Music Recommendations

Project Objective

The objective of this project is to develop a music recommendation system that predicts the perceived appreciation of music based on Twitter analyses and user feedback. This will involve leveraging contextual data from the nowplaying-RS dataset, which contains 11.6 million music listening events from 139K users and 346K tracks collected from Twitter. Additionally, sentiment values associated with hashtags will be analyzed to understand the emotional context.

Data Sources

nowplaying-RS Dataset

Contains rich context and content features of listening events, including timestamps, user location, tweet language, and sentiment analysis of hashtags. It provides insights into user preferences and cultural origins based on listening behavior.

Spotify Tracks Dataset

Includes audio features and genre information for Spotify tracks, enabling genre-based analysis and recommendation system development.

Project Scope

Data Exploration and Cleaning

- Understand the structure of the nowplaying-RS dataset.
- Address missing values, outliers, and inconsistencies in the data.
- Analyze sentiment values associated with hashtags to extract emotional context.

Data Analysis and Visualization

- Perform exploratory data analysis (EDA) to uncover patterns and relationships within the datasets.
- Visualize data using graphs and charts to gain insights into user preferences and music trends.

Feature Engineering and Modeling

- Extract relevant features from the datasets for building predictive models.
- Develop machine learning models to predict music appreciation based on user context and content features.

Final Report

- Compile final project report integrating previous renderings, including conclusions and future directions.
- Clean and comment code on GitHub, refine model and modeling pipeline if time permits.

Data Exploration and Cleaning

In the first step of this project to gain a comprehensive understanding of the datasets, we will examine the following tables to describe their structures.

The **“sentiment_values.csv”** dataset provides sentiment analysis results for hashtags encountered in the listening events. It includes sentiment scores from various lexicons (Vader, AFINN, Opinion Lexicon, SentiStrength) such as minimum, maximum, sum, and average scores. The hashtag column links sentiment scores to specific hashtags associated with the listening events:

Name of the Column	**Variable's Type**	**Description**	**Categorical / Quantitative**
hashtag	Float	Hashtag associated with the sentiment information	Quantitative
vader_min	Float	Minimum sentiment score using Vader lexicon	Quantitative
vader_max	Float	Maximum sentiment score using Vader lexicon	Quantitative
vader_sum	Float	Sum of sentiment scores using Vader lexicon	Quantitative
vader_avg	Float	Average sentiment score using Vader lexicon	Quantitative
afinn_min	Float	Minimum sentiment score using AFINN lexicon	Quantitative

afinn_max	Float	Maximum sentiment score using AFINN lexicon	Quantitative
afinn_sum	Float	Sum of sentiment scores using AFINN lexicon	Quantitative
afinn_avg	Float	Average sentiment score using AFINN lexicon	Quantitative
ol_min	Float	Minimum sentiment score using Opinion Lexicon	Quantitative
ol_max	Float	Maximum sentiment score using Opinion Lexicon	Quantitative
ol_sum	Float	Sum of sentiment scores using Opinion Lexicon	Quantitative
ol_avg	Float	Average sentiment score using Opinion Lexicon	Quantitative
ss_min	Float	Minimum sentiment score using SentiStrength Lexicon	Quantitative
ss_max	Float	Maximum sentiment score using SentiStrength Lexicon	Quantitative
ss_sum	Float	Sum of sentiment scores using SentiStrength Lexicon	Quantitative
ss_avg	Float	Average sentiment score using SentiStrength Lexicon	Quantitative

Table 1. variables of 'sentiment_values' dataset

The **"user_track_hashtag_timestamp.csv"** dataset captures associations between users, tracks, hashtags, and timestamps of listening events. Key columns include user_id, track_id, hashtag, and created_at. This dataset enables us to explore user engagement and interactions with music content through hashtags on social media:

Name of the Column	Variable's Type	Description	Categorical / Quantitative
user_id	Object	Identifier for the user	Unique Values
track_id	Object	Identifier for the track	Unique Values

hashtag	Object	Hashtag associated with the listening event	Categorical
created_at	Object	Timestamp of the listening event	N/A

Table 2. variables of 'user_track_hashtag' dataset

The **"context_content_features.csv"** dataset contains contextual and content features related to music listening events collected from Twitter. It includes information such as coordinates, instrumentalness, liveness, speechiness, danceability, valence, loudness, tempo, acousticness, energy, mode, key, artist_id, place, geo, tweet_lang, track_id, created_at, lang, and time_zone. These variables encompass a range of audio features and contextual information associated with each listening event:

Name of the Column	**Variable's Type**	**Description**	**Categorical / Quantitative**
coordinates	Object	N/A	Unique Values
instrumentalness	Float	Indicating the absence of vocals in a track	Quantitative
liveness	Float	Indicating the presence of an audience in a track	Quantitative
speechiness	Float	Indicating the presence of spoken words in a track	Quantitative
danceability	Float	Describing the suitability of a track for dancing	Quantitative
valence	Float	Indicating the musical positiveness	Quantitative
loudness	Float	The overall loudness of a track in decibel	Quantitative
tempo	Float	Beat per minute representation of the overall estimated tempo of a track	Quantitative
acousticness	Float	Describing whether the track is acoustic or not	Quantitative

energy	Float	Describing the intensity level of a track	Quantitative
mode	Float	Indicating the main key of a track with 1.0 and 0.0	Categorical
key	Float	Representing the group of notes by using standard Pitch class notation	Categorical
artist_id	Object	Identifier for the artist	Unique Values
place	Object	N/A	
geo	Object	N/A	
tweet_lang	Object	Indicating in which language the tweet was written.	Categorical
track_id	Object	Identifier for the track	Unique Values
created_at	Object	N/A	
lang	Object	Hashtag Language (?)	
time_zone	Object	representing the time zone of the area where the listening event occurred.	
user_id	Float	Identifier for the user	Unique Values
id	Int	N/A	Unique Values

Table 3. variables of 'context_content_features' dataset

Data Cleaning

We begin by cleaning the dataset **"sentiment_values.csv"** using the following steps:

Renaming misplaced columns: As can be seen in Fig. 1, the hashtag column is misplaced and it should be in the first column, then we have 4 columns that are not explicitly specified. We rename them to the sentiment scores. This renaming step enhances the interpretability of the dataset, providing clearer and more meaningful column names that align with the specific sentiment analysis metrics used.

Dropping Unnecessary Columns: Furthermore, the columns 'vader_min', 'vader_max', 'vader_sum', 'afinn_min', 'afinn_max', 'afinn_sum', 'ol_min', 'ol_max', 'ol_sum', 'ss_min', 'ss_max', and 'ss_sum' should be dropped because they do not provide valuable information

for the analysis and modeling process. These columns represent minimum, maximum, and sum values for sentiment scores ('vader', 'afinn', 'ol', and 'ss'), but since their values are already available in separate columns ('vader_score', 'afinn_score', 'ol_score', and 'ss_score'), including them would introduce redundancy and increase the complexity of the dataset without adding meaningful insights. Therefore, removing these redundant columns helps streamline the dataset and focus on the relevant sentiment score metrics for further analysis.

Addressing missing values: In the subsequent step, we address missing values within the sentiment scores (vader_score, afinn_score, ol_score, ss_score) by imputing them with average values where available. This approach ensures that our data remains representative and complete. The remaining unnecessary score columns (vader_avg, afinn_avg, ol_avg, ss_avg) are then dropped, as they do not offer additional valuable information for our analysis. Additionally, any rows containing missing values are dropped entirely to maintain data integrity and consistency.

Correlation matrix: Upon visualizing the data using a correlation matrix (Fig. 2), we observed strong correlations among certain sentiment scores ('vader_score', 'afinn_score', 'ss_score'), with correlation coefficients exceeding 80%. This high correlation indicated redundancy in the information captured by these scores. To address this, we made the decision to drop these three highly correlated scores, retaining only the 'ol_score', which we then renamed to 'sentiment_score' for clarity and focus.

				hashtag	vader_min	vader_max	vader_sum	vader_avg	afinn_min	afinn_max	afinn_sum	afinn_avg	ol_min	ol_max	ol_sum	ol_avg	ss_min	ss_max	ss_sum	ss_avg
relaxtime	0.8	0.8	2.4	0.8	NaN	NaN	NaN	NaN	0.7375	0.7375	0.7375	0.7375	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
melovechilicheese	0.8	0.8	0.8	0.8	NaN	NaN	NaN	NaN	0.9000	0.9000	0.9000	0.9000	1.0	1.0	1.0	1.0	0.8	0.8	0.8	0.8
greatmusic	0.8	0.8	2.4	0.8	1.0	1.0	1.0	1.0	0.8875	0.8875	0.8875	0.8875	1.0	1.0	1.0	1.0	0.8	0.8	0.8	0.8
rockballad	0.7	0.7	0.7	0.7	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
amonamarth	0.3	0.3	0.3	0.3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	0.0	0.0	0.0	NaN	NaN	NaN	NaN

Figure 1. First 5 rows of 'sentiment_values.' dataset showing the misplaced columns

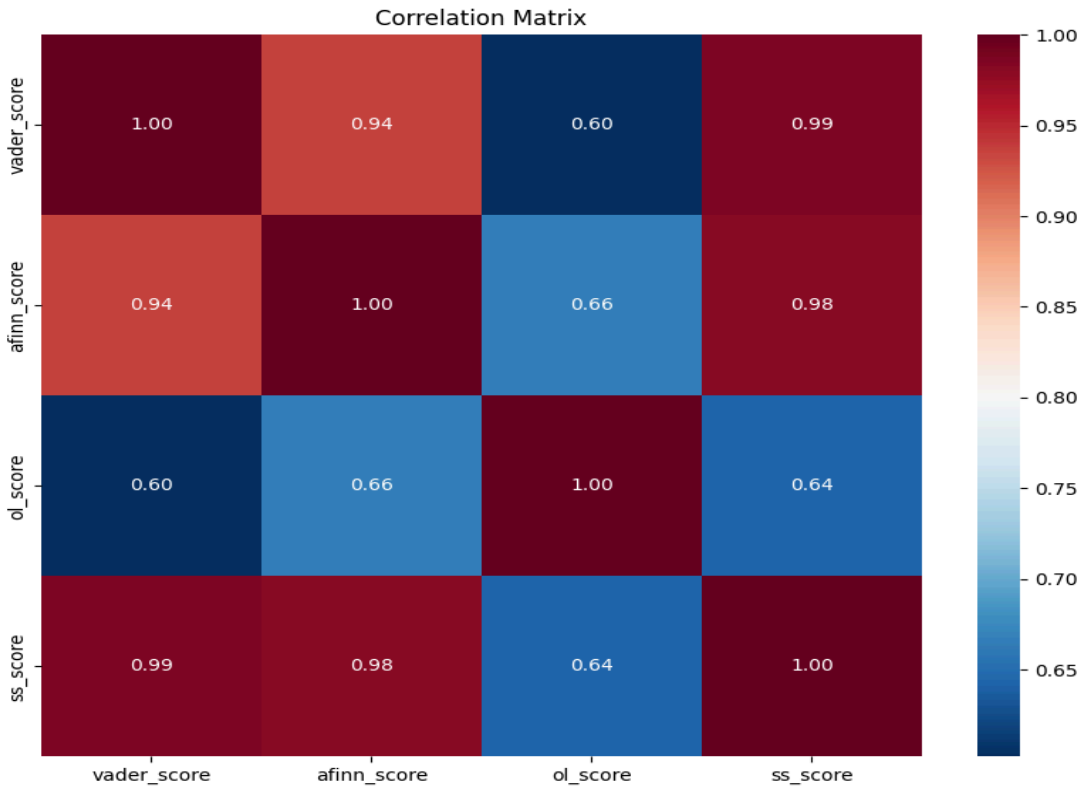

Figure 2. Correlation matrix between the columns of 'sentiment_values' dataset including vader_score, afinn score, ol_score and ss_score

By executing these data preprocessing steps, we ensured that our dataset was cleaned, consolidated, and optimized for further analysis, with a specific emphasis on retaining meaningful sentiment information while eliminating redundancy.

Next, we detail the preprocessing steps applied to the **"user_track_hashtag_timestamp.csv"** dataset. This preprocessing aims to ensure data cleanliness, integrity, and relevance for subsequent analyses. We followed the steps below for cleaning this dataset:

Loading the Dataset: The dataset was loaded into a Pandas DataFrame (df2) from the provided file path. This initial step allowed us to access and manipulate the dataset using Python's data manipulation tools.

Handling Null Values: Null values in the 'hashtag' column were identified and removed using the dropna() function. This step was crucial for maintaining data integrity and ensuring that subsequent analyses were not compromised by missing values.

Filtering Tracks by Usage Count: To focus on tracks with significant user interactions, tracks that were played less than 50 times were filtered out. This filtering process helped reduce noise in the dataset and prioritize tracks with higher usage counts for further analysis.

Merging with Cleaned Sentiment Dataset: The cleaned dataset was merged with sentiment scores obtained from a separate dataset (df1). This merge was based on the

common 'hashtag' column using an inner join, ensuring that only hashtags present in both datasets were retained. The merged dataset (df_sentiment) combined information about user interactions with tracks and associated sentiment scores.

Confirming Dataset Integrity: Several checks were performed to validate the integrity of the merged dataset. These checks included verifying the absence of null values, confirming the dataset's dimensions, and examining the distribution of hashtags. These steps ensured that the merged dataset was complete, consistent, and ready for further analysis.

The last dataset is **"context_content_features.csv"**. The following steps have been taken to clean this dataset:

Loading Necessary Columns: We load the dataset and limit it to the necessary columns to reduce memory usage and computational overhead. This step ensures that only relevant data is loaded into memory, improving processing efficiency.

Removing Tracks with Fewer Plays: Tracks that were played fewer than 50 times are removed. This filtering helps focus the analysis on more popular tracks, which are likely to provide more meaningful insights and trends. Additionally, removing infrequently played tracks can help reduce noise in the data.

Dropping Unnecessary Columns & Removing Null Values: Unnecessary columns such as coordinates, id, place, and geo and also all null values are dropped. These columns do not provide valuable information and we use 'time_zone' instead of 'coordinates', 'place', and 'geo', and 'user_id' instead of 'id'.

Filtering English Language Entries: The dataset is filtered to include only English language entries. This step ensures consistency in language for analysis and modeling purposes, as mixing multiple languages could introduce complexity and potentially skew results.

Merging with Sentiment Data: The dataset is merged with df_sentiment based on specified columns (track_id, created_at, user_id). This merge combines sentiment information with the context content features, enriching the dataset with additional insights about user sentiment towards tracks.

Converting and Dropping Columns: Certain columns like hashtag and user_id are converted to string type for consistency and ease of handling. Additional unnecessary columns are dropped post-merge to streamline the dataset and remove redundant information.

Filtering USA Time Zones: The dataset is filtered to include only USA time zones and their names are simplified. This step focuses the analysis on a specific geographical region, providing more targeted insights relevant to a particular audience or market segment.

Creating Binary Sentiment Column: A binary sentiment column is created based on the sentiment_score. This simplifies sentiment analysis by categorizing sentiment as either

positive or negative, making it easier to interpret and incorporate into further analysis or modeling tasks.

Reordering Columns: Columns are reordered to create the MVP dataset, arranging them in a logical sequence for better readability and usability. This step helps organize the data for easier analysis and model building.

One-Hot Encoding Time Zone: The time_zone column is one-hot encoded to convert categorical data into numerical format, which is required by many machine learning algorithms. This transformation preserves the ordinal relationship between different time zones while preventing the model from interpreting them as continuous variables. The original time_zone column is then dropped to avoid multicollinearity issues.

Spotify Tracks Dataset:

Name of the Column	Variable's Type	Description	Categorical / Quantitative
track_id	Object	Identifier for the track	Unique Values
artists	Object	The names of the artists	Categorical
album_name	Object	The name of the album in which the track appears	Categorical
track_name	Integer	The name of the track	Quantitative
popularity	Integer	The popularity of the track on a scale from 0 to 100, based on the total number of plays and recency	Quantitative
duration_ms	Integer	The length of the track in milliseconds	
explicit	Boolean	Indicates whether the track has explicit lyrics	Categorical
danceability	Float	Describing the suitability of a track for dancing	Quantitative
energy	Float	Describing the intensity level of a track	Quantitative
key	Integer	Indicating the main key of a track with 1.0 and 0.0	Quantitative
loudness	Float	The overall loudness of a track in decibel.	Quantitative
mode	Integer	Indicating the main key of a track with 1.0 and 0.0	Quantitative
speechiness	Float	Indicating the presence of spoken words in a track .	Quantitative
acousticness	Float	Describing whether	Quantitative

		the track is acoustic or not	
instrumentalness	Float	Indicating the absence of vocals in a track	Quantitative
liveness	Float	Indicating the presence of an audience on a track	Quantitative
valence	Float	Indicating the musical positiveness	Quantitative
tempo	Float	Beat per minute representation of the overall estimated tempo of a track	Quantitative
time_signature	Integer	An estimated time signature, indicating how many beats are in each bar	Quantitative
track_genre	Object	The genre in which the track belongs	Categorical

Table 4. Variables of 'spotify_tracks' Dataset

Unlike the previous datasets, **"Spotify Tracks Dataset"** doesn't require a very detailed cleaning process. Only two steps are pursued to reach our final dataset.

Loading the Dataset: Upon loading the dataset as 'Spotify', we see that it includes 114000 rows and 20 columns.

Handling Duplicates and Null Values: The result of the duplicated method showed us that 450 duplicates exist in the dataset. After dropping the duplicates, null values were searched and only one row of 3 missing values was found and the dataset was reduced to 113549 rows and 20 columns.

These preprocessing steps collectively prepare the datasets for analysis or modeling tasks, ensuring data quality, consistency, and relevance for deriving meaningful insights or building predictive models.

Data Analysis and Visualization

nowplaying-RS Dataset

The purpose of this Exploratory Data Analysis (EDA) is to understand the underlying patterns and relationships within the nowplaying-RS dataset. We aim to explore the data,

identify and address issues such as multicollinearity and data imbalance, and prepare the dataset for further predictive modeling.

We start by examining the distribution of the target variable sentiment to understand its balance and the distribution of other features to understand their spread and central tendency. As Fig.3 shows the data is highly imbalanced, with a significant majority of positive sentiments (1). This imbalance can affect model training, leading to biased predictions. Therefore, balancing the dataset is necessary.

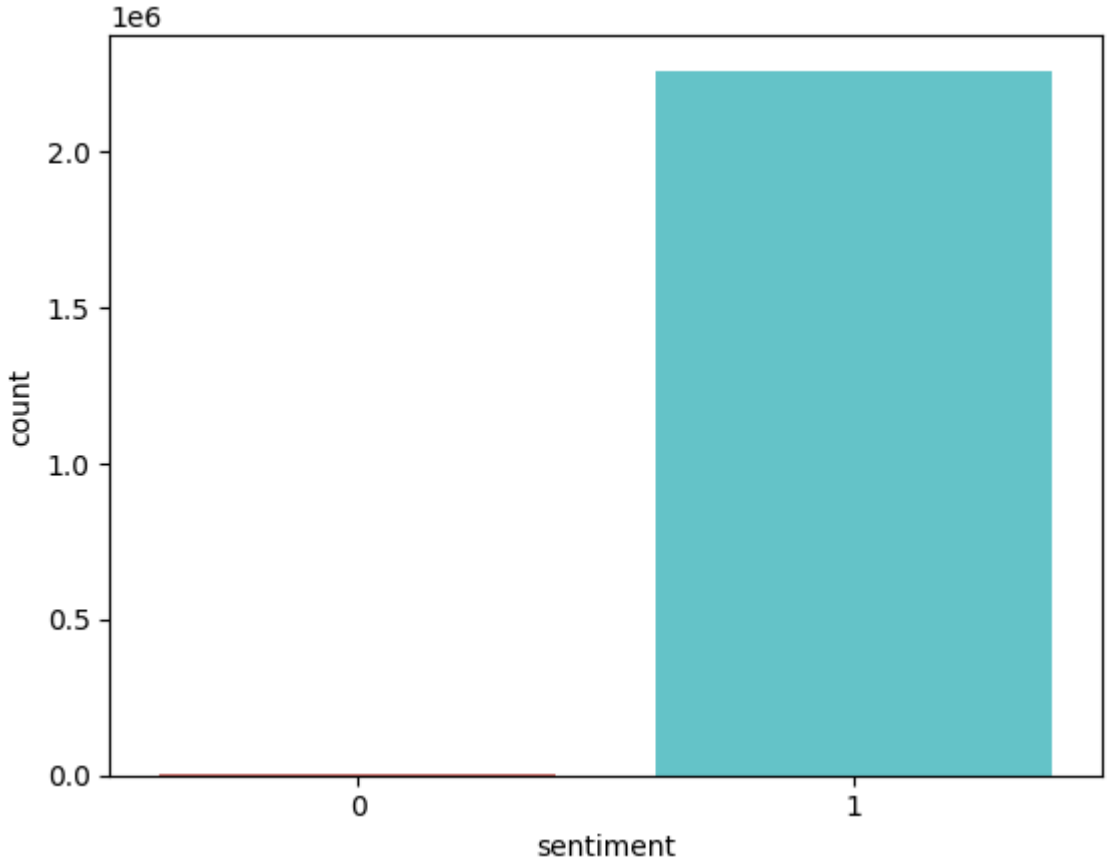

Figure 3. Sentiment distribution

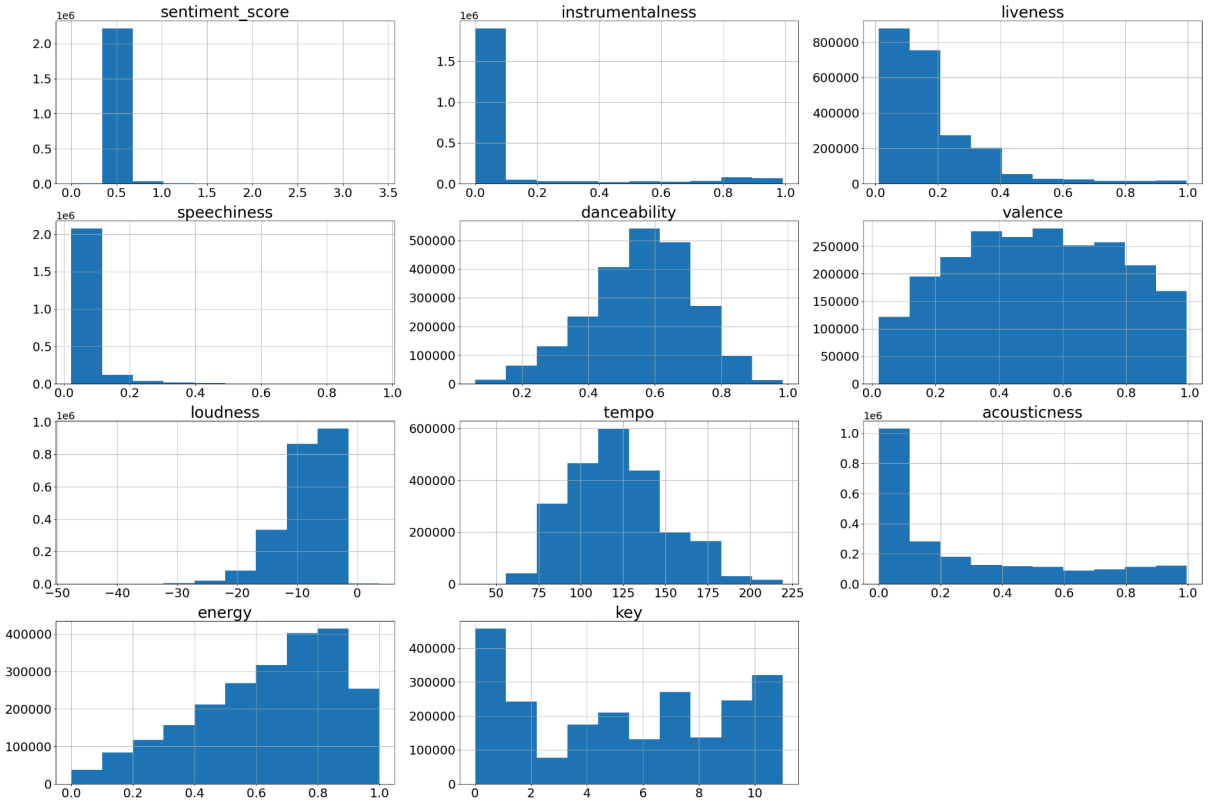

Figure 4. Distribution of the quantitative audio features in the nowplaying-RS

Then, we create a subset of the dataset with continuous variables and visualize their distributions using histograms (Fig. 4). Each feature exhibits different distribution patterns (some features are normally distributed with positive or negative skew). All features need to be normalized or scaled before modeling.

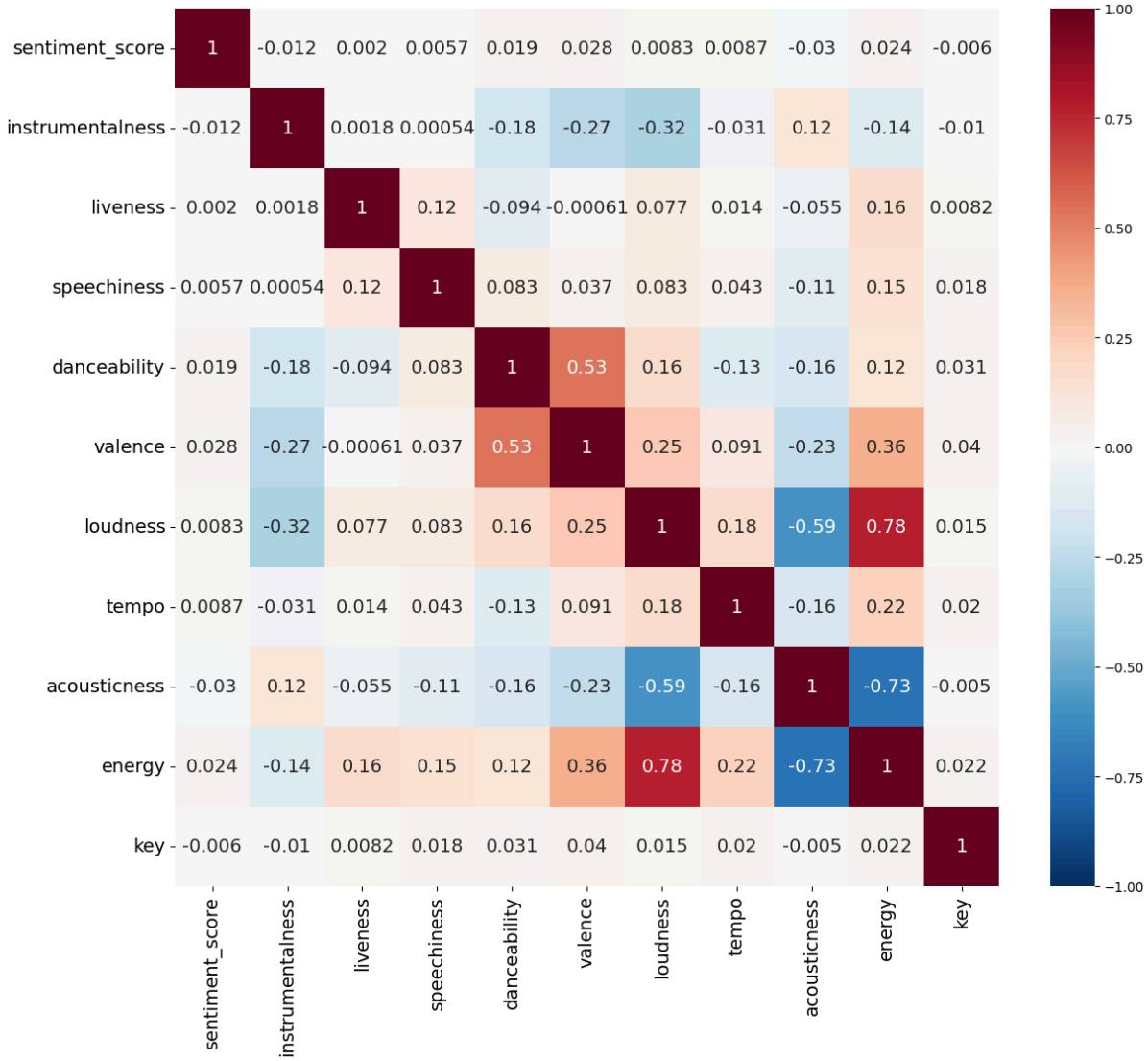

Figure 5. Correlation matrix between the chosen qualitative values of nowplaying-RS dataset

For the next step, we analyzed the correlations between numerical features to identify potential multicollinearity, which could affect model performance and interpretation. The heatmap (Fig. 5) shows the correlation coefficients between features. Notably, loudness and energy exhibit a high correlation, indicating multicollinearity. High multicollinearity can lead to instability in model coefficients, making it difficult to interpret their impact on the target variable. Apart from that, there is a strong negative correlation between acousticness and energy. Additionally, there is a moderate positive correlation between danceability and valence, as well as a moderate negative correlation between acousticness and loudness.

The strong correlation between **energy** and **loudness** and between **acousticness** and **energy** suggests multicollinearity. This can be addressed by removing one of the correlated features or using dimensionality reduction techniques.

Spotify Tracks Dataset

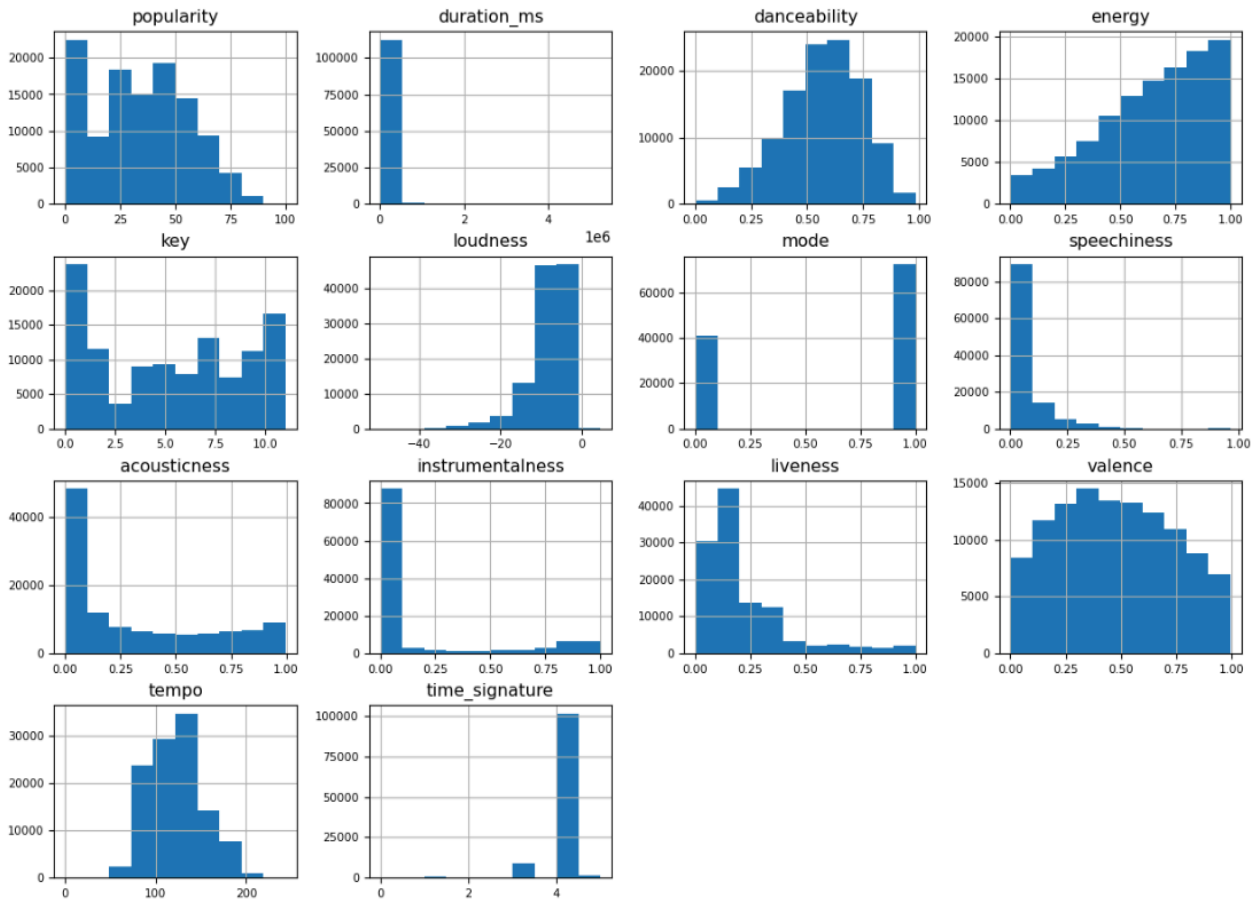

Figure 6. Distribution of qualitative features in the Spotify Tracks dataset (left to right, top to bottom: popularity, duration_ms, danceability, energy, key, loudness, speechiness, acousticness, instrumentalness, liveness, valence, tempo and time_signrature)

Similar to the nowplaying-RS dataset, Fig. 6 shows variables like danceability, valence, and tempo exhibiting quasi-normal distribution. However, the other features are distributed very unequally, notably, duration_ms, instrumentalness, and speechiness. This suggests the presence of a few extreme values in this category.

It must be noted that the tracks in this dataset are not chosen completely at random, as there are exactly 1000 entries for each of the 114 genres. This might explain why some variables don't seem to be randomly or normally distributed, notably energy in this dataset.

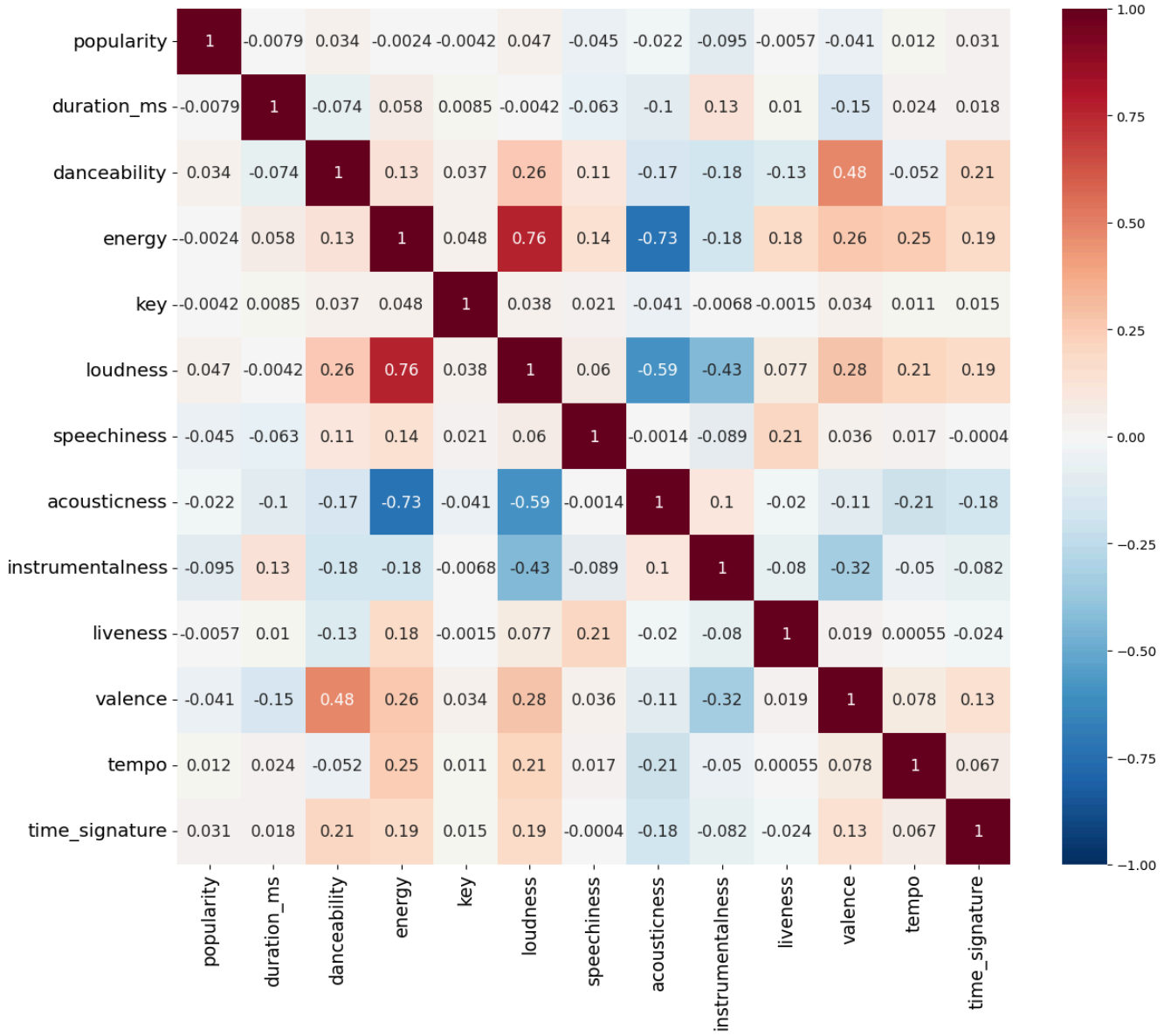

Figure 7. Correlation matrix between the chosen qualitative values of Spotify Track dataset

The correlation pattern of the Spotify dataset shows similarities to the nowplaying-RS dataset. The energy-loudness correlation and the negative acousticness-energy correlation are also observed here. (Fig. 7). As this dataset also contains the categorical variable “explicit”, which takes either the value 0 or 1, we can see another slight correlation: Tracks labeled as “explicit” score slightly higher in terms of “speechiness”. A possible explanation might be that rap songs are more “speechy” and also tend to be more explicit.

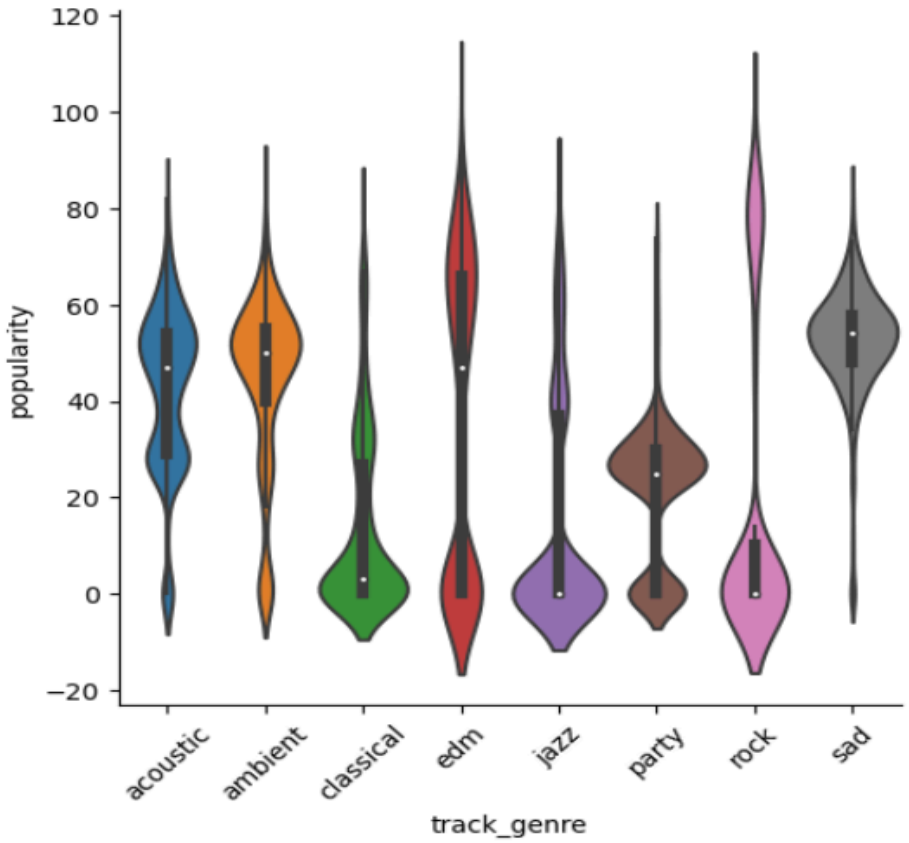

Figure 8. Popularity of the tracks by genres

Figure 8 shows the visualization of an exemplary selection of genres and their respective popularity from the Spotify dataset. As per the dataset's documentation, "[t]he popularity is calculated by algorithm and is based, for the most part, on the total number of plays the track has had and how recent those plays are."

This gives us an insight into possible interpretations of how the popularity is distributed across the genres. For example, EDM seems to have equally many very popular tracks and less popular ones. This might suggest that there are some well known hits in this genre that are listened to by a broader audience, and some lesser known that are enjoyed by fans of the genre. A similar effect can be seen with the jazz and classical genres: There are a few songs that are considered quite popular, but most of the songs are possibly lesser known and enjoyed by connaisseurs.