IN SUPREMÆ DIGNITATIS

1343

UNIVERSITÀ DI PISA

DATA MINING 1

Project Report

Marco Palumbo (682968)
Kodjo Flaurent Dekadjevi (566556)
Bruce Charles Omogbolahan (683186)

Academic Year 2023/2024

Abstract

The report focuses on the analysis of a music data set containing songs on Spotify. It is divided into four parts. Initially, we perform exploratory data analysis on the data set providing: various information on the meaning of the variables, quality of the data, distribution of the variables, correlations between variables, and elimination of some of them. Secondly, we applied four different clustering methods, also applying Principal Component Analysis (PCA) for better 2D visualization. The next part concerns the classification of a target variable, based on three distinct methods and evaluating them based on the quantitative performance of the algorithms. After that, we perform association analysis on our dataset to see further hidden relationships in it in a way that can bring more knowledge and understanding of how different attributes and attribute values behave in our dataset. We conclude our report with regression analysis, where we utilize various regression techniques to predict continuous variables, providing a comprehensive model evaluation based on performance metrics such as R2 score, Mean Squared Error (MSE), and Mean Absolute Error (MAE).

Contents

1 Data Understanding & Preparation

This phase begins with a thorough exploration of the dataset to understand the nuances and meanings behind the variables, thereby ensuring the data's overall quality. Initially, we conduct a detailed analysis of the variable distributions, incorporating both discussions and visualizations to identify and interpret any outliers. This is followed by an examination of the correlations between variables, leading to the careful pruning of certain variables to enhance the dataset's relevancy and accuracy.

1.1 Data Semantics

Our study utilizes the Spotify dataset, comprising 15,000 tracks (rows) and 24 attributes (columns) per track. These attributes range from basic information like track name and artist to more complex features like acousticness, danceability, and energy levels. The dataset's size and diversity provide a rich ground for analysis, enabling us to draw meaningful insights about music trends and preferences. Each attribute will be meticulously examined for its significance and impact on our analysis, ensuring a comprehensive understanding of the dataset's structure and content.

1.1.1 Categorical Attributes

- **name**: Text identifier for each track, giving its title. It's a nominal variable that uniquely identifies the content.
- **explicit**: Binary variable indicating whether a song contains explicit lyrics/content (True or False).
- **artists**: Nominal variable containing the name(s) of the artist(s) who performed the track.
- **album_name**: Nominal variable that indicates the album a track belongs to.
- **key**: Ordinal variable indicating the key of the track. It has a numerical value that corresponds to a specific musical key.
- **mode**: Binary variable indicates the modality (major or minor) of the track, which can affect the mood conveyed by the song.
- **time_signature**: Ordinal variable that indicates the time signature of the track.
- **genre**: Nominal variable indicating the genre of the track.

1.1.2 Numerical Attributes

- **duration_ms**: Represents the length of the track in milliseconds.
- **popularity**: A quantitative measure reflecting how popular a song is, typically based on streams, downloads, or user ratings.
- **danceability**: Measures how suitable a track is for dancing based on tempo, rhythm stability, and overall regularity.
- **energy**: A measure of a track's intensity and activity.
- **speechiness**: Indicates the presence of spoken words in a track.
- **acousticness**: Measures the acoustic nature of a track.
- **instrumentalness**: Predicts the likelihood of a track having no vocals.
- **liveness**: Detects the presence of an audience in the recording.
- **valence**: Describes the musical positiveness conveyed by a track.
- **loudness**: Measures the average loudness of a track in decibels (dB).
- **tempo**: The overall estimated tempo of a track, measured in beats per minute (BPM).
- **features_duration_ms**: Appears to duplicate the duration_ms column.
- **n_beats**: Counts the total number of beats in the track, derived from the tempo and duration.
- **n_bars**: Measures the number of musical bars in the track.
- **popularity_confidence**: Reflects the reliability or confidence in the popularity score.
- **processing**: The purpose of this column is unclear. It could relate to data or audio processing steps taken in the track's production.

1.2 Assessing Data Quality

Assessing the quality of data is a critical first step in any data analysis process. It ensures the reliability and validity of the results obtained. In our project, we focused on the following key aspects to enhance our data quality:

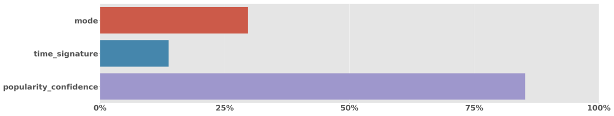
1.2.1 Errors

There are songs that have a loudness greater than 0. Since loudness is based on the dB scale, which does not allow positive values, we decided to remove these rows immediately.

1.2.2 Missing/NaN Values

We encountered missing values in three columns:

- mode: missing 4,450 values (29.7%)
- time_signature: missing 2,062 values (13.7%)
- popularity_confidence: missing 12,783 values (85.2%)



For the 'mode' column, NaN values were imputed based on the existing distribution:

- 1.0 (major mode) at 63.14
- 0.0 (minor mode) at 36.86

In the 'time_signature' column, we removed rows where the time signature was 0 (92 rows) or 1 (149 rows), as these are not standard in music. The remaining time signatures (4, 3, and 5) are more common in music, making the dataset more consistent for analysis. NaN values in this column were also filled based on their existing distribution:

- 4.0 at 87.89%
- 3.0 at 9.15%
- 5.0 at 2.95%

By imputing missing values in both columns based on their proportional distribution, we preserved the original data distribution, which is crucial for maintaining statistical accuracy in further analysis.

1.2.3 Data Types

Ensuring that each column in our dataset has the appropriate data type is crucial for accurate data analysis. Here's an overview of the data types and our adjustments to enhance consistency:

- **Object**: Columns like 'name', 'artists', 'album_name', and 'genre' are of type object. This is expected, as they contain string data representing categorical information.
- **Integer**: 'duration_ms', 'popularity', and 'features_duration_ms' are int64, suitable for numerical values that are integer counts.
- **Float**: Musical feature columns such as 'danceability', 'energy', 'loudness', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo', 'n_beats', 'n_bars', and 'processing' are float64. This type is appropriate for measurements that can have decimal values.
- **Boolean**: The 'explicit' column is bool, indicating binary categorical data (True or False).

Adjustments made for consistency:

- The 'mode' column, representing a binary category (major or minor), was initially a float. We converted it to bool, with True for Major and False for Minor, to accurately reflect its binary nature.
- 'time_signature', initially a float, contains discrete integer values (like 3, 4, 5). Considering its nature, we changed it to an integer type to represent these values more appropriately.

1.2.4 Semantic Inconsistencies

The "processing" column in our dataset lacked clear definition and relevance, making it unsuitable for analysis. Therefore, we removed this column to maintain data quality.

1.3 Distribution of the variables and statistics

We therefore begin with the distribution of the variables, which we distinguish into the 2 well-known macrocategories: categorical and numerical.

1.3.1 Categorical attributes

- **name** The Data Set originally contains 15.000 unique track names, however, after the data cleaning contains 14738.
- **explicit** The majority of tracks in the Data Set are non-explicit(93.6%), with a smaller portion being explicit.(Figure: 1.3.1)
- **artists** The Data Set features tracks from 6.172 unique artists. The figure 1 shows the uneven distribution of track counts per artist with Vybz Kartel leading with 78 tracks.
- **album_name** The Data Set contains tracks from 9.705 unique albums. It has a distribution similar to the artists bar chart.
- **key** The keys are fairly well distributed throughout the data set, with G major (Key 7) and C major (Key 0) being the most common.(Figure: 1.3.1)
- **mode** The majority of tracks are in a major mode(63,3%), with a smaller portion in a minor mode.(Figure: 1.3.1)
- **time_signature.** The predominant time signature is 4/4, followed by 3/4 and 5/4.(Figure: 1.3.1)
- **genre** The Data Set contains 20 genres with each one having 750 songs.

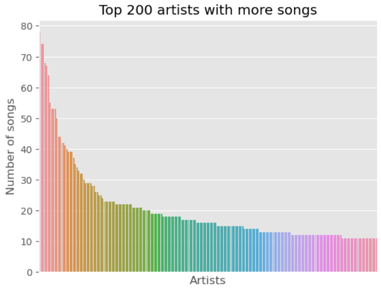
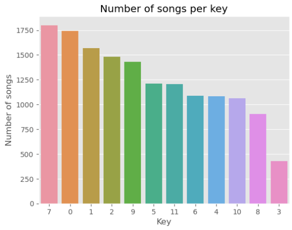
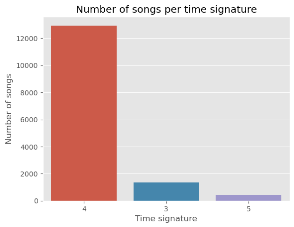
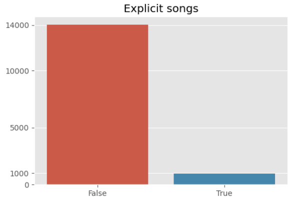
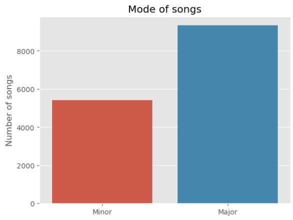


Figure 1









1.3.2 Numerical attributes(and relative outliers)

- **duration_m** The attribute's range is 4,111,672 ms, with a positive skew due to many outliers. Its median is 227,826 ms, and the MAD(Median Absolute Deviation) is 52,986.5 ms.
- **popularity:** Characterized by a positive skew, this attribute ranges from a minimum of 0 to a maximum of 94. The median value is 24, with a Mean Absolute Deviation (MAD) of 14.
- **danceability:** This attribute spans from 0 to 0.98, exhibiting a slight negative skew. The mean is 0.24, and the median is 0.58. The Mean Absolute Deviation (MAD) is 0.124, and the Standard Deviation (STD) is 0.194.
- **energy:** With a negative skew, this attribute's mode is 0.961, near the maximum. The median is 0.709 and the mean 0.656231, as shown in the energy histogram (Figure 2).
- **loudness:** The distribution of Loudness is left-skewed, with a mean of -8.895 dB and a median of -7.303 dB. Most songs range from -10.636 dB to -5.101 dB, but there's a tail extending to -49.531 dB, indicating quieter songs.
- **speechiness:** Right-skewed in the Spotify dataset, speechiness mostly ranges from 0.037 to 0.088, with a median of 0.051, suggesting a focus on lower levels, but including higher values (above 0.088).
- **acoustiness:** The acousticness distribution in the dataset is skewed to the right, with most songs having a relatively low acousticness score. This suggests that the dataset contains less acoustic music. The median acousticness score is 0.155.
- **instrumentalness:** The distribution in the dataset is notably right-skewed. The mean instrumentalness is 0.2867, with a standard deviation of 0.3829.

- **liveness:** This attribute's distribution is right-skewed, suggesting a prevalence of songs with lower liveness values. The mean liveness is approximately 0.21679, with a standard deviation of about 0.1952.

- **valence:** The distribution of valence is roughly symmetrical with a notable peak between 0 and 0.05, reflecting a balanced mix in the positivity of songs. The mean valence is 0.4368, with a standard deviation of 0.2772.

- **tempo:** The tempo plot has a normal distribution, with a peak of around 124 BPM, indicating that most songs have a moderate tempo.

- **features Duration (ms):** Most songs in the dataset fall within a duration range of 1.8 to 2.9 minutes, with a median duration of 2.3 minutes. This suggests a predominance of relatively short songs, alongside some that are shorter (below 1.8 minutes) or longer (above 2.9 minutes).

- **n_beats:** The distribution of n_beats is right-skewed with a mean of 501.86 and a standard deviation of 280.69. Additionally, 25% of songs have fewer than 327 beats, and 50% have fewer than 461 beats.

- **n_bars:** The distribution of `n_bars` is right-skewed with a mean of 128.39 and a standard deviation of 75.11. The 25th percentile is at 83 bars, indicating that 25% of songs have fewer bars, and the median (50th percentile) is at 117 bars.

1.3.3 Outliers

Our analysis of the Spotify dataset revealed a significant presence of outliers across various attributes. Outliers are data points that significantly deviate from the majority of data, often representing unusual or rare occurrences. In our dataset, these outliers manifest differently across attributes and require careful consideration.

Outliers in attributes like 'popularity' are mostly legitimate and explainable. For instance, tracks with extremely high popularity scores represent a small number of widely recognized and frequently played songs, standing out distinctly from the general distribution of this attribute. This deviation is not indicative of noise or error but reflects the real-world popularity dynamics in the music industry.

As depicted in the boxplot (Figure 4), nine attributes exhibit notable outliers, predominantly near their maximum values. Attributes such as 'speechiness', 'liveness', 'n_beats', 'n_bars', 'duration_ms', and 'popularity' show this trend. Conversely, 'danceability' and 'loudness' have outliers near their minimum values. The 'tempo' attribute uniquely exhibits outliers on both ends of its spectrum.

The identification and understanding of these outliers are crucial in the preprocessing phase. It informs our decision on the most appropriate transformation methods to standardize the dataset. Additionally, it emphasizes the need for robust statistical measures that are not overly sensitive to outliers, such as median or interquartile range, over mean or standard deviation. This approach ensures a more accurate and representative analysis, accommodating the diverse nature of musical attributes while minimizing the distortion caused by extreme values.

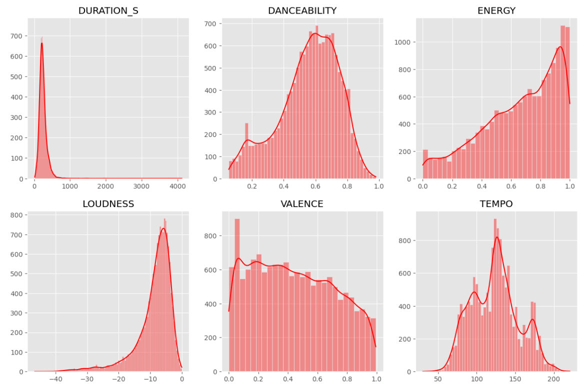


Figure 2: Histograms of different attributes.

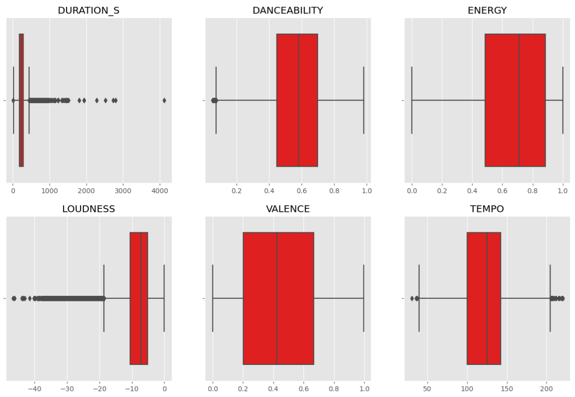


Figure 3: Boxplots of different attributes.

1.3.4 Variable Transformation

In the process of preparing our Spotify dataset for more effective comparability and correlation analysis, we have employed variable transformation techniques, with a specific focus on standardization using the Robust Scaler. This method is particularly suited to our dataset's characteristics, which include a significant presence of outliers. The choice of the Robust Scaler over more conventional scaling methods, such as Z-score normalization, is informed by the nature of our data. Outliers in our dataset can greatly skew the mean and standard deviation, leading to inaccurate scaling results. The Robust Scaler, in contrast, uses the median and the interquartile range (IQR) for scaling. The IQR, defined as the range between the 25th and 75th percentiles, offers a measure of statistical dispersion that is less influenced by outliers.

The formula for the Robust Scaler is:

$$\boldsymbol{x}_{\textbf{scaled}} = \frac{\boldsymbol{x} - \textbf{median}(\boldsymbol{x})}{\textbf{IQR}(\boldsymbol{x})}$$

1.4 Pairwise Correlation and Elimination of Variables

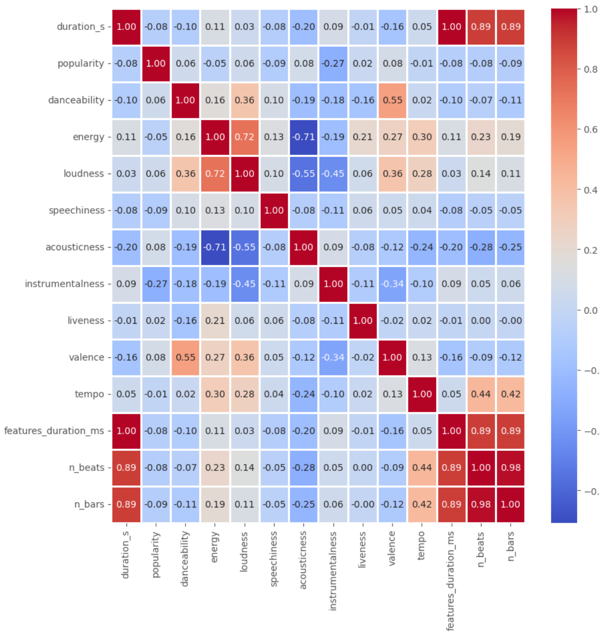
The pairwise correlation matrix provides insights into the relationships, among numerical variables in our dataset. Here, we highlight some notable observations:

duration_ms and n_beats/n_bars: There's a strong positive correlation between 'duration_ms' (or 'features_duration_ms') and 'n_beats'/'n_bars'. Longer tracks naturally have more beats and bars, so this correlation was expected.

Energy and Loudness: A notable positive correlation is observed between 'energy' and 'loudness'. Tracks with higher energy usually have greater loudness, aligning with our understanding of these musical attributes.

Danceability and Valence: There's a moderate positive correlation between 'danceability' and 'valence'. Tracks that are more danceable tend to exhibit higher levels of positivity or happiness.

Acousticness and Energy: An inverse relationship exists between 'acousticness' and 'energy'. This indicates that tracks with a higher degree of acousticness generally exhibit lower energy levels.



	duration_s	popularity	danceability	energy	loudness	speechiness	acousticness	instrumentalness	liveness	valence	tempo	features_duration_ms	n_beats	n_bars
duration_s	1.00	-0.08	-0.10	0.11	0.03	-0.08	-0.20	0.09	-0.01	-0.16	0.05	1.00	0.89	0.89
popularity	-0.08	1.00	0.06	-0.05	0.06	-0.09	0.08	-0.27	0.02	0.08	-0.01	-0.08	-0.08	-0.09
danceability	-0.10	0.06	1.00	0.16	0.36	0.10	-0.19	-0.18	-0.16	0.55	0.02	-0.10	-0.07	-0.11
energy	0.11	-0.05	0.16	1.00	0.72	0.13	-0.71	-0.19	0.21	0.27	0.30	0.11	0.23	0.19
loudness	0.03	0.06	0.36	0.72	1.00	0.10	-0.55	-0.45	0.06	0.36	0.28	0.03	0.14	0.11
speechiness	-0.08	-0.09	0.10	0.13	0.10	1.00	-0.08	-0.11	0.06	0.05	0.04	-0.08	-0.05	-0.05
acousticness	-0.20	0.08	-0.19	-0.71	-0.55	-0.08	1.00	0.09	-0.08	-0.12	-0.24	-0.20	-0.28	-0.25
instrumentalness	0.09	-0.27	-0.18	-0.19	-0.45	-0.11	0.09	1.00	-0.11	-0.34	-0.10	0.09	0.05	0.06
liveness	-0.01	0.02	-0.16	0.21	0.06	0.06	-0.08	-0.11	1.00	-0.02	0.02	-0.01	0.00	-0.00
valence	-0.16	0.08	0.55	0.27	0.36	0.05	-0.12	-0.34	-0.02	1.00	0.13	-0.16	-0.09	-0.12
tempo	0.05	-0.01	0.02	0.30	0.28	0.04	-0.24	-0.10	0.02	0.13	1.00	0.05	0.44	0.42
features_duration_ms	1.00	-0.08	-0.10	0.11	0.03	-0.08	-0.20	0.09	-0.01	-0.16	0.05	1.00	0.89	0.89
n_beats	0.89	-0.08	-0.07	0.23	0.14	-0.05	-0.28	0.05	0.00	-0.09	0.44	0.89	1.00	0.98
n_bars	0.89	-0.09	-0.11	0.19	0.11	-0.05	-0.25	0.06	-0.00	-0.12	0.42	0.89	0.98	1.00

Figure 4: Heatmap of Correlation Matrix

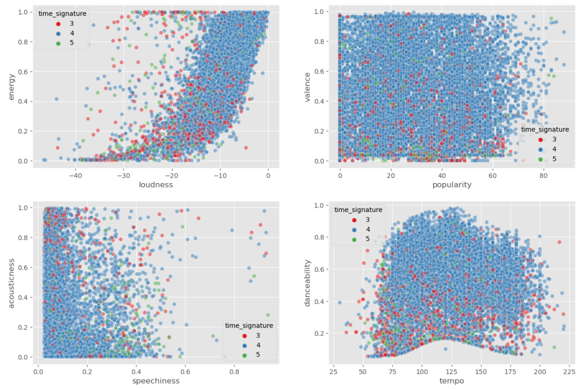


Figure 5: Scatter plots for different attributes using time_signature as hue.

Based on these results, we have decided to eliminate certain variables to streamline our dataset. Specifically, we remove:

- **feature_duration_ms**: This variable is eliminated to avoid redundancy with **duration_ms**, as both represent the track length and contain highly correlated values.

2 Clustering

This chapter explores the dual roles of clustering in both understanding and practical application. We focus on center-based methods for identifying cluster prototypes (objects representative of their respective clusters) and examine density-based and hierarchical clustering for understanding how data is divided into classes, primarily based on proximity.

2.1 Analysis by Centroid-Based Methods

In centroid-based clustering, a cluster is defined by objects closer to their own cluster's prototype than to any other. This prototype is typically the centroid, or the average of all points in the cluster, especially in datasets with continuous attributes like ours.

For our analysis, we utilized K-means and Bisecting K-means algorithms.

Choice of Attributes Determining the optimal feature set for these algorithms, we settled on eight features: duration_s, popularity, danceability, loudness, speechiness, instrumentalness, liveness, and tempo.

Features showing high correlation (defined as a correlation exceeding |0.5|) with others were eliminated. Our experiments with varying feature sets indicated that additional features, such as n_beats and n_bars, primarily correlated with duration and did not significantly enhance the algorithm's performance. Additionally, although relationships between loudness and acousticness, as well as danceability and energy, were noted, their inclusion increased computational demands without substantial improvement in clustering outcomes.

2.1.1 K-means

In K-means clustering, we determine the number of clusters (k) upfront. We chose the optimal k by applying the Elbow method and calculating Silhouette scores for k values from 2 to 30. The ideal number emerged as k=5. This choice is justified as increasing k beyond 5 resulted in sub-clusters within the initial clusters, while decreasing k significantly increased the Sum of Squared Errors (SSE) and obscured clear differences between clusters.

The analysis yielded the following results:

- $SSE \approx 61583$

- $Silhouette Score \approx 0.2397$

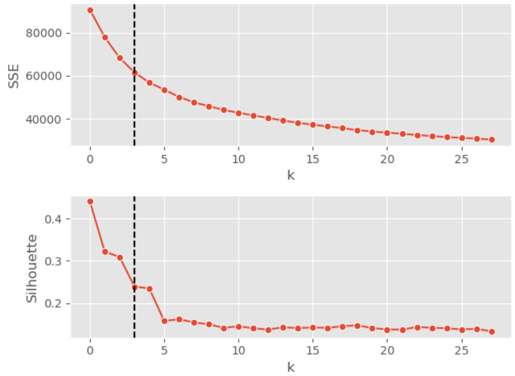


Figure 6: SSE and Silhouette for different k values.

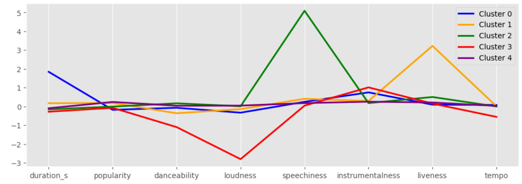


Figure 7: Parallel Coordinates plot shows the differences between centroids.

We can summarize our findings as follows:

- Cluster 0 (2071 objects) is primarily characterized by longer song durations compared to other clusters.
- Cluster 1 (1240 objects) stands out due to its higher liveness in the songs relative to those in other clusters.
- Cluster 2 (1351 objects) is distinguished by a higher speechiness in its songs compared to the other cluster prototypes.
- Cluster 3 (1284 objects) is notable for its lower loudness (i.e., very loud songs), along with lower danceability and tempo, but higher instrumentalness.
- Finally, Cluster 4 (8792 objects), the largest group, appears to encompass songs with 'standard' values when compared to the other clusters.

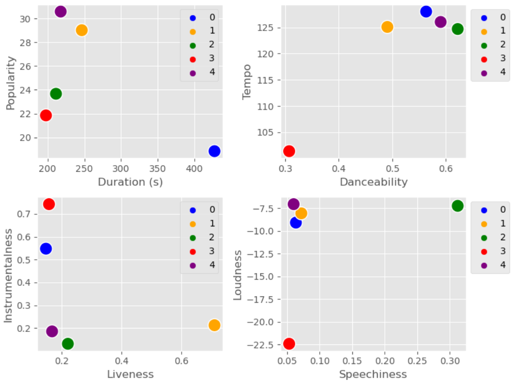


Figure 8: Scatter plot show the different centroids in the different dimensions.

2.1.2 Bisecting K-means

Also for this algorithm, we employ the Elbow method and calculate the Silhouette scores for k values ranging from 2 to 30. The optimal k appears to be 5 once again.

The analysis yielded the following results:
- $SSE \approx 64012.74$
- $SilhouetteScore \approx 0.189$

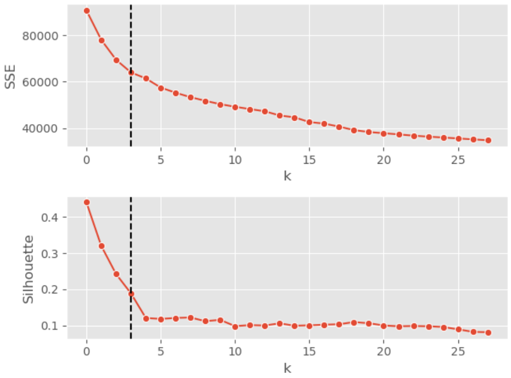


Figure 9: SSE and Silhouette for different k.

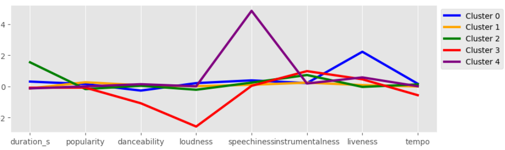


Figure 10: Parallel Coordinates plot shows the differences between centroids.

The clusters identified by the algorithm display similar distinguishing characteristics to those observed with the k-means algorithm. However, there is a notable difference in quantitative balance among the clusters.

The respective cluster sizes are as follows:

- Cluster 0: 2100 objects
- Cluster 1: 7442 objects
- Cluster 2: 2056 objects
- Cluster 3: 1632 objects
- Cluster 4: 1508 objects

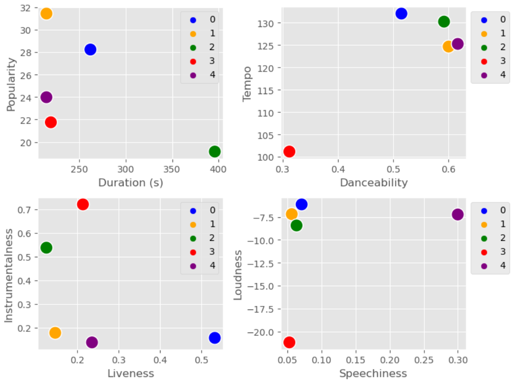


Figure 11: Scatter plot show the different centroids in the different dimensions.

Nonetheless, it is noteworthy that the centroids across various dimensions do not exhibit significant differences from those identified by the k-means algorithm.

2.2 Analysis by density-based clustering

Density-based clustering identifies regions of high density, which are distinctly separated from one another by areas of low density.

For this type of clustering's methods we focus on DBScan.

2.2.1 DBScan

The DBScan's algorithm requires 2 parameters: Eps and $min\_samples$.

To select the appropriate parameters, we developed a code that iterates over various combinations of features, Eps, and $min\_samples$.

Ultimately, we selected the same features as those used for the K-means algorithm. This decision was based on the code's suggestion that these features yield the best results with $Eps = 3$ and $min\_samples = 8$, achieving a silhouette score of approximately 0.75.

Additionally, we applied these parameters to the k-distance graph for visualization purposes.

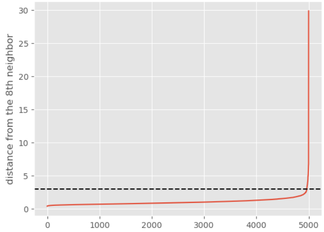


Figure 12: K-distance graph with distance from the 8th neighbor, depicted with a black line set at an $Eps = 3$.

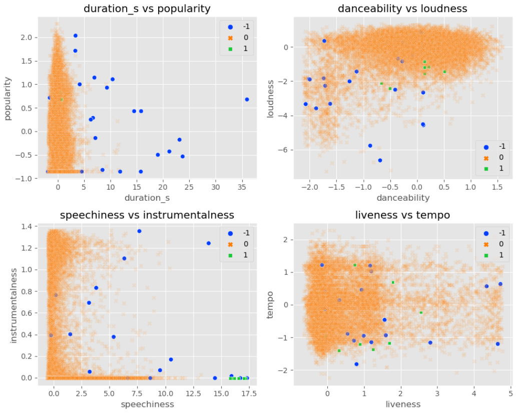


Figure 13: Scatter plot that shows the points in the dimensions used for DBScan.

Clusters discussion. The DBScan algorithm, with the tuned parameters, identified two clusters containing 14703 and 9 objects, respectively, along with 26 noise points.

It is observed that the second, smaller cluster is distinguished primarily by high values of speechiness. However, the presence of noise points near this cluster suggests that its formation is influenced by a combination of speechiness and other features. The 26 noise points are markedly distinct, representing songs that deviate significantly from the main cluster. These points are characterized by unusual durations and atypical feature values, standing apart from the 'standard' or 'common' characteristics of the larger cluster. In conclusion, the DBScan algorithm tends to face challenges in situations with highly variable cluster densities and in high-dimensional data.

PCA. For enhanced two-dimensional visualization, we applied Principal Component Analysis (PCA) to the features used in the DBScan algorithm.

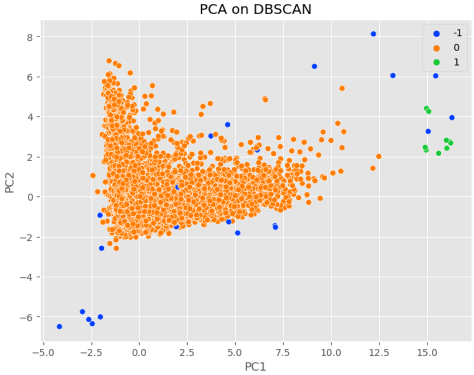


Figure 14: PCA on DBScan.

The PCA visualization clearly demonstrates that the second cluster is distinctly separated from the first. Moreover, most noise points are also well separated from both the first and second clusters. This observation reinforces our earlier conclusion that DBScan identifies a primary cluster within the 'mass' of songs, characterized by 'standard' feature values. In contrast, cluster 2 is defined by higher values of specific features, such as speechiness. The noise points, meanwhile, are scattered around these two clusters, further emphasizing their distinct nature.

2.3 Analysis by hierarchical clustering

If we allow clusters to encompass subclusters, we arrive at hierarchical clustering, which is a collection of nested clusters arranged in a tree-like structure.

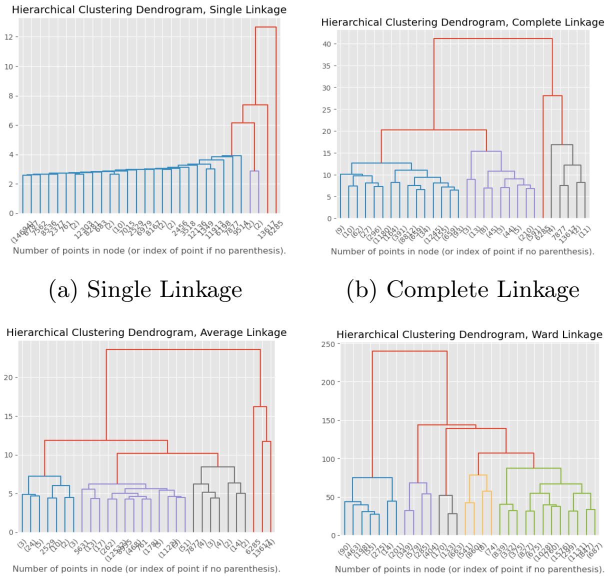
In our study, we specifically focus on agglomerative hierarchical clustering.

2.3.1 Agglomerative Hierarchical Clustering

The agglomerative approach to generating a hierarchical clustering begins by treating each point as an individual cluster. At each step, the closest pair of clusters are merged.

The features selected for this approach are identical to those used in the other clustering techniques.

Regarding the parameters, we chose "euclidean" as the *metric* due to our use of numerical values. We also experimented with various *linkage methods*, including single link, complete link, group average, and Ward's method.



(a) Single Linkage

(b) Complete Linkage

(c) Average Linkage

(d) Ward's method

Figure 15: Dendograms shows the results by different linkage methods.

MIN (single link). With a distance threshold of 4, three distinct clusters are identified, exhibiting a high silhouette score of approximately 0.8. This score indicates notable separation and cohesion among these clusters. The dendrogram reflects single linkage clustering's tendency to form elongated clusters, characterized by a chaining effect where clusters merge at varying heights.

MAX (complete link). A distance threshold of 20 yields 4 clusters with a moderate silhouette score of 0.48. Complete linkage clustering, focusing on the maximum inter-cluster distance, typically forms compact and well-separated clusters, as evidenced in the dendrogram. This method is less susceptible to chaining and favors clusters that are compact and distantly positioned.

Group Average technique. Setting the threshold at 10 reveals 4 clusters with a silhouette score of 0.636, indicating good separation and cohesion. Average linkage, which uses the mean inter-cluster distance, results in a balanced dendrogram structure. This method avoids the aggressive chaining of single linkage and suggests a more natural cluster definition.

Ward's method. A threshold of 100 delineates 5 clusters with a silhouette score of 0.214, signifying some limitations in separation and cohesion. Ward's method, aimed at minimizing within-cluster variance, tends to produce evenly spaced, compact clusters. The dendrogram shows clusters merging at comparable distances, and color coding identifies distinct clusters.

2.4 Conclusion

After analyzing the statistics and distributions of various clusters, our conclusions are as follows:

- *K-Means and Bisecting K-Means* provide insightful distinctions in song characteristics, despite the presence of one disproportionately large cluster alongside four smaller, balanced ones.
- *DBScan* effectively highlights outliers but fails to offer distinct class separations, rendering it less effective for our purposes.
- *Hierarchical Clustering* excels in class division, with Ward's method creating more evenly balanced clusters than average linkage. This is noteworthy even though average linkage achieves a higher silhouette score, suggesting a trade-off between balance and silhouette performance.

In summary, each algorithm contributes uniquely to our understanding of the dataset, which is crucial for the further development of this report.

3 Classification

This part focuses on the classification of the 'popularity' attribute and aims to categorize songs as 'popular' or 'unpopular' based on a range of musical features. We leverage the strengths of three distinct machine learning algorithms: Decision Trees, K-Nearest Neighbors (KNN), and Naive Bayes, each bringing its unique approach to the challenge. Our goal is to discern which best serves our predictive needs.

3.1 Feature Selection and Label Creation

Feature selection was conducted based on a median importance threshold, where features below this threshold (0.064) were removed. This allowed us to focus on features with the most predictive power. The selected features for the classification models include 'genre', 'instrumentalness', 'speechiness', 'valence', 'tempo', 'acousticness', 'loudness', 'danceability'.

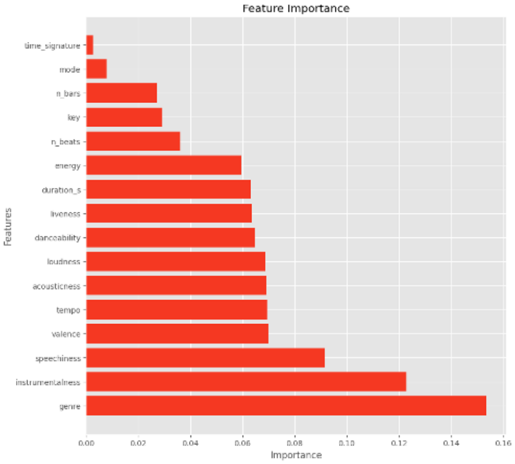


Figure 16: Feature Importance ranking

A binary label was created based on the median of the 'popularity' column, defining a new column 'popular'. Tracks with a 'popularity' score greater than or equal to the median (24.0) were labeled as 'popular' (1), and the rest as 'unpopular' (0).

3.1.1 Criteria for Model Evaluation

The most important values for us when evaluating our models are:

1. **Accuracy**: The proportion of total correct predictions (both popular and unpopular) relative to all predictions made.

2. **F1 Score**: The harmonic mean of precision and recall, balancing the two in the presence of class imbalances.

3. **Precision and Recall**: Reflecting the models' ability to correctly identify positive instances and their robustness against false positives.

4. **AUC**: The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the Receiver Operating Characteristic (ROC) curve.

3.2 Model Development

3.2.1 Decision Trees

Decision trees required little to no data preparation. The only requirement is that null values must be addressed. It was carried out using three techniques, as summarized below.

- **Untuned**: The initial model was a baseline Decision Tree without any hyperparameter adjustments.

- **Grid Search with Cross Validation**: In this method an exhaustive search over a predefined grid of hyper-parameters is conducted. Each combination is evaluated using cross-validation, ensuring a comprehensive assessment across different data splits

- **Randomized search with cost complexity pruning**: It offers a balanced approach to model tuning. It involves randomly sampling hyper-parameter combinations for efficiency and applies ccp_alpha pruning to control tree complexity, thereby preventing overfitting.

The best gain criterion was found to be '`Entropy`', as it produces the most consistent results. After comparing these methods, we found that the **Grid Search with Cross Validation** method was the most effective as it produced the best balance between F1 score, Recall, Precision and accuracy, with a training accuracy of 99.8% and test accuracy of 73.22% leading to a model that was more accurate and reliable for predicting whether a song is popular.

The Decision tree, confusion matrix, classification report and ROC curve when evaluated on the test set are shown below.

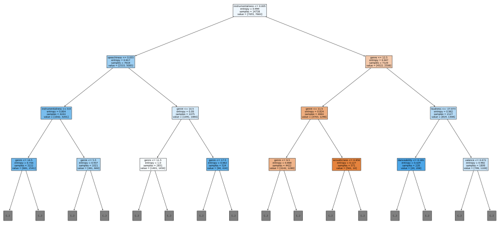


Figure 17: Decision Tree

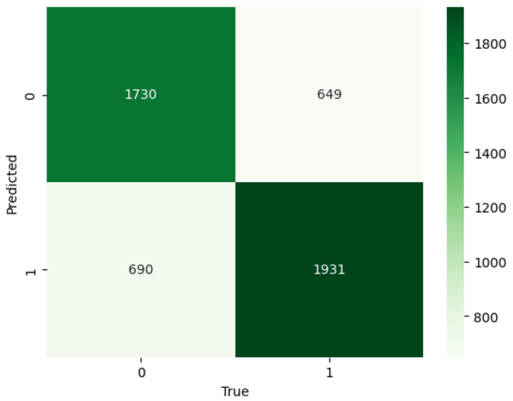


Figure 18: Decision Tree Confusion Matrix

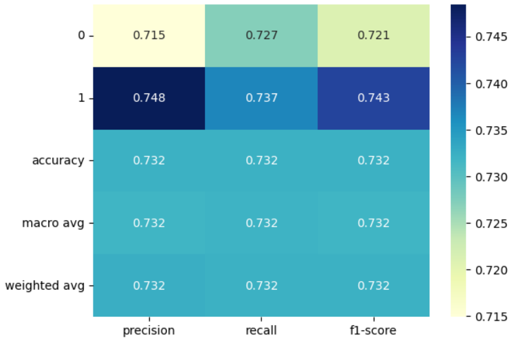


Figure 19: Classification Report

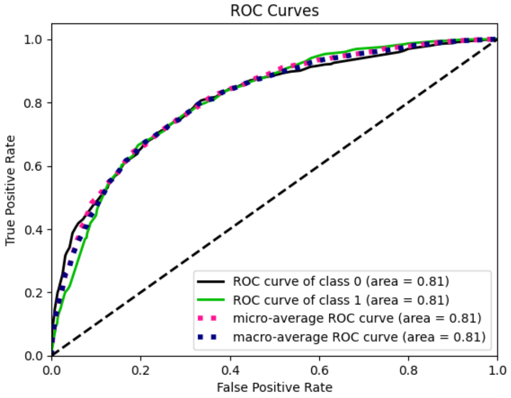


Figure 20: ROC Curve

3.2.2 K-Nearest Neighbors

The K-Nearest Neighbors model is an instance-based classifier that, for each record, selects the class label based on the majority of its nearest neighbors. We applied the Robust Scaler and then evaluated the model on 3 different techniques as shown below:

- **Untuned**: A baseline k-NN model with default parameters.
- **Manually Tuned for K**: Adjusted the number of neighbors (k) manually to find the optimal value.
- **Grid Search**: Applied `GridSearchCV` for a more comprehensive search of hyperparameters, including `n_neighbors`, `weights`, and distance metric.

The Grid search for $n\_neighbors$ showed the best parameter configurations as:

- **Best Parameters**: {'metric': 'manhattan', 'n_neighbors': 13, 'weights': 'distance'}

With these values, we have a training accuracy of 99.8% and a validation accuracy of 71.3%. The summarized results when evaluated on the test set are shown below.

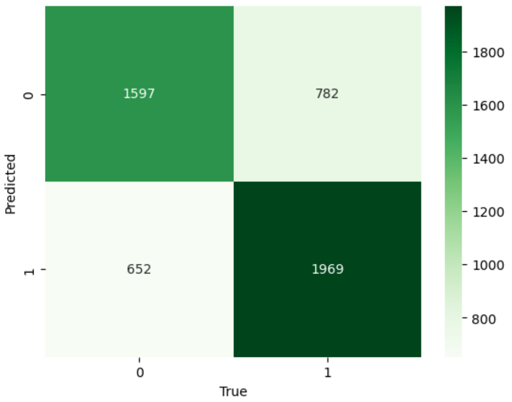


Figure 21: K-NN Confusion Matrix

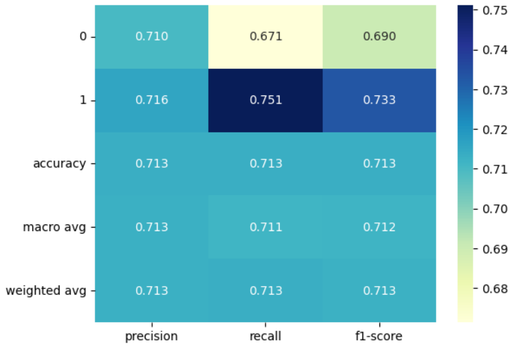


Figure 22: Classification Report

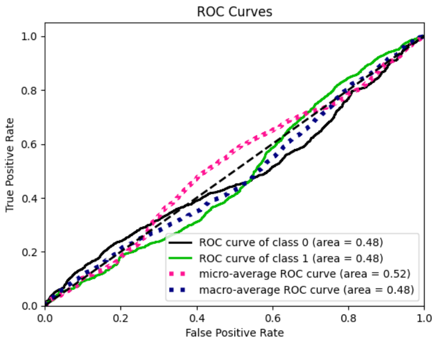


Figure 23: ROC Curve

3.2.3 Naive Bayes

The Naive Bayes Classifier estimates the class conditional probability for each feature using Bayes' theorem, with the assumption that all features are conditionally independent. We utilized the Gaussian Naive Bayes, which assumes the features' likelihood is Gaussian
The summarized results when evaluated on the test set are shown below .

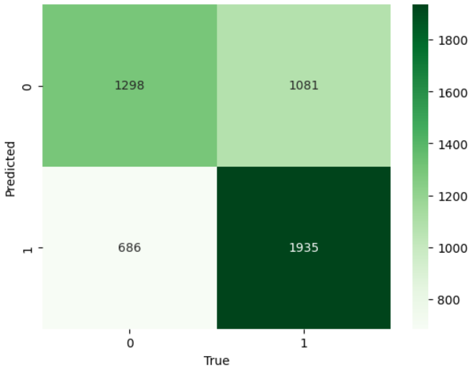


Figure 24: Naive Bayes Confusion Matrix

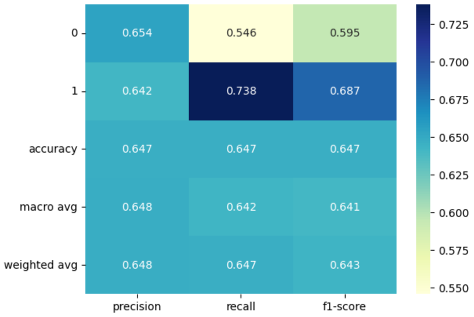


Figure 25: Classification Report

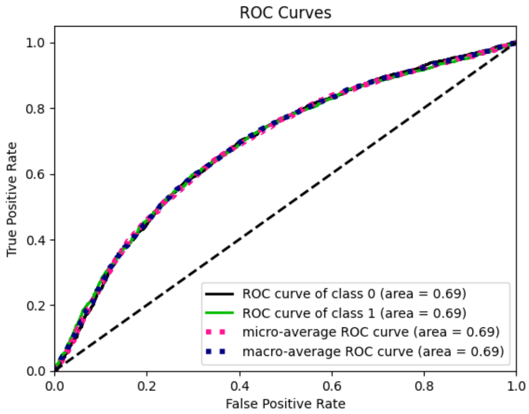


Figure 26: ROC Curve

3.3 Model Evaluation and Results

Metric	Decision Tree	KNN	Naive Bayes
Accuracy	0.732	0.713	0.647
Precision	0.743	0.716	0.642
Recall	0.737	0.751	0.738
F1	0.748	0.733	0.687
AUC	0.81	0.48	0.69

Table 1: Comparative Performance Metrics of Classifiers

3.4 Comparative Analysis

The superior performance of the **Decision Tree optimized with Grid Search and Cross-Validation** can be attributed to its effective hyper-parameter tuning, which ensured an optimal balance between model complexity and generalization. Additionally, Decision Trees' inherent strength in handling non-linear relationships and mixed data types, along with their robustness to outliers, likely made them particularly suitable for the specific characteristics of the dataset. The clear and interpretable structure of Decision Trees also contributed, especially if the dataset demanded a high level of model transparency.

4 ASSOCIATION

In this section, we delve into our Spotify dataset to uncover hidden behavioral patterns using association analysis. While typically applied to market basket transactions, association analysis is equally relevant for our dataset, treating each track as a customer's basket and its attribute values as itemsets. This approach aims to reveal relationships manifesting as frequent, maximal, and closed itemsets, or insightful association rules. Before the analysis, as detailed in Section 1.2, we ensured the dataset was free from NaNs and inconsistencies. We've omitted irrelevant attributes like 'name', 'album_name', and 'artists'. The 'popularity' attribute was categorized into four labels: 'not famous' (0-25), 'moderately known' (26-50), 'well known' (51-70), and 'famous' (71-94). After discretizing continuous values into bins and converting them to strings, our data is aptly formatted for generating frequent itemset using the Apriori algorithm.

4.1 Frequent itemset

In our analysis, we initially consider all itemsets as potential candidates for rule generation, leading to an exponential increase in candidates. The Apriori algorithm efficiently narrows this down by eliminating itemsets below a chosen minimum support threshold. Consequently, only itemsets with support at or above this threshold qualify as frequent. In our dataset, setting the minimum supports at 40%, 30%, 20%, and 10% yielded 9, 16, 166, and 896 frequent itemset, respectively, demonstrating that a lower threshold increases the number of frequent itemset identified. The plot in Figure(27) could serve as a guide to determine the appropriate support threshold for discovering meaningful patterns while avoiding an unmanageably large number of itemset.
We chose a 20% support threshold to achieve a balance in our analysis. This threshold is sufficiently inclusive to capture a wide range of itemsets, and also capture those frequent at higher thresholds. It allows us to investigate a more diverse range of itemset combinations, enhancing the potential for more comprehensive rule generation. Below are examples of some frequent patterns identified at this support level in our data set:

Number	Frequent Itemset	Support(%)
125	((35.999, 331.0]_nbeats, (7.999, 84.0]_nbars, . . .)	22.207898
158	(not_famous, True_mode, False)	29.963360
164	(True_mode, False)	59.675668
165	(4_t.sgn, False)	82.358529

- Itemset at the number 165 is the most frequent one, featuring a 4-beat time signature and non-explicit (False) content ('lack of lyric'), it holds a high support of 82.36%. This suggests that these attributes are dominant in our popular Spotify music dataset, likely catering to a broader audience and reflecting prevailing listener preferences and market trends, particularly where explicit content is less desirable.
- Itemset in line 158, present in 29.97% of the dataset, combines not famous, major mode, and non-explicit content, this might reflect the presence in our dataset of emerging artists producing mainstream-sounding tracks.
- The third frequent itemset, with 59% support, shows many tracks are in a major mode and non-explicit, hinting at a preference for cheerful, family-friendly music, as a major mode in music is often associated with a happier, brighter sound compared to minor modes.
- Frequent itemset number 125, found in 22.21% of the dataset, represents a specific song length and structure as n_beats and n_bars are number of time intervals of beats and number of time interval bars throughout the track, indicating a common preference for certain rhythmic patterns and track lengths.

These insights from frequent pattern reveal listener preferences and music trends in our dataset. The frequent 'False' (non-explicit content) in patterns corroborates Section 1's (data understanding) observation of predominantly non-explicit tracks in our dataset.

4.1.1 Maximal and closed itemset

In the Apriori analysis, we categorize itemsets as frequent, maximal, or closed, all of which are subsets of frequent itemsets:

- Maximal Itemsets: In our dataset, there are 61 maximal itemsets; these are frequent itemsets with no frequent supersets. The count is lower than the general frequent itemsets due to their definition.
- Closed Itemsets: With 165 closed itemsets, their number falls between the total frequent (166) and maximal itemsets (61). Closed itemsets are frequent itemsets without any supersets sharing the same support.

The plot demonstrates that as support increases, the counts of closed and maximal itemsets converge, suggesting that at higher support levels, many closed itemsets are also maximal.

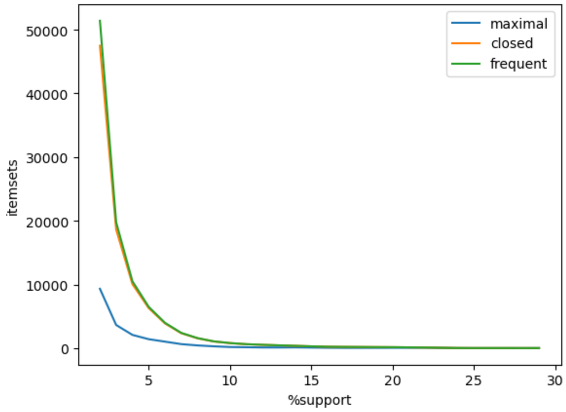


Figure 27: Frequents patterns curves

4.2 Rules

The Apriori algorithm's second step, after identifying frequent patterns, is to create association rules that reveal hidden relationships between itemsets, based on a confidence level. Confidence indicates how often the consequent itemset appears in transactions containing the antecedent itemset. Despite reducing candidates, this step can still produce many rules. To filter out less relevant patterns, a minimum confidence threshold is set. Higher confidence is preferred, even with lower support. As the confidence threshold increases, the number of rules decreases. This relationship between the number of rules and the confidence threshold is illustrated in Figure (28).

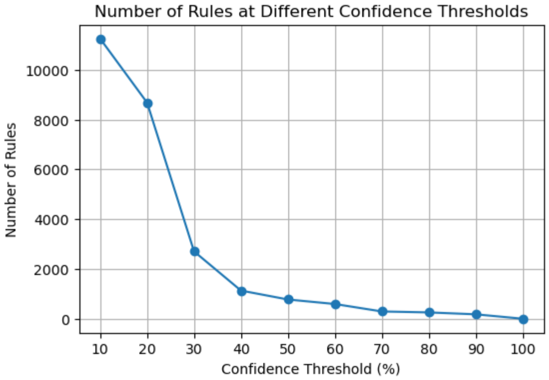


Figure 28: number of rule & confdence

The reliance on confidence in the Apriori algorithm can produce misleading rules, as it doesn't account for the overall frequency of the consequent.

This issue is addressed by using the lift metric, which evaluates both the rule's support and the individual frequencies of the antecedent and consequent. A lift value above 1 indicates a positive association, distinguishing true relationships from those occurring by chance. For generate Rules we choose a confidence threshold of 60%, meaning only rules with a confidence level from 60% to the maximum were considered. This threshold generated 592 rules. Many of these rules, especially those with the highest lift and confidence, involved features like 'n_beats', 'n_bars', and 'duration_s'. These are known to be related features from data understanding part. Rules that exclusively contain these related features (n_beats, n_bars, duration_s) provide detailed insights into their interrelationships. However, these insights are not particularly surprising since it's well-known that these features are correlated. here below we will show a couple of the rules we think are useful especially the last tre(3) rule even with a lift value less than the first two rules, they seem to provide us with more unexpected information:

Number	consequent	antecedent	support	confidence	lift
48	(627.0, 7348.0]_nbeats	((160.0, 2170.0]_nbars, 4_t.sgn)	20.993351	0.999354	4.007749
45	(627.0, 7348.0]_nbeats	((160.0, 2170.0]_nbars, 4_t.sgn, False)	20.077351	0.999325	4.007631
190	(0.565, 0.996]_acoust	((-0.0009798, 0.485]_energy, False)	16.929027	0.687139	2.751168
447	4_t.sgn	((0.00955, 0.153]_acoust, False)	21.352965	0.924501	1.052959
565	True_mode	((-0.001, 0.00267]_instru, False)	30.614737	0.672529	1.062351

- We prioritize rules with lift values above 1 in our selection process, as they indicate stronger associations. The first two rules we examined both have a lift significantly greater than 1, highlighting their relevance. With a support value of around 21%, these rules apply to about a fifth of the tracks in our dataset. Their exceptionally high confidence levels suggest that, when the antecedent conditions are met (such as the specific range of bars and a 4/4 time signature, in rule number 48),the consequent (like a high range of beats) is almost always true. The key metric, lift, confirms the likelihood of these features occurring together is higher than by chance, emphasizing the rules' robustness.

- The first rule indicates that tracks with many beats and bars, combined with a 4/4 time signature, are likely longer and rhythmically complex, characteristic of genres like progressive rock or electronic music, known for their lengthy and rhythmically intricate compositions, reflected in the high counts of beats and bars typical of their tracks.

- Rule 45 parallels Rule 48 in its itemset but uniquely includes non-explicit tracks, suggesting a preference for instrumental compositions emphasizing musicality over vocals. This distinction implies suitability for varied listening contexts, from public spaces to family-friendly environments, where non-explicit content is essential

- Rule 190 highlights a trend for lower-energy, non-explicit tracks within the (-0.0009798, 0.485] energy range, correlating with a high acoustic presence in the (0.565, 0.996] range. This pattern suggests a leaning towards acoustic genres like folk or soft rock, suitable for a wide audience due to their calm and non-explicit nature. This rule has a Moderate strength reflected by its support, decent confidence, and positive but not strong association as it's lift is less than the first two rules.

- Rule 447 identifies a trend towards tracks with a higher acoustic feature in the (0.00955, 0.153] range, often lacking explicit content, and being associated with a 4/4 time signature. This suggests a preference for rhythmically regular tracks that are likely non-explicit, fitting into various accessible genres. The rule shows high confidence in this association, indicating a strong correlation between these elements. However, the lift value is close to 1, suggesting that while the association is consistent, it's not exceptionally stronger than random chance.

- Rule 565 indicates a correlation between tracks with very low instrumental content in the (-0.001, 0.00267] range, often non-explicit, and their likelihood of being in a major mode. This pattern suggests a preference for simpler, major key music in various accessible genres. While the rule has moderate confidence, its lift slightly above 1 indicates a positive but not exceptionally strong association

These rules can be particularly useful for music streaming services and record labels for curating playlists, understanding listener preferences, and identifying market trends. They can also aid musicologists and researchers in analyzing music patterns across genres

4.2.1 Test of rule on test set

We applied Rule 565, which predicts a track's mode as major (True_mode) based on its non-explicit nature and specific instrumentalness range, to a test dataset. This approach is particularly feasible due to the binary nature of the 'mode' attribute. By using the rule's antecedent (explicitness and instrumentalness range) as predictors, we evaluated its performance in predicting the mode. The obtained metrics from this test in Table 2 below offer insights into the rule's practical applicability and accuracy

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0.0	0.40	0.61	0.48	1822
1.0	0.68	0.47	0.56	3178
Accuracy				0.52
Macro Avg				0.54
Weighted Avg				0.57

Table 2: Performance Metrics of Rule 565 on Test Set

- **Accuracy**: 0.52 (Overall, the rule correctly predicts the mode 52% of the time).

- **Macro Avg**: 0.54 (Average performance across classes, not accounting for class imbalance).

- **Weighted Avg**: 0.57 (Average performance with consideration for class imbalance).

we know that the Accuracy gives an overall effectiveness of the rule but can be misleading if classes are imbalanced in our case mode attributes are imbalanced so we will rely more on precision, recall, and f1-score.

- **Precision**: 0.40 (40% of predicted *False_ mode* are correct), 0.68 (68% of predicted *True_ mode* are correct)

- **Recall**: 0.61 (61% of actual *False_ mode* are correctly predicted), 0.47 (47% of actual *True_ mode* are correctly predicted)

- **F1-Score**: 0.48 (A balance between precision and recall, leaning towards recall), 0.56 (A balance between precision and recall, leaning towards precision)

Rule 565 provides some predictive power, particularly for identifying *False_ mode* (non-major key tracks). The rule's effectiveness is moderate, reflecting its moderate confidence of 67% and its lift slightly above 1, indicating a positive but not exceptionally strong association. However, it could benefit from refinement or combination with other rules or predictors for improved accuracy.

4.3 Regression

In the regression phase, we used different regression methods, both univariate and multivariate. For feature selection, we chose only those with high correlation from our **Pearson correlation matrix**. These features include n_beats, n_bars, acousticness, duration_s, energy, and loudness, as they gave better results. For univariate regression, we looked at how energy affects loudness. This helped us understand the relationship between these two features.

In multivariate regression, we used loudness, and acousticness as input features. We wanted to see how they together predict the output features, duration_s and energy.

Our results, including the coefficients, intercept, R-squared score, mean squared error, and mean absolute error for each regression type, are shown in the table below.

Regression Model	Independent Variable(s)	Dependent Variable	R2 Score	MSE	MAE
Linear Regression	Energy	Loudness	0.516	17.950	2.866
Ridge Regression	Energy	Loudness	0.516	17.950	2.866
Lasso Regression	Energy	Loudness	0.079	34.125	4.041
Decision Tree Regressor (Simple)	Energy	Loudness	0.580	15.575	2.684
KNN Regressor (Simple)	Energy	Loudness	0.569	15.974	2.760
Linear Regression (Multiple)	Loudness, Acousticness	Energy	0.661	0.024	0.118
Ridge Regressor (Multiple)	Loudness, Acousticness	Energy	0.439	0.040	0.142
Decision Tree Regressor (Multivariate)	Loudness, Acousticness	Energy , Danceability	0.053	0.045	0.157

Table 3: Comparative Performance Metrics of Classifiers

As seen in Table 3, the Linear Regression (Multiple) model with "Loudness" and "Acousticness" as independent variables and "Energy" as the dependent variable outperforms the other models in terms of R2 Score, MSE, and MAE. This suggests a stronger predictive power, where both the loudness and acoustic properties of a track are significant predictors for its energy level. To illustrate this model's predictive quality, a plot showcasing its regression fit is provided below:

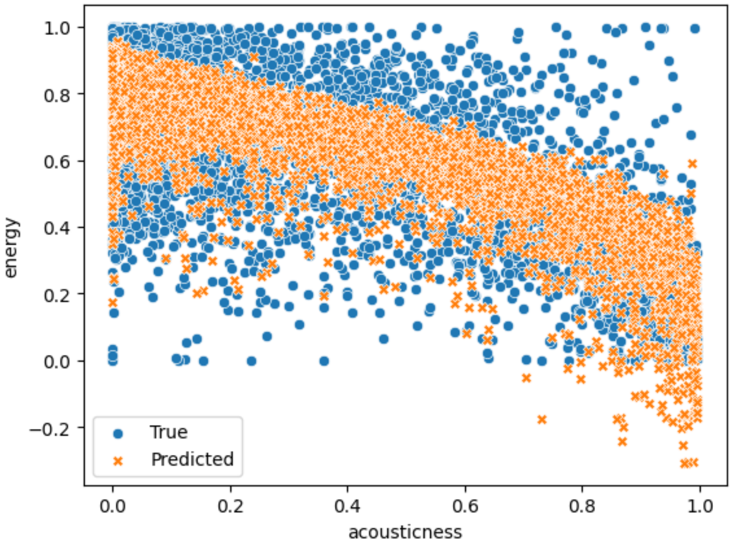


Figure 29: Performance of Regression Models

5 Conclusion

The analysis of the Spotify dataset using data mining techniques revealed significant insights into the characteristics of musical tracks. Clustering methods such as K-means, Bisecting K-mens, Hierarchical, and DBSCAN uncovered distinct patterns and groupings within the dataset. K-means was found to be particularly effective. During the classification phase, we aimed to predict the 'popularity' of tracks using algorithms such as Decision Trees, KNN, and Naive Bayes. The Decision Tree model was the most successful in predicting popularity, while the KNN classifier also improved significantly after optimization through grid search methods. The Apriori algorithm was crucial in extracting meaningful association rules for pattern mining, which proved especially useful in classifying songs as either explicit or non-explicit. Despite the low support level for selected rules, the high confidence levels indicated the algorithm's effectiveness in uncovering significant patterns within the dataset. Regression analysis was used to capture relationships between various musical features. Linear and ridge regression models were found to be potentially effective in delineating these associations. This project has improved our understanding of the complexity inherent in music data and the general efficacy of data mining techniques in extracting meaningful information.