Università di Pisa

Data Mining 1 - Project

R. Anello, F. Corsini, M. Poiani

A.Y. 2023/2024

Contents

Chapter 1

Data Understanding and Preparation

1.1 Introduction

This report presents a comprehensive overview of our investigative endeavours and discoveries. We conducted an in-depth examination of our dataset to achieve a holistic comprehension of its components, covering entities, their affiliated attributes, and their corresponding numerical or categorical representations. The original dataset consists of 20000 rows and 23 columns. It is additionally divided into two smaller datasets: "train", which contains 15000 records, and "test", comprising the remaining 5000. This subdivision plays a pivotal role in various activities within Supervised Learning. Nonetheless, for the Unsupervised Learning tasks, we opted to utilize the training set exclusively. Each row within the dataset denotes a distinct Spotify track, while the columns delineate the specific characteristics or attributes associated with each individual item.

1.2 Data semantics

In Table 1.1, we provide details about the dataset variables, including their names, concise descriptions, and types.

Variable name	Description	Type
name	Indicates the title of the track.	Nominal categorical variable
duration_ms	Denotes the track's length in milliseconds.	Discrete numerical variable
explicit	Indicates whether the track contains explicit lyrics or not.	Binary categorical variable
popularity	Quantifies the popularity of a track with a value ranging from 0 to 100.	Discrete numerical variable
artists	Lists the name of the artists who participated in the track's performance.	Nominal categorical variable
album_name	Specifies the album name in which the track is featured.	Nominal categorical variable
danceability	Quantifies the track's danceability on a scale from 0.0 to 1.0.	Continuous numerical variable
energy	Measures the level of energy on a scale from 0.0 to 1.0, serving as a way to evaluate intensity and activity levels.	Continuous numerical variable
key	Indicates the key of the track, expressed as integers corresponding to pitches in the Pitch Class notation.	Ordinal categorical variable
loudness	Quantifies the overall loudness level of a track in decibels (dB).	Continuous numerical variable

mode	Specifies the modality of a track, with 1 representing major and 0 representing minor.	Binary categorical variable
speechiness	Quantifies the presence of spoken words in a track. Values close to 1.0 indicate predominantly spoken content (e.g., talk shows), values close to 0.0 indicate a focus on non-speech content.	Continuous numerical variable
acousticness	Quantifies the confidence level, on a scale from 0.0 to 1.0, representing the likelihood that the track is acoustic.	Continuous numerical variable
instrumentalness	Predicts the absence of vocals in a track. As the value approaches 1.0, the likelihood of the track being entirely instrumental increases.	Continuous numerical variable
liveness	Detects the presence of an audience in the recording, measured on a scale from 0.0 to 1.0.	Continuous numerical variable
valence	Quantifies the musical positivity of a track on a scale from 0.0 to 1.0. Higher valence values indicate a more positive emotional tone.	Continuous numerical variable
tempo	Specifies the estimated tempo of a track, measured in beats per minute (BPM).	Continuous numerical variable
features_duration_ms	Denotes the duration of the track in milliseconds.	Discrete numerical variable
time_signature	Specifies the time signature of a track, indicating the number of beats within each bar or measure. ”0.0” designates non-musical tracks, while ”1.0”, ”3.0”, ”4.0” and ”5.0” represent songs with the corresponding beats per bar.	Ordinal categorical variable
n_beats	Quantifies the total number of time intervals corresponding to beats across the duration of the track.	Discrete numerical variable
n_bars	Quantifies the total number of time intervals corresponding to bars across the duration of the track.	Discrete numerical variable
popularity_confidence	Quantifies the confidence in the song's popularity level, measured on a scale from 0.0 to 1.0.	Continuous numerical variable
genre	Specifies the genre to which the track belongs.	Nominal categorical variable
processing	*Absence of description in the documentation.*	Discrete numerical variable

Table 1.1: Variables description

1.3 Data quality and variable transformations

1.3.1 Duplicate rows assessment

Upon evaluation, it can be confidently asserted that there are no duplicate rows in the dataset.

1.3.2 Semantic inconsistencies

loudness: In the domain of digital audio, surpassing 0 dB is virtually unattainable. Any attempt to exceed this threshold results in a form of distortion known as digital clipping, which can produce harsh and abrasive auditory artefacts. Therefore, 0 dB still serves as the maximum attainable output. This semantic inconsistency has been addressed by replacing the 21 values that exceeded this limit with their true output value of 0 dB.

1.3.3 Missing values

- **mode:** A total of 4450 missing values, constituting 29.67% of the dataset, have been identified. Those missing values have been replaced with the mode value of 1.0.
- **time_signature:** A total of 2062 missing values, constituting 13.75% of the dataset, have been identified. Those missing values have been replaced with the mode value of 4.0.
- **popularity_confidence:** A total of 12783 missing values, constituting 85.22% of the dataset, have been identified. However, due to the substantial presence of missing values, deriving meaningful insights from this attribute is unfeasible. Consequently, the variable will be excluded in Section 1.5.2.

1.3.4 Outliers detection

In our analysis, we employed the Interquartile Range (IQR) method to detect outliers, selecting it for its effectiveness in identifying records that significantly deviate from the central tendency of the dataset. Various methods, encompassing statistical and visualization approaches, can be utilized for outlier detection. We initiated the process by examining boxplots. Figure 1.1 illustrates that some variables exhibit outliers exclusively below the lower bound (e.g. loudness), some exclusively above the upper bound (e.g. popularity), and some on both sides (e.g. duration_ms). Using this method, here is the count of outliers we detected:

- **duration_ms**: 616 Outliers detected.
- **popularity**: 5 Outliers detected.
- **danceability**: 104 Outliers detected.
- **loudness**: 1004 Outliers detected.
- **speechiness**: 1679 Outliers detected.
- **liveness**: 1149 Outliers detected.
- **tempo**: 113 Outliers detected.
- **n_beats**: 362 Outliers detected.
- **n_bars**: 402 Outliers detected.

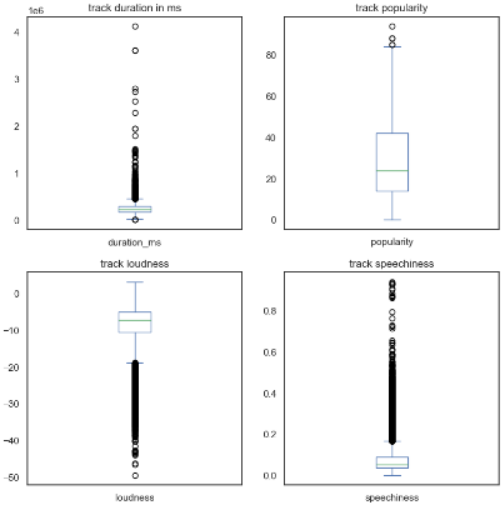


Figure 1.1: Outliers detection through boxplot

In our analytical approach, we intentionally refrained from prematurely removing outliers from the dataset. Recognizing the potential insights and contribution to a comprehensive understanding of the data that outliers might offer, we chose to focus on detecting their presence.

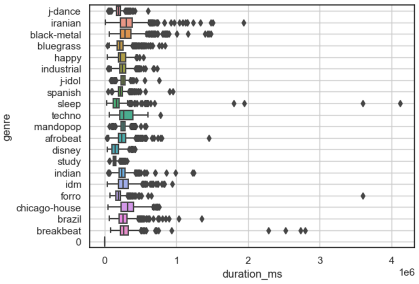


Figure 1.2: Boxplot relation between ”duration_ms” and ”genre”

Consequently, our decision on whether to retain or remove outliers is contingent upon the specific analysis, ensuring that our data preprocessing aligns with the goals of each analytical task. For instance, although we attempted to remove some outliers before clustering, the negligible impact on results led us to retain all records to avoid information loss. A noteworthy observation emphasises the relative nature of outlier designation. A value may be considered an outlier in relation to one feature but not in relation to another. Therefore, once you are aware that outliers exist in your dataset, you can remove or maintain them selectively. In Figure 1.2, the values of the ”duration_ms” feature grouped by the different genres are presented. The boxplots affirm that the duration of a track may be deemed an outlier if the track belongs to one genre, while the same duration in another genre may not be considered an outlier. The graphical representation underscores the importance of the context on how a variable should be evaluated.

1.4 Variable transformations

In the initial phase of our data preparation, we intentionally refrained from applying any variable transformation. This decision was made with the foresight of reserving such transformations for subsequent analyses. By deferring variable transformations to later stages, we maintain flexibility in tailoring these adjustments, and this allows us to adapt to the nuances of each analytical task, ensuring that any transformation applied is purposeful and aligned with the objectives of the specific analysis at hand.

1.5 Pairwise correlations and eventual elimination of variables

1.5.1 Matrix correlation analysis

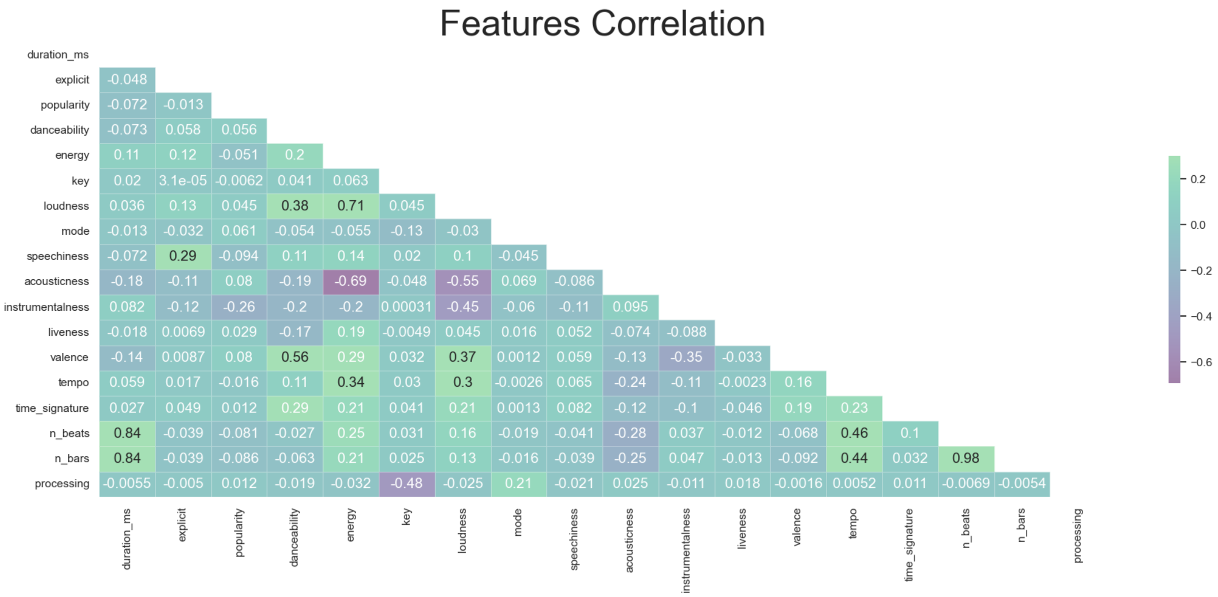


Figure 1.3: Triangular Matrix for Features Correlation

In Figure 1.3, we present the correlation matrix for all numerical variables in our dataset. Beginning with the most substantial correlation coefficients, a notable correlation of 0.98 is evident between ”n_bars” and ”n_beats”. This strong association is easily explained by the inherent connection between the number of beats and bars in a song, with the latter forming the structural framework for beats. Another noteworthy correlation arises between a track's duration in milliseconds and its count of beats or bars, yielding a coefficient of 0.84. This relationship is straightforward, with longer songs naturally containing more beats and bars, while shorter tracks exhibit fewer musical elements. This reasoning guided the decision to drop the attributes ”n_beats” and ”n_bars”. An insightful observation reveals a significant correlation (0.71) between ”loudness” and ”energy”. This result is entirely anticipated, emphasizing the interplay between a track's intensity and activity and its overall loudness. Continuing, a valuable correlation coefficient emerges in the connection between ”valence” and ”danceability”, indicating a moderate correlation of 0.56. This finding suggests that, in most cases, tracks well-suited for dancing tend to possess a more musically positive character. Finally, two noteworthy negative correlations are observed within the dataset. These correlations link the ”acousticness” variable with ”loudness” (-0.55) and ”energy” (-0.69). These negative relationships imply that as a track's acoustic characteristics become more prominent, its loudness decreases, and its energy level diminishes.

1.5.2 Removed attributes

- **features_duration_ms**: This attribute has been removed from the dataset due to its substantial redundancy with the "duration_ms" attribute, resembling a near-perfect one-to-one copy.
- **popularity_confidence**: This variable has been removed from the dataset due to an exceptionally high percentage of missing values, amounting to 85.22%.
- **n_beats**: This variable has been removed from the dataset due to its high correlation coefficient (0.84) with "duration_ms". Furthermore, considering that the variables "n_beats" and "n_bars" offer information similar to that of "time_signature", the decision was made to retain the latter for a more cohesive representation.
- **n_bars**: Similar reasoning as the aforementioned variable.
- **processing**: We encountered difficulty in assigning a clear interpretation to the variable "processing" due to its absence in the documentation. Despite our attempts at analysis, no substantive meaning could be extracted. It constitutes a discrete numerical variable composed of 12 distinct floating-point numbers which range from a minimum of 0.748116 to a maximum of 4.067086. During our exploration, we noted a modest correlation of 0.21 with the "mode" variable. Further investigation revealed intriguing patterns: when the "mode" variable is 0.0 (minor modality), the median of the "processing" variable hovers around 1.28, with the mode value being the minimum of the entire distribution (0.748116). Conversely, when the "mode" variable is 1.0 (major modality), the median of the "processing" attribute is approximately 2.37, and its mode corresponds to the maximum value of the distribution (4.067086). However, since no other meaningful insights could be obtained from the variable, it has been removed from the dataset.

1.6 Distribution of the variables and statistics

In this segment, we explore the distribution of variables and associated statistics to unveil patterns, central tendencies, and variability within the dataset. We've chosen to split our analysis into two sections: the former focuses on numerical variables, and the latter explores the categorical ones.

1.6.1 Numerical variables

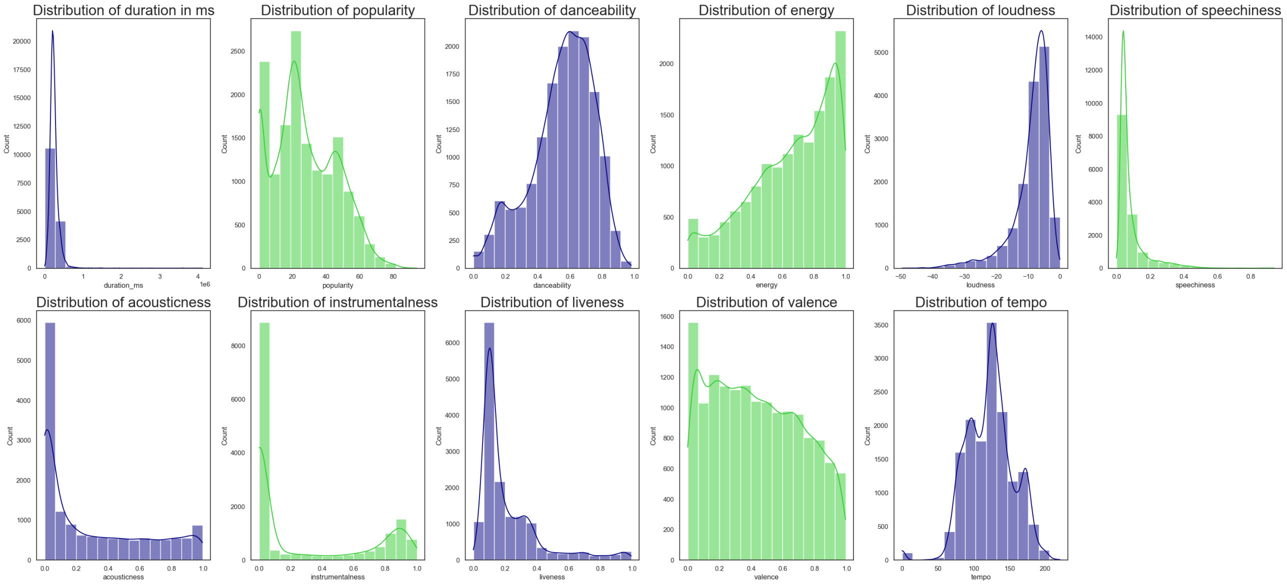


Figure 1.4: Numerical variables distribution

In Figure 1.4, we showcase the frequency distribution of the remaining 11 numerical attributes following the Data Preparation phase. The insights derived from this figure are noteworthy, and we will swiftly delve into them in this section. Employing 15 bins, in accordance with Sturges' rule, the histograms offer a comprehensive view. Starting from the top-left subgraph, let's discuss the first attribute, "duration_ms". It emerges as a highly imbalanced variable with a median of 227826 (just under 4 minutes) and a mean of 246807 (slightly over 4 minutes). The positively skewed distribution is evident, signifying that the majority of data points huddle on the lower side, with a few extreme values pulling the mean to the right. More than 75% of the tracks fall below the 5-minute mark, but there are impactful outliers, including the maximum value exceeding 68 minutes. A similar skewed pattern is observed in "speechiness", while variables like "popularity", "acousticness", "liveness", and "valence" exhibit slightly more subtle positive skewness. Interestingly, "instrumentalness" leans towards positive skewness, but the graph also suggests a potential argument for a bimodal distribution, especially evident when examining different genres. As a matter of fact, the genres "black-metal", "breakbeat", "chicago-house", "disney", "idm", "iranian", "sleep", "study", and "techno" exhibit an average instrumentalness of approximately 0.52, a value that aligns well with the inherent traits of these genres; while the remaining genres average around 0.09. On the flip side, we encounter negatively skewed distributions in "danceability", "energy", and "loudness", where the tail extends more to the left, and the mean is lower than the median. This aligns with expectations; for instance, "loudness" commonly hovers around -8dB to -10dB, with an average value of -8.89, mirroring industry standards. Lastly, the variable "tempo" presents a more normal distribution, coherent with a median and mean around 123.11 and 124.19, respectively.

1.6.2 Categorical variables

Concerning categorical variables, the four most informative bar charts are depicted in Figure 1.5.

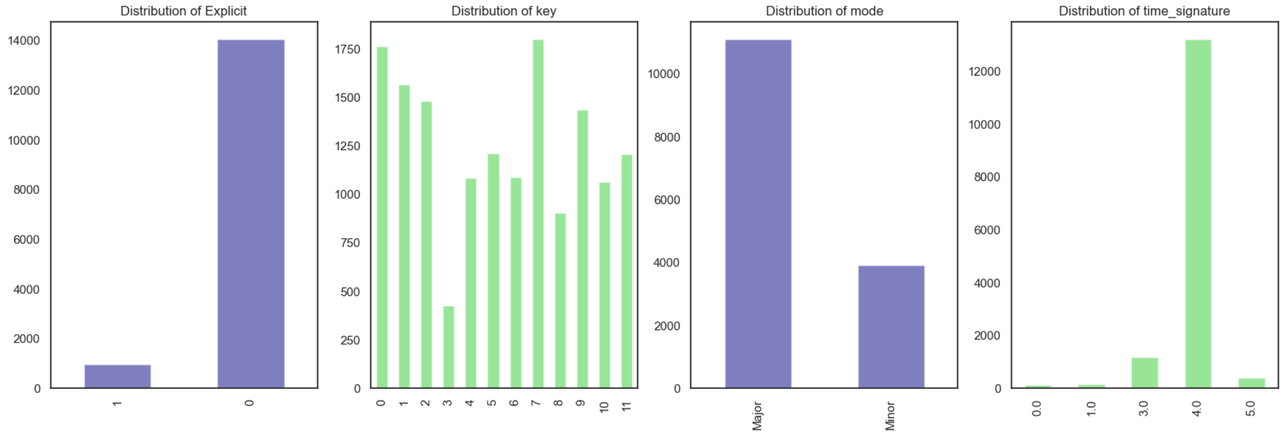


Figure 1.5: Categorical variables distribution

The insights from the figure are quite straightforward. It's evident that the number of tracks containing explicit language is significantly lower compared to those without this trait. The distribution of keys appears well-balanced, with the most common pitch class being situated a whole step (or two half steps) above the reference pitch, specifically at number 7 in Integer notation. Additionally, the prevalent track modality is Major and the most frequent time signature allocates 4 beats within each measure. Lastly, it's worth noting that the "genre" attribute is evenly distributed, meaning that the 15000 records are divided into 20 genres, each containing 750 records.

Chapter 2

Clustering

In this section, we will delve into a detailed cluster analysis, employing the three primary methods: centroid-based, density-based, and hierarchical clustering. Common algorithms like K-means for centroid-based, DBSCAN for density-based, and hierarchical agglomerative clustering for hierarchical methods will be applied. Moreover, exploring notable variants within these frameworks will offer a comprehensive overview of the clustering landscape. Our analysis initiation involved preparing the dataset for clustering, aiming for comprehensive results encompassing the entire dataset. We chose to include all numerical attributes, whether continuous or discrete, that remained after the data preparation phase. The initial working dataset for clustering comprises 11 attributes: ”duration_ms”, ”popularity”, ”danceability”, ”energy”, ”loudness”, ”speechiness”, ”acousticness”, ”instrumentalness”, ”liveness”, ”valence”, and ”tempo”. While we acknowledge the possibility of reducing the number of features, such as dropping the slightly redundant ”loudness”, correlated with ”energy” at a 0.71 correlation coefficient, our initial clustering decision was to incorporate all nuances each variable offers for a complete dataset perspective. This approach holds for all the clustering methods utilised. Subsequently, we applied the MinMax normalization method to the working dataset, resulting in records with values ranging from 0.0 to 1.0.

2.1 Analysis by centroid-based methods

2.1.1 K-means

In the initial phase of the K-means algorithm, selecting the optimal ’K’, representing the desired number of clusters and the corresponding centroids, is essential.

Various methods are available to identify the best ’K’, each providing complementary insights to enhance accuracy and reliability. A correct determination of the number of clusters is crucial for the success of the clustering process, as an incorrect ’K’ value can compromise the integrity of the analysis. Two key metrics, the Sum of Squares Error (SSE) and the Silhouette score, help us in evaluating an appropriate ’K’ Since there is no predefined range for SSE, we autonomously judge its acceptability. In contrast, the Silhouette score must fall within a range of -1 to 1, with a higher value being preferable. However, relying solely on SSE may be misleading, as a higher ’K’ corresponds automatically to a lower SSE. Therefore, our objective is to achieve an acceptable SSE while minimizing the number of clusters. Figure 2.1 depicts line plots for both SSE and Silhouette scores corresponding to different numbers of clusters (K), from 2 to 50. Upon analyzing the curve’s elbow, as we can see in the figure 2.1, we determined 7 as an acceptable value for ”K”, a conclusion supported by the silhouette score. This suggests that, for our dataset, 7 could be considered one of the optimal numbers of clusters. With the ”K” established, we proceeded to conduct clustering using the K-means algorithm configured for 7 clusters. The resulting outcome

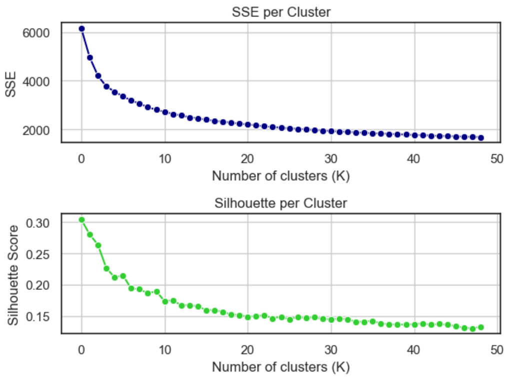


Figure 2.1: SSE and Silhouette score for K-Means

looks promising, showcasing a well-balanced distribution of records within the clusters, as illustrated in Figure 2.2.

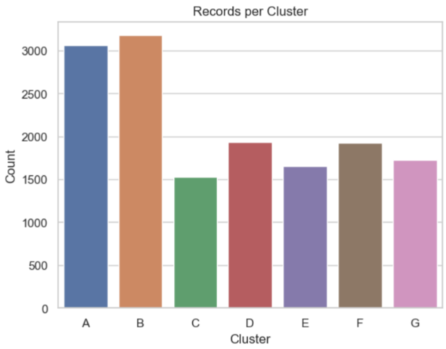


Figure 2.2: Number of records per cluster (K=7)

Figure 2.2 reveals that the larger cluster comprises 3700 elements, whereas the smaller one consists of 800 elements. The obtained results demonstrate favourable cluster sizing, underscoring the efficacy of our current approach. This advantageous cluster formation enables us to derive an efficient summary of our data. To enhance the validation of our findings, we extended our analysis by incorporating randomization. In this supplementary investigation, we generated 500 datasets mirroring the structure of our primary dataset. Each of these datasets comprised records confined within the lower and upper bounds of our actual data, ensuring a comprehensive exploration of randomized scenarios. Analyzing the frequency distribution of Sum of Squares Errors (SSE) and Silhouette scores in the random datasets, we found that the median SSE for 7 clusters was 10820, in contrast, our dataset exhibits a significantly lower SSE of 3319. This pattern holds true for the Silhouette score metric as well. The median silhouette value calculated from random datasets for 7 clusters was 0.0596, whereas our dataset surpasses this substantially with a score of 0.2148. Therefore, it is evident that our clustering analysis provides meaningful information about our working dataset, as indicated by the pronounced disparities in SSE and Silhouette scores between the actual and random datasets. From an empirical standpoint, the likelihood of obtaining SSE and silhouette score values comparable to those of our data through a random dataset is practically nonexistent. At this point according to Figure 2.1, we opted to conduct the same analyses again, before using 5 clusters (K=5) and after 4 clusters (K=4). Despite observing that the SSE in both cases was higher than the value obtained for 7 clusters, as per definition, the Silhouette score showed a slight improvement, as we can apprehend from Table 2.1.

Table 2.1: Results for K-means Analyses

N°of clusters (K)	**SSE**	**Sil Score**	**Randomized SSE**	**Randomized Sil score**
K=4	4163.7596	0.2640	11757.4870 (median)	0.0598 (median)
K=5	3731.2934	0.2269	11403.3426 (median)	0.0588 (median)
K=7	3319.4778	0.2147	10820 (median)	0.0596 (median)

To validate our final decision, we conducted randomization experiments for both K=5 and K=4, and the results have been inserted in advance in Table 2.1. The frequency distributions of the abovementioned experiments are shown in Figure 2.3 and 2.4.

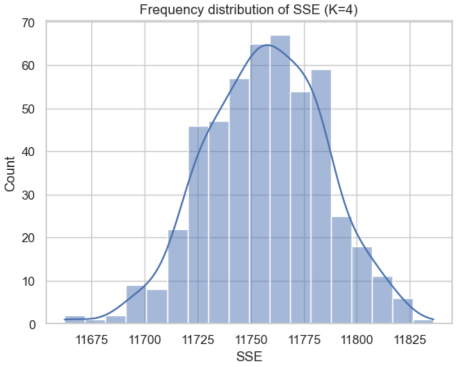


Figure 2.3: Randomized SSE for K=4

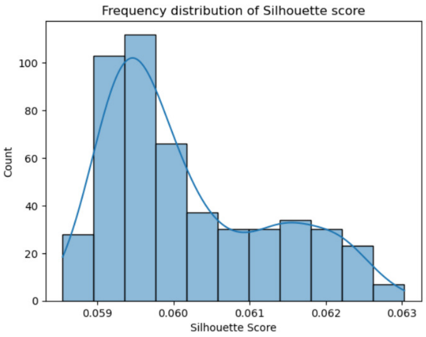


Figure 2.4: Randomized Silhouette Score for K=4

The randomization results affirmed the acceptability of our values in both cases. However, the preference for a smaller number of clusters, achieving a higher Silhouette score and slightly worse SSE, emerged as a valuable compromise. Exploring clustering with less than 4 clusters was not pursued because, despite a superior Silhouette score, the SSE was excessively high in such cases. Therefore, summarising our dataset using just 4 clusters could be considered an improvement. To conclude our analysis, we computed the centroids using K=4 and added a new column to our working dataset, "cluster_labels", which indicates the cluster affiliation for each record. The visual representation in Figure 2.5 highlights that the most diverse features across clusters are "danceability", "energy", "acousticness", and "valence".

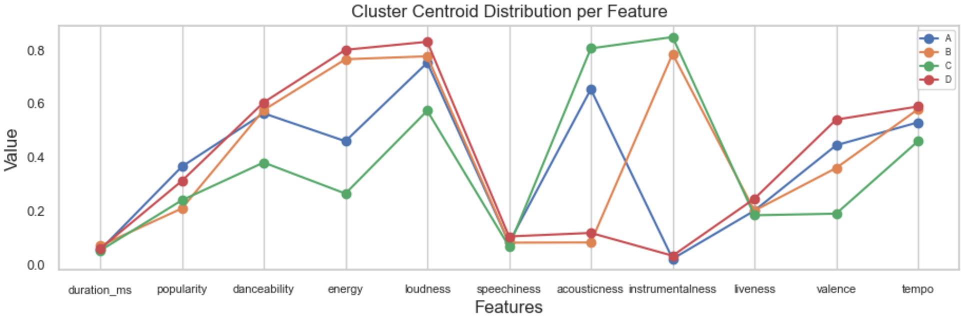


Figure 2.5: Cluster Centroid Distribution per Feature

After conducting K-means clustering with K=4, clusters were visually represented using the "energy" and "instrumentals" variables, guided by the correlation triangular matrix and centroid distribution graph (Figure 1.3, Figure 2.5). In Figure 2.6, we observe the four distinct clusters, each revealing only a minimal overlap that enhances the clarity of visualization, enabling a clear separation of their centroids. In contrast, the subsequent experiment with K=5 reveals a consistent overlap in two clusters. This observation further underscores the preference for the K=4 configuration, emphasizing its effectiveness in maintaining distinct cluster separations. From the previously mentioned visualization, our aim was to extract effective insights on the composition of each cluster. Upon examining the genres associated with the tracks, we discerned distinctive characteristics for each cluster, as depicted in Figure 2.6

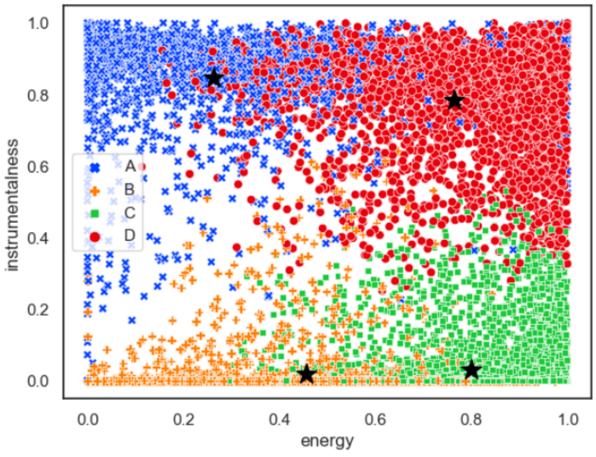


Figure 2.6: Clusters distribution in a bidimensional space

The blue cluster (Cluster A), showcasing records with high instrumentalness but low energy, mostly encompassed genres like "sleep", "study", "iranian", and "disney". The orange cluster (Cluster B), exhibiting both low energy and low instrumentalness, predominantly consisted of genres like "spanish", "j-idol", "j-dance", and "industrial". The green cluster (Cluster C), characterized by high energy and low instrumentalness, mainly comprised genres such as "mandopop", "bluegrass", and "indian". Lastly, the red cluster (Cluster D), marked by high levels of both energy and instrumentalness, was largely composed of genres such as "techno", "black-metal", "idm", "chicago-house", and "breakbeat". All these genres are notably known for having the particular characteristics highlighted by the different clusters. This genre-based analysis highlights the significance of the clustering approach in capturing and identifying the distinct musical characteristics within each cluster. As a result, we are content with the outcomes, emphasizing the noteworthy contribution of clustering analysis in grouping similarities, aligning with the primary goal of the task.

2.1.2 Bisecting K-means

After conducting the K-means analysis, we ventured into implementing Bisecting K-means. Despite achieving a more balanced distribution with the same number of clusters for the K-means, in comparison, we observed a lower silhouette and a slightly higher SSE, as shown in Table 2.2. Consequently, we deemed K-means to be a superior choice over bisecting K-means for our analysis.

Table 2.2: Results for bisecting K-means

N°of clusters (K)	SSE	Sil Score
K=4	4185.1579	0.2598
K=5	3789.1119	0.2153
K=7	3399.3470	0.1790

2.2 Analysis by density-based clustering

The second approach utilized in the clustering analysis is based on the concept of density, defined as the number of points within a specific radius (ϵ). As stated before, our analysis employs the same dataset chosen for K-Means Clustering. Identifying the optimal combination of MinPts, which signifies the minimum number of points within (ϵ) required to form a cluster, and (ϵ) itself requires a thorough exploration and experimentation process. Following the idea that points in clusters share a similar distance to their k^{th} nearest neighbours, our initial step involved plotting the previously mentioned distance of each point to its k^{th} nearest neighbours, with k = MinPts. This operation was performed for a broad range of MinPts values, although only k values ranging from 2 to 9 will be illustrated in Figure 2.7. We limited the upper range to nine because, when MinPts = 10, only one cluster emerged. It's crucial to emphasize that we had pre-computed the algorithm using the optimal (ϵ) parameter, and this prior analysis indicated that utilizing MinPts = 10 would not provide meaningful results for our analysis.

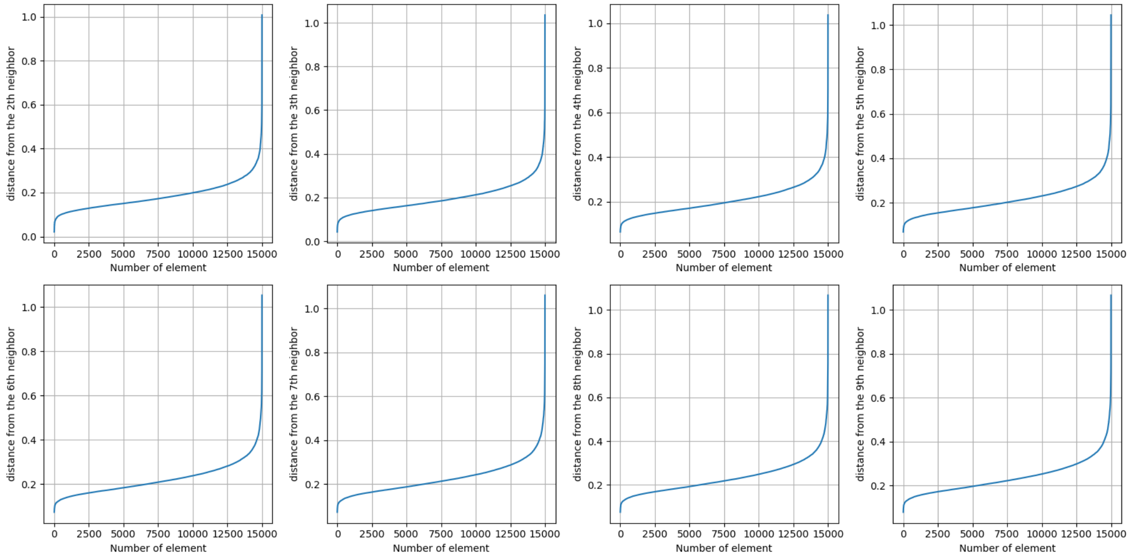


Figure 2.7: Distance of each point to its k^{th} nearest neighbors

Figure 2.7 serves a crucial role in elucidating the reasoning behind the selection of the (ϵ) parameter. In our DBSCAN algorithm, we maintained a consistent (ϵ) value of 0.4 for all "k" values. This decision was based on the insight that beyond this threshold, each graph distinctly shifts from a relatively gradual slope to a considerably steeper one, forming the discernible "elbow" or "knee" of the curve. This characteristic facilitated an effective differentiation between noise points and cluster points.

After determining the optimal (ϵ), we proceeded to run the algorithm for all MinPts values up to 10. Unfortunately, none of the obtained results proved satisfactory. Specifically, the resulting clustering consistently featured a large cluster comprising over 14800 points, alongside several small clusters with a minimal number of elements. Despite experimenting with different (ϵ) values, the results did not improve. Consequently, we concluded that DBSCAN was not well-suited for our dataset. Similar outcomes were expected when using the OPTICS algorithm and our expectations were realized. This confidently led us to assert that our dataset is not conducive to clustering techniques based on the density concept.

2.3 Analysis by hierarchical clustering

After completing analyses with both K-means and DBSCAN methods, we shifted our focus to hierarchical clustering. We employed the agglomerative clustering method, utilizing the same datasets as in the previous two instances.

2.3.1 Single linkage

Our exploration commenced by evaluating clustering using Single linkage as the linkage criterion, employing Euclidean distance to calculate distances between points. In our initial approach, we constructed the dendrogram, without specifying a threshold. Upon reviewing the dendrogram, we decided to assess clustering with 7 clusters. Subsequently, we computed silhouette scores for clusters within the range of 2 to 50. Analyzing the graph, with a specific emphasis on the ”elbow” or ”knee” of the curve, we determined that 7 clusters represented a suitable cut for our clustering. In this instance, the silhouette score was calculated to be 0.2166. During the clustering analysis with the number of clusters set to 7, a noticeable imbalance emerged, characterized by the first cluster containing 14994 elements, while all other clusters consisted of only one element each. This imbalance can be attributed to the sensitivity of the Single linkage criterion to outliers or noise points. Upon inspecting the dendrogram, we acknowledged that this type of cluster sizing quality would likely persist even with improved silhouette values achievable by reducing the number of clusters. Furthermore, the outcome would remain consistent even after eliminating outliers from the dataset. Concluding our analysis with the Single linkage criterion, we assert that, for our dataset, hierarchical clustering using this method doesn’t furnish meaningful insights due to persistent imbalance issues.

2.3.2 Complete linkage

Transitioning from Single linkage, we implemented Agglomerative Clustering using the Complete linkage criterion. The procedure commenced with the construction of the dendrogram, in Figure 2.8, adhering to the same steps as in the previous approach. The silhouette scores per number of clusters are depicted in Figure 2.9.

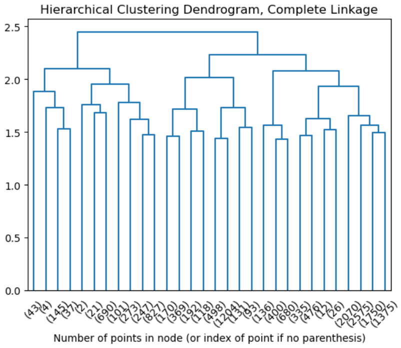


Figure 2.8: Dendrogram C-L

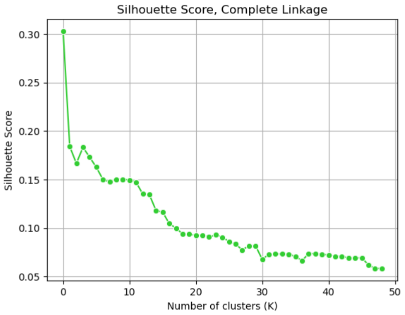


Figure 2.9: Silhouette for dendrogram C-L

Upon inspecting the dendrogram, we opted to conduct our analyses with 5 clusters, a choice corroborated by the silhouette graphic. This time, the silhouette value reached 0.1831, and the points exhibited a more favourable distribution compared to the hierarchical single linkage method. The resulting clusters comprised 8619, 2775, 2161, 1216, and 229 points. We attempted the analysis with 3 clusters, but the results proved inconsequential. The silhouette scores showed remarkable similarity. However, when opting for 5 clusters, the outcome was more meaningful and displayed a discernible improvement in terms of cluster sizing quality. The result achieved with the Complete linkage criterion is unsurprising, considering its lower sensitivity to outliers or noise points.

2.3.3 Average linkage

In our ultimate assessment employing the Average linkage criterion, we adhered to the same steps as with the two preceding criteria. We began by opting for 5 clusters based on the dendrogram. Upon examining the silhouette graph, we observed a significant improvement, reaching a value of 0.2938 – the highest achieved in hierarchical clustering so far. However, akin to Single linkage criterion, the challenge persisted regarding the quality of cluster sizings. Notably, one cluster encompassed 13249 points, while all other clusters included only a handful or just one point. Consequently, we abstained from this method as it proved challenging to derive any meaningful insights from it.

2.3.4 Hierarchical agglomerative clustering randomization

After establishing that the most effective linkage criterion, considering both the Silhouette Score and well-distributed cluster sizes, is the Complete linkage, it is imperative to scrutinize and validate this outcome. We conducted a parallel task similar to what was done for K-Means, using randomization. Figure 2.10 illustrates the frequency distribution of the Silhouette scores from 500 randomly generated datasets. The graph provides valuable insights, revealing a normal distribution with a median of 0.302993 and a mean of 0.309356. This indicates that the Silhouette Score of our dataset, at 0.1831, is notably lower than the average value one would obtain by randomly generating 500 datasets with equivalent dimensions and populating them with values ranging from 0 to 1. This significant result elucidates why our data is not conducive to hierarchical clustering. However, it aligns with expectations, given the pronounced imbalance in our dataset and the presence of outliers. This explains why randomized datasets, exhibiting a more even distribution between 0 and 1, yield higher Silhouette Scores than our dataset in Hierarchical Clustering, a method susceptible to such outliers.

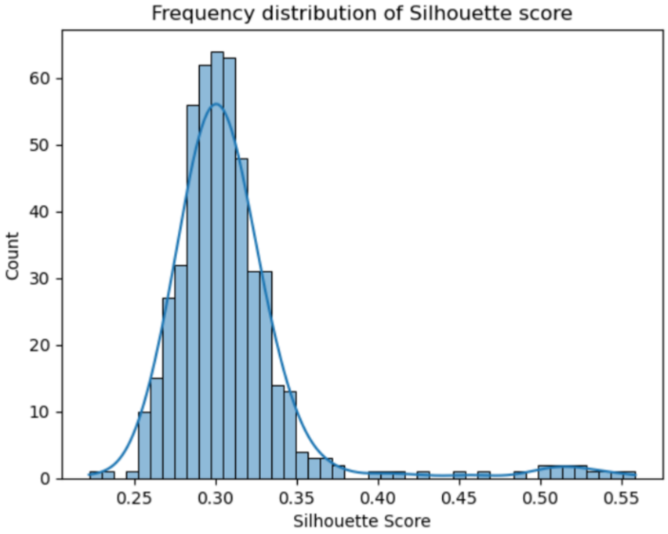


Figure 2.10: SSE and Silhouette score for K-Means

2.4 Final discussion

In summary, K-means clustering with four clusters emerged as the optimal choice, supported by both Sum of Squares Error and Silhouette score analyses. Randomization experiments were fundamental in confirming its effectiveness and the quality of the abovementioned validation metrics. Density-based methods like DBSCAN and OPTICS proved unsuitable due to persistent imbalances and sensitivity to outliers. Hierarchical clustering, specifically using Complete linkage, offered valuable insights. However, challenges in cluster sizings persisted, and randomization revealed limitations in the suitability of hierarchical clustering. However, it's noteworthy that K-means clustering with five or seven clusters also yielded favourable results. This flexibility suggests that, even though the four-cluster solution is deemed optimal, there is room for a more adaptable choice based on specific analytical needs.

Chapter 3

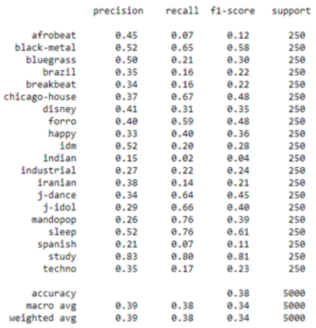
Classification

In this chapter, we explore the application of three classification algorithms - Naïve Bayes, K-Nearest Neighbors(KNN), and Decision Tree. Our primary aim is to predict the 'genre' attribute, and secondarily, we aim to streamline the classification task by predicting four classes associated with the identified clusters in the prior section. This is motivated by the small size of our training dataset, making the accurate classification of 20 genres challenging due to their similarities. To address this, we introduce a new column, "grouped_genres", assigning labels ranging from 0 to 3, where 0 corresponds to the green cluster which, in Figure 2.6, was characterized by high energy and low instrumentalness; class 1 corresponds to the blue cluster with low energy and high instrumentalness; class 2 represents the red cluster with both high energy and instrumentalness, and finally class 3 is associated to the yellow cluster with both low energy and instrumentalness. Subsequently, we conduct data preparation, excluding features with negligible information contribution, handling missing values, and normalizing the datasets. The holdout technique is employed to partition the dataset into a "train set" (80% for training) and a "validation set" (20% for validation). Stratification ensures balanced class instances in both sets, enhancing result reliability and avoiding overlap. Occasionally, to enhance the robustness of our security measures, we opt for employing cross-validation with a k-fold value of 5. Our chapter concludes with a comparative analysis of the three algorithms, considering metrics and overall performance to identify the best model for our dataset and task.

Cluster name	Class label
high energy and low instrumentalness	0
low energy and high instrumentalness	1
high energy and high instrumentalness	2
low energy and low instrumentalness	3

Table 3.1: 4 classes labels mapping

3.1 Naïve Bayes



```
               precision    recall  f1-score   support

     afrobeat       0.45      0.07      0.12       250
  black-metal       0.52      0.65      0.58       250
    bluegrass       0.50      0.21      0.30       250
       brazil       0.35      0.16      0.22       250
    breakbeat       0.34      0.16      0.22       250
chicago-house       0.37      0.67      0.48       250
       disney       0.41      0.31      0.35       250
        forro       0.40      0.59      0.48       250
        happy       0.33      0.40      0.36       250
          idm       0.52      0.20      0.28       250
       indian       0.15      0.02      0.04       250
   industrial       0.27      0.22      0.24       250
      iranian       0.38      0.14      0.21       250
      j-dance       0.34      0.64      0.45       250
       j-idol       0.29      0.66      0.40       250
     mandopop       0.26      0.76      0.39       250
        sleep       0.52      0.76      0.61       250
      spanish       0.21      0.07      0.11       250
        study       0.83      0.80      0.81       250
       techno       0.35      0.17      0.23       250

     accuracy                           0.38      5000
    macro avg       0.39      0.38      0.34      5000
 weighted avg       0.39      0.38      0.34      5000
```

Figure 3.1: Scores for Naïve Bayes - Test

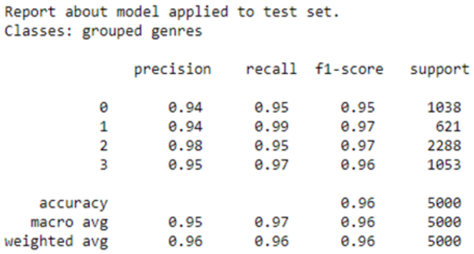
For this classification task, we tested Naïve Bayes with both Gaussian and Categorical classifiers, considering the mix of categorical and continuous variables in our dataset. The initial classification focused on normalized continuous variables using GaussianNB, involving all the numerical features, except popularity, and 1 target variable ("genre"). The accuracy achieved on the validation set was 0.38, with a slight improvement to 0.39 on the test set as shown in Figure 3.1. Analysing the classification report revealed varying accuracies across genres, with "study" showing the highest accuracy and "Indian" the lowest. The Confusion Matrix highlighted discrepancies in true positive counts among genres, indicating differences in classification performance. The overall performance, also supported by the AUC-ROC curve, was satisfactory. The multiclass curve, implemented through One-vs-Rest, showed AUC values ranging from 0.76 for "afrobeat" to 0.99 for "study". To assess the model in different scenarios, we applied repeated holdout to create 100 different partitions, without a specified random state but maintaining the stratified approach. The resulting median accuracy remained consistent at 0.38, mirroring the accuracy obtained with our initial partition.

3.1.1 Binary classification

Extending our analysis to predict binary classes, specifically targeting firstly ”mode” and successively “explicit”, we encountered a challenge with class imbalances. For ”mode”, 74% of records were labelled as ”1” and the remaining as ”0”. In predicting ”mode”, the model achieved an apparent high accuracy of 0.73 on the validation set. However, a deeper inspection of the confusion matrix revealed a bias, as 98% of predictions were labelled as ”1”. This result is misleading, and accuracy is high only because in this circumstance almost the totality of the tracks have mode ”1”. This trend persisted in the test set due to its similar disproportion. A parallel issue emerged with the ”explicit” target variable. The substantial class imbalance led the model to predominantly predict all records as ”0”. This situation further compromises the reliability of the analysis for both target variables. In conclusion, the limited number of instances for the smaller class undermines the reliability of our analysis. Addressing this imbalance through increased instances in the underrepresented class is imperative for robust and generalizable predictive models.

Naïve Bayes applied at result emerged by clustering

Next, we performed an analysis using the Gaussian Classifier, on four distinct groups in our dataset identified through prior clustering. Given the dataset size, predicting four classes was deemed more suitable. While three groups were balanced, ”group 2” had a predominant imbalance with 5478 records out of 15000. To address this, we applied a stratified holdout technique, effectively managing class distribution. The model demonstrated commendable performance, achieving 0.96 accuracy on both the validation and test sets, as we can see in Figure 3.2. Insights from the confusion matrix highlighted the model's ability to navigate imbalanced classes with high precision, we can notice this behaviour in Figure 3.3.



```
Report about model applied to test set.
Classes: grouped genres

              precision    recall  f1-score   support

           0       0.94      0.95      0.95      1038
           1       0.94      0.99      0.97       621
           2       0.98      0.95      0.97      2288
           3       0.95      0.97      0.96      1053

    accuracy                           0.96      5000
   macro avg       0.95      0.97      0.96      5000
weighted avg       0.96      0.96      0.96      5000
```

Figure 3.2: Scores for Naïve Bayes 4 classes

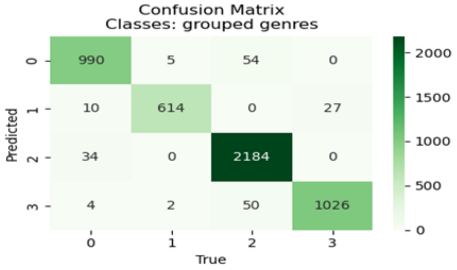


Figure 3.3: Naïve Confusion Matrix 4 classes

Supplementing our findings, the AUC-ROC curve yielded the highest score so far for the Naïve Bayes classifier, affirming its discriminative prowess in the four-class prediction task.

3.1.2 Categorical classifier

The utilization of the Categorical classifier within the Naïve Bayes framework to predict the four classes using the same features as before yielded less satisfactory results compared to previous analyses. The obtained accuracy of 0.77 falls below the expected threshold for a model tasked with discerning among a set of four classes.

3.2 K-Nearest Neighbors (K-NN)

We performed the second classification employing the K-Nearest Neighbors (K-NN) classifier, utilizing the Euclidean distance metric as a proximity measure. The performance of the K-NN classifier varies significantly depending on the chosen parameter K. The selection of an appropriate value for K is crucial. If K is too small, the model may be sensitive to noise in the data; if K is too large, the model might not capture complex patterns.

3.2.1 K-NN applied at Genre

The precision of the algorithm in predicting the ”genre” of tracks exhibits an apex value of 0.324 when varying the parameter K, with optimal performance observed at K=30. Notably, the model demonstrates enhanced accuracy in predicting specific genres, such as the ”sleep” genre reaching an accuracy of 0.72. However, the overall results are deemed unsatisfactory, suggesting that the K-NN classifier may not be optimally suited for this classification task. Subsequent implementation of cross-validation aimed to discern potential overfitting, yet the comparative analysis did not unveil significant differences in results.

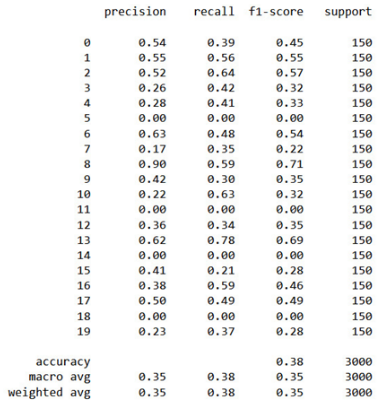
3.2.2 K-NN applied at result emerged by clustering

Here are presented classification results obtained for the clustering labels using the K-NN classifier, with overarching conclusions articulated in the last paragraph of this chapter. Since the assignment of our labels uses straightforward criteria, the K-NN classifier demonstrated high accuracy in predicting these labels, reaching 0.95 accuracy with K=2, but these results must be approached with caution as we will discuss later.

3.3 Decision tree

3.3.1 Decision tree applied on Genre

In the validation phase of the model aiming to predict all 20 genres based on 15 features, the training set achieved a perfect accuracy of 1.0 across all five folds of the cross-validation process. However, this seemingly exceptional performance raises concerns about potential overfitting, as a perfect fit to the training data may not generalize well to unseen instances. The limited size of the dataset may contribute to the model essentially ”memorizing” the training set rather than learning the underlying patterns. Subsequently, when assessing the accuracy on the validation sets, the average accuracy across the five folds was 0.43, indicating a substantial drop from the flawless training accuracy. In pursuit of improved accuracy and a balanced trade-off between model complexity and generalization, a two-fold approach was adopted. Firstly, the random search method was employed for hyperparameter tuning, resulting in an enhancement of accuracy to 0.49. Notable were the improvements in F1 scores for specific classes, with a significant boost for the ”black_metal” class (”2”), reaching 0.71, as well as impressive scores of 0.76 and 0.78 for the ”sleep”(”8”) and ”study”(”13”) classes, respectively. This highlighted the efficacy of the random search method in fine-tuning the model’s performance for specific class predictions. Secondly, post-pruning using a ccp_alpha of 0.004 was implemented to curb overfitting and strike a balance in model complexity. However, this strategy led to a decline in overall accuracy to 0.38 (reported in Figure 3.4) and demonstrated worse F1 scores for the classes of interest. This trade-off underscores the delicate balance required in optimizing a decision tree model, considering both hyperparameter tuning and post-pruning strategies to achieve the desired performance across diverse classes. Upon the introduction of the test set, initial results mirrored those observed in the validation set, manifesting a train accuracy of 1.0 and a test accuracy of 0.44. However, following the implementation of the random search method, the train accuracy decreased to 0.59 while the test accuracy increased to 0.48. This adjustment can be attributed to the randomized exploration of hyperparameter space leading to a more generalized model.



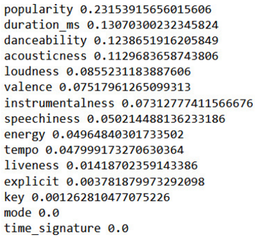
```
              precision    recall  f1-score   support

           0       0.54      0.39      0.45       150
           1       0.55      0.56      0.55       150
           2       0.52      0.64      0.57       150
           3       0.26      0.42      0.32       150
           4       0.28      0.41      0.33       150
           5       0.00      0.00      0.00       150
           6       0.63      0.48      0.54       150
           7       0.17      0.35      0.22       150
           8       0.90      0.59      0.71       150
           9       0.42      0.30      0.35       150
          10       0.22      0.63      0.32       150
          11       0.00      0.00      0.00       150
          12       0.36      0.34      0.35       150
          13       0.62      0.78      0.69       150
          14       0.00      0.00      0.00       150
          15       0.41      0.21      0.28       150
          16       0.38      0.59      0.46       150
          17       0.50      0.49      0.49       150
          18       0.00      0.00      0.00       150
          19       0.23      0.37      0.28       150

    accuracy                           0.38      3000
   macro avg       0.35      0.38      0.35      3000
weighted avg       0.35      0.38      0.35      3000
```

Figure 3.4: Post-pruning scores in val set

The consequential F1 scores were: class "black_metal"(2) achieving 0.68, "sleep"(8) reaching 0.74, and "study"(13) also attaining 0.74. The top features influencing predictions, as indicated by feature importance, were identified as popularity, duration_ms, danceability, and acousticness, as depicted in the accompanying Figure 3.5. Subsequent post-pruning with ccp_alpha set at 0.004 induced a further reduction in train accuracy to 0.35 and test accuracy to 0.34. Notably, specific classes, such as "afrobeat"(11), "indian"(14), "brazil"(18), and "breakbeat"(19), had trouble recognizing any records post-pruning. This phenomenon underscores the impact of pruning in simplifying the decision tree and, in some cases, limiting the model's ability to discern certain classes.



```
popularity 0.23153915656015606
duration_ms 0.13070300232345824
danceability 0.1238651916205849
acousticness 0.1129683658743806
loudness 0.0855231183887606
valence 0.07517961265099313
instrumentalness 0.07312777411566676
speechiness 0.050214488136233186
energy 0.04964840301733502
tempo 0.047999173270630364
liveness 0.01418702359143386
explicit 0.003781879973292098
key 0.001262810477075226
mode 0.0
time_signature 0.0
```

Figure 3.5: Features Influencing predictions

3.3.2 Decision Tree applied at result emerged by clustering

In the context of the 4-class decision tree, initial observations during the training phase unveiled a 1.0 average accuracy across the 5-fold cross-validation and a commendable 0.96 accuracy for the validation set, which further increased to 0.97 following random search. However, a subsequent application of ccp_alpha set at 0.0025 led to a reduction in accuracy to 0.91. Intriguingly, in the testing set, the initial training accuracy remained at an impeccable 1.0, while the test accuracy slightly decreased to 0.97. Random search induced minimal alterations, leaving both train and test accuracy at 0.93 after ccp_alpha adjustment. The resulting decision tree after post-pruning is illustrated in Figure 3.6

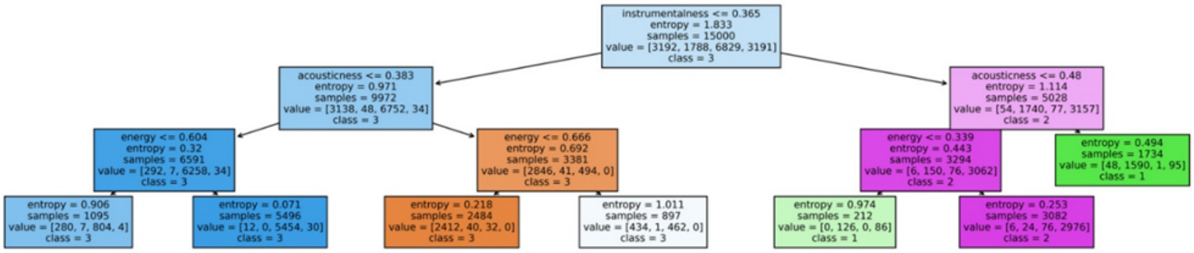


Figure 3.6: Final decision tree

3.3.3 Conclusions on the classification of the classes emerged by clustering

The remarkably high accuracy observed, in predicting the 4-classes, could stem from various factors. One potential explanation is that the classes formed through clustering might exhibit distinctive patterns or characteristics that the model has effectively learned and generalized. Clustering techniques often result in well-defined and separable groups, making it easier for many classifier models to discern and predict these clusters accurately. However, it's essential to approach such high accuracy with caution, as it could also indicate a degree of overfitting, additional datasets would be valuable to ensure the model's robustness and generalizability.

3.4 Conclusions and evaluation of algorithms

For Genre Classification, Naïve Bayes exhibited satisfactory performance, providing reliable and consistent results across different scenarios. Although Decision Tree showed promise after hyperparameter tuning, concerns about overfitting and a decline in accuracy post-pruning make Naïve Bayes a more robust choice for this task. In the context of Four-Class Classification, each algorithm exhibited strengths and areas for improvement. Naïve Bayes, with its consistently high accuracy, effective handling of imbalanced classes, and discriminative prowess, stands out as a robust performer. KNN showcased high accuracy but raised concerns about overfitting, necessitating careful consideration in its application. The Decision Tree, after tuning, demonstrated improved accuracy but faced challenges in maintaining generalization post-pruning.

Chapter 4

Pattern Mining

In this section, we outline the Apriori and FP-Growth algorithms' process and results. We first identified frequent patterns, including closed and maximal item sets. Next, we applied rules to reveal deeper associations and utilized these patterns for predicting a target variable and filling in missing values.

4.1 Data Processing

We began by identifying key variables in our dataset, with 11 out of 15 being the numerical variables. We discretized these values into four ranges. The remaining 4 features were explicit, mode, genre, and grouped_genres. For binary features (explicit and mode), we adjusted labels without discretization. After analyzing different thresholds, we removed the unbalanced explicit and mode variables, as we can see from Figure 4.1, one of the two states of mode appears only when the support threshold is very low. The same happened with "explicit". The feature "genre" was also removed as single genres never appeared in frequent itemsets; thus, grouped_genres served as its discretized version. In the end, we worked with 12 features.

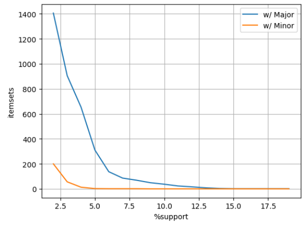


Figure 4.1: Support comparison between mode: Minor, mode: Major

4.2 Apriori Algorithm

4.2.1 Frequent pattern recognition

In the initial task, we focused on identifying frequent itemsets, experimenting with various thresholds for min_support and the minimum number of elements per itemset. Optimal parameters were found with a support between 8% and 10% and a minimum of 3 elements per itemset. An 8% support with a minimum of 3 items yielded 40 frequent patterns. Notably, the number of closed itemsets matched the number of frequent itemsets (40), but the count of maximal itemsets was slightly lower, 36, aligning with theoretical expectations. Figure 4.2 illustrates the variation in itemset numbers at different support thresholds.

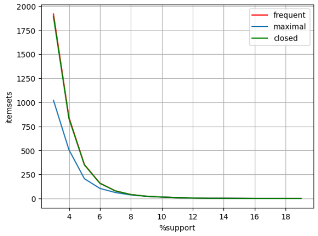


Figure 4.2: N°of itemset per supp.

4.2.2 Rules extraction

Extracting rules, a key aspect of pattern mining was performed with a 9% support threshold, a minimum of 3 items per itemset, and a 60% confidence threshold, resulting in 204 rules. Key metrics, like confidence and lift, were crucial for assessing rule strength and quality. Notably, over half of the rules had a lift greater than 2, indicating a positive impact. The heatmap in Figure 4.3 illustrates the distribution of rules (Figure 4.4) across various confidence and support levels.

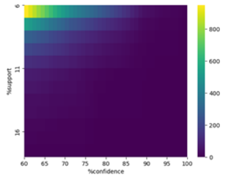
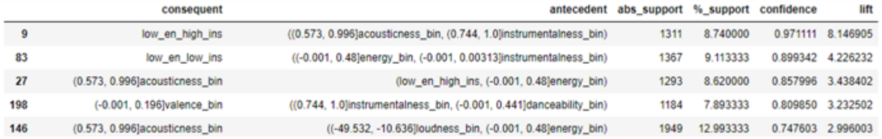


Figure 4.3: N° of itemset generated by different thresholds



	consequent	antecedent	abs_support	%_support	confidence	lift
9	low_en_high_ins	((0.573, 0.996]acousticness_bin, (0.744, 1.0]instrumentalness_bin)	1311	8.740000	0.971111	8.146905
83	low_en_low_ins	((-0.001, 0.48]energy_bin, (-0.001, 0.00313]instrumentalness_bin)	1367	9.113333	0.899342	4.226232
27	(0.573, 0.996]acousticness_bin	(low_en_high_ins, (-0.001, 0.48]energy_bin)	1293	8.620000	0.857996	3.438402
198	(-0.001, 0.196]valence_bin	((0.744, 1.0]instrumentalness_bin, (-0.001, 0.441]danceability_bin)	1184	7.893333	0.809850	3.232502
146	(0.573, 0.996]acousticness_bin	((-49.532, -10.636]loudness_bin, (-0.001, 0.48]energy_bin)	1949	12.993333	0.747603	2.996003

Figure 4.4: Items generated by rules

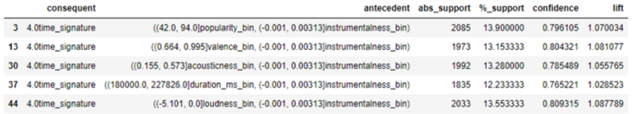
4.2.3 Target variable prediction

After identifying rules, we employed the 4 classes of ”grouped_genres” as the consequent values for predicting the track’s group. This unsupervised learning task yielded the results reported in Table 4.1.

Table 4.1: ”grouped_genre” recognition by pattern mining

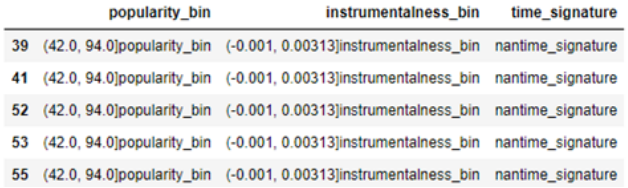
Consequent	Antecedent	Support	Confidence	Lift
high_en_low_ins	(0.00974, 0.155]aco_bin, (-0.001, 0.00313]ins_bin	14.34	0.995	2.186
low_en_high_ins	(0.573, 0.996]acou_bin, (0.744, 1.0]ins_bin	8.74	0.9711	8.146
high_en_high_ins	(0.744, 1.0]ins_bin, (-0.001, 14.0]popu_bin	6.66	0.616	2.898
low_en_low_ins	(0.573, 0.996]acou_bin, (-0.001, 0.00313]ins_bin	11.58	0.978	4.596

The most reliable pattern for predicting the ”high_en_low_ins” class occurs when acousticness_bin is (0.00974, 0.155] and instrumentalness_bin is (-0.001, 0.00313], appearing 2151 times with a confidence of 0.99 and a lift of 2.86. The ”low_en_high_ins” class also shows high values, while other classes have lower matrices, yet confidence is consistently above 0.60, and lift exceeds 2. We applied the same technique to fill in missing values in the ”time_signature” feature. Initially, it had 2062 missing values, that, up until now, we filled with the mode. Without imputing missing values, we used different patterns to predict them, successfully filling in the dataset by applying rules based on high-probability itemsets.



	consequent	antecedent	abs_support	%_support	confidence	lift
3	4.0time_signature	((42.0, 94.0]popularity_bin, (-0.001, 0.00313]instrumentalness_bin)	2085	13.900000	0.796105	1.070034
13	4.0time_signature	((0.664, 0.995]valence_bin, (-0.001, 0.00313]instrumentalness_bin)	1973	13.153333	0.804321	1.081077
30	4.0time_signature	((0.155, 0.573]acousticness_bin, (-0.001, 0.00313]instrumentalness_bin)	1992	13.280000	0.785489	1.055765
37	4.0time_signature	((180000.0, 227826.0]duration_ms_bin, (-0.001, 0.00313]instrumentalness_bin)	1835	12.233333	0.765221	1.028523
44	4.0time_signature	((-5.101, 0.0]loudness_bin, (-0.001, 0.00313]instrumentalness_bin)	2033	13.553333	0.809315	1.087789

Figure 4.5: Rules generated with ”4.0time_signature” as consequecne



	popularity_bin	instrumentalness_bin	time_signature
39	(42.0, 94.0]popularity_bin	(-0.001, 0.00313]instrumentalness_bin	nantime_signature
41	(42.0, 94.0]popularity_bin	(-0.001, 0.00313]instrumentalness_bin	nantime_signature
52	(42.0, 94.0]popularity_bin	(-0.001, 0.00313]instrumentalness_bin	nantime_signature
53	(42.0, 94.0]popularity_bin	(-0.001, 0.00313]instrumentalness_bin	nantime_signature
55	(42.0, 94.0]popularity_bin	(-0.001, 0.00313]instrumentalness_bin	nantime_signature

Figure 4.6: Filling in nan time signature with pattern mining

4.3 FP-Growth

FP-growth is one of the two most common algorithms in pattern mining. The main difference with the Apriori is how the patterns or itemsets are created since FP-growth generates a tree while the Apriori joins the items based on a minimum support threshold. In practice, the difference concerns the computational costs, but considering our dataset differences are irrelevant, in fact, the results generated by the Apriori are the same generated by FP-growth. The only difference that emerged is related to dimensions of frequent, closed, and maximal itemset, as all of which have 36 itemsets. The rules generated are the same in both algorithms and when adjusting thresholds, they mirror each other while maintaining equality.

Chapter 5

Regression

To conduct regression analysis, we prepared both a training and a test dataset using standard preprocessing techniques akin to those employed in the Classification section. However, categorical variables were excluded for computational simplicity and adherence to regression assumptions. The resulting set of variables includes ”duration_ms”, ”popularity”, ”danceability”, ”energy”, ”loudness”, ”speechiness”, ”acousticness”, ”instrumentalness”, ”liveness”, ”valence” and ”tempo”.

5.1 Simple Regression

5.1.1 “popularity” as dependent variable

Initial attempts to establish simple regression models with popularity as the dependent variable (y) and various numerical attributes as independent variables (x) proved unsuccessful. Despite diligent efforts, including outlier removal, no significant correlations were observed. Subtle improvements were noted with speechiness or loudness as independent variables, but they were not noteworthy.

5.1.2 ”loudness” as dependent variable

Guided by both matrix correlation analysis and domain knowledge, the focus shifted towards predicting loudness using energy as the independent variable. The resultant linear regression model exhibited significant enhancement, yielding an impressive R-squared value of 0.516 and a mean squared error (MSE) of 17.929. These metrics underscore the model’s effectiveness in capturing the intricate relationship between energy and loudness. The result is depicted in Figure 5.1. Nonlinear models, specifically Decision Tree and K-NN, outperformed linear models with R-squared values of 0.578 and 0.564. That can be attributed to their intrinsic advantages, such as their independence from the presupposition of linearity or their robustness to outliers.

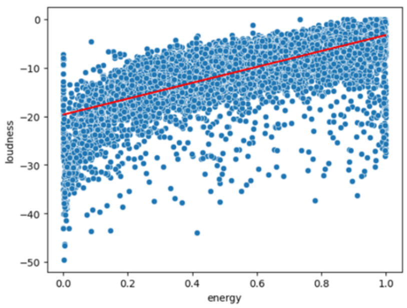


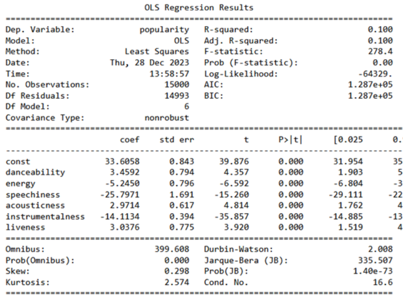
Figure 5.1: Simple linear regression: energy, loudness

5.2 Multiple Regression

5.2.1 ”popularity” as dependent variable

A multiple regression analysis was initially conducted with popularity as the dependent variable. The first attempts, incorporating all numerical variables as independent variables, yielded unsatisfactory results. Subsequent steps involved feature selection using backward elimination, considering statistical significance, and addressing multicollinearity issues.

The selected features ('acousticness', 'instrumentalness','liveness', 'energy', 'danceability', 'speechiness') resulted in marginal improvements in linear models, as reflected by an R-squared value of 0.109. However, the disparity between the R-squared values of the training model (0.100) and the test set (0.109) raises concerns about potential overfitting or the limitations posed by the dataset size, scores are reported in Figure 5.2. Notably, even regularization techniques such as Lasso or Ridge proved unsatisfactory in addressing this discrepancy. Nonlinear models exhibited significant underperformance, especially the Decision Tree with an exceptionally low R-squared value of -0.556.

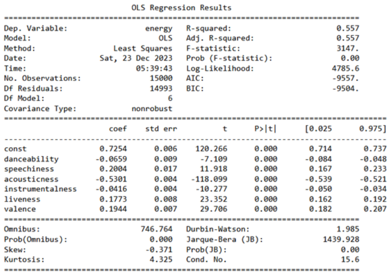


```
                            OLS Regression Results
==============================================================================
Dep. Variable:             popularity   R-squared:                       0.100
Model:                            OLS   Adj. R-squared:                  0.100
Method:                 Least Squares   F-statistic:                     278.4
Date:                Thu, 28 Dec 2023   Prob (F-statistic):               0.00
Time:                        13:58:57   Log-Likelihood:                -64329.
No. Observations:               15000   AIC:                         1.287e+05
Df Residuals:                   14993   BIC:                         1.287e+05
Df Model:                           6
Covariance Type:            nonrobust
====================================================================================
                       coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
const               33.6058      0.843     39.876      0.000      31.954      35.258
danceability         3.4592      0.794      4.357      0.000       1.903       5.015
energy              -5.2450      0.796     -6.592      0.000      -6.804      -3.685
speechiness        -25.7971      1.691    -15.260      0.000     -29.111     -22.483
acousticness         2.9714      0.617      4.814      0.000       1.762       4.181
instrumentalness   -14.1134      0.394    -35.857      0.000     -14.885     -13.342
liveness             3.0376      0.775      3.920      0.000       1.519       4.557
==============================================================================
Omnibus:                      399.608   Durbin-Watson:                   2.008
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              335.507
Skew:                           0.298   Prob(JB):                     1.40e-73
Kurtosis:                       2.574   Cond. No.                         16.6
==============================================================================
```

Figure 5.2: Multiple regression scores: popularity (y)

5.2.2 "energy" as dependent variable

Given the dissatisfaction with popularity as the dependent variable, the focus shifted to energy. After normalization and feature selection, specifically involving backward elimination considering statistical significance and mitigation of multicollinearity issues, the selected features ('danceability', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence') produced promising results. Linear models achieved an R-squared of 0.567, despite the training model exhibiting an R-squared of 0.557, as reported in Figure 5.3 due to the issues previously outlined. In contrast, Lasso regression demonstrated less effectiveness (R-squared: -0.001) due to its feature selection mechanism aggressively penalizing important variables, leading to a suboptimal fit. Notably, Decision Tree and K-NN outperformed linear models with R-squared values of 0.427 and 0.614, respectively. It's crucial to highlight that the Decision Tree achieved a perfect R-squared of 1.000 in the training set, a result influenced by the inherent characteristics of the algorithm, while K-NN attained a training R-squared of 0.748. Considering these results, and its additional impressive metric such as MSE: 0.027 and MAE: 0.125 it is appropriate to express that K-NN emerged as the best-performing model. Its high R-squared values on both the training and test sets indicate a strong ability to capture the underlying patterns in the data, as evidenced by the scatter plot illustrating residuals for K-NN predictions vs. actual values, in Figure 5.4. The colour map employed in the plot provides insights into areas where the model might be less accurate. Notably, the vast majority of points are set at a colour corresponding to residuals between 0 and 0.2. This concentration of points in a specific colour range implies that a significant portion of predictions falls within a small margin of error, showcasing the model's accuracy in capturing the true values.



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 energy   R-squared:                       0.557
Model:                            OLS   Adj. R-squared:                  0.557
Method:                 Least Squares   F-statistic:                     3147.
Date:                Sat, 23 Dec 2023   Prob (F-statistic):               0.00
Time:                        05:39:43   Log-Likelihood:                 4785.6
No. Observations:               15000   AIC:                            -9557.
Df Residuals:                   14993   BIC:                            -9504.
Df Model:                           6
Covariance Type:            nonrobust
====================================================================================
                       coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
const                0.7254      0.006    120.266      0.000       0.714       0.737
danceability        -0.0659      0.009     -7.109      0.000      -0.084      -0.048
speechiness          0.2004      0.017     11.918      0.000       0.167       0.233
acousticness        -0.5301      0.004   -118.099      0.000      -0.539      -0.521
instrumentalness    -0.0416      0.004    -10.277      0.000      -0.050      -0.034
liveness             0.1773      0.008     23.352      0.000       0.162       0.192
valence              0.1944      0.007     29.706      0.000       0.182       0.207
==============================================================================
Omnibus:                      746.764   Durbin-Watson:                   1.985
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1439.928
Skew:                          -0.371   Prob(JB):                         0.00
Kurtosis:                       4.325   Cond. No.                         15.6
==============================================================================
```

Figure 5.3: Multiple regression scores: energy (y)

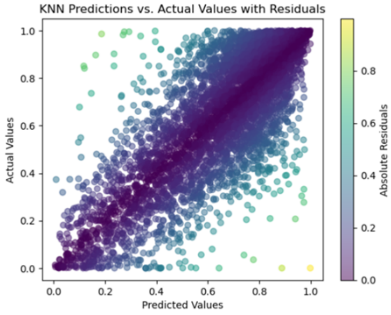


Figure 5.4: K-NN vs Actual Values with residuals

5.3 Conclusion

In summary, multiple regression analysis on the variable "popularity" resulted in modest improvements with selected features, yet concerns were raised regarding potential overfitting and limitations imposed by the dataset size. Shifting the focus to "energy" as the dependent variable, K-NN emerged as the better-performing model, demonstrating improved predictive accuracy and resilience to nonlinear patterns, as reflected in a relatively high R-squared of 0.614 and noteworthy metrics such as MSE: 0.027 and MAE: 0.125. The concentration of residuals within a narrow range further underscored K-NN's precision in capturing the true values of the dependent variable.