UNIVERSITÀ DI PISA

Department of Informatics
Master’s Degree in Data Science and Business Informatics

Project Report
Data Mining-1

Data Preparation, Understanding and Clustering

Authors:

Guido Trentacapilli, 668551

Mohan Upadhyay, 667179

Professor:

Riccardo Guidotti

Academic Year 2023-24

Contents

1 Data Understanding

2 Data Quality

3 Clustering

4 Classification

5 Pattern Mining

1. Data Understanding

The dataset for Spotify Tracks comprises information about audio tracks that are available in the Spotify catalog. These tracks belong to 20 different genres, including techno, idm, study, and black-metal. Each track comes with essential details such as its name, artist, album name, and popularity within the Spotify catalog. Moreover, the dataset incorporates audio-related characteristics, covering elements like danceability, energy, key, and loudness.

1.1 Data semantics

The dataset contains 15000 rows and 24 columns: each row represent an audio track, the columns represent the attributes of the tracks. In the following tables are shown the attributes, making a distinction among numerical and categorical attributes:

Attribute Name	Min Value	Max Value	Explanation
Duration_ms	8566	4120258	The length of the track in milliseconds.
Popularity	0	94	Describes how popular the track is from 0 to 100.

Danceability	0.0	0.98	Describes how suitable a track is for dancing from 0.0 to 1.0.
Energy	0.0	1.0	A measure from 0.0 to 1.0 representing the perceptual intensity and activity of the track.
Key	0	11	The key in which the track is.
Loudness	-49.531	3.156	The overall loudness of the track in decibels (dB).
Mode	0.0	1.0	Indicates the modality of the track, where 1 represents major and 0 represents minor.
Speechiness	0.0	0.939	Detects the presence of spoken words in a track.
Acousticness	0.0	0.996	A confidence measure from 0.0 to 1.0 indicating whether the track is acoustic.
Instrumentalness	0.0	1.0	A measure from 0.0 to 1.0 representing whether a track contains no vocals,
Liveness	0.0	0.994	Detects the presence of an audience in the recording.
Valence	0.0	0.995	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track.
Tempo	0.0	220.525	The overall estimated tempo of the track in beats per minute (BPM).
Features_duration_ms	8587	4120258	The duration of the track in milliseconds.
Time_signature	0.0	5.0	An estimated time signature.
N_beats	0.0	7348.0	The total number of time intervals of beats throughout the track.
N_bars	0.0	2170.0	The total number of time intervals of bars throughout the track.
Popularity_confidence	0.0	1.0	The confidence, ranging from 0.0 to 1.0, of the popularity of the song.

Table 1: Numerical attribute

Eighteen attributes are numeric:

- duration_ms and features_duration_ms: These two columns both appear to represent the duration of the track but with different names so are reduntant features.
- danceability, energy, loudness, speechiness, acousticness, instrumentalness, liveness and valence: These audio features can be essential for analyzing the musical characteristics of the tracks and may be valuable for genre classification or recommendation systems.
- key and mode: These can help analyze the musical key and modality of the tracks, which might be relevant for understanding musical patterns.
- time_signature, n_beats, and n_bars: These features can provide insights into the rhythm and structure of the tracks.
- popularity and popularity_confidence: Then popularity and the confidence level associated with it could be useful for assessing the reliability of popularity ratings.

In the following table (Figure 2) we summarize the categorical attributes, with their domain and explanation:

Attribute Name	**Type**	**Explanation**
Name	String	Name of the track.
Explicit	Boolean	Whether or not the track has explicit lyrics.

Artists	String	Whether or not the track has explicit lyrics.
Album name	String	The album name in which the track appears.
Genre	String	The genre in which the track belongs.

Table 2: Categorical attribute

1.2 Distribution of Variables and statistics

To enhance our comprehension of the dataset, we opted to depict the distributions of all variables using histograms and boxplots, which are showcased below.

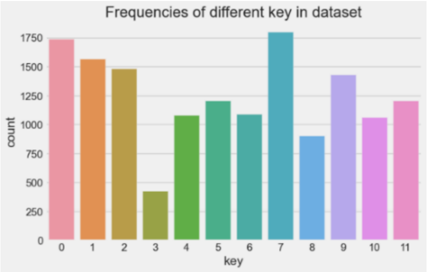


Figure 1: Distributions of key and count

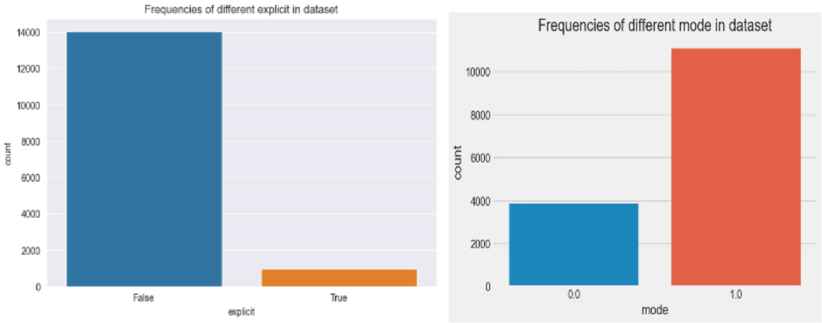


Figure 2: distributions of the explicit and mode attribute

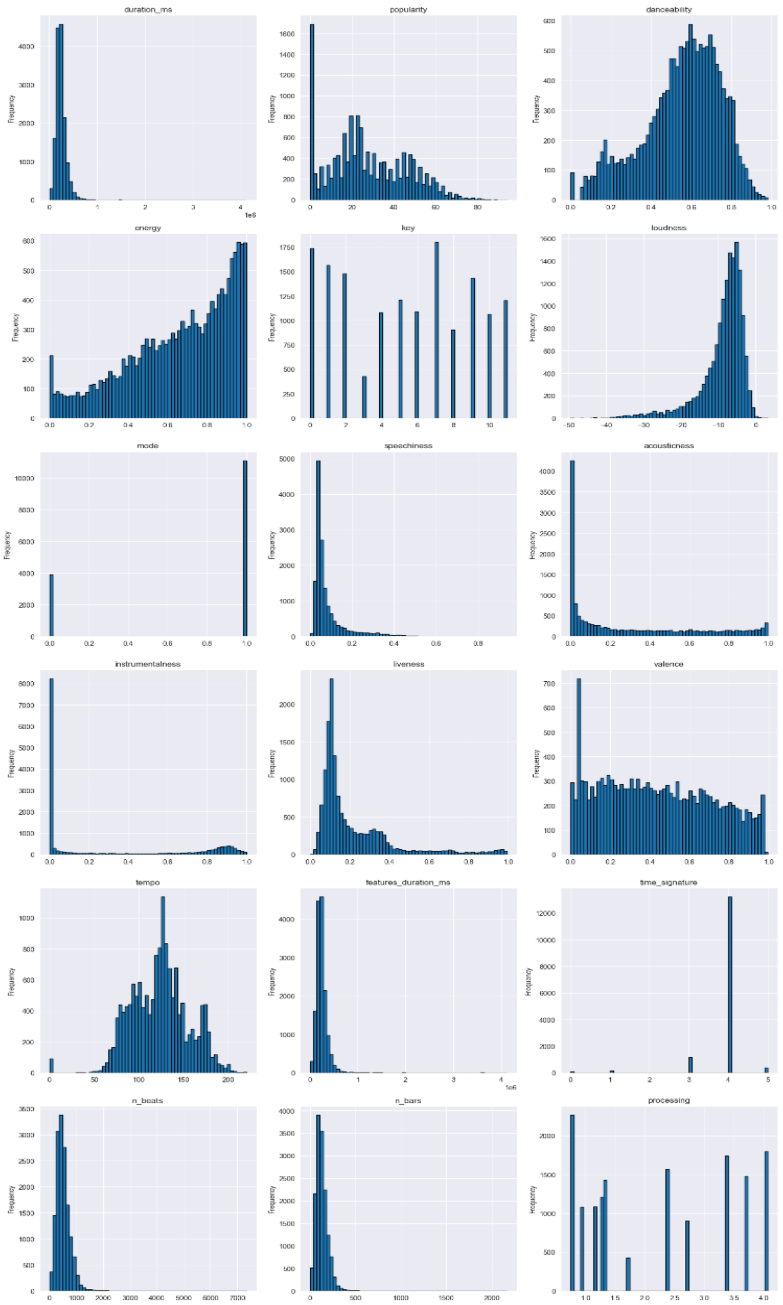


Figure 3: distributions of the numerical attributes

We further explored the relationship between various genres and their average popularity and duration in milliseconds. The visual representation highlighted that Indian, Brazilian, and Mandopop genres emerged as the top three most popular, while Chicago House, Iranian, and Black Metal genres secured the top positions for the highest mean duration in milliseconds.

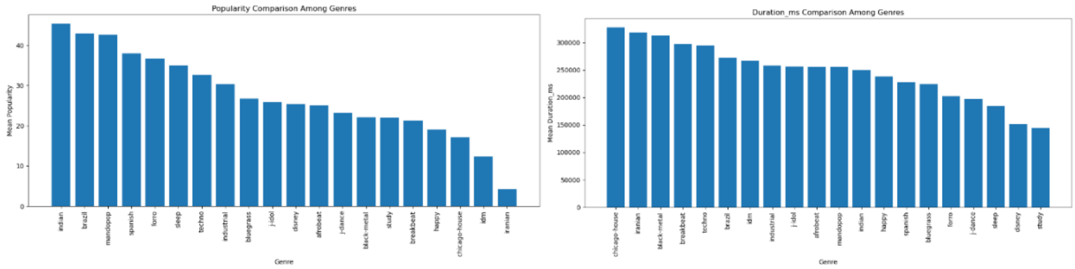


Figure 4: genres by mean popularity and genres by duration_ms

2. Data Quality

2.1 Missing values

Upon analyzing the dataset, it became evident that several variables contain missing values. Specifically, the 'mode' attribute has 4450 missing values, constituting 29.67% of all values; 'time_signature' has 2062 missing values, accounting for 13.75%; and 'popularity_confidence' has a substantial 12783 missing values, approximately 85.22% of its values.

Our strategy for handling missing values was straightforward. Firstly, considering the significant absence of values (85%) in 'popularity_confidence' and its redundancy with the 'popularity' attribute, we chose to drop the 'popularity_confidence' attribute. Secondly, for the 'mode' attribute with only two values (0 and 1), we filled the missing values with its mode, which is 1. Regarding 'time_signature,' both its median and mode are 4, and the mean is approximately 3.8, close to 4. Consequently, we opted to fill all the missing values in 'time_signature' with the value 4.

Attributes	Missing values in percent
time_signature	13.75%
mode	29.67%
popularity_confidence	85.22%

Table 3: percentages of missing values in the dataset

Given below is the bar plot of before and after treating the missing values.

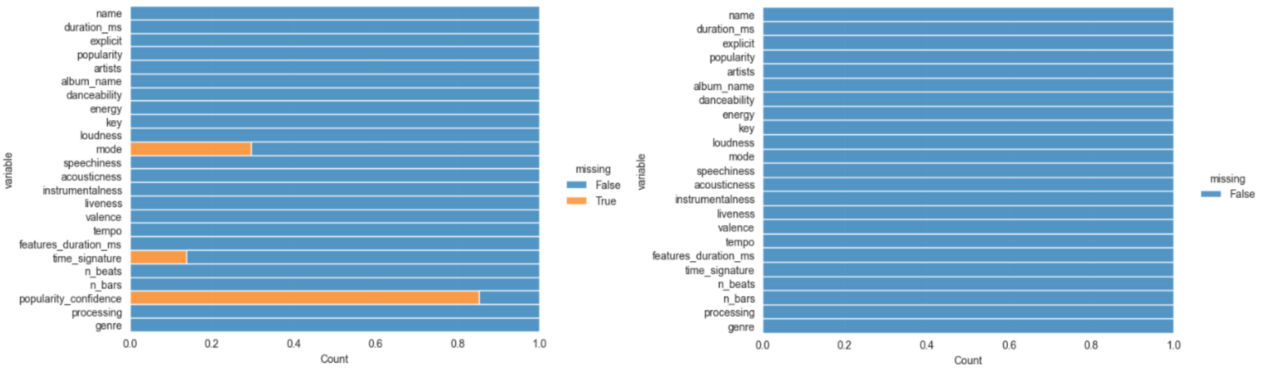


Figure 5: count of missing values in the dataset before and after treating

2.2 Variable Transformation

Our dataset encompasses categorical attributes, and it is crucial to transform them into numerical formats before applying clustering algorithms. This transformation is essential for the proper functioning of clustering algorithms and to facilitate the exploration of correlations among these attributes.

Specifically, we focused on transforming the 'explicit' and 'genre' attributes into numerical representations. For 'genre,' we encoded the values using an ordinal range from 1 to 20, corresponding to its 20 genres. This mapping schema proves valuable throughout our analysis, simplifying data manipulation tasks and aiding in the ease of plotting.

Similarly, for 'explicit,' we mapped its 'yes' and 'no' values to 1 and 0, respectively. This numerical encoding enhances the algorithm's ability to process the data effectively.

Additionally, we converted the 'duration' attribute, originally in milliseconds, to seconds. This conversion was implemented to streamline the dataset and reduce complexity, aligning it with common time units for improved interpretability.

2.3 Correlation Matrix

Having addressed missing values and converted relevant variables into numerical formats, we procee d to create a correlation matrix, offering a comprehensive view of relationships within the dataset. We explored two methods for plotting the correlation matrix—Pearson correlation and Spearman correlation—and found their outputs to be nearly identical in our analysis. Consequently, the choice between the two methods holds negligible significance, with only slight fluctuations observed.

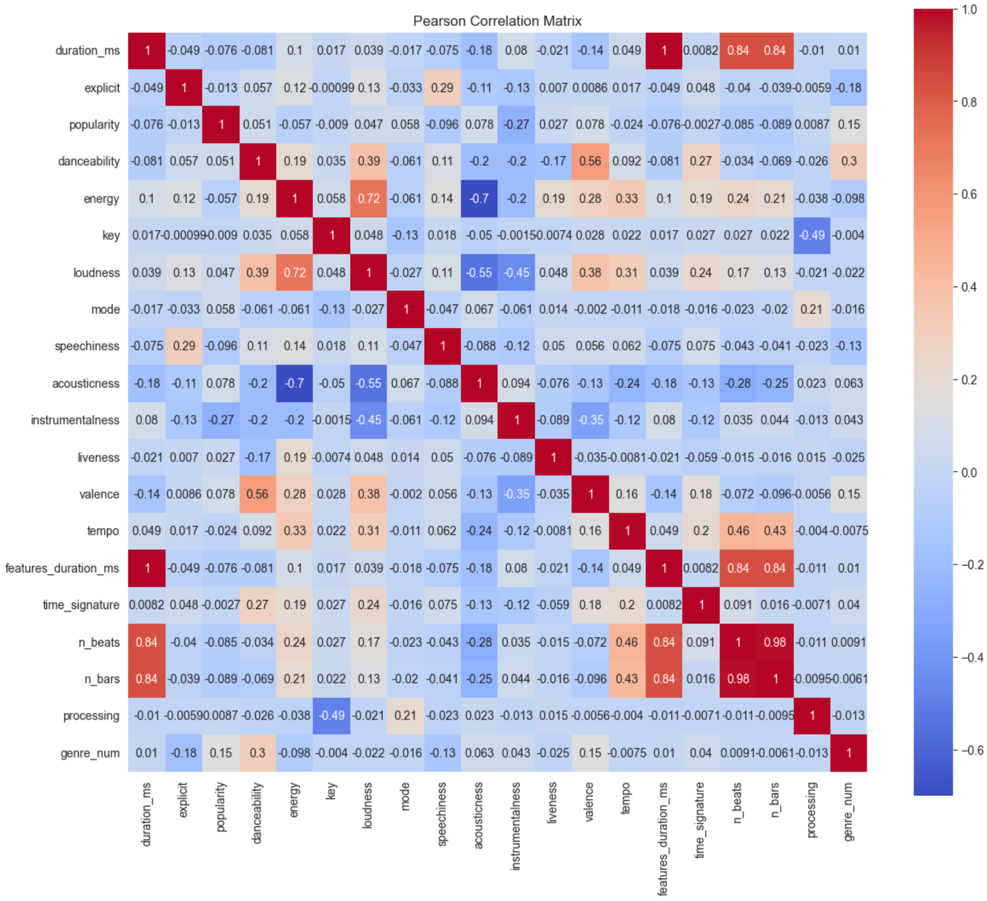


Figure 6: Correlation Matrix

As we may deduce from (Figure 3), very high levels of correlation, 98%, have emerged for the variables n_beats and n_bars: this is explained because the bar is a unit of measurement that defines a specific number of beats. The relationship between beats and bars contributes to the overall structure and timing of the music. Duration_ms is also highly correlated to this variables, 84%, because the higher is the number of beats, the longer the track will last. Lastly there is a good correlation between n_beats, n_bars and tempo, respectively 46% and 43%, because in faster-paced music more beats (and so bars) occur within a given time frame.

An interesting positive correlation also emerged between the attributes energy and loudness, 72%. This means that as the perceived loudness of the track increases, the intensity and activity (energy) of the music also tend to increase. This is intuitive, as louder music is often associated with more energetic and intense sound. Loudness has also good positive correlation with valence, 38%, and danceability, 39%, which are also positively correlated with a 56% correlation index:

- Loudness and Valence: This is explained because some genres may generally have louder or softer music that can be linked to the emotional tone .
- Loudness and Danceability: Higher loudness might be associated with more danceable tracks, especially in genres like electronic dance music (EDM) where loudness and danceability often go hand in hand.
- Valence and Danceability: In this case tracks with higher valence values (more positive or happy) could be perceived as more danceable.

Moving on to the side of negative correlations:

- Negative Correlation between loudness and acousticness (-55%):
  A negative correlation suggests that louder tracks are less likely to be acoustic. This aligns with the general expectation that acoustic music tends to be softer and less loud compared to non-acoustic or electronic genres.
- Negative Correlation between loudness and instrumentalness (-45%):
  A negative correlation implies that louder tracks are less likely to be purely instrumental. This is consistent with the notion that instrumental tracks, which may be more subdued, are not typically as loud as tracks with vocals or a full band.
- Negative Correlation between acousticness and energy (-70%):
  A negative correlation suggests that acoustic tracks are less likely to be highly energetic. This aligns with the expectation that acoustic music often has a more subdued and relaxed energy compared to non-acoustic genres.

2.4 Outliers

Examining the presence of outliers in the dataset holds comparable significance to addressing missing values. It ensures a more accurate representation of the data by preventing undue influence from extreme data points. As illustrated in the boxplot chart below, numerous attributes exhibit outliers. However, in our analysis, we opted to retain these outliers in the dataset, refraining from implementing any techniques to mitigate their impact. Although we experimented with removing outliers and subsequently conducting clustering, the outcomes proved unsatisfactory. Managing outliers requires careful consideration, and in our approach, we chose not to intervene with them to preserve the integrity of the data.

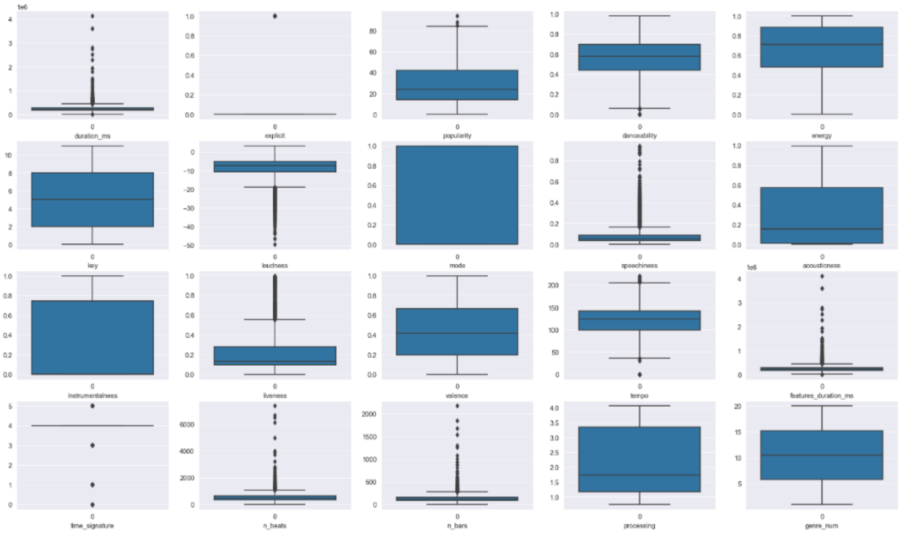


Figure 7: boxplots of the attributes

3. Data Clustering

In this segment, we will perform clustering to identify meaningful clusters within the dataset attributes. The objective is to discover common characteristics and uncover unique correlations that may have eluded detection in the previous phase.

Features Selection- The approach employed for attribute selection in our clustering methodology is straightforward. Initially, we conducted a comprehensive analysis of the correlation matrix, identifying attributes that exhibited significant correlations. Subsequently, from this pool of highly correlated attributes, we deliberately opted to include only one. The rationale behind this decision was to avoid redundancy in clustering, ensuring that the same correlations observed earlier would not be reiterated. Our aim was to cultivate distinct clusters, and therefore, the chosen attributes are intentionally devoid of any notable correlations among themselves or exhibit only minimal correlation. This meticulous attribute curation process enhances the uniqueness and specificity of the clusters generated in our project. Following attributes are taken "duration_sec", "danceability", "energy", "explicit", "popularity", "key", "mode", "speechiness", "instrumentalness", "liveness", "time_signature", "genre_num"

3.1 K-means

In the KMeans clustering phase of our project, we employed a strategic approach to determine the optimal number of clusters (k) for our dataset. Initially, we conducted a thorough analysis by plotting the distortion elbow graph against the range of k values from 1 to 11 (Figure 6). This exploration revealed a distinct elbow at k=3, with a corresponding distortion score of 83063.635, signifying a point of diminishing returns in terms of reducing within-cluster variance. To enhance the precision of our clusters, we applied the robust scaling technique, as it consistently yielded superior results. Subsequently, in our pursuit of evaluating cluster quality, we generated a second graph. This graph (Figure 5), plotting silhouette coefficient values against cluster labels, provided valuable insights. For the optimal k value of 3, we observed a silhouette coefficient of 0.29, indicating a reasonable degree of separation and cohesion among the clusters. These findings contribute to the robustness and reliability of our KMeans clustering results, offering a nuanced understanding of the underlying patterns within the dataset.

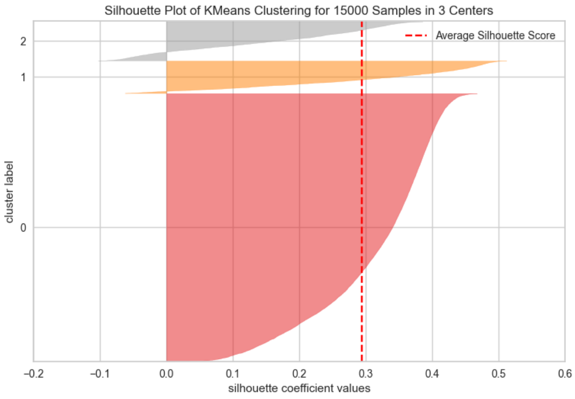


Figure 11: silhoulette of KMeans

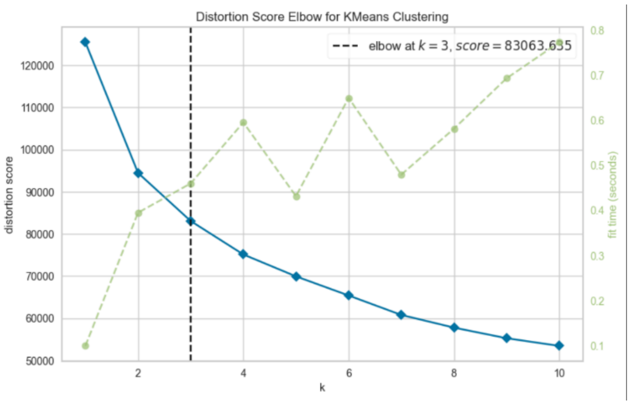


Figure 12: distortion score

In our comprehensive exploration of the dataset, considering all available attributes, we systematically experimented with various attribute combinations to achieve optimal visualizations. Notably, the most distinct and well-separated clusters emerged when plotting the attributes of speechiness and liveness. In the ensuing analysis, a graphical representation was generated, illustrating the positions of three centroids across all attributes. Each attribute distinctly revealed the separation among clusters A, B, and C. Specifically, the clusters exhibited notable distinctions in danceability, speechiness, and liveness. The scatter plot further visualized the clear demarcation between these clusters concerning liveness and speechiness. It's noteworthy that clusters B and C demonstrated similarities in several attributes, including Duration_sec, explicit, popularity, instrumentalness, and genre_num. Notably, the pronounced peaks of centroid A in speechiness and centroid C in liveness may be influenced by existing outliers. Despite the presence of outliers, our clustering approach successfully yielded well-separated clusters. Additionally, attempts to refine the results by removing outliers and conducting clustering did not yield satisfactory outcomes. This underscores the robustness of our methodology in accommodating data variability and outliers while achieving meaningful cluster separation.

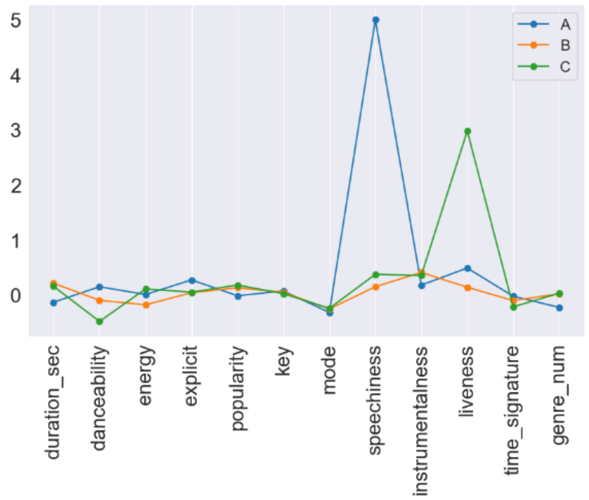


Figure 8: evaluation of attributes for clustering

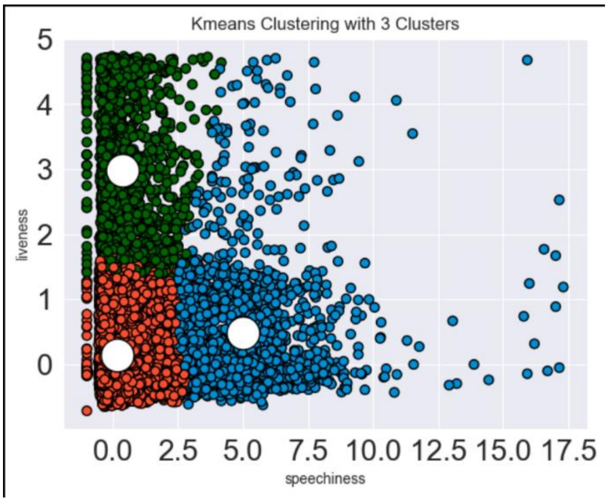


Figure 9: K-means Clustering

3.2 DBSCAN

DBSCAN stands out as a widely acknowledged clustering algorithm, prominently featured in scientific literature. This algorithm, requiring only the tuning of parameters epsilon (eps) and min points, plays a crucial role in identifying clusters within datasets. To determine optimal parameter values, we employed distance plots depicting the kth nearest neighbor distances against data point indices. Notably, the plot revealed an elbow point around 1.9. Consequently, we selected an epsilon value of 1.8 and set min points to 7. With this parameter combination, the algorithm achieved a silhouette coefficient of 0.326, indicative of well-defined clusters, and identified three distinct clusters.

In the resulting scatterplot (Figure 9), points were color-coded, with violet denoting noise points (labeled as -1) primarily concentrated in the upper regions of the plot. This observation suggested an elevated level of speechiness, indicating potential outliers. Other colors, specifically 0, 1, and 2, represented the three identified clusters.

Further exploration involved a comparison of clusters with categorical attributes. Notably, the attribute "explicit" exhibited a significant correlation, with 91% of its values aligning with the clusters formed by our DBSCAN algorithm. This finding underscores the algorithm's ability to reveal patterns and relationships within the dataset.

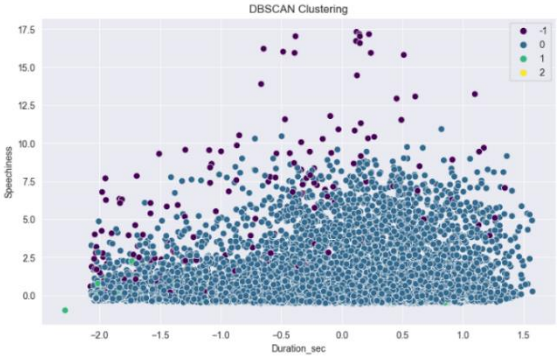
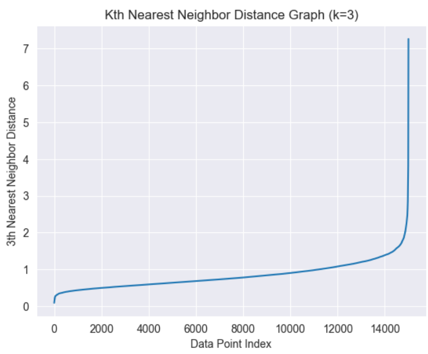


Figure 10: DBSCAN Clustering



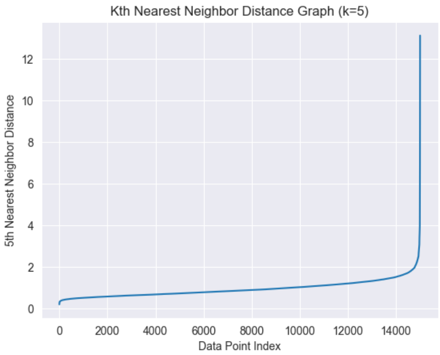


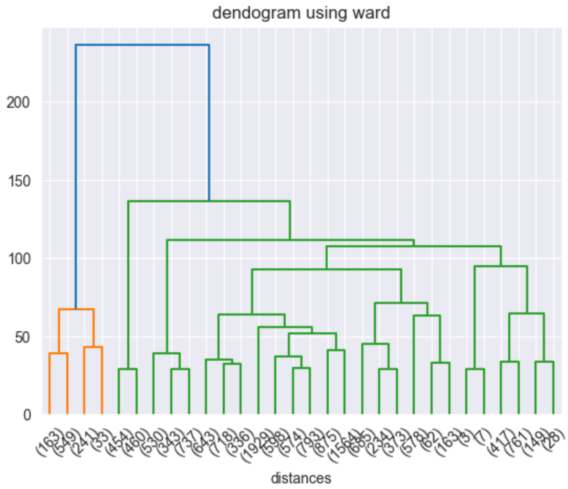
Figure 11: k nearest neighbour with k=3　*Figure 12: k nearest neighbour with k=5*

3.3 Clustering with agglomerative hierarchical technique

In our exploration of the Agglomerative Hierarchical Clustering technique, we leveraged the same set of 12 variables employed in both K-means and DBSCAN analyses. Employing the Euclidean distance function, we evaluated the performance of four linkage methods: single linkage, complete linkage, average linkage, and Ward's method. The dendrogram visualizations below illustrate the resulting hierarchical structures.

Upon examination, we found that this clustering technique did not yield significantly improved results. The clustering, observed across methods, exhibited a high degree of imbalance among clusters. While the group average method demonstrated the highest silhouette coefficient, indicating better separation, it led to very imbalanced clusters. Hence, it became apparent that relying solely on the silhouette coefficient might not provide a comprehensive assessment.

Ward's method emerged as the most promising among the hierarchical clustering techniques, boasting a silhouette score of 0.294. Despite the presence of imbalanced clusters, Ward's method demonstrated a more balanced distribution with cluster dimensions of 13,100 in the first cluster, 986 in the second, and 914 in the third. This finding positions Ward's method as a preferred choice for hierarchical clustering in our analysis.



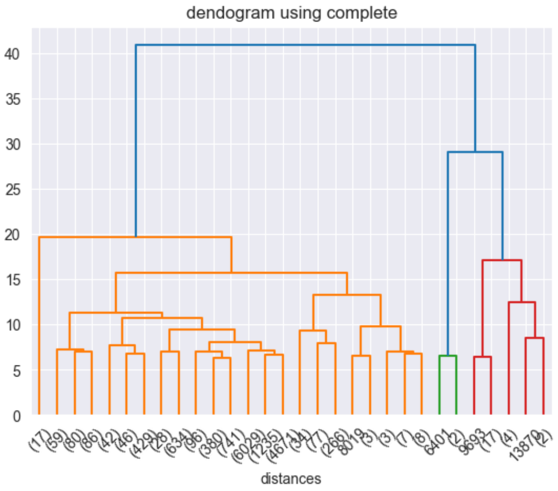
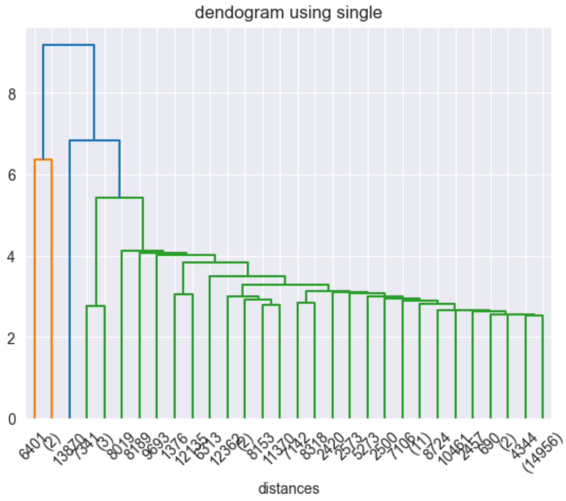


Figure 13: Hierarchical clustering with complete linkage　*Figure 14: Hierarchical clustering with average linkage*



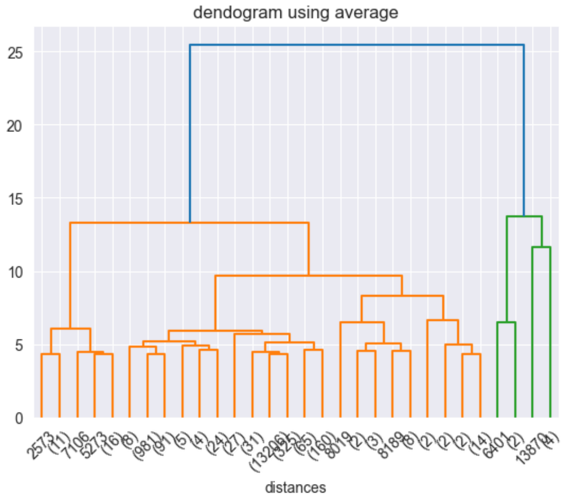


Figure 15: Hierarchical clustering with complete linkage

Figure 16: Hierarchical clustering with average linkage

	Silhouette Coefficient	Cluster Dimension		
		0	1	2
COMPLETE	0.716	14972	25	3
SINGLE	0.797	3	1	14996
GROUP AVERAGE	0.826	14992	5	3
WARD’S METHOD	0.294	13100	986	914

Table 5: comparison of different linkage methods

3.4 Final Evaluation of the best clustering algorithm

The table presented below details the silhouette scores obtained from various clustering techniques. Notably, the K-means technique yielded three clusters, each with balanced dimensions. However, the silhouette score was relatively low at 0.29. In contrast, hierarchical clustering produced a similar silhouette score of 0.29; however, the inherent imbalance in the clusters led us to dismiss it as a favourable result for our analysis.

Our primary inclination is towards DBSCAN, which boasts a higher silhouette score of 0.33. Despite the proximity in silhouette scores between K-means and DBSCAN, our preference leans towards DBSCAN due to its ability to identify balanced clusters and outliers effectively. In our comprehensive analysis, both K-means and DBSCAN emerge as reliable techniques for cluster identification. Nevertheless, the slightly superior silhouette score of DBSCAN and its adeptness at handling balanced clusters and outliers position it as the preferred clustering technique in our study.

Clustering Algorithms	**No. of Clusters**	**Silhouette Score**
K-Means	**3**	**0.29**
DBSCAN	**3**	**0.32**
Heirarchical	**3**	**0.29**

Table 6: comparison of different clustering algorithms

4. CLASSIFICATION

This section describes the results of the following classification algorithms: Naive Bayes(Gaussian), K-NN, and Decision Tree. The above models require the setting of a target variable that will be the starting point of the classification procedure. We choose to select the “genre” attribute as target variable and to run the classification alghoritms on it: since we have a dataset of songs with a lot of technical attributes, and with 750 songs for each genre (so the train set is perfectly balanced) we tried to predict the genre they belong building a multiclass classification problem.

4.1 Pre-processing

Before creating and testing our models we made some changes in our datasets.

1) Regarding the predictor models of the “genre” variable we performed a One Hot Encoding operation to substitute the “string” values of the target variable “genre” with numerical values.
2) We decided to drop some features that were not suitable or not useful for the classification process. We dropped all the categorical variables, except for “explicit” and, depending on the model, some numerical features, that we verified weren’t relevant for the predictions.
3) Regarding the “explicit” attribute we choose to substitute the “True” and “False” values with “1” and “0”, in order to simplify the analysis and modelling process.
4) We applied a standard scaling to our dataset, to eliminate scale differences that can negatively affect some algorithms, giving more weight to variables with larger scales.
5) Except for NaiveBayes models, we applied GridSearchCV (using as input the training set data) to find the best configuration of hyper-parameters that can maximize the performance evaluation metrics. Parameter optimization is inferred from the use of the” Repeated K-Fold cross-validation” methodology (with k equal to 5).

4.2 K-nearest-neighbours

We apply, as announced, the GridSearchCV algorithm on the classifier obtaining the following results:

	Best parameters values
N_neighbours	24
Metric	Manatthan
Weight	distance

Table 7: best configurations for GenreK-NN

Then we applied the inferred hyperparameters to the classifier that is trained on the training set and predicts on the test set. For the performance evaluation of the same, using the conventional metrics, we obtain the following results:

Model	**Accuracy**	**Recall**	**Precision**	**F1-score**	**AUC**
GenreK-NN	0.521	0,521	0.523	0.52	0.7478

Table 8: performance evaluation for GenreK-NN

The previous results can be verified looking at the Confusion matrix and at the ROC curves for the classification classes:

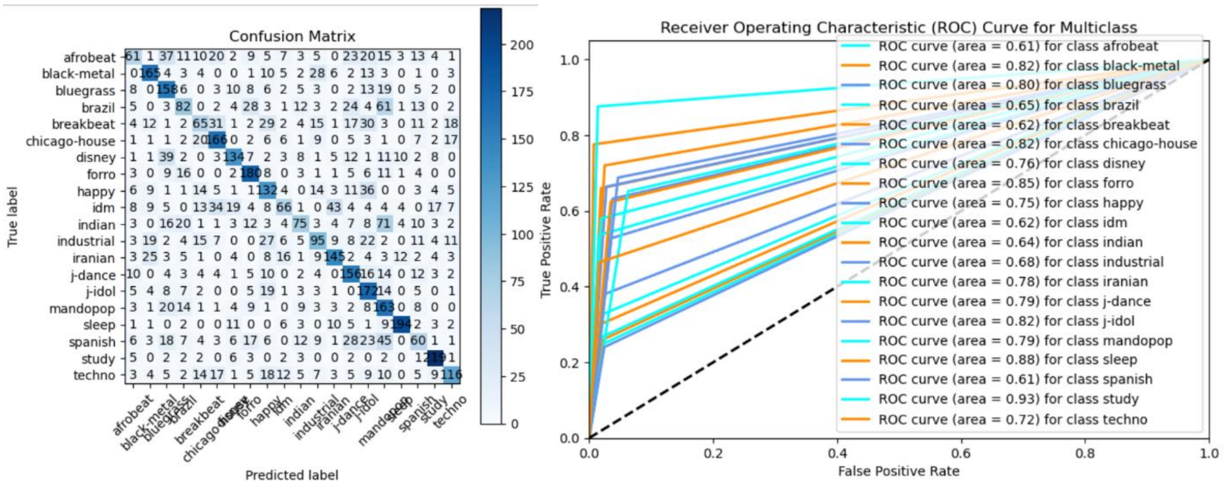


Figure 17: Confusion Matrix and ROC curves fro GenreK-NN

4.3 Naive-Bayes

For this type of classifier there aren't hyperparameters to set. but we evaluated the general performance

that the classifier achieves in class prediction:

Model	Accuracy	Recall	Precision	F1-score	AUC
GenreNB	0.2248	0.2248	0.467	0.22	0.592

Table 9: performance evaluation for GenreNB

An accuracy of 0.2248 means that the model correctly predicted the class for approximately 22.48% of the total instances. A recall of 0.2248 indicates that the model is capturing only about 22.48% of the instances belonging to the positive class. This suggests a high number of false negatives. A precision of 0.4668 means that out of all instances predicted as positive, approximately 46.68% are true positives. This suggests a relatively high rate of false positives. An AUC of 0.59 suggests that the model has some ability to discriminate between classes, but it may not be highly effective.

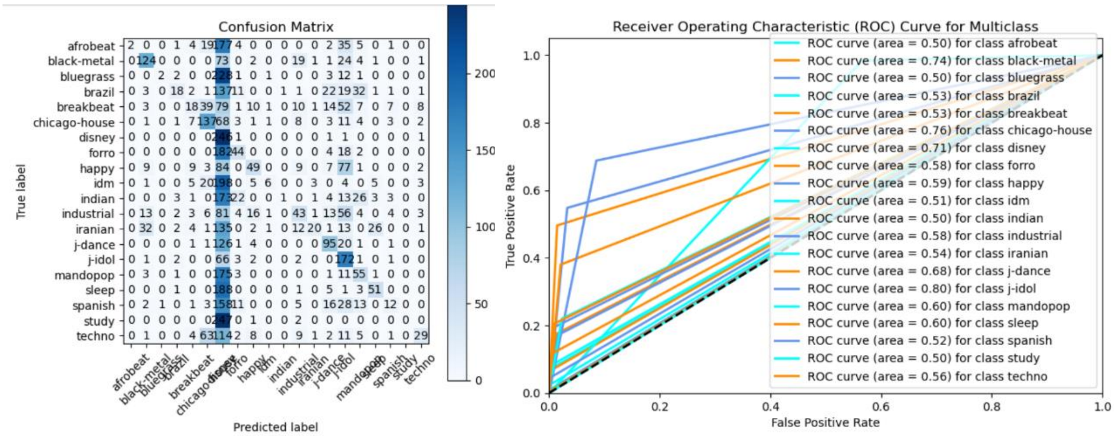


Figure 18: Confusion Matrix and ROC curves fro GenreNB

4.4 Decision-tree

For the decision-tree we applied, as announced, the GridSearchCV algorithm on the classifier obtaining the following results:

	Best parameters values
Min_samples_split	2
Min_samples_leaf	11
Max_depht	12

Table 10: best configurations for GenreDT

Then we applied the inferred hyperparameters to the classifier. For the performance evaluation of the same, using the conventional metrics, we obtain the following results:

Model	**Accuracy**	**Recall**	**Precision**	**F1-score**	**AUC**
GenreDT	0.4752	0,4752	0.4782	0.48	0.7238

Table 11: performance evaluation for GenreDT

The previous results can be verified looking at the Confusion matrix and at the ROC curves for the classification classes:

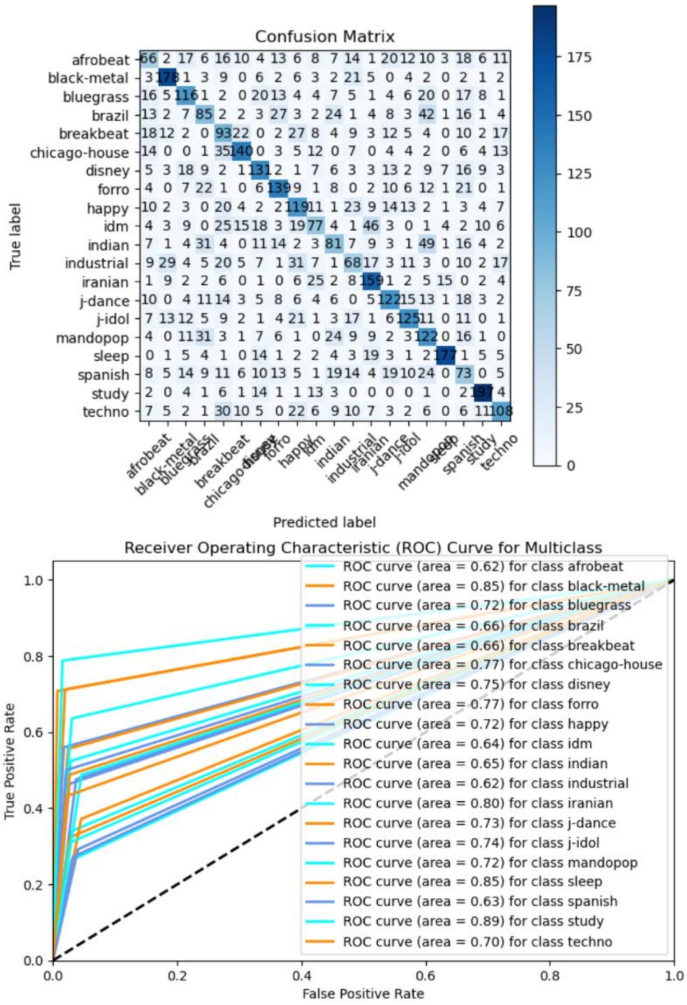


Figure 19: Confusion Matrix and ROC curves for GenreDT

For the decision tree algorithm, we found also the most relevant features and how much they influence the classification process. The results are shown in the following table:

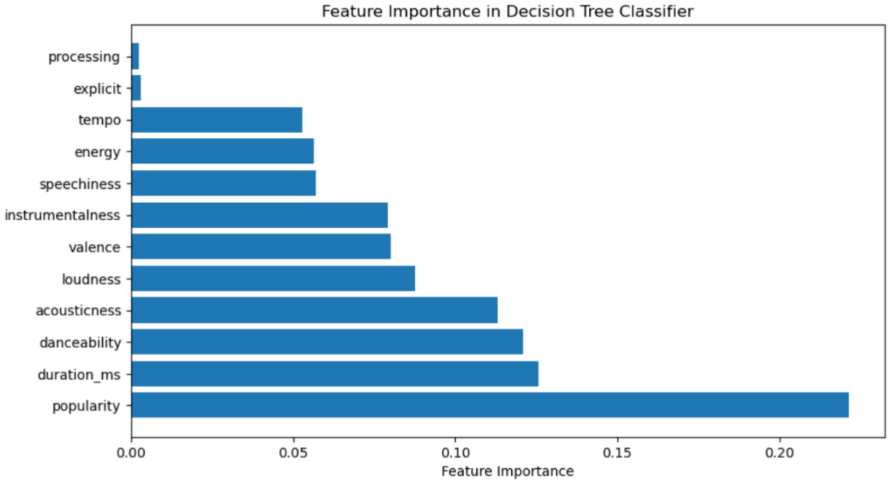


Figure 20: Feature importance for GenreDT

As we can see the most relevant feature is “popularity”, that influences the classification process by 22%, while the less relevant are “explicit” and “processing”, respectly by 0.3% and 0.25%.

4.5 Conclusions

The predictive algorithm that gave us the best results turns out to be the K-NN with an overall F1 macro level of 52%, although however the Decision Tree also record similar values. The Naive Bayes instead is the worst one with an obverall F1 score of 22%. As could be seen from the ROC curves, both K-NN and Decision Tree had curves with good values (74.78% and 73.28%), so they were able to distinguish the classes in a lot of cases. For the Naive-bayes we have an AUC of just 59.2%, so it couldn’t distinguish the class in more than 40% of the instances.

Moreover, looking at the AUC of the individual classes for the three classifiers, we observe that the easiest classes to predict were: black-metal (average AUC of 0.8), j-idol (average AUC of 0.786) and chicago-house (average AUC of 0.783). On the other hand, the classes that were more difficult to predict were: bluegrass (average AUC of 0.573), afrobeat (average AUC of 0.576) and spanish (average AUC of 0.586).

5. PATTERN MINING

In this segment, we opted to discretize the variables using the pandas qcut function and established frequency classes according to the quartiles of each variable's distribution. We chose to include all variables except "album_name," "name," "artists," and "features_duration_ms," as they were deemed unnecessary for our analysis.

5.1 Extraction of frequent patterns and analysis of the number of patterns with respect to the MinSup parameter

We endeavoured to identify frequent patterns employing various support values and typologies (frequent, closed, and maximal). Upon extracting the most frequent item sets, it became immediately apparent that with a low support (0.10), the initial item set in the extensive list (comprising 1754 item sets) showed that 47% of the item sets contained the variable "mode" with values "major" and "minor". Increasing the support to 0.15 led to a reduction in the list (now comprising 676 item sets), with 31% of the items retaining the "mode" variable. Subsequently, we experimented with incrementally raising the minimum support to 0.25, resulting in a further reduction to 15 item sets. Detailed values and calculations can be found in the table below.

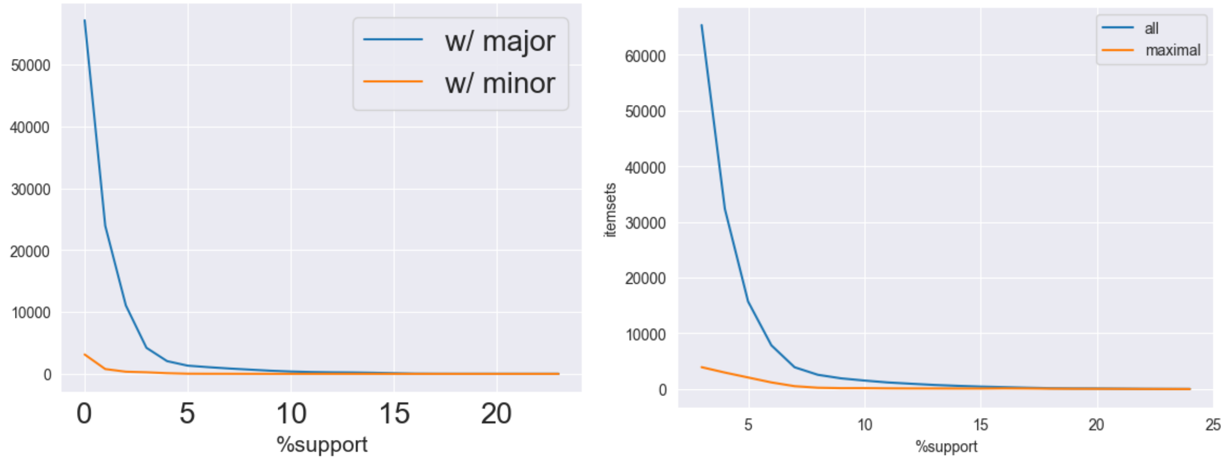


Figure 21

Support (supp)	**Number of Frequent Itemset**	**Value “Major”**	**Value “Minor”**
Supp =10	1754	826	13
Supp =15	676	212	1
Supp =20	94	12	1
Supp =25	15	11	0

Table 12: frequent patterns

With a support of 0.10, only 47% of the total frequent patterns observed include the necessary target values. When the support is increased to 0.15, this percentage decreased to 31% of the total frequent patterns. However, with a further increase in support to 0.20, only 13% of the total frequent patterns contain the required target values. Finally, at a support value of 0.25, we are left with 16 frequent itemsets, where 73% of the total frequent patterns include the necessary target values, specifically with the value "major" and the absence of "minor."

This trend is also evident when examining the number of maximal and closed itemsets for the two values of the variable as the support increases, as depicted in Figure 1. Below are some of these itemsets with support greater than 0.25, provided for further analysis and commentary (16 itemsets).

PATTERN				**SUPPORT**
(3.349, 4.067]_Proc	major	4.0_time	-	16.340000
(1.171, 1.739]_Proc	-	False	4.0_time	17.113333
(3.349, 4.067]_Proc	major	False	-	17.313333
(-0.001, 0.1]_instr	major	4.0_time	False	39.106667

Table 13: itemsets with support greater than 0.25

We experimented with various minimum support values and identified a subset of frequent patterns for analysis. To assess their significance, we specifically chose three frequent patterns, each possessing a support value exceeding 15. In the first pattern, three attributes—processing, mode, and time processing—are present. The processing value falls within the bin range of (93.349, 4.067], the mode value is major, and the time signature value is 4. Upon examining the correlation matrix, a noticeable positive correlation emerges between processing and mode, suggesting their tendency to co-occur. However, time signature exhibits a marginal negative correlation with both attributes. Despite this, in specific genres, instances of this combination may still be plausible.

Similarly, the second combination involves processing with explicit and time signature. Here, it can be rationalized by a modest positive correlation between time signature and explicit. Other variations may be conceivable in certain scenarios. However, an interesting observation is that, in all four patterns, the occurrence of "major" is linked to a higher value of processing. In the second pattern, where the processing value is smaller, "major" does not appear, hinting at a potential relationship or rule between higher processing values and the presence of "major." In the third pattern, a positive correlation between processing and mode seems to be the driving factor. In the fourth pattern, the possibility of the combination is influenced by a slight negative correlation between time signature, instrumentalness, and explicit.

5.2 Extraction of associative rules for different values of MinConf: RULES ACCORDING TO THE PERCENTAGE OF CONFIDENCE AND SUPPORT

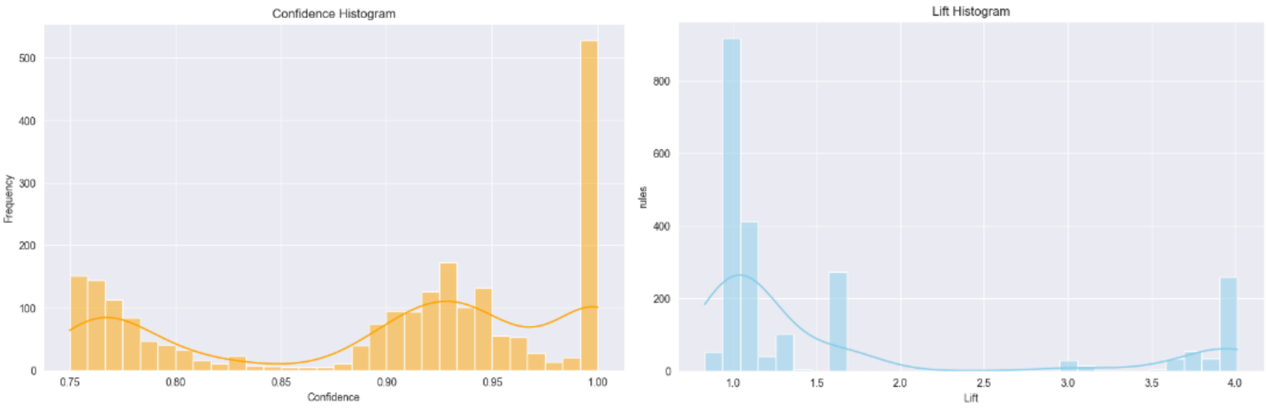


Figure 22: histograms of confidence and lift

In the context of extracting association rules, it was determined that a support threshold of 0.15 and a minimum confidence threshold of 75 were appropriate. Additionally, a minimum requirement of 3 sets per set was set to prevent the generation of trivial rules with excessively low thresholds. With these specified parameters, a total of 2222 rules were obtained. Furthermore, as illustrated in the graph presented in Figure 1, titled "Rules according to Confidence and Support," it becomes evident that opting for a higher confidence threshold would lead to a substantial reduction in the number of discovered rules. Examining the lift histograms (refer to Figure 1), it is noticeable that beyond the peak around 1.6, there were limited rules exhibiting high lift values.

Some rules are given below to comment on the semantics and any matches with other sections of the project.

RULES	ABS_SUPP	SUPP%	CONF	LIFT
“Major” (3.349, 4.067]_Proc, 4.0_time, False)	2304	15.360000	0.847059	1.143541
“Major” (3.349, 4.067]_Proc, 4.0_time)	2451	16.340000	0.846632	1.142965
“Major” (3.349, 4.067]_Proc, False)	2597	17.313333	0.845103	1.140900
“Major” (117.0, 159.0]_nbars, (461.0, 625.0]_nbeats, (-0.001, 0.1]_instr, 4.0_time)	1748	11.653333	0.765324	1.033198

Table 14: extracted rules

The presented rules follow the format Y » X, and each rule includes information on absolute support (the number of transitions), support percentage, confidence, and lift. An essential metric to consider is lift, as it represents the ratio between the observed support and the expected support if X and Y were independent. Rules with lift values close to 1 suggest independence between the antecedent and consequent, indicating a random association.

Analyzing the initial rule, a notable high lift is observed, coinciding with a higher value of processing, consistent with a previously identified trend. In this pattern, the positive correlation between processing and mode is evident. If any of these trends experiences an increase or alteration, there is a high likelihood of the mode being "major." The first rule, characterized by a confidence of 0.84, a lift above 1.10, and very high absolute support, implies a substantial probability for the consequent ("mode") to be "major" rather than "minor." In this context, the rule suggests that when processing falls within the interval (3.349, 4.067] and the time signature is 4, the mode of the song is highly likely to be major.

The subsequent rules follow a similar pattern, where the appearance of mode as major depends more on the value of processing within a specific interval than on the explicit language used in a song. Another intriguing pattern emerges, independent of processing, involving specific ranges of n_beats and n-bars, along with instrumentalness in the interval (-0.001, 0.01] and time signature as 4. Although the support for this rule is comparatively lower, it still presents a possibility.

5.3. Predicting Mode (Classification)

For predicting the mode, we used a K-NN classifier. First, we removed the categorical features except for genre and explicit, that were replaced by integers like for the previous classifiers. After that we used a balanced resampling with replacement too fill the missing values of the mode attribute. In the end we applied the GridSearchCV algorithm on the classifier obtaining the following results:

	Best parameters values
N_neighbours	10
Metric	Euclidean

Weight	uniform

Table 15: best configurations for ModeK-NN

Then we applied the inferred hyperparameters to the classifier that is trained on the training set and predicts on the test set. For the performance evaluation of the same, using the conventional metrics, we obtain the following results:

Model	**Accuracy**	**Recall**	**Precision**	**F1-score**	**AUC**
ModeK-NN	0.998	0,997	0.997	0.998	0.998

Table 16: performance evaluation for ModeK-NN

The previous results can be verified looking at the Confusion matrix:

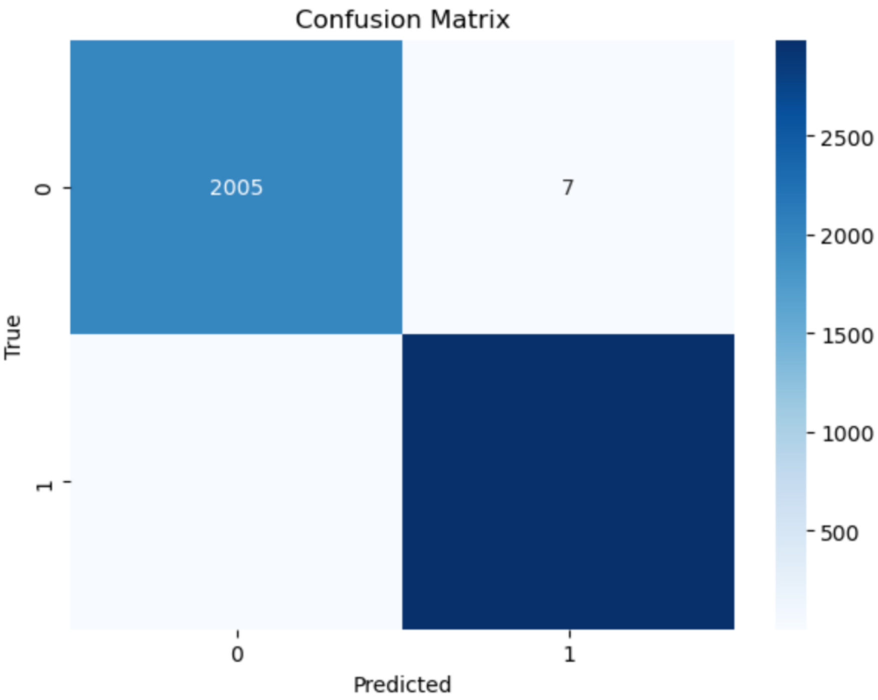


Figure 23: Confusion Matrix for ModeK-NN

6 Regression

The objective of this task was to forecast popularity. In univariate regression, we employed instrumentalness as the dependent variable. For multivariate regression, we utilized instrumentalness and genre_num, as they exhibited the highest correlation with the popularity attribute, as evident from the correlation matrix. Various regression techniques, including linear, ridge, lasso, as well as non-linear methods like k-nearest neighbors (KNN) and decision tree, were employed. We computed R-squared, mean squared error (MSE), and mean absolute error (MAE) values, and subsequently compiled them into a table for analysis.

6.1 Univariate regression

The initial experiment involved treating the independent variable X as popularity, which was identified as the feature most strongly correlated with the selected Y, instrumentalness. The results obtained from employing various regressors are presented below.

Measures	**Linear**	**Ridge**	**Lasso**	**KNN**	**Decision Tree**
R^2	**-0.68**	**0.071**	**0.052**	**-0.039**	**-0.139**
MSE	**366.285**	**312.626**	**318.972**	**349.695**	**383.271**
MAE	**15.820**	**14.586**	**14.794**	**15.211**	**15.763**

Table 17: comparison of regressors performances

As evident from the table, the ridge regression stands out as the optimal model, exhibiting the highest R^2 value along with lower MSE and MAE values compared to other regressors.

6.2 Multivariate Regression

In addressing this task, the variables instrumentalness and genre_num were considered due to their high correlation with popularity. The table below illustrates the outcomes obtained from employing different regressors in the analysis.

Measures	**Linear**	**Ridge**	**Lasso**	**KNN**	**Decision Tree**
R^2	**0.123**	**0.123**	**0.092**	**0.234**	**0.037**
MSE	**305.285**	**305.288**	**316.003**	**266.435**	**335.156**
MAE	**14.261**	**14.261**	**14.592**	**11.962**	**13.116**

Table 18: comparison of regressors performances

The table reveals that, among the various regressors, K-nearest neighbours (KNN) emerges as the superior model, displaying the highest R^2 value and lower MAE and MSE values in comparison to the other.

7 Conclusion

In summary, our primary objective was to predict popularity, considering it as the most crucial measure for gauging a song's potential success. We focused on the attributes most strongly correlated with popularity, namely instrumentalness and genre_num. Our observations indicate that the multivariate regressor outperformed the univariate regressor, demonstrating higher R^2 values and lower MAE and MSE values. Notably, negative R^2 values in the univariate regression may be attributed to outliers within our dataset, as discussed earlier.