UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

Algorithmics for Data Mining

Exploratory Data Analysis and Preprocessing for Spotify Dataset

Abstract

In this report we perform exploratory data analysis on a Spotify song dataset from Kaggle. We conduct univariate analyses and encounter issues of scales and non-normality. While performing multivariate analyses, we find interesting relationships between variables such as valence and danceability, or energy and acousticness. Through PCA we are able to discover that most of the variability in the data is derived from the volume of a track and its overall danceability/liveliness. In the second phase of this report we conduct preprocessing which includes, Z-score scaling, univariate outlier imputation through K-nn, and multivariate outlier removal. All of these steps are aided by the guidance of related literature, and implemented with the goal of preparing the dataset for machine learning models.

Darryl Abraham

Riccardo Paciello

darryl.abraham@estudiantat.upc.edu
riccardo.paciello@estudiantat.upc.edu

May 10, 2024

Contents

Link to Github repository, containing all code used to obtain the results found in this report: https://github.com/darryl-abraham/adm/

1 Introduction

Music is an art form that transcends societal and cultural boundaries having always held a significant place in society. Songs possess the ability to evoke emotions, convey complex narratives, and serve as channels for cultural development. In the digital age, the proliferation of streaming platforms like Spotify has revolutionized how we access and consume music. In the midst of this endless amount of musical content, the challenge of algorithmically evaluating songs for analyses and recommendations has become a significant part of the music industry. This challenge underscores the importance of research aimed at understanding song data and developing robust methodologies for modelling and recommendation.

Exploratory data analysis (EDA) is the starting point for our research. We aim to offer insights into the underlying structure and characteristics of the data. Employing data science techniques and principles we perform univariate and multivariate analysis, outlier detection, and principal component analysis (PCA). We discover patterns, relationships, and anomalies within song data. Understanding the data is a crucial part of the process and guides us through preprocessing and ultimately modelling the data.

Univariate analysis goes into the distributions and summary statistics of individual song attributes. Multivariate analysis, on the other hand, explores relationships between multiple song features, unveiling dependencies and correlations. The identification of outliers is crucial in maintaining data integrity. It ensures that unusual observations do not negatively influence subsequent analyses. Principal component analysis facilitates extracting meaningful patterns from high dimensional song datasets.

Preprocessing is an essential precursor to machine learning which involves a series of steps aimed at enhancing data quality for machine learning models. Scaling, which involves normalizing numerical attributes, reduces issues stemming from varying scales and units. Additionally, robust handling of univariate and multivariate outliers is necessary to prevent skewed inferences and model inaccuracies.

In this report we outline a comprehensive exploration of song data, employing a multitude of EDA techniques. Subsequently, we explain the significance of preprocessing in preparing song data for downstream clustering analyses. These methodologies informed by academic research will help us build a strong foundation for deeper analyses such clustering, investigating song recommendation systems, and advancing our understanding of musical content in the digital era.

1.1 Dataset

The dataset features a broad selection of Spotify tracks and is available on Kaggle [1]. It consists of 114000 examples (each genre has 1000 songs in the dataset) and 20 variables. Among these variables, the majority are numerical (15), along with 5 categorical variables (not including track ID). Some of the numerical features such as mode, key, time signature, explicit are in reality encoded categorical features.

- Numerical variables: popularity, duration, danceability, energy, loudness, speechiness, acousticness, instrumentalness, liveness, valence, tempo.
- Categorical variables: artists, album name, track name, genre, explicit, key, mode, time signature.

Please visit Kaggle [1] for detailed explanation of the variables.

2 Related Literature

Our research into this dataset revealed a limited number of related scholarly works. Given the limited availability of literature, we understand the necessity for comprehensive review of the existing studies becomes imperative to identify gaps and opportunities for further exploration.

In a study by Pareek et al. [2] the same dataset was used for the purposes of classifying songs in terms of popularity to be able to predict the popularity of songs. For the exploratory data analysis they very briefly investigate the distribution of popularity, and plot a heatmap to identify correlations. We believe that a mix of univariate and multivariate analysis is crucial to understand the data however, we aim to build upon their work with a more thorough investigation into various features and their relationships, in a similar fashion.

Pulling from the work of Luo [3], we found an interesting analysis of Spotify data through Principal Component Analysis. Although the paper is sparse in its data exploration and preprocessing, the use of PCA has shown interesting insights into relationships between features. They show that energy, loudness, and acousticness account for most of the variability (information) in the data. We believe that such methods are useful to provide us with a comprehensive understanding of our data.

Tran et al. [4] used a similar dataset (same features, different instances), collected independently from the Spotify API. Their objective was K-means clustering to enhance music recommendation, and personalisation. In the article, they do not engage in exploratory data analysis, and move straight to preprocessing which is subpar. The only preprocessing they perform is the popular Z-score standardization of data (moving the mean to 0 and the standard deviation to 1). From their discussion we understood the importance of this, especially for a clustering task.

3 Exploratory Data Analysis

In this section we focus on exploratory data analysis (EDA), a fundamental step in understanding the underlying patterns and anomalies of our dataset. By using some statistical techniques, we aim to collect insights that will help us in later stages to take decisions.

3.1 Univariate Analysis

Among all the variables, only three (artists, album_name, track_name) exhibit null values. Specifically, each of these variables is missing only one value. Upon further investigation, it is discovered that the missing values all belong to the same observation. Due to the singular null valued row in the midst of 114000 instances we immediately remove this row from the dataset.

Using the inbuilt *pandas* describe function we get a quick glance into the structure of the data, find the results in Table 1.

Table 1: Summary Statistics

Variable	mean	st. dev.	min	25%	50%	75%	max
popularity	33.2	22.3	0.0	17.0	35.0	50.0	100.0
duration_ms	228031.2	107296.1	8586.0	174066.0	212906.0	261506.0	5237295.0
danceability	0.6	0.2	0.0	0.5	0.6	0.7	1.0
energy	0.6	0.3	0.0	0.5	0.7	0.9	1.0
loudness	-8.3	5.0	-49.5	-10.0	-7.0	-5.0	4.5
speechiness	0.1	0.1	0.0	0.0	0.0	0.1	1.0
acousticness	0.3	0.3	0.0	0.0	0.2	0.6	1.0
instrumentalness	0.2	0.3	0.0	0.0	0.0	0.0	1.0
liveness	0.2	0.2	0.0	0.1	0.1	0.3	1.0
valence	0.5	0.3	0.0	0.3	0.5	0.7	1.0
tempo	122.1	29.9	0.0	99.2	122.0	140.1	243.4

The metrics were trimmed for conciseness. Here we can already determine some issues that arise in terms of data quality, several notable observations emerge. Firstly, the mean *popularity* score is 33.2, with a large standard deviation of 22.3 which is an indication of notable variance in popularity across the dataset. Moreover, the range extends from a minimum of 0.0 to a maximum of 100.0.

Moving on to the *duration_ ms'* variable, the mean duration is 228,031.2 milliseconds, yet the standard deviation is substantial at 107,296.1 milliseconds. The range is extremely large going from 8,586 milliseconds to a maximum of 5,237,295 milliseconds, indicating a very wide diversity in song lengths.

Additionally, considering 'loudness', the mean is -8.3, with a standard deviation of 5.0. Notably, the minimum value is -49.5, which appears strikingly low, possibly indicating an outlier or a song with very low volume.

Furthermore, examining the 'tempo' variable, the mean tempo is 122.1 beats per minute (BPM), with a standard deviation of 29.9 BPM. However, the minimum tempo is listed as 0.0 BPM, which seems anomalous and warrants further investigation or clarification.

Danceability, energy, speechiness, acousticness, instrumentalness, liveness, and valence all have a mean and standard deviation close to 0. The difference is difficult to see

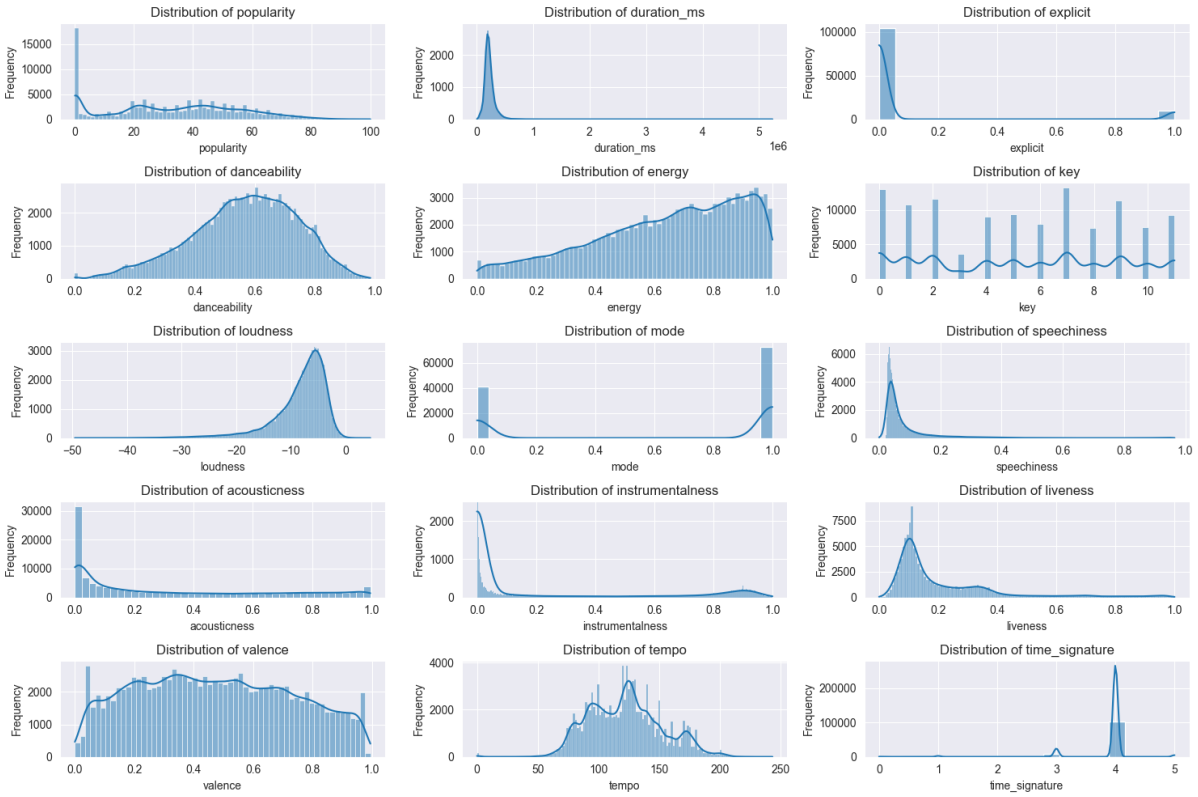


Figure 1: Feature distributions

in Table 1 as the values have been trimmed to accomodate the physical restrictions of the report.

Figure 1 shows the distribution of each (numeric) variable. Some interesting observations:

- Large number of songs are not popular
- There are far less explicit songs than not
- Danceability seems to follow a relatively normal distribution
- The distribution of key seems almost uniform, with only key 3 not being represented
- Instrumentalness is bimodal with the peaks at opposite ends of its range (most songs do not have instrumantalness
- Liveness is bimodal
- Tempo has a vague shape of a normal distribution but with many local peaks throughout
- Most songs follow a time signature of 4, and few songs with 3

In Table 2 we can see artists are the most common (most songs) in the dataset.

Table 2: Top 10 artists by number of songs

	count
The Beatles	279
George Jones	271
Stevie Wonder	236
Linkin Park	224
Ella Fitzgerald	222
Prateek Kuhad	217
Feid	202
Chuck Berry	190
Håkan Hellström	183
OneRepublic	181

3.2 Multivariate Analysis

In this phase, we conducted a multivariate analysis on the numerical variables of the dataset with several objectives in mind. Our first aim was to gain a deeper understanding of the relationships between variable pairs and to identify early patterns or trends in the data. This analysis enables us to find highly correlated features and potentially eliminate some of them. Highly correlated features can lead to multicollinearity in models contributing redundant information to the clustering process, which could lead to clusters being formed based on this redundant information rather than meaningful differences in the data.

In the Python pandas library, the available correlation coefficients include Pearson, Kendall, and Spearman. For the heatmap, we chose the Pearson coefficient, which is calculated using the following formula:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{1}$$

The choice was made to utilize the Pearson correlation coefficient as it is best suited for continuous data. This method is most often used to assess the strength between variables that are assumed to be normally distributed, however it is suitable for non-normal data [5].

In the multivariate analysis of the numerical variables, a strong positive correlation of 0.76 is evident between energy and loudness, as illustrated in figure 2. We find that significant correlation of 0.48 exists between valence and danceability. Regarding negative correlations, there is a pronounced inverse relationship of -0.73 between energy and acousticness. This is followed by weaker inverse correlations, such as those between loudness and acousticness (-0.59) and between loudness and instrumentalness (-0.43). We will further discuss the handling of correlated data in Section 5.

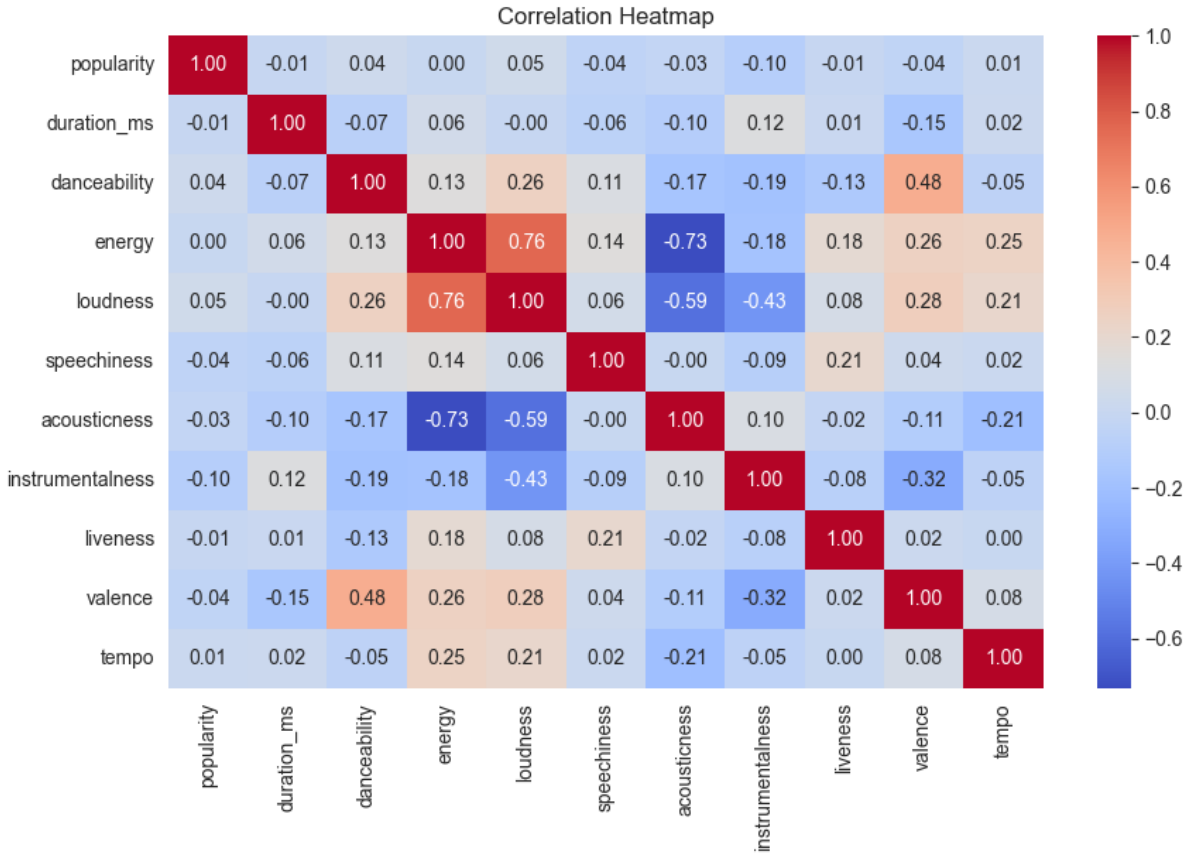


Figure 2: Correlation Heatmap between numeric variables.

Please find the pairwise plotted variables in Appendix 5.1 for detailed inspection of pairwise relationships.

3.3 Outlier Detection

To identify outliers for each numerical variable, we considered any values that deviated by more than 3 times the inter-quartile range (IQR) from quartile 1 and quartile 3. The variable that stands out for having the highest total number of outliers is instrumentalness, with 23,354 outliers exceeding the established boundary. This is followed by speechiness (7643), liveness (3410), loudness (1872) and duration_ms (1223). On the contrary, popularity, danceability, energy, acousticness, valence and tempo do not present any severe outliers.

Upon closer examination, as shown in Figure 6, it can be observed that although the duration variable does not have the highest number of outliers, it does exhibit the most extreme outliers.

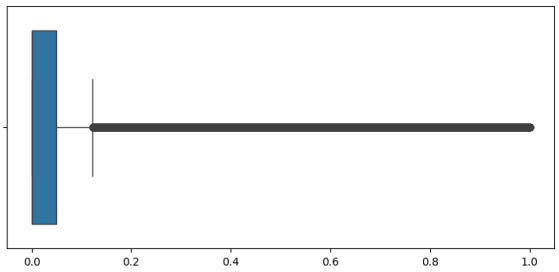


Figure 3: Instrumentalness

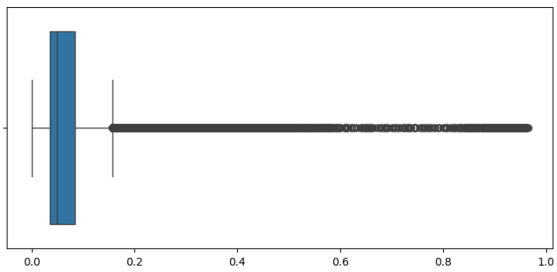


Figure 4: Speechiness

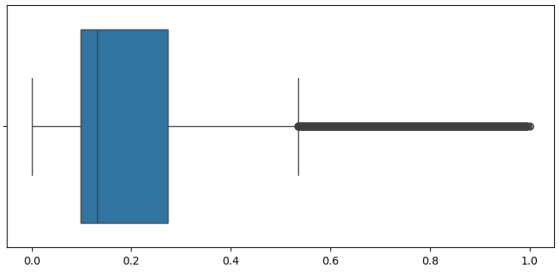


Figure 5: Liveness

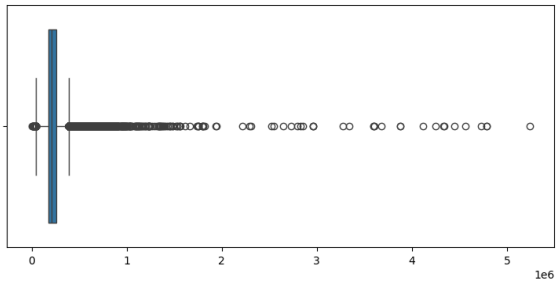


Figure 6: Duration in ms

Please find boxplots for all variables in Appendix 5.1.

To detect multivariate outliers, we divided the dataframe by musical genre. Within each segmentated group, we then searched for values that significantly diverged from the average across multiple variables. We utilized the Mahalanobis distance as the measurement criterion, which is a measure of the distance between a point x and the mean μ of the distribution D and it is calculated in the following way:

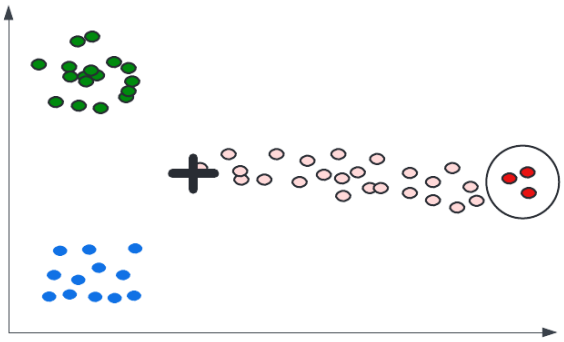


Figure 7: Illustrative example of outlier detection method

$$D_M(\mathbf{x}, \mu) = \sqrt{(\mathbf{x} - \mu)^T \mathbf{S}^{-1} (\mathbf{x} - \mu)} \tag{2}$$

The threshold used to identify outliers was set at the 99th percentile of the chi-squared (χ^2) distribution, every observation in the remaining 1% will been classified as a multivariate outlier.

The segmentation of the dataframe was implemented to account for the substantial differences in song attributes across different genres. Without such segmentation, genres with notably extreme traits might have resulted in a disproportionately high number of outliers, which could have introduced a bias in our data. By organizing the data in this manner, we significantly enhance our ability to accurately identify truly anomalous observations, while keeping into consideration the context specific to each musical genre. True anomalies lie in their difference to the genre they belong to rather than the difference to the mean values of the entire dataset.

For an illustrative example please note Figure 7. If we are to measure outliers using the entire data, we would possibly mark songs as outliers (circled points), although they follow the distribution of the genre specific to that song. Whereas our interest lies in understanding the clustering properties in relation to genres. This calculus led to the identification of 15,331 outliers, with an average of 134 outliers per genre. The methods for addressing there outliers will be further discussed and decided upon in Section 4.2.

3.4 Principal Component Analysis

Principal Component Analysis (PCA) is a statistical technique primarly applied for reducing the dimensionality of a dataset while preserving as much of the data's variance as possible. Reducing the dimensionality is essential as an high-dimensional dataset can often lead to several problems such as overfitting or waste of resouces. The process involves identifying the directions, called principal components, along which the variance of the data is maximized. These components are orthogonal to each other, each representing a linear combination of the original variables.

To perform PCA, initially the data is standardized (more in Section 4) to have zero mean and unit variance, so that the analysis will not be influenced by different measure-

ment scales. Subsequently, the covariance matrix is calculated and it is decomposed into its eigenvalues and eigenvectors. Finally, by ranking eigenvectors in order of their eigenvalues, we obtain a table with the principal components in order of how much variance in the data is explained by them.

The outcomes of this analysis will be presented in this subsection. The first ten components account for 97% of the variance, and details of each of the first ten components can be found in Figure 8.

From Table 3, it is evident that the first and second principal components account for 40% of the data variance. When the third component is included, this figure reaches 53%. This distribution of explained variance suggests that dimension reduction will not be a trivial problem in this dataset. The relatively gradual decline in the variance explained by successive components indicates a complex interplay of features.

Principal Component	Explained Variance Ratio
PC1	0.2614
PC2	0.1386
PC3	0.1125
PC4	0.0954
PC5	0.0874
PC6	0.0785
PC7	0.0756
PC8	0.0670
PC9	0.0412
PC10	0.0296

Table 3: Explained variance ratios for the principal components

Concentrating the first two principal components and their representation on a x-y plane shown in Figure 9, it is evident that PC1 has a strong negative association with the variables 'loudness' and 'speechiness', while showing a positive correlation with 'instrumentalness'. This component could be measuring how quiet a music track is and it could correspond to genres or styles of music that are typically instrumental, classical or ambient music.

As for PC2, it demonstrates a significant positive relationship with 'danceability' and 'liveness', at the same time it exhibits strong negative correlations with both 'tempo' and 'energy'. This component could be measuring how engaging a song is in terms of liveliness while being slower and more relaxed in its nature

This finding is partially surprising, as we might have expected pairs of features such as danceability and tempo, or danceability and energy to have a strong correlation, at least in the most important components. However, PC2 significantly challenges this assumption.

For a 2D plot of instances projected on PC1 and PC2 please refer to Appendix 5.1.

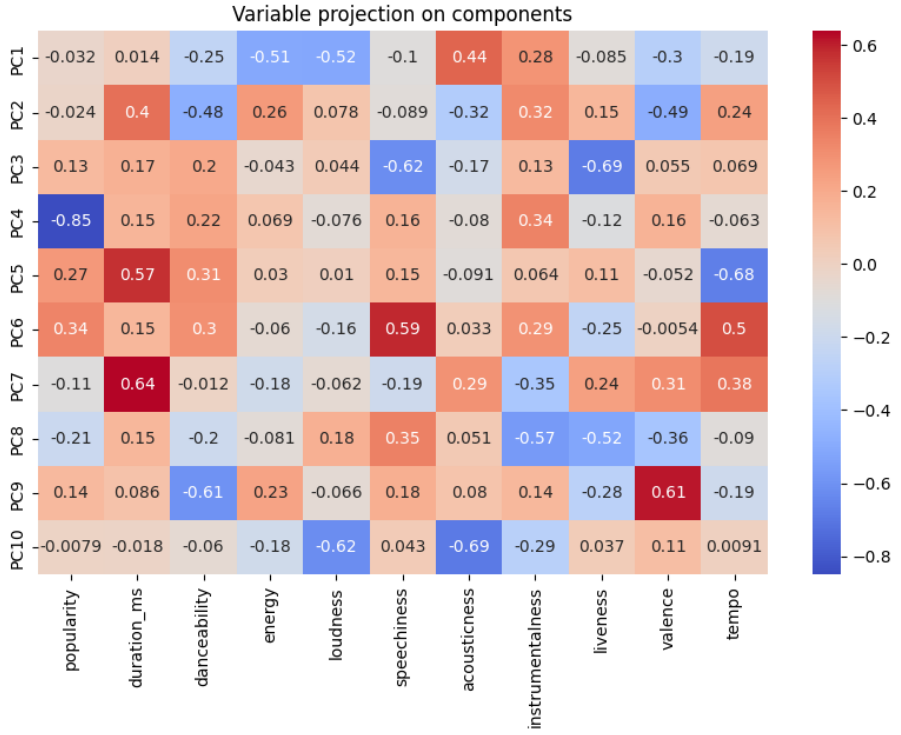


Figure 8: Top 10 Principal Components and their associations with variables

Further analyses on the relationship between the first *three* principals were conducted, but we decided to abandoned this in the interest of time and focused research. Please find the 3D plots for these analyses in Appendix 5.1.

4 Preprocessing

In this section we detail and justify the methods we used to preprocess our data for future machine learning applications. The preprocessing is executed to a *python* script, wherein the *Preprocess* class is defined with methods to execute all the steps. Please refer to the script and the *README.md* file for information on how to run.

4.1 Scaling

Scaling data in machine learning is important because it ensures that all features contribute equally to the learning process. When features have different scales, those with larger scales can dominate the learning process, introducing bias to the models. Scaling allows for a fair comparison between features, stopping any single feature from overpowering others simply because of its scale. Scaling can also help algorithms converge faster during training, as it reduces the variance in features. This ultimately improves the model's performance and stability, leading to more accurate predictions on unseen data.

From the EDA, we could see that a lot of the numerical variables in our dataset where on vastly different scales. Features such as duration in miliseconds were in the range of a

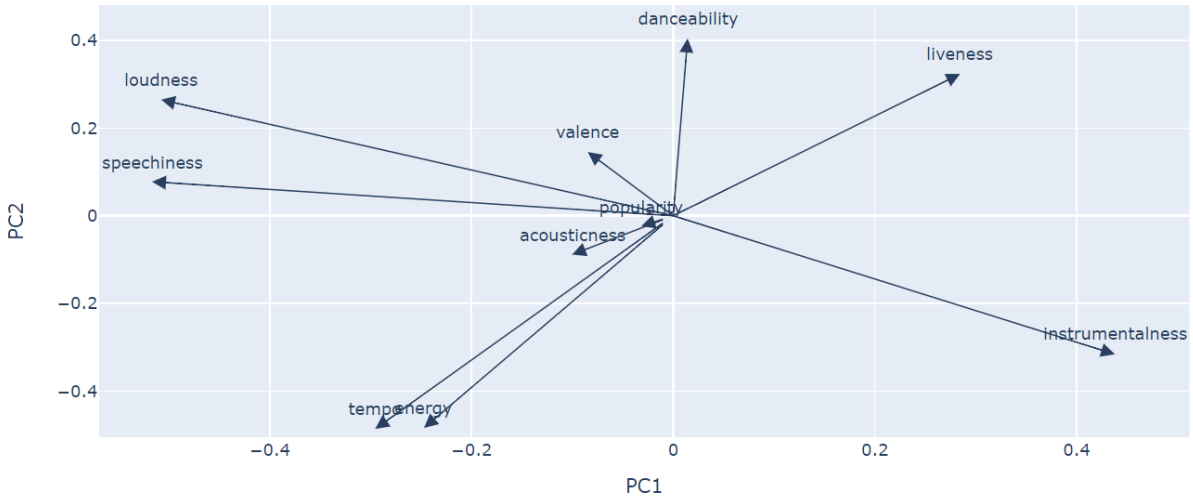


Figure 9: Variable Projections on PC1 and PC2

couple thousands to millions, while others like energy where in the range of 0 to 1. The scaling in our case would prevent features like duration from having more of an impact in machine learning models than energy just due to their size. To scale the data we employed the most common standardization technique, whereby we appply the following function to each of our features,

$$z = \frac{x - \mu}{\sigma}$$

Using the z-score normalization we transfarm all features such that their mean is 0 and their standard deviation is 1, while simultaneously preserving the relative distance between data points within a feature. To apply this function to each numeric column of our dataframe with used the *Standard Scaler* function from the *Scikit-learn* library.

4.2 Outlier Handling

Outliers can significantly affect machine learning models by skewing the training process and distorting the learned patterns. Their can influence the estimation of statistical parameters, which introduces bias to models. In our case specifically, outliers can also influence decision boundaries, leading to suboptimal performance in classification tasks. Models trained on data with outliers may not generalize well to unseen data, as they might tend to overfit to the noise introduced by these outliers. Thus it is crucial to identify and appropriately handle outliers during the data preprocessing stage to ensure the robustness and generalization ability of machine learning models.

For this project we have decided to univariate and multivariate outliers differently. To preserve the integrity of the data, we have chosen to impute univariate outliers by imputing them. We believe this to be the best choice, as there may be valuable information in the data instance that can help modelling, only stained by a single value that can

introduce some bias. Imputing the data means we additionally preserve the distribution, and are able to retain these rows to train the model. On the other hand, multivariate outliers are wholly removed from the dataset. Due to their nature, such instances can significantly distort the relationships between variables and affect model performance, which for classification tasks can heavily affected decision boundaries. Imputation of all features for by replacing outlier values with imputed values might introduce artificial patterns or correlations that do not truly exist in the data. In contrast to univariate outliers, the effect of imputation will not be as strong.

4.2.1 Univariate Outliers

To handle univariate outliers we first detect them by using the interquartile range method, which is robust to different distributions of data. It identifies outliers based on the variability of the middle 50% of data which makes it less sensitive to extreme values in the tails of the distribution. Once we detect the instances that are further than 3 times the IQR (Q3 - Q1) below Quartile 1 and above Quartile 3 are marked and replaced by *null* values.

To impute these values there are several methods. The one selected is K-nn imputation. K-nn according to research by Jadhav et al. [6] has shown the be the best performing imputation method for numeric datasets, evaluated by the root mean squared error for regression tasks. Zainuri et al. [7] reached similar conclusions determining K-nn to be one of the best imputation methods for air quality data. Their research alongside an earlier study by Batista and Monard in 2003 [8] serve as validation of the performance of K-nn imputation.

K-nn imputation is a method used to fill in missing values in a dataset by using the values of neighboring data points. It calculates the distance between the data point containing *null* values and its nearest neighbors with complete information. It assigns a value to the missing data point based on the values of its nearest neighbors. In our case, we have selected the mean value of the 5 closest neighbours. This method leverages similarities between data points to estimate missing values, making it useful for datasets with complex relationships between variables.

4.2.2 Multivariate Outliers

To detect multivariate outliers we have chosen to go with the Mahalanobis distance metric. The Mahalanobis distance can be used for multivariate outlier detection, and is a popular tool to do so [9] because it accounts for correlations between variables while adjusting for the scale of each variable. This makes it robust to non-normal data as shown by previous research [10], and is suitable for datasets with complex relationships between variables. By considering the covariance structure of the data the Mahalanobis distance can identify outliers based on their deviation from the typical patterns observed in the dataset.

In the *preprocessing* python script this method is implemented by means of the *Sci-kit learn* library which provides functions to compute the Mahalanobis distance. Using the inverse of a χ^2 cumulative distribution function, with a significance level of 0.99, we find and delete all the instances from the dataset that have a Mahalanobis distance above this threshold.

5 Discussion and Conclusion

The research conducted in this report have shown a more nuanced perspective of the data used. The univariate analyses showing most features are on similar scales with some exceptions, such as duration and popularity. The multivariate analyses have shown some significant relationships between *valence* and *danceability*, *energy* and *acousticness*. Through PCA we have discovered that most of the variance (information) is held in the speechiness, loudness, and instrumentalness in PC1, followed by tempo, energy and danceability in PC2. We find that PC1 could be measuring how quiet a music track is and it could correspond to genres or styles of music that are typically instrumental, classical or ambient music. While PC2 could be measuring how engaging a song is in terms of its liveliness and danceability while having a slow tempo and a relaxed energy. PCA has allowed us to understand that the main variability, or information of a song comes from its overall volume, and its liveliness.

Using this information we have conducted an appropriate preprocessing of the data. Wherein the data is scaled, univariate outliers are imputed by K-nn, and multivariate outliers are removed. In Section 3 we found variables that show some significant correlation. Consciously, we decide to retain these variables and not outright remove them. The decision to leave the removal for the next step in the larger project is to give the opportunity to test models using different variables and find the best subset, suited to the model chosen.

In summary, our research has provided us with a strong foundation and useful, clean data for any subsequent analyses following this first part of the project.

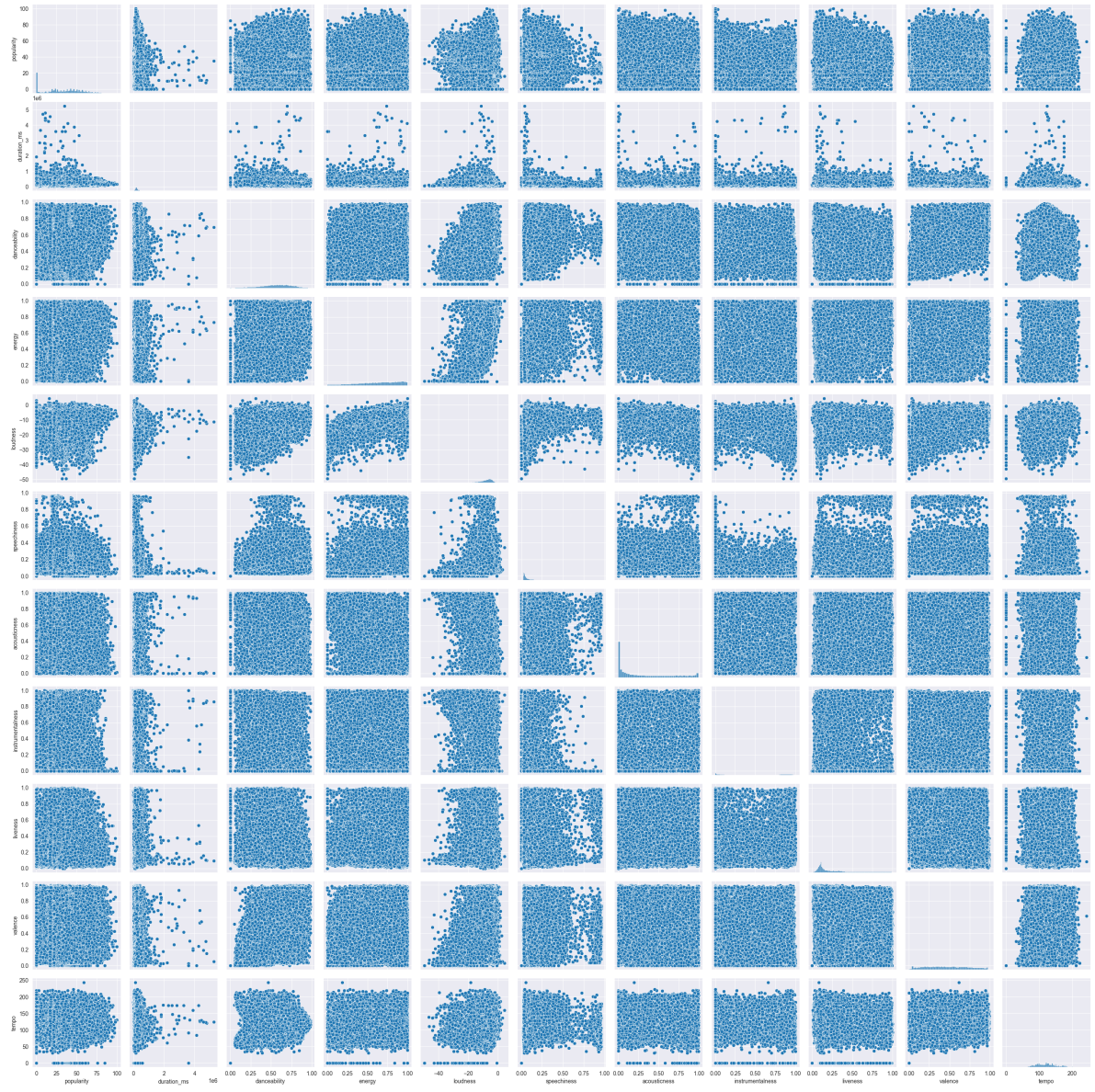
5.1 Future Work

This report details the exploratory analysis and the preprocessing of a song/track dataset, with the goal of providing suitable data for any machine learning model to exploit. The next step for our project will be to use the information discovered here to develop a clustering model, understand the relationships between clusters and genres, and investigate how recommender systems, such as Spotify's personalised music mixes work.

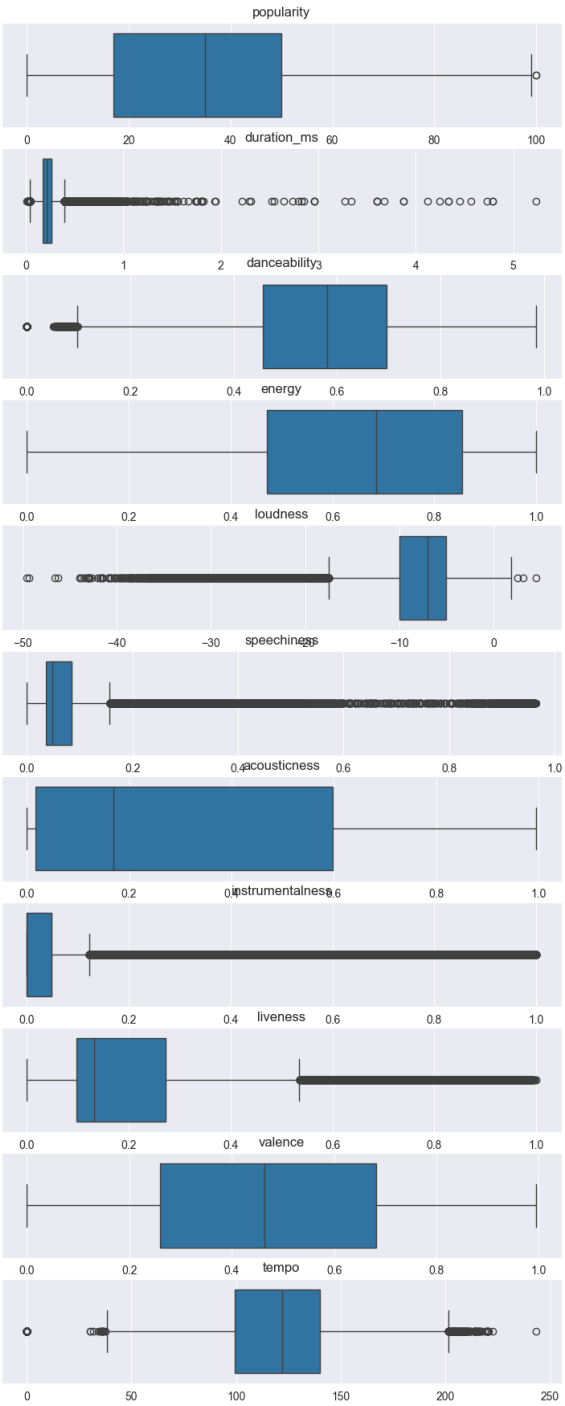
We have conducted this report with the goal of clustering in mind, but have generalised our preprocessing to the extent that any type of analytic learning task can benefit from.

Appendix

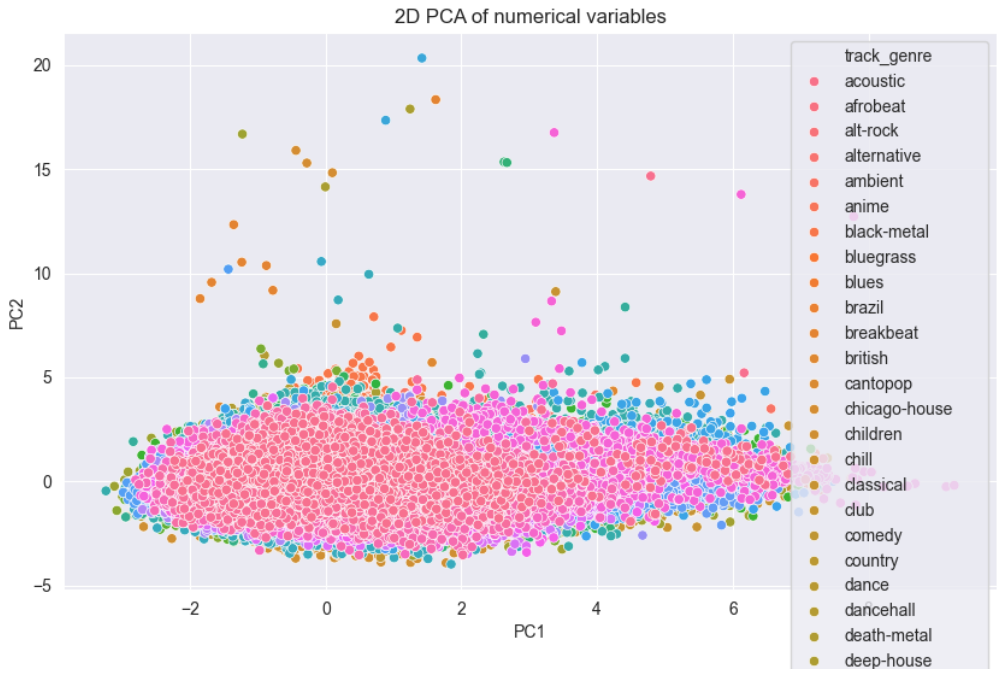
A. Pair-wise variable scatter plots



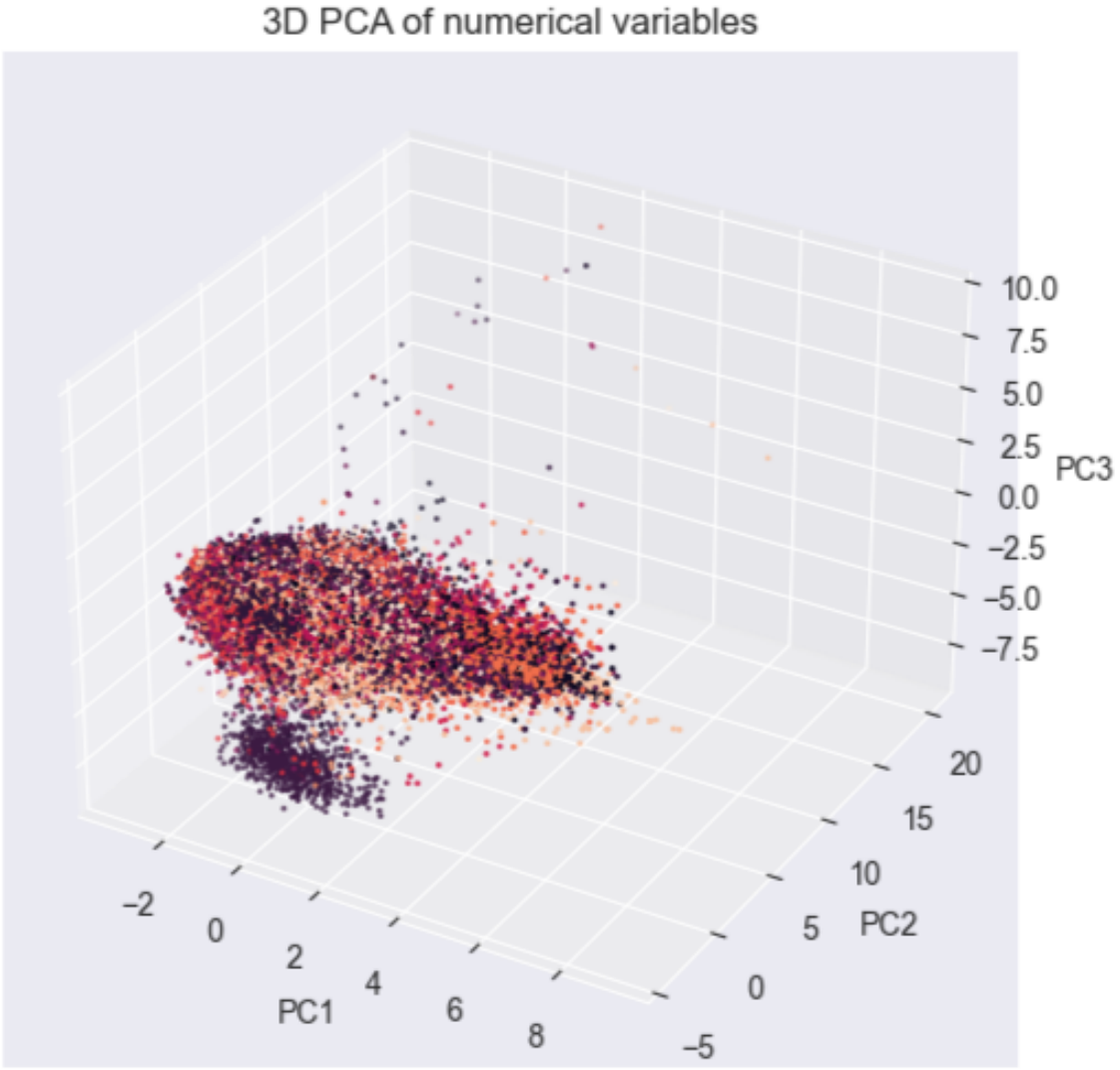
B. Variable Boxplots

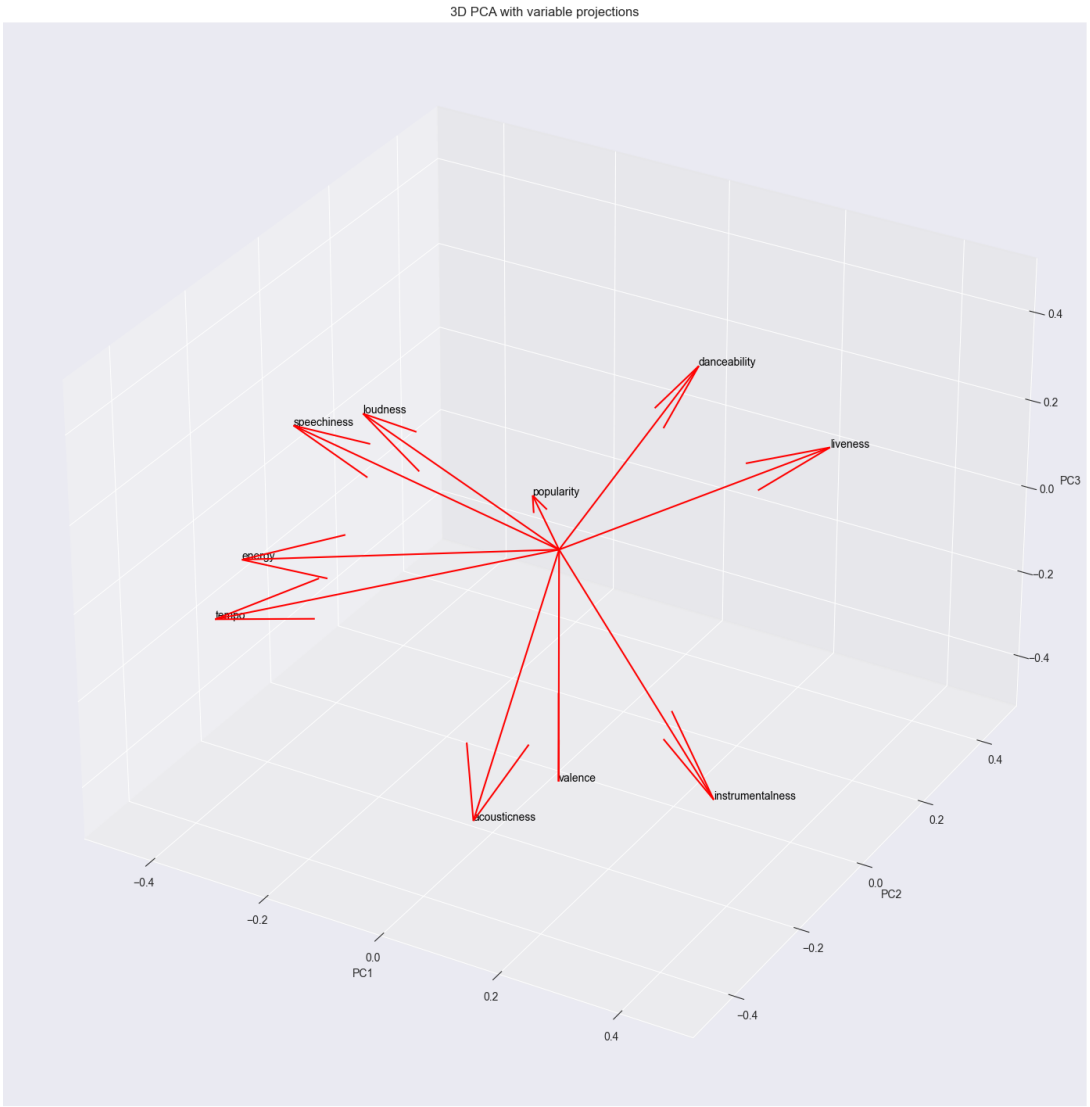


C. 2D PCA plot (PC1, PC2)



D. 3D PCA plots (PC1, PC2, PC3)




3D PCA with variable projections
danceability
loudness
speechiness
liveness
popularity
energy
tempo
valence
instrumentalness
acousticness
PC1
PC2
PC3
−0.4
−0.2
0.0
0.2
0.4

References

[1] M. Pandya, "Spotify tracks dataset," 2022. [Online]. Available: https://www.kaggle.com/dsv/4372070

[2] P. Pareek, P. Shankar, M. P. Pathak, and M. N. Sakariya, "Predicting music popularity using machine learning algorithm and music metrics available in spotify," *J. Dev. Econ. Manag. Res. Stud. JDMS*, vol. 9, pp. 10–19, 2022.

[3] K. Luo, "Machine learning approach for genre prediction on spotify top ranking songs," 2018.

[4] N. Tran, T. Nguyen, and D. Dinh, *Personalization of Music Streaming Services Using K-means—A Business User-Oriented Model for Spotify Soundtracks Clustering*, 12 2023, pp. 369–378.

[5] N. S. Chok, "Pearson's versus spearman's and kendall's correlation coefficients for continuous data," September 2010. [Online]. Available: http://d-scholarship.pitt.edu/8056/

[6] D. P. Anil Jadhav and K. Ramanathan, "Comparison of performance of data imputation methods for numeric dataset," *Applied Artificial Intelligence*, vol. 33, no. 10, pp. 913–933, 2019. [Online]. Available: https://doi.org/10.1080/08839514.2019.1637138

[7] N. A. Zainuri, A. A. Jemain, and N. Muda, "A comparison of various imputation methods for missing values in air quality data," *Sains Malaysiana*, vol. 44, no. 3, pp. 449–456, 2015.

[8] G. E. Batista and M. C. Monard, "An analysis of four missing data treatment methods for supervised learning," *Applied artificial intelligence*, vol. 17, no. 5-6, pp. 519–533, 2003.

[9] H. Ghorbani, "Mahalanobis distance and its application for detecting multivariate outliers," *Facta Universitatis Series Mathematics and Informatics*, vol. 34, p. 583, 10 2019.

[10] R. Warren, R. F. Smith, and A. K. Cybenko, "Use of mahalanobis distance for detecting outliers and outlier clusters in markedly non-normal data: A vehicular traffic example," 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:127938294