of speechiness and a high level of instrumentalness. Indeed this can be seen in the majority of the songs shown in the graph.Finally, an even smaller number of songs are found in the middle of the graph, meaning that they have an intermediate level of speechiness and instrumentalness,.

Explicit and Genre

Subsequently, an investigation motivated primarily by intellectual curiosity, rather than explicit research objectives, concerns the Boolean variable denoted *explicit*. This variable is correlated with the musical genre, as shown in Figure 4, leading to the observation that a predominant proportion of the songs categorised as explicit belong to J-dance (269 cases), industrial (161 cases), happy (88 cases), brazil (60 cases) and spanish (58 cases). It is worth noting that the distribution of the variable genre appears to be uniform, suggesting that the result is not influenced by any asymmetry inherent in the dataset.

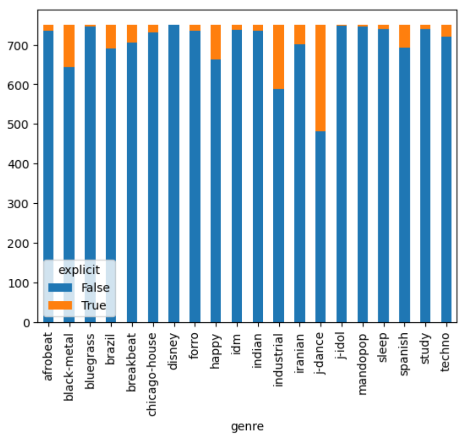


Figure 4: Explicit songs grouped by genre.

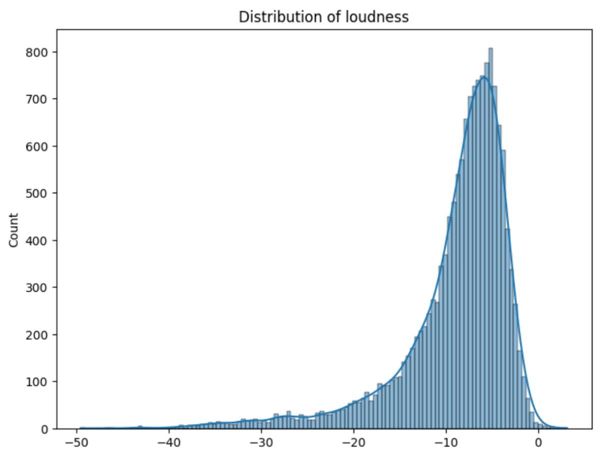


Figure 5: Loudness distribution.

Loudness

The skewness of the loudness distribution is positive, indicating that most of the songs in the dataset are skewed towards higher loudness levels. The right skew is visually manifested in the graph, with the majority of the data clustered at higher loudness values and a tail extending towards lower values. The mean loudness of -8.8950 indicates a generally high volume level, while a lower median of -6.202000 suggests a possible influence of very low values. The range from -49.5310 to 3.1560 shows a variety of volume levels, and 75% of the tracks exceed -5.1010, according to the percentiles.

2.3 Assessing data quality

In preparing the dataset, it has been recognised that the efficiency and coherence of our analysis is greatly influenced by the quality of the dataset. Meticulous data preparation is essential for a more straightforward and reliable investigation. To ensure this, it is imperative to address key aspects such as identifying missing data points, handling outliers and resolving semantic inconsistencies in the data. These tasks were therefore carried out in turn:

- Missing data points;
- Outliers;
- Semantic inconsistencies in the data.

2.3.1 Missing values

During our analysis were found missing values for certain attributes: *mode*, *popularity_confidence* and *time_signature*. For the first two variables, where the missing values were 4450 and 12783 respectively, it was decided to completely remove them , as any modification could have led to further imbalance in the data. Instead, information from two other variables, *n_beats* and *n_bars*, was used to manage the *time_signature* attribute. The time signature is actually the ratio between the number of beats and the number of bars, rounded to the nearest integer.

2.3.2 Outlier removal

Outliers are calculated using the *z-score* with a threshold of 3 and are considered outliers if they deviate more than 3 standard deviations from the mean. Once identified, outliers were removed from the dataset to prevent them from negatively affecting future analyses. The number of outliers for each variable is shown in the table below 2.

Attribute	**duration_ms**	**popularity**	**loudness**	**speechiness**	**liveness**	**tempo**	**n_beats**	**n_bars**
Number of outliers	127	7	388	429	484	94	120	130

Table 2: Number of outliers for each attribute.

2.3.3 Semantic inconsistencies in the data

Among the features there was one called *processing*. This, was removed from the dataset due to semantic inconsistency: on the one hand, there was no description of the feature within the dataset presentation file, and on the other hand, an initial quantitative analysis revealed a high presence of repeated decimal values, leading suspicions of poor data recording.

2.4 Transformation of variables

Regarding the transformation of the variables, the characteristic *explicit*, which contained the binary values *True* and *False*, was converted from a string to a numeric value. In fact, this attribute was made 1 and 0 respectively.

2.5 Pairwise correlation

This section will presents the correlation matrix of the entire dataset without outliers or inconsistent variables, such as *processing*. Figure 6 shows the correlation heatmap calculated with the Pearson coefficient.

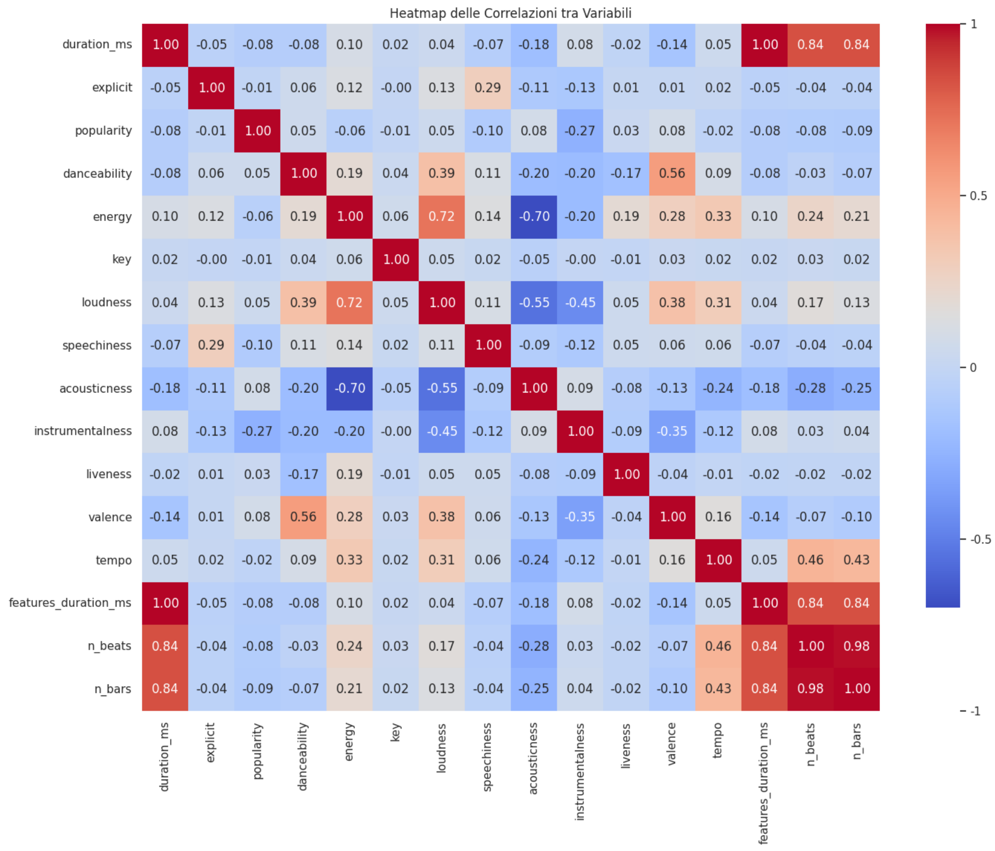


Figure 6: Correlation matrix of the features without outliers.

Within this rich framework, a significant revelation emerged from our analysis. There are several variables that have a correlation or anti-correlation greater than (±)0.60. Among these variables, it is possible to identify in particular *energy* and *loudness*, *energy* and *acousticness*, *duration_ms* with *n_bars* and *n_bars*, and the last mentioned variables between them. The *PCC* (Pearson Correlation Coefficient) is a statistical measure that quantifies the degree of linear association between two sets of data. It can therefore be assumed that the correlations found are indicative:

- The observed correlation between the characteristics *duration_ms*, *n_beats* and *n_bars* suggests that longer songs tend to have more beats and more bars. Consequently, the correlation between beats and bars could be due to the fact that songs are generally composed of a sequence of bars, each of which contains a fixed number of beats.

- The observed correlation between *energy* and *loudness* suggests that more energetic pieces of music tend to be louder. Similarly, the anti-correlation between *energy* and *acousticness* can be explained by the fact that acoustic songs tend to sound softer and more natural.

The high correlation between the variables *n_beats* and *n_bars* conveys similar information about the temporal organisation of a song. This finding led us to reconsider the efficiency of using both variables at the same time, so it was decided to remove n_bars. Similarly, the variable *features_duration_ms* was removed because of redundancy with the feature *duration_ms*, where the correlation is 1.

3 Clustering

3.1 Data preparation

After data cleaning, a clustering analysis was performed on the data set. First, a new set with only the numerical labels was taken from the data set obtained by data cleaning. After standardization with the *MinMaxScaler*, the search for the best set of features was started. The application of metrics such as SSE (Sum of Squared Errors) and Silhouette was studied at different values of k to evaluate their changes. To avoid the curse of dimensionality of the data, the initial sample of variables had to be reduced. In fact, there were 15 numerical variables in the entire data set. It was necessary to make a selection among these variables, trying to find the best set to obtain satisfactory clustering results, but on the other hand being careful not to lose too much information from the reduced data set. The main criteria used for the selection are the semantic importance of the variables and the avoidance of redundancy (*n_beats/n_bars*, *feature_duration_ms/durations_ms*). In the end, 7 variables were selected: *danceability*, *energy*, *loudness*, *acousticness*, *instrumentalness*, *tempo*, *n_beats*. Three clustering algorithms were used: K-Means, DBSCAN and Hierarchical. Finally, the optimal parameters were evaluated and the quality of the results obtained was assessed.

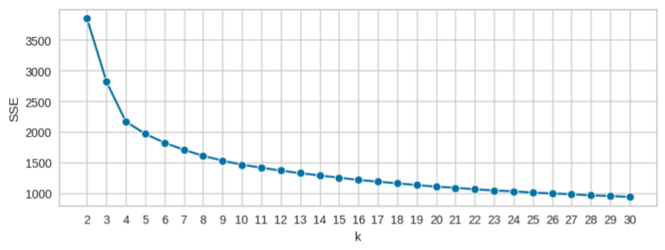
3.2 K-Means

The K-means algorithm is a widely used clustering technique in data mining that aims to divide a data set into distinct groups. It starts by randomly selecting K initial centroids, where K is the predefined number of clusters. Each data point is assigned to the nearest cluster based on Euclidean distance. New centroids are computed and the process is repeated iteratively until a stopping criterion is met, such as stabilized centroids or a predefined maximum number of iterations. The goal is to minimize the sum of the squared distances between data points and cluster centroids. K-means can converge to a local minimum, and results can be affected by the initial selection of centroids, so it is common to run the algorithm several times with different initializations.

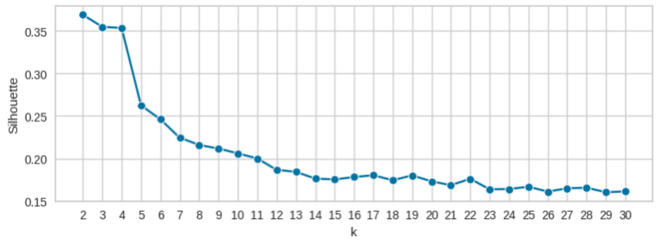
3.2.1 Finding the best k

The study to find the best value of k was done using *SSE* and *Silhouette*. The SSE is calculated by measuring the squared Euclidean distance of each data point to its nearest centroid and summing these squared errors. The Silhouette Score measures how well an object is aligned with its own cluster (cohesion) as opposed to other clusters (separation).

Below are the graphs with the metrics for the best K-study:



(a) SSE with k from 2 to 30



(b) Silhouette with k from 2 to 30

Figure 7: Comparison of SSE and Silhouette

Analyzing the graph 7a, a decreasing trend of the SSE value is clearly observed as the value of k increases. The values shown on the y-axis decrease rapidly, indicating a gradual decrease in the error. This decrease shows that as k increases, there is a significant improvement in how well the data has been clustered. Using the elbow method, it is possible to choose where to cut the graph to find the optimal k value. In this case, the choice was the third point, which corresponds to the value 4 on the x-axis. It's here that the change in the curve during the descent is most noticeable; in fact, the tail gets closer after that.

In the other graph, 7b shows the dynamics of the silhouette in response to changes in the number of clusters (k). The silhouette shows an initial increase to a maximum point and a subsequent decrease. Initially, it increases as the differentiation between clusters becomes more pronounced, reaching a peak suggesting optimal cohesion. Beyond this point, however, the silhouette decreases, indicating that an excessive number of clusters leads to overlap or clusters with few points. In conclusion, the graph suggests that the best k is 4 to optimize the number of clusters, as the value of the silhouette decreases rapidly after this point.

K	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
SSE	3840	2820	2165	1967	1822	1702	1609	1528	1467
Silhouette	0.37	0.35	0.35	0.26	0.24	0.22	0.21	0.21	0.20

Table 3: Values of SSE and Silhouette for different numbers of K

In conclusion, the analysis led to a sophisticated selection of the optimal number of clusters (K) within the considered dataset for the optimization of the K-means clustering algorithm. Although the parameter K=2 may have appeared to be the best choice at the initial analysis, further investigation revealed that by evaluating both the intrinsic cluster cohesion and the inter-cluster discrimination, the optimization materialized with K=4. This conclusion is based on specific quantitative evidence presented in the table 3, which shows a balanced trade-off between SSE and Silhouette for K=4. It is worth noting that while for K=5 the values of SSE do not differ significantly from those of K=4, the value of Silhouette shows a significant deterioration (-0.09). Conversely, for K=3, the Silhouette values remain unchanged from K=4, but the SSE shows a significant increase. Therefore, the final choice was K=4 as the optimal configuration for the clustering model.

3.2.2 Analysis of clusters obtained

This section continues with a detailed analysis of the clusters resulting from the application of the K-means algorithm. The table 4 gives an overview of the number of elements in each cluster. There is a significant imbalance, as cluster 1 contains twice as many elements as cluster 2. Despite this predominant imbalance, the other three clusters show a more even balance.

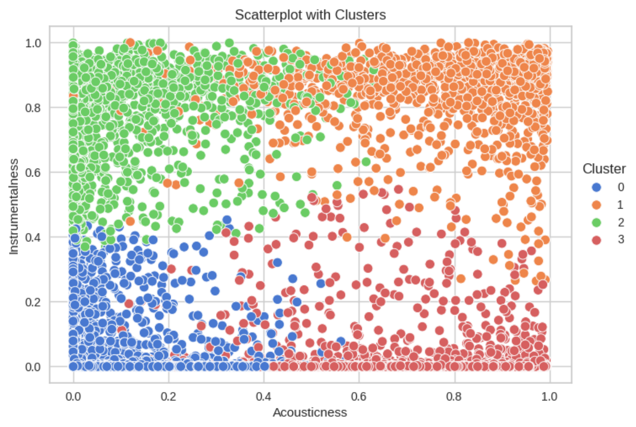
Examining the distance between the centroids in the different clusters, the graph 8b shows that the centroids of cluster 1 and cluster 3 tend to converge, with the exception of the *instrumentalness* features. This convergence is particularly evident for the features *danceability* and *tempo*, leading to an overlap of the clusters. However, outside of these cases, the centroids generally maintain a significant distance. This differentiation is most evident in the

Cluster	**0**	**1**	**2**	**3**
Elements in the cluster	1400	6183	2874	3067

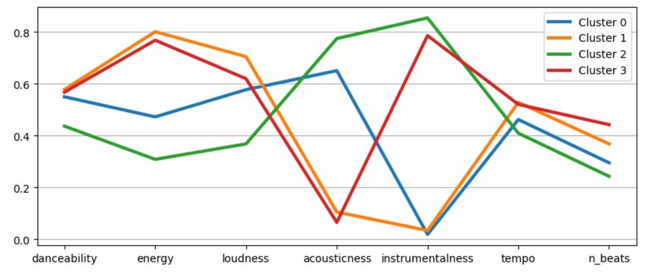
Table 4: Number of elements for each cluster obtained

acousticness and *instrumentalness* features, as shown in Figure 8b, where the divergence of the centroids allows for an accurate visual representation.

Chart 8a shows a scatterplot based on the above features, revealing a clear division of the data into four distinct groups, each skewed toward a specific corner of the graph. In the upper left, less vocal and more instrumental acoustic music is observed; in the upper right, even less vocal but more instrumental electric music is observed. In the lower left is less acoustic instrumental and more vocal, while in the lower right is electronic music.



(a) Pairplot with cluster



(b) Distance of centroids

Figure 8: Comparison of Pairplots with cluster and centroid.

In light of these considerations, it was interesting to examine the distribution of genres within the same clusters. As highlighted in Figure 9, Cluster 1 accommodates a prevalence of genres associated with music with a vocal predominance, including: j-idol (642), industrial (531), happy (495), and spanish (462). The second cluster, on the other hand, is characterized by a more pronounced acoustic component, attributable to genres such as: mandopop (478), bluegrass (425), indian (386), and brazil (260). The third cluster, closer to the electronic genre, has high frequencies for genres such as: techno (460), chicago-house (354), IDM (335), and black metal (323).

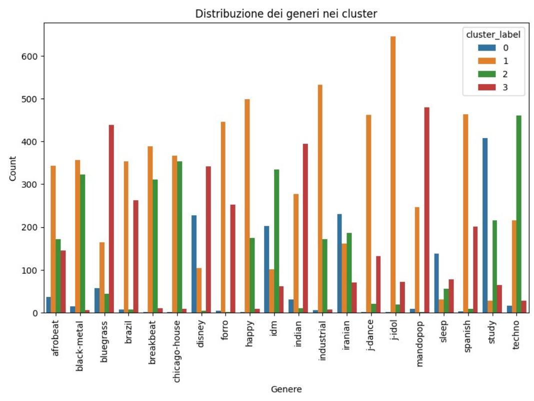


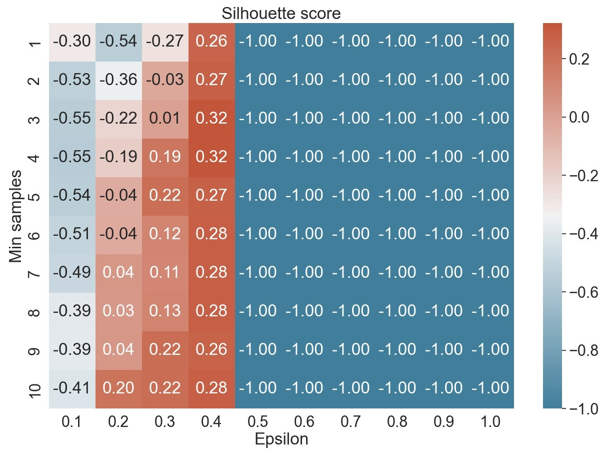
Figure 9: Distribution of clusters in the genre

Finally, the last cluster is predominantly characterized by instrumental music, although the precise data regarding this representation remains unspecified. In summary, the four clusters delineate different musical domains: cluster 1 corresponds to *pop* music (strong vocal component), cluster 2 to *traditional folk/acoustic* music (where mandopop, bluegrass, indian are present), cluster 3 to *electronic dance* music (techno, chicago house, IDM) and cluster 0 to compositions characterized by a strong instrumental nature (like genre *study*, close to lo-fi).

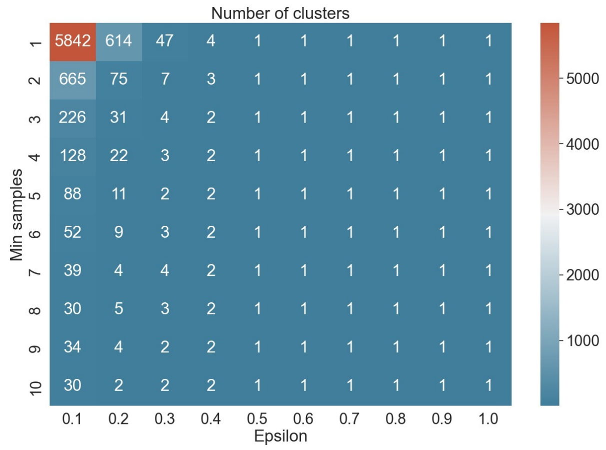
3.3 DBSCAN

Next the DBSCAN clustering algorithm was then applied. This clustering technique is based on the concept of density, as it groups points based on their relative density. The main parameters of the algorithm are ϵ (eps), which defines the radius of the "neighborhood" used to determine the density of points relative to a point x, and *minimum number of points*, which determines the minimum density required for a point to be considered a core point.

3.3.1 Study of clustering parameters



(a) Matrix for the study of Silhouette



(b) Matrix for the study of eps

Figure 10: Study of parameters for the DBSCAN

This subsection focuses on the analysis and study aimed at identifying the optimal parameters for the DBSCAN algorithm. Table 10a was created by comparing the silhouette values when varying the minimum number of samples and the Epsilon parameter. The most significant results are obtained with an Epsilon value of 0.4 and a minimum number of samples of 4, resulting in a Silhouette score of 0.32.

In the second matrix 10b, keeping the axes of ordinates and abscissas constant but using the number of clusters as focus, it has been observed that with the same values of Epsilon and minimum number of samples, the number of clusters is 2. In conclusion, the choice of parameters led to the adoption of Epsilon equal to 0.4 and the minimum number of samples equal to 4, configuring the optimal choice for the application of the DBSCAN algorithm.

3.3.2 Analysis of clusters obtained

As shown in the previous matrix analyses, the adoption of the optimal parameters mentioned above leads the algorithm to the formation of two main clusters. As can be seen in Figure 11, it is clear that the cluster labeled 0 turns out to be the most numerous, containing practically the entirety of the data. On the other hand, the other cluster, labeled -1 according to the specific DBSCAN operation, is identified as noise and consists of only 4 elements.

However, these results cannot be considered satisfactory. The presence of a very large cluster and a second very limited cluster highlights the limitations of DBSCAN, which is closely related to the concept of density and therefore negatively sensitive to situations of overlapping data. The presence of noise can also affect the overall performance of the algorithm.

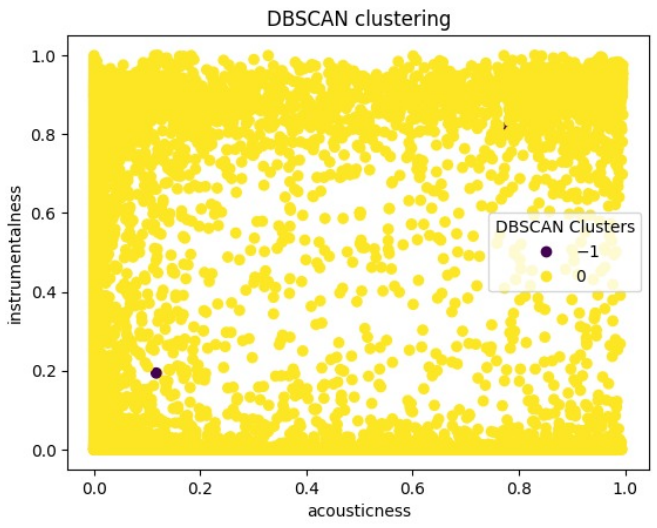


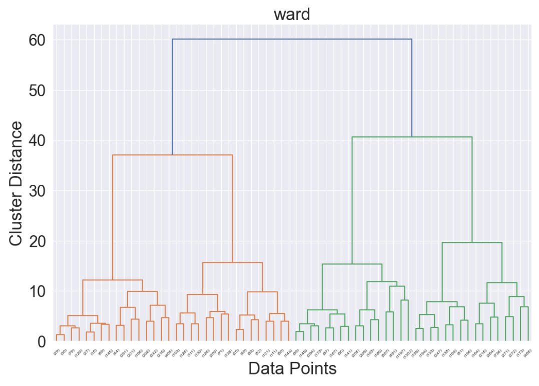
Figure 11: DBSCAN plot

3.4 Hierarchical

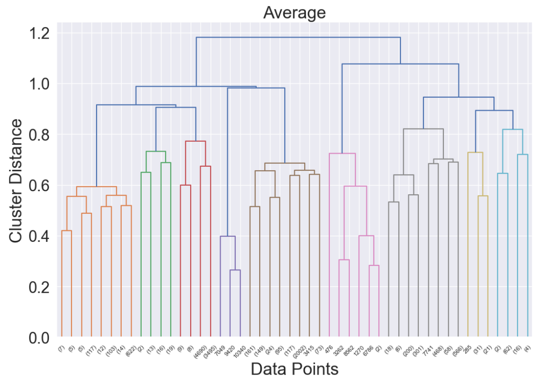
The third clustering method used, is the hierarchical method, which in turn allows for several possible types: single linkage, complete, average, and ward.

The first method used was the **single-linkage**, which turned out to be the worst, as it groups as many as 13,000 points into a single cluster. In fact, it is observed that the distances between them are quite small, so that the root of the tree is only slightly separated from the previous grouping, and so on. The high amount of noise may have created extensive chains, i.e. elongated or irregular clusters connecting distant points that are not really similar. For this reason, it was not considered appropriate to perform further analysis. The **average method** is found to be better, but still does not provide a clear and distinct differentiation of clusters.

The most effective results were obtained with the **Ward method** (Figure 12a), where the cluster merging minimizes the increase of the SSE, like the k-means algorithm. It is therefore no coincidence that four well-defined clusters can be observed with frequencies very similar to the four clusters generated by k-means: two of medium size, one larger and one smaller. Using always four clusters, the highest silhouette value was obtained with the **complete method** (Figure 13b). The problem is that setting a threshold of 1.75 to generate four clusters doesn't make much sense, since the subsequent mergers (which would form two clusters) would be very close in distance. A possible explanation is that it is indeed preferable to have two large clusters.

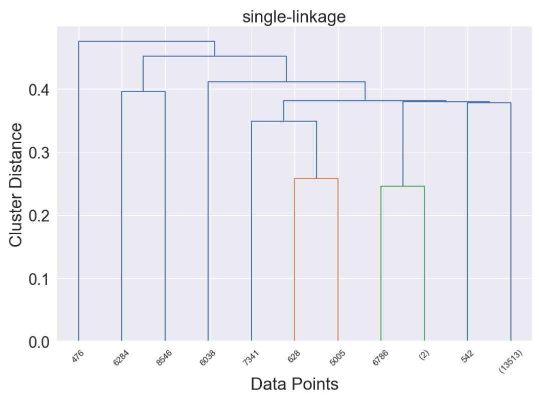


(a) Dendrogram with Ward method

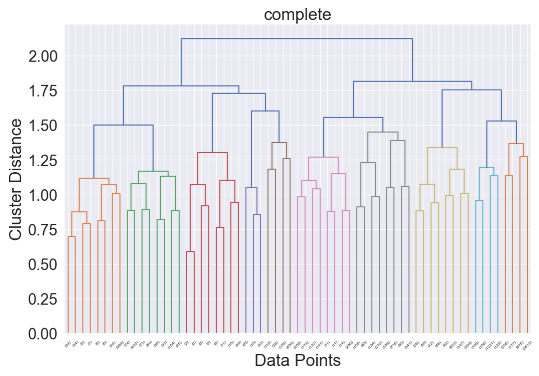


(b) Dendrogram with Average method

Figure 12: Comparison of Dendrograms with Different Linkage Methods



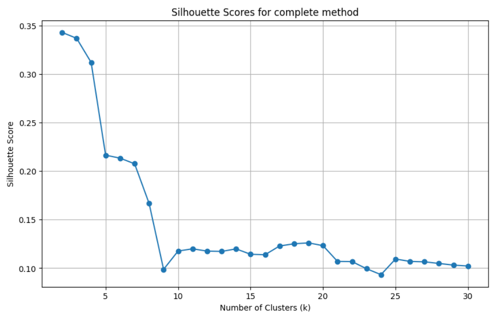
(a) Dendrogram with Single-linkage method



(b) Dendrogram with Complete Linkage method

Figure 13: Comparison of Dendrograms with Different Linkage Methods

The following are the best silhouette values for hierarchical clustering, as shown in the graphs below:



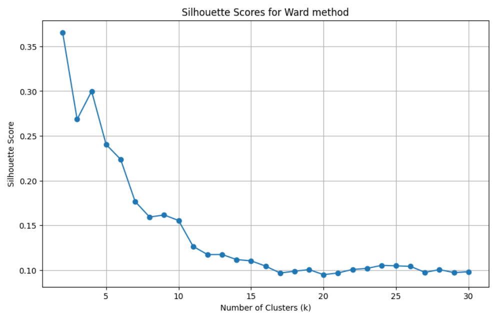


Figure 14: Number of rules based on confidence values

3.5 Conclusion

The best results were obtained with the k-means algorithm (silhouette: 0.35), from which three clusters with well-defined characteristics originated, plus another with more blurred boundaries, as described in section 3.2.2. DBSCAN showed some limitations in distinguishing between clear-cut clusters, probably due to the high density. During the data preparation for DBSCAN, the attempt to remove rows from the data frame belonging to cluster 0 obtained with k-means, which was generally identified as a source of annoyance for DBSCAN and a "noisy" cluster, i.e. with characteristics that are difficult to categorize, was unsuccessful.

As for hierarchical clustering, the Ward and Complete methods gave acceptable results. For the Ward method, when analyzing the genre distribution (with a threshold of 20), it is noted that the results are very similar to those of k-means, especially regarding merging, as already mentioned in section 3.4. For the Complete method, on the other hand, a discrepancy in the population level between all clusters is observed. However, even in this case, a qualitative similarity is observed: indeed, the most frequent genres in each cluster remain the same.

In summary, with the exception of DBSCAN, each cluster (except for the cluster 0, that is more heterogeneous) generated by the k-means and hierarchical clustering algorithms seems to convey some useful semantic information for grouping similar genres, through the features considered.

4 Classification

In this section, we will discuss the evolution of classification using three supervised learning algorithms: Decision Tree, KNN and Naive Bayes. The target variable chosen for the task was *genre*, on which work had already been done in the clustering task. Using only the results of the clustering, it was decided to train a classifier considering the top 5 genres for each of the three main clusters, excluding the cluster 0: j-idol, industrial, happy, spanish, j-dance, techno, chicago-house, idm, black-metal, breakbeat, mandopop, bluegrass, indian, disney and brazil. However, moving from the general (cluster) to the particular (specific genre) is always a risk: some genres, containing songs with diverse styles, are prominently present in both clusters. Therefore, opting for a simple reduction of the dataframe without considering all the possible shades between songs belonging to the same genre could be an oversimplification (this aspect will be further explored in the next section on data preparation).

4.1 Data preparation

The dataset was divided into a training set (70%), which is the part used to train the model, and a test set (30%), which is used to evaluate the model's performance. The variables chosen were those identified as "primary features" in paragraph 2.1 of the Data Understanding chapter: *danceability*, *valence*, *liveness*, *n_beats*, *tempo*, *energy*, *loudness*, *time_signature*, *acousticness*, *speechiness* and *instrumentalness*, not including *n_bars* because of its high correlation with *n_beats*.

As mentioned above, the reduction of the dataframe based on the given values of *genre* could be a trivialization. For this reason, it was decided to make a comparison between two different dataframes to understand if the performance of the classifier varies with the change of certain conditions. Both were filtered on the 15 musical genres, but while the first one was obtained by extracting only songs belonging to one of the 15 genres, the second one also considered an additional feature, *cluster_label*, where was placed a label of the song's cluster membership, based on the results of the k-means algorithm (for example, songs with the genre 'Black Metal' were added to the dataframe only if they were labeled with cluster_label = 2, which is the cluster where they were more frequently distributed). In the end, this dataframe contains a total of 6302 rows; for reasons of experimental consistency, it was decided to also standardize the number of rows for the other dataframe (without the label's cluster). Furthermore, the latter was randomized for each algorithm (selecting different songs each time) for 100 iterations to avoid any bias, showing the average results.

4.2 Decision Tree

For the initial classifier, the decision tree, parameter optimization was performed by applying the *random search algorithm* to the original training set. The goal of this technique, also known as grid search, is to find the optimal combination of parameters that maximizes the performance of the model. For the decision tree, the following parameters were considered:

- **min_samples_leaf:** minimum number of samples in a leaf;
- **min_samples_split:** minimum number of a samples for a split;
- **max_depth:**: the maximum allowed depth for the decision tree, representing the maximum number of levels between the root and the leaves;
- **criterion**: Gini or Entropy.

	Models			
	df without clusters	**df without clusters**	**df with clusters**	**df with clusters**
Criterion	Gini	Entropy	Gini	Entropy
Min Sample Split	50	30	5	30
Min Sample Leaf	1	5	10	10
Max Depth	9	8	8	10
Training Accuracy	1.0	1.0	1.0	1.0
Test Accuracy	0.39	0.38	0.51	0.49

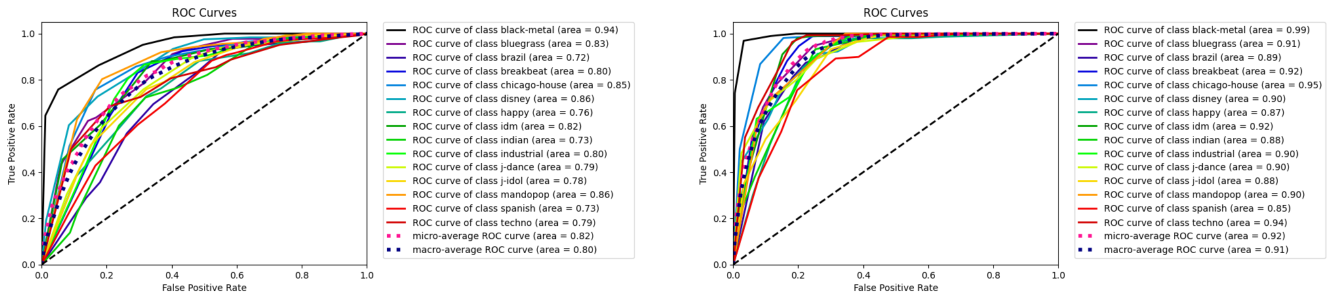
Table 5: Decision Tree's hyperparameters

As can be seen from table 5, using both Gini and Entropy as measures of impurity, the classification does not exceed 40% accuracy on the test for either dataframe without any association between genre and clusters. On the training set, the model clearly overfits in all four cases, but the performance on the test set is significantly improved by extracting genre in the manner described in section 4.1.

Delving deeper, the best results are achieved using Gini as the impurity measure and the following hyperparameters:

- Min_samples_split: 5
- Min_samples_leaf: 10
- Max_depth: 8

Below are the algorithm's Receiver Operating Characteristic (ROC) curves. This type of graph represents the trade-off between the true positive rate and the false positive rate as the model's decision threshold changes. A larger area under the ROC curve indicates better discrimination by the model.



(a) ROC curve of DT without cluster label (b) ROC curve of DT with cluster label

Figure 15: Comparison of ROC curves

In these graphs, the ROC curves of the two datasets can be seen and compared. In the first graph of Figure 15a, the curve values are quite good, with a micro and macro average of 0.82 and 0.80 respectively. On the other hand, as far as the ROC curve (Figure 15b) of the dataset with the clusters is concerned, the values improve significantly for the different classes, so that the macro and micro average also improve significantly, with values of 0.92 and 0.91. The improvements specifically concern the classes that were difficult to recognise with the first dataset, such as brazil where there is a + 0.17, Indian with + 0.15 or even techno with + 0.15.

4.3 K-NN algorithm

The K-NN algorithm, used as the second classification model, works by classifying a data point based on its proximity to neighboring data points. This proximity is determined by the parameter *n-neighbors*, which was systematically varied from 3 to 30 during implementation. In addition, value scaling was applied to ensure the effectiveness of the algorithm.

Similar to the decision tree classifiers, the K-NN algorithm performed a grid search to identify the optimal parameters. The results of this search, shown in Table 6, indicate the best performing configuration for the K-NN classifier. As with the Decision Tree, the best results are obtained with the frame with clusters.

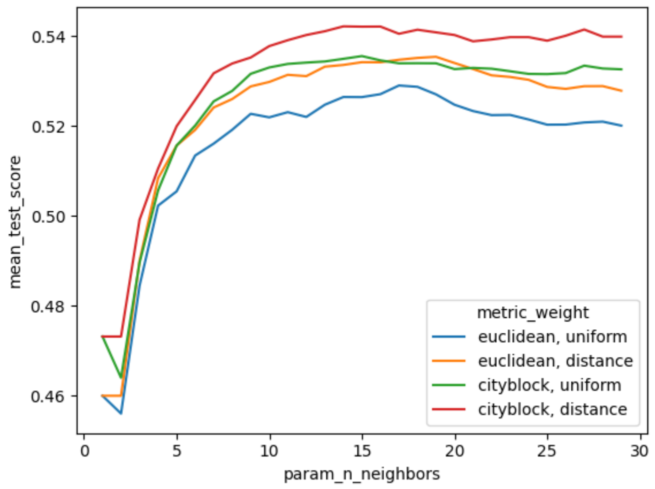


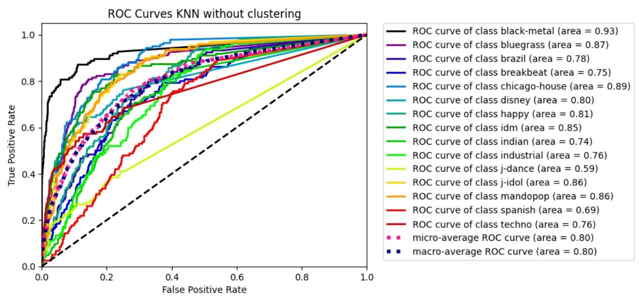
Figure 16: Different metrics results for df with cluster

The graphical representation in Figure 16 provides insight into the impact of different distance metrics on the number of neighbors and the accuracy score. Notably, the *cityblock* metric with a weighting scheme of *distance* yielded the best results, demonstrating the importance of carefully selecting these parameters for optimal K-NN performance.

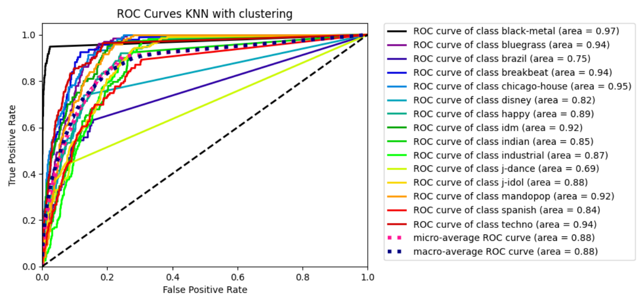
	Models	
	df without clusters	**df with clusters**
Metric	cityblock	cityblock
K-neighbors	22	14
Weights	distance	distance
Test accuracy	0.41	0.54

Table 6: Best hyperparameters for K-NN for the 2 df

As can be seen in the table 6, the specific hyperparameter values produced the best performance for the classifier. In particular, a metric *cityblock*, 22 neighbors and weights *distance* produced optimal results for the dataset without clusters, while the dataset with clusters benefited from 14 neighbors with the same metric and weights. In conclusion, the best results for the K-NN algorithm were obtained with the clustered dataset and the choice of hyperparameters just mentioned.



(a) ROC curve of K-NN



(b) ROC curve of K-NN'

Figure 17: Comparison of ROC curves

In the two graphs presented, we can see the ROC curves of the algorithm. Interestingly, the df with clusters improves the classification performance. A difference between macro and micro average of +0.08. The average improvement compared to the df without clusters among the classes is +0.10. At the same time, a data point that goes against the trend is the *brazil* element, which presents a deterioration (-0.03) in the df with clusters, contrary to the previous observations.

4.4 Naive Bayes

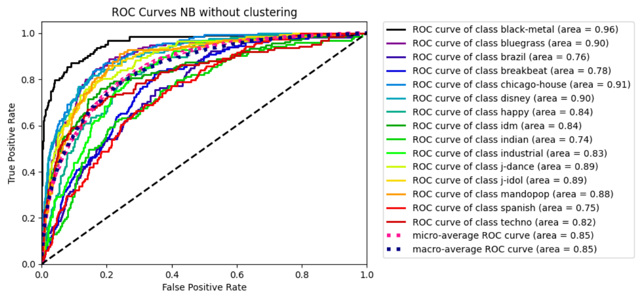
The Naive Bayes algorithm was subjected to a grid search focusing on a single parameter known as *var_smoothing*. This parameter represents the fraction of the largest variance of all features that is added to the variances during the computations, contributing to the stability of the model.

In particular, the initial results on the non-clustered data set were suboptimal. However, a significant improvement is observed when applied to the clustered dataset. This discrepancy suggests that the Naive Bayes classifier may benefit from the inherent structure provided by clustered data.

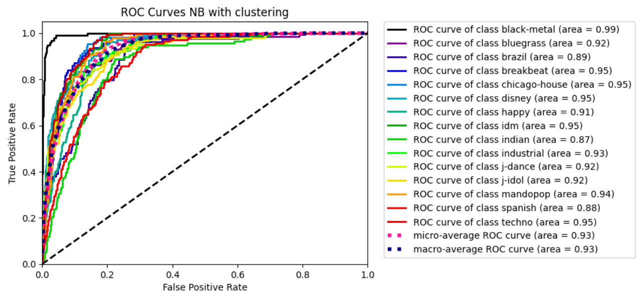
Models	df without clusters	df with clusters
Test accuracy	0.38	0.49

Table 7: Test accuracy for the two dataframes

As can be seen in the table 7, the evaluation metric, represented by the accuracy score, improved significantly when moving from the dataset without clusters to the one with clusters. Specifically, the test accuracy increased from 0.38 to 0.49, indicating that the Naive Bayes classifier performs better when applied to clustered data.



(a) ROC curve of NB



(b) ROC curve of NB'

Figure 18: Comparison of ROC curves

In these two ROC plots (Figures 18a - 18b) it is possible to compare the classification performance obtained by the two datasets. Again, an improvement of the AUC values is observed for the classification performed with the *df* containing the clusters. In particular, the micro and macro averages for the *df* without the cluster attribute are 0.85. However, for the second classifier, there is an improvement in both values of +0.08, reaching a total of 0.93 for both indicators of performance.

4.5 Conclusion

To tackle the classification task, it was decided to use the clustering results, in particular those of the k-means algorithm. For each different classification algorithm, two smaller data frames were compared, both reduced in size compared to the main frame, with the same number of rows and the same types of musical genres. The only difference, as mentioned in section 4.1, was in the filtering of the rows: in one case the rows were randomly sampled, while in the other the clustering labels were used (of course the column corresponding to 'cluster_label' was then removed).

The hypothesis anticipated in the clustering conclusions was that the representation of clusters could help (at least partially) to distinguish between different genres. "Partially" because songs belonging to the same genre do not always share the same characteristics. In the end, the hypothesis proved to be correct: training the classifier on songs with different characteristics, and thus easier to distinguish, improved the performance of all three algorithms.

In a first analysis, comparing the accuracy values among the different algorithms studied, it could be said that the best results in df with clusters were obtained with K-NN (0.54), then DT (0.51) and finally NB (0.49). But deepening the analysis with ROC curves revealed interesting observations. In fact, it is well known that accuracy is calculated as the ratio of the number of correct predictions to the total number of predictions. While ROC curve calculation is done by studying the relationship between TPR (*True Positive Rate*, the percentage of correctly classified positive examples) and FPR (*False Positive Rate*, the percentage of incorrectly classified negative examples). In light of these considerations, accuracy cannot be considered a fully reliable measure, but rather an indicative measure of general predictive ability. Instead, the ROC curve, taking into account the TPR and FPR, provides a more comprehensive view of the classification errors generated by the model. In fact, comparing the values of the *Area Under the Curve* (AUC), we can see that the best model for performance is the NB with micro and macro average of 0.93, then we have the DT with 0.91 and finally the K-NN with 0.88. Although the K-NN has a better accuracy, the AUC values, averaged also by errors, are lower than the other models. It is also interesting to note that the DT is the model that improves the most, micro and macro average, between df without clusters and with clusters (+0.11). In conclusion, although in a first analysis the K-NN might have been the best model for classification, further analysis showed that the NB and DT are better in classification than even the FPRs.

5 Pattern mining and association rules

This section focuses on *pattern mining*. After a first phase of *data preparation*, the most frequent itemsets have been extracted. Subsequently, employing the Apriori function, association rules have been derived from these itemsets.

5.1 Preparation of the dataset

For the analysis, the following attributes were taken into consideration: *explicit*, *key*, *genre*, *n_beats*, *duration_ms*, *danceability*, *energy*, *acousticness*, *valence*, and *tempo*. All variables, except for *explicit*, *key*, and *genre* (already inherently categorical or ordinal), have been discretized. For *tempo* and *danceability*, a two-level discretization (low and high) was applied. *Energy* and *valence* were discretized into three levels (low, medium, and high). Finally, the remaining variables (*duration_ms*, *acousticness*, *instrumentalness*, *liveness*, and *n_beats*) were discretized into four levels (very low, low, medium, and high). The values of explicit attribute, 1 and 0, were converted into *explicit* and *not explicit*

5.2 Frequent itemset and support

As a preliminary step, it was decided to determine the optimal minimum support parameter (min sup) for representing the dataset. To achieve this, it was chosen to correlate the number of generated itemsets with the minimum support threshold, varying it within a range from 0 to 100.

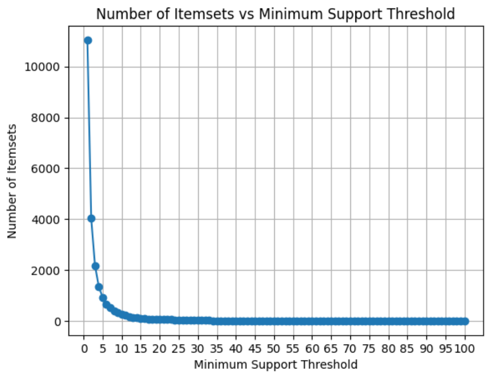


Figure 19: Minimum support in a range from 0 to 100

The obtained curve (Figure 19) reveals a discernible "elbow" in the range of values between 2 and 10. Consequently, the number of itemsets was computed for various minimum support thresholds identified within the specified interval of Table 8:

Min_support	**2**	**4**	**6**	**8**	**10**
Number of itemset	4033	1345	658	398	266

Table 8: Number of itemset for different values of min_support

Given the obtained results, it has been decided to use a minimum support of 2, allowing us to use more frequent itemset. From the results in the table, it can be observed that all the most frequent itemsets (ranging from length 1 to 7) always include the attributes danceability_low and not_explicit. Furthermore, it can be noted that the most frequent itemsets from 4 to 7 also contain the attributes acoustic_high and danceability_low.

Len_Trx.	**Itemset**	**Supp.**
2	danceability_low, not explicit	47.30%
3	tempo_low, danceability_low, not explicit	25.27%
4	acoustic_high, energy_low, danceability_low, not explicit	12.56%
5	danceability_low, acoustic_high, energy_low, tempo_low, not explicit	7.86%
6	acoustic_high, energy_low, valence_low, danceability_low, tempo_low, not explicit	5.24%
7	acoustic_high, duration_low, n_beats_low, energy_low, danceability_low, tempo_low, not explicit	2.67%

Table 9: Frequent Itemsets and Support

From this, it can be inferred that the majority of songs in the Spotify Track dataset exhibit the following characteristics:

- They are not very danceable (*danceability_low*);
- The lyrics of the songs don't contain vulgar or obscene language (*not explicit*);
- They make use of acoustic instruments (*acousticness_high*);
- They are perceived as having low energy (*energy_low*).

5.3 Association rules

From the obtained frequent itemsets, association rules were extracted by varying the confidence from 60 to 100. Confidence is simply the conditional probability that a certain pattern occurs given another: rules with a confidence below 60% are therefore not very informative. The relationship between Association Rules and confidence is depicted in the graphs below:

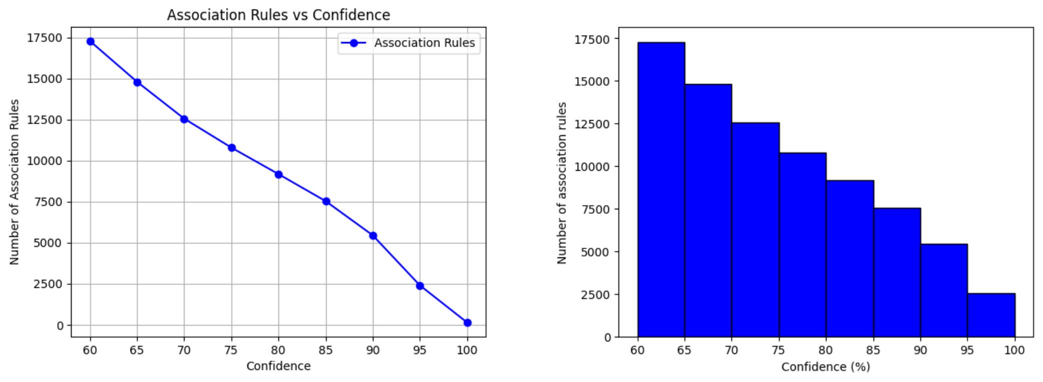
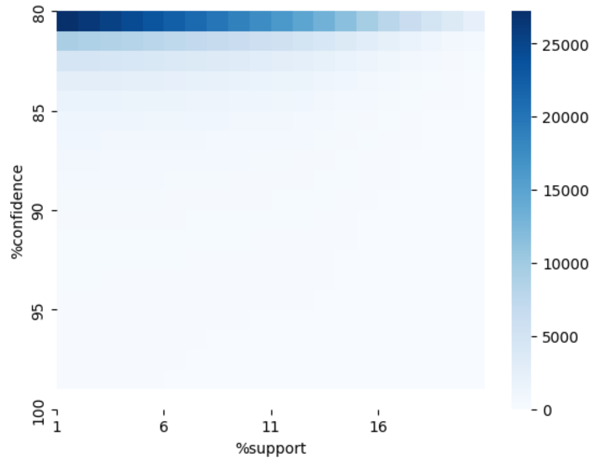
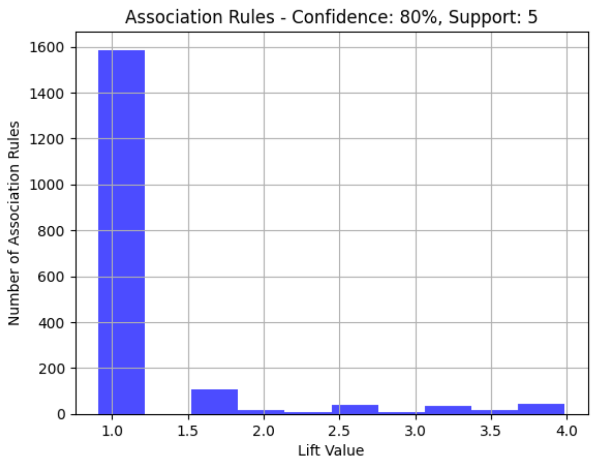


Figure 20: Number of rules based on confidence values

As it can be observed, the number of association rules naturally decreases with the increase in confidence. However, a substantial number of rules exhibit confidence values that are quite high, equal to or exceeding 80%. By narrowing down the analysis to this range, the relationship between confidence and support was then examined in depth. It turns out that Association Rules with high confidence exhibit elements that co-occur only a few times (support roughly exceeding 11%).



(a) Correlation between *confidence* and *support*



(b) *Lift* of association rules

Figure 21: Graphs about association rules

As regards the lift, the performance were evaluated with a support of 5 and a confidence of 80%. Many association rules are enclosed around the value of 1, indicating rules where the antecedent and consequent are independently associated. Only a small number of rules have a lift exceeding 1.5 (Figure 21a).

Among these, the most interesting ones have been selected, which are shown in Table 10:

Association Rules
n_beats_high, tempo_low → duration_ms_high (conf = 0.89) (lift = 3.97)
acousticness_high, duration_ms_low, energy_low, tempo_low, not explicit → n_beats_verylow (conf = 0.89) (lift = 3.98)
acousticness_verylow, tempo_high, 4.0_time_signature, not_explicit, duration_ms_high → n_beats_high (conf = 0.87) (lift = 3.90)
black_metal, danceability_low → acousticness_verylow (conf = 0.88) (lift = 3.5)

Table 10: Most interesting rules

5.4 Prediction of the target variable

The extracted rules were used to build a predictive model for our target variable 'explicit', which consists of the values *explicit* and *not explicit*. The accuracy of the model is 91%. However, due to the imbalance in the dataset, it was not possible to find rules containing the attribute *explicit*, since there are no rules with a confidence greater than 60% which have as a consequent the value *explicit*.

For this reason, undersampling was used, a technique that involves the removal of rows from the majority class: in our case, one-third of the rows in the dataframe containing the value *not explicit* were removed. Despite this, no improvements were obtained, so it was not possible to find rules with *explicit* as consequent. Therefore, songs could only be classified as *not explicit*.

6 Regression analysis

Regression is a statistical technique used to understand and predict the relationship between a dependent variable and one or more independent variables through a mathematical model that approximates this relationship. This model can then be used to make predictions on new data or to fill in gaps between existing data. The study of linear regression involves finding the equation of the line that best describes my data, bearing in mind that, given a set of data, I can draw infinitely many lines in space. Of course, such an analysis only makes sense if there is a moderate to strong correlation between the data. For this reason, it was decided to consider for our univariate regression analysis the variables *energy* (dependent variable) and *loudness* (independent variable), whose Pearson correlation coefficient is 0.31, in order to understand whether louder tracks, with higher volume, might be perceived as more energetic.

To assess the goodness of the model, it was used an index called the coefficient of determination (R-squared), which is the square of the Pearson correlation coefficient. This index explains what proportion of the variance of the dependent variable is explained by the independent variable: the higher it is, the more it explains the variability. Other measures for assessing model fit include MSE (Mean Squared Error) and MAE (Mean Absolute Error). Respectively, they represent the average of the squared differences between the model's predicted values and the observed values, and the average of the absolute differences.

For our two variables, as shown in Table 11, the R^2 is equal to 0.590 for both *conventional* linear regression and for Ridge and Lasso regression, two techniques of coefficient penalization in the model to prevent overfitting. Slightly lower results were obtained with Decision Tree Regressor (R^2: 0.300) and K-NN Regressor (R^2: 0.538).

For the multivariate regression analysis, instead, *energy* and *danceability* were chosen as independent variables, and *acousticness* as the dependent variable. The R^2 value of 0.301 indicates that the linear regression model can explain approximately 30.1% of the observed variation in the dependent variable *acousticness*. While this value is modest, it suggests that *energy* and *danceability* significantly contribute to predicting *acousticness*. Compared to linear regression, the decision tree model appears (as in the case to univariate regression) to explain less variation in the data (R^2: 0.091).

Model	R^2	**MSE**	**MAE**
Univariate Regression			
Linear Regression	0.590	0.027	0.129
Ridge	0.590	0.027	0.129
Lasso	0.590	0.027	0.129
Decision Tree Regressor	0.300	0.046	0.163
KNN Regressor	0.538	0.030	0.134
Multivariate Regression			
Linear Regression	0.301	0.073	0.226
Decision Tree Regressor	0.091	0.095	0.214

Table 11: Regression Performance Metrics