Project 1

Anxin Yi, Anqi Wu, Yuantingyu Li, Xiaoying Wang

2025-02-19

Introduction and Dataset description

As you settle onto a New York City subway and look around, you'll see a variety of groups of people, each staying in their own musical experience via different headsets. Music has always been a fundamental element of human entertainment—previously experienced live at concerts and performances, or via radio and television transmissions. In today's world, with the emergence of portable devices and user-friendly music apps, individuals have liberty to choose and relish any content that aligns with their personal preferences. This transition from shared listening experiences to individualized streaming illustrates a wider change in our interaction with music, highlighting the relevance and importance of studying digital music consumption.

Numerous music apps such as Spotify, Prime Music, and others can be found in the market. Nonetheless, Spotify distinguishes itself from among all the other offerings. Spotify platform enables skilled artists globally to upload their work without needing a record label and offers an interactive interface to foster significant connections between users and artists. Additionally, Spotify also offers music spanning different genres and artists: including Indie Rock, Top 40 Pop, film soundtracks, classical tunes, inspirational tracks, and podcasts across various categories. Spotify has more than 450 million users and around 195 million subscriptions globally.

As we know, with its vast user base and notable subscription amounts, Spotify has emerged as a central point for researchers and music creators to study. In this project, we utilized a dataset from Kaggle named "Nearly a Million Spotify Tracks." This dataset consists of an extensive array of metadata and audio characteristics for almost one million songs accessible on Spotify. By analyzing and exploring the dataset, we can grasp important information about music trends, song popularity, and different audio features.

In this report, our study focuses on three main goals: cleaning the data by resolving problems like duplicate records, inconsistent formats, and absent values; enhancing the dataset via feature engineering to create new variables; and conducting thorough exploratory data analysis (EDA) to reveal patterns and trends that illuminate the factors influencing digital

music consumption.

Data acquisition methodology

This Dataset was collected using Spotify's Web API to retrieve metadata, which provides programmatic access to Spotify's extensive music database. The dataset contains 899,702 entries and 33 features. Every row signifies a distinct Spotify track, whereas the columns encompass an extensive array of metadata and audio characteristics, such as track titles, performers, albums, release dates, and various musical features like danceability, energy, and tempo.

Cleaning and preprocessing steps

Check & Fix Incorrect Data Formats

The initial examination of the dataset revealed no duplicates. In order to maintain consistency and accuracy in our dataset, we examined the data types for each column and subsequently standardized the columns by changing them into suitable types. In particular, we transformed columns like "streams", "artist_followers", "album_total_tracks", "artist_popularity", "tempo", "energy", "key", "popularity", "duration_ms", "track_track_number", "rank", "mode", "time_signature", "speechiness", "danceability", "valence", "acousticness", "liveness", "instrumentalness", and "loudness" into numeric values, substituting any invalid entries (such as "unknown") with null values.

Subsequently, we standardized the date columns "album_release_date" and "added_at" by transforming them into datetime objects, guaranteeing a consistent date format throughout nearly millions entries of tracks. We then ensured uniformity in text fields by converting all values to uppercase and substituting string representations of boolean values ("True", "False") with actual booleans.

Furthermore, we change columns such as "chart", "region", "track_album_album", "album_name", "trend", and "track_artists" into categorical data types to restrict their values to specific categories. Specifically for the column "chart", we aimed to maintain consistency in categorical data, which avoids duplicate categories and enhances data precision.

Ultimately, for columns with list-like data represented as strings, we transformed these strings into actual Python lists, allowing for improved data management.

Handle Missing Values

To handle missing values across various data types, we initially displayed the count of missing values for each column to establish a clear baseline evaluation prior to imputation. For numeric columns, absent values were substituted with the median, since it is more resilient to outliers than the mean. In columns with categorical data, missing values were then replaced with "unknown" to prevent gaps; when a column is categorical, "unknown" is specifically included in its category list to ensure consistency. For date and text columns, missing values in "album_release_date" were discarded because of their minimal share, while substantial gaps in "added_at" were filled with a default date of "1970-01-01," and an indicator column ("added_at_is_missing") is added to monitor these imputations. For textual information like the "name" column, absent values were substituted with "Unknown Title" to maintain completeness. Further processing involves populating the "track_track_number" column with 0 as a numeric placeholder, replacing empty lists for absent values in list-like columns (such as "gerne" and "available_markets"), and assigning False to missing boolean entries in the "explicit" column.

Detect and Handle Outliers

To improve data quality and avoid extreme values skewing our analysis, we initially conducted outlier detection through visual inspection by utilizing a Seaborn boxplot—specifically creating a boxplot for the "streams" column to spot possible irregularities. Drawing on these visual insights, we lessened the effect of outliers through Winsorization, modifying the lowest 5% and highest 5% of "streams" values to minimize the impact of extreme data points while maintaining the general data structure. For the "website_visits" feature, we utilized a capping method based on percentiles by determining the 99th percentile and limiting any values that surpass this level, thus ensuring that outliers do not have an undue impact on statistical metrics or predictive models. These changes enhance model performance and aid in avoiding overfitting while also guaranteeing that our later analysis is influenced by significant trends instead of outliers.

Exploratory Data Analysis (EDA)

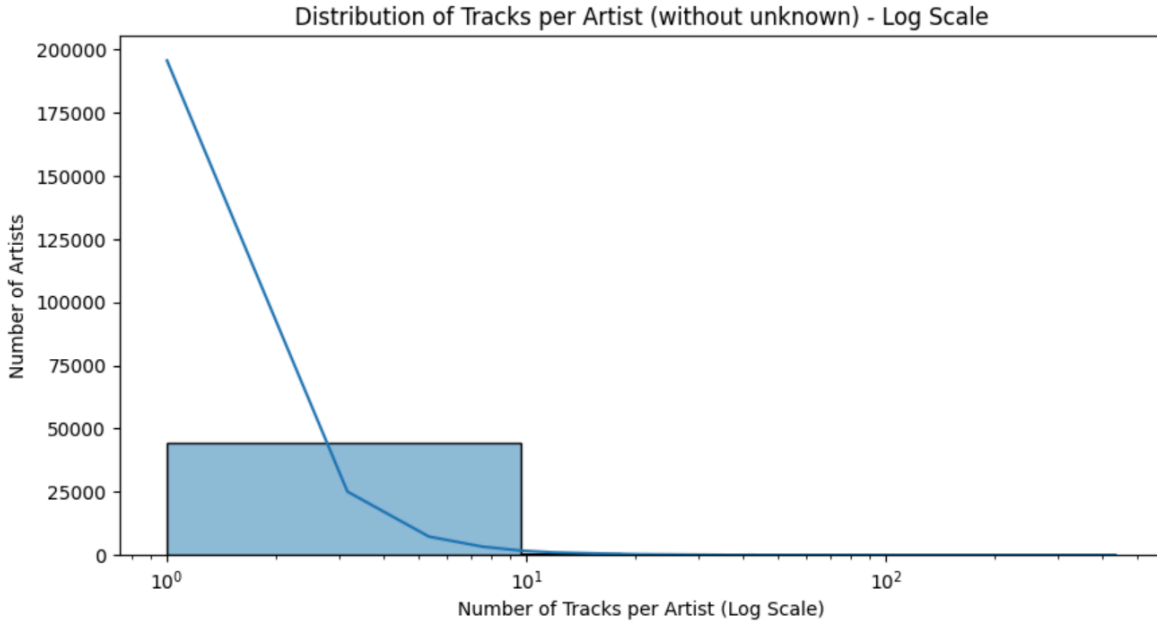
Generate summary statistics and descriptive insights

We started by pinpointing the categorical columns—specifically, track_artists, track_album_album, album_name, region, and trend—and the boolean columns, chart_top200 and chart_viral50. The numerical information was summarized by utilizing df_encoded.describe(), which offers statistical details like the mean, standard deviation, minimum, and maximum for every numerical attribute.

For categorical data, we determined the count of distinct values in each column to evaluate feature diversity and cardinality; likewise, for boolean columns, value counts indicated the occurrence of each binary category. The numerical summary revealed a significant disparity in song popularity, featuring an average of 21.88 and a peak of 98, indicating that although many tracks garner minimal attention, a select few lead the charts. Streaming figures showed a similarly uneven distribution, varying from just 1,001 streams to a maximum of 1.36 million. Regarding audio features, the typical tempo of 119 BPM corresponded with popular pop and dance genres, though anomalies like songs listed with a tempo of 0 BPM could suggest data inaccuracies or exceptionally slow pieces. Energy levels were moderate, with an average of 0.535 on a scale from 0 to 1, while loudness levels averaged -10.83 dB, demonstrating the compression characteristic of contemporary tracks. Furthermore, the majority of songs seem to be written in major keys, with an average key value of 5.2—implying that keys such as F and G were prevalent—and the primary time signature is 4/4, as shown by an average value of 4. Ultimately, danceability ratings hovered around 0.55, suggesting that numerous songs were ideal for rhythmic motion, whereas acousticness figures differed, indicating a blend of completely acoustic and fully electronic creations.

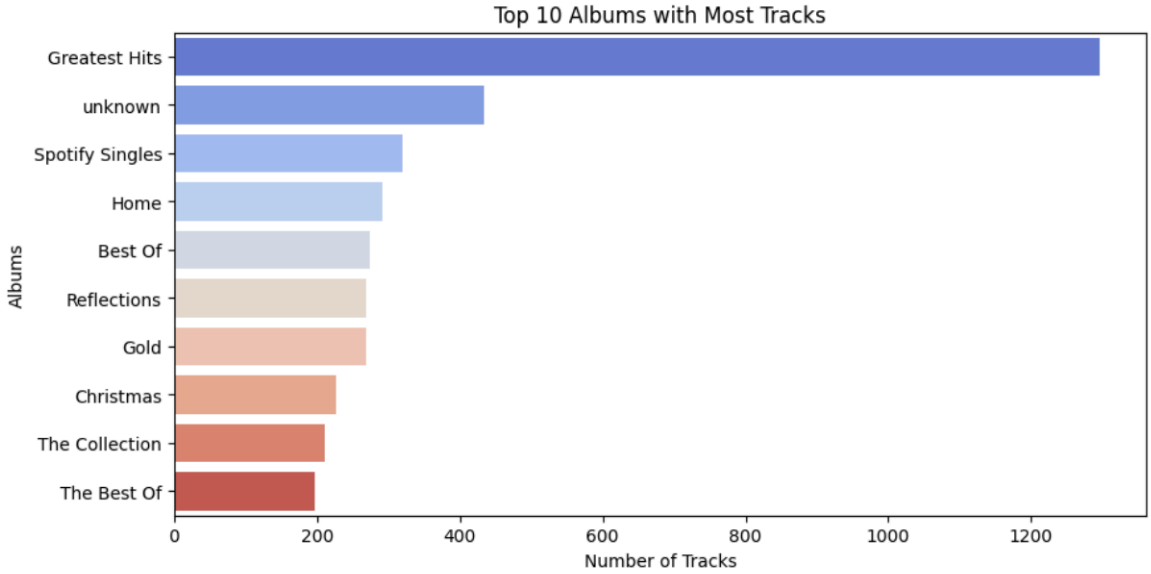
Categorical Variables: trend with track artists and track album

In this step, we first conclude the top ten top 10 most frequent artists in this dataset without unknown artists using bar plot. Then, we figure out the distribution of Tracks per Artist. From the graph, we conclude that the majority of artists have only a few tracks, with a few having over 10-100 tracks, as seen from the steep decline in artist count as the number of tracks per artist increases.



We started by analyzing the distribution of tracks among artists, showing that the majority

have just a handful of tracks, with this trend dropping sharply as the number of tracks per artist rises. Significantly, our analysis indicated that 88.89% of tracks fall under the "Unknown" category, underscoring considerable deficiencies in the artist metadata. Turning our attention to album data, a bar chart displaying the Top 10 Albums with the Most Tracks indicates that "Greatest Hits" is the most prevalent album title, which makes sense considering the fame of compilation albums. Nevertheless, "Unknown" is the second most frequent album title, suggesting that numerous tracks do not have appropriate album identification. Moreover, numerous generic album titles including "Spotify Singles," "Best Of," "Gold," and "Christmas" occur often, indicating that the widespread use of vague album names could hinder additional analyses.



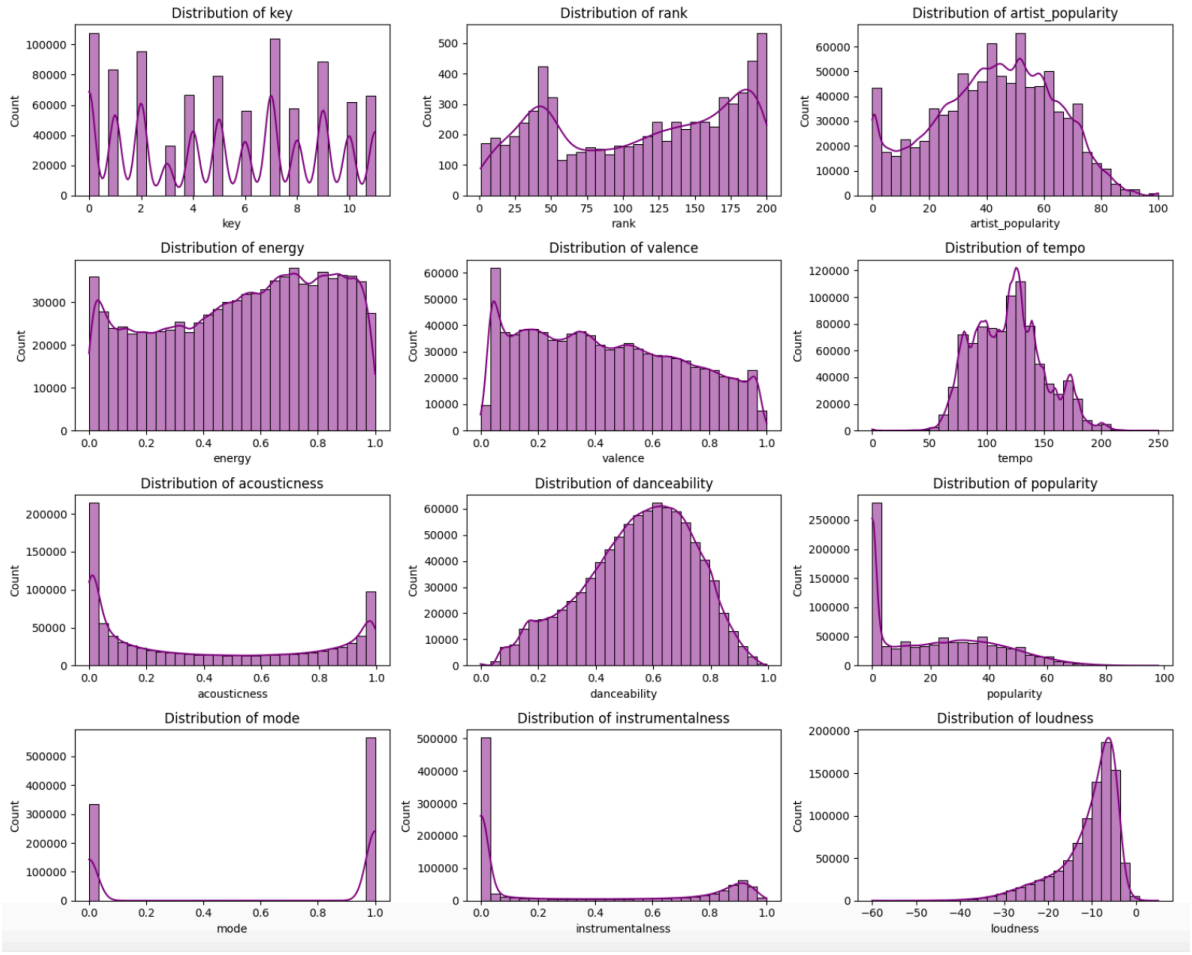
Categorical Variables: Regions, Album types distribution

This dynamic choropleth map illustrated the worldwide distribution of tracks, with color intensity representing the number of tracks—darker hues signify greater counts. In our dataset, nations like Australia, Argentina, and Brazil are prominent with the highest track counts, indicated by deep red shades, while certain areas appear empty, probably due to inadequate data coverage. This illustration clearly emphasized the disparities in music accessibility and streaming patterns across regions.

In summary, other than the album titles, the majority of categorical variables in the dataset were significantly imbalanced, with a large portion marked as "unknown." This absence of variability restricts their effectiveness in forecasting future trends, since these characteristics offered little differentiating strength. As a result, although the dataset provided important insights into global music distribution, its categorical variables might not serve as dependable indicators for analyzing future trends or making predictions.

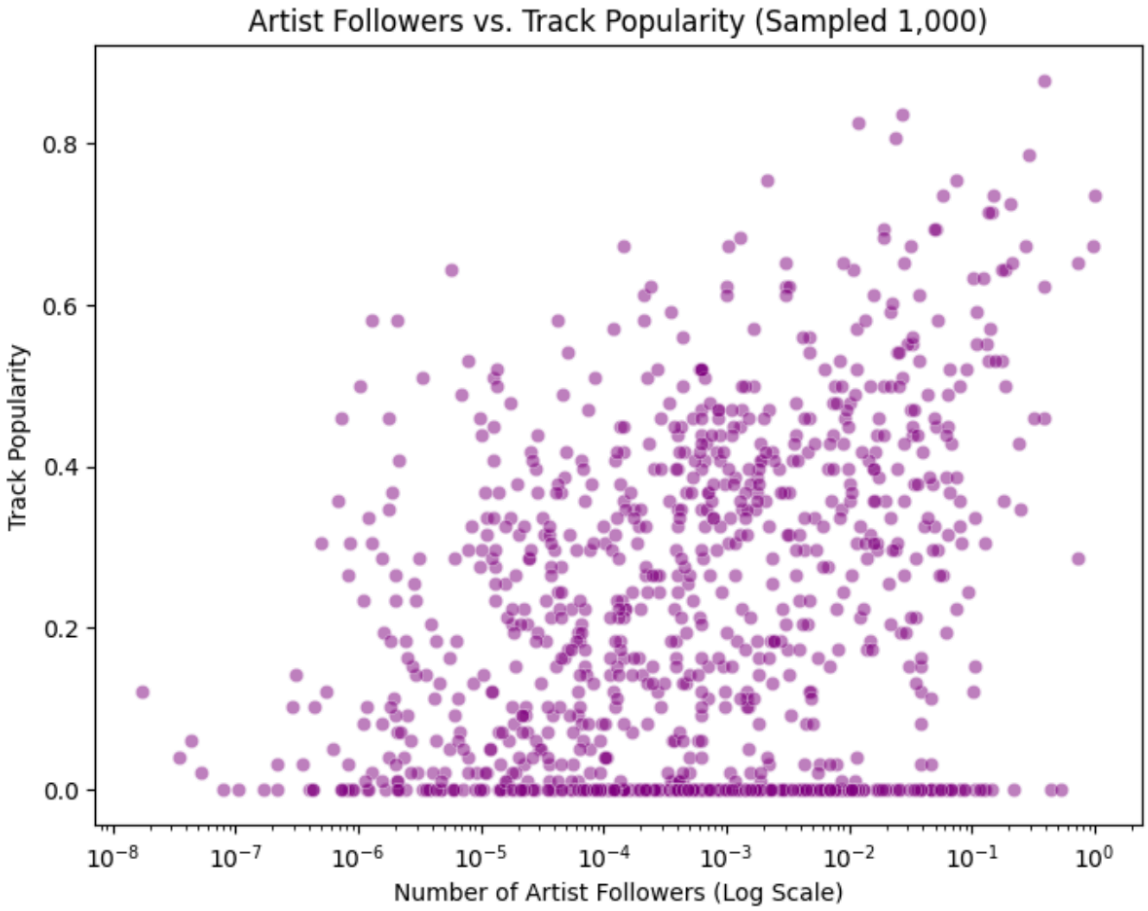
Numerical value distribution and trend analysis

The graph showed the distribution of all numerical features in details. In our assessment, we started by looking at the spread of track popularity—an essential measure of a song's recognition. The statistical data indicated a significantly right-skewed distribution of popularity, with 25.44% of tracks receiving a score of zero. This suggested that a large number of songs get little to no interaction from Spotify users, evident in the sharp drop in frequency as popularity rises. The kernel density estimation (KDE) highlighted this trend even more, showing a significant peak at zero and a secondary increase in the mid-range (approximately 0.2 to 0.4), indicating that just a portion of tracks reach moderate popularity. Conversely, when analyzing the skewness of alternative musical features, we observed that the 'key' of tracks—which indicates musical tonality—displays a nearly uniform distribution and is the variable with the least skew. This was anticipated because musical keys are categorical and generally show a uniform distribution across compositions. In general, although certain audio characteristics like danceability and energy showed distributions that were close to normal, others, including popularity and instrumentalness, were significantly skewed, suggesting some variables may need transformation prior to modeling, whereas others fitted well with anticipated musical trends.

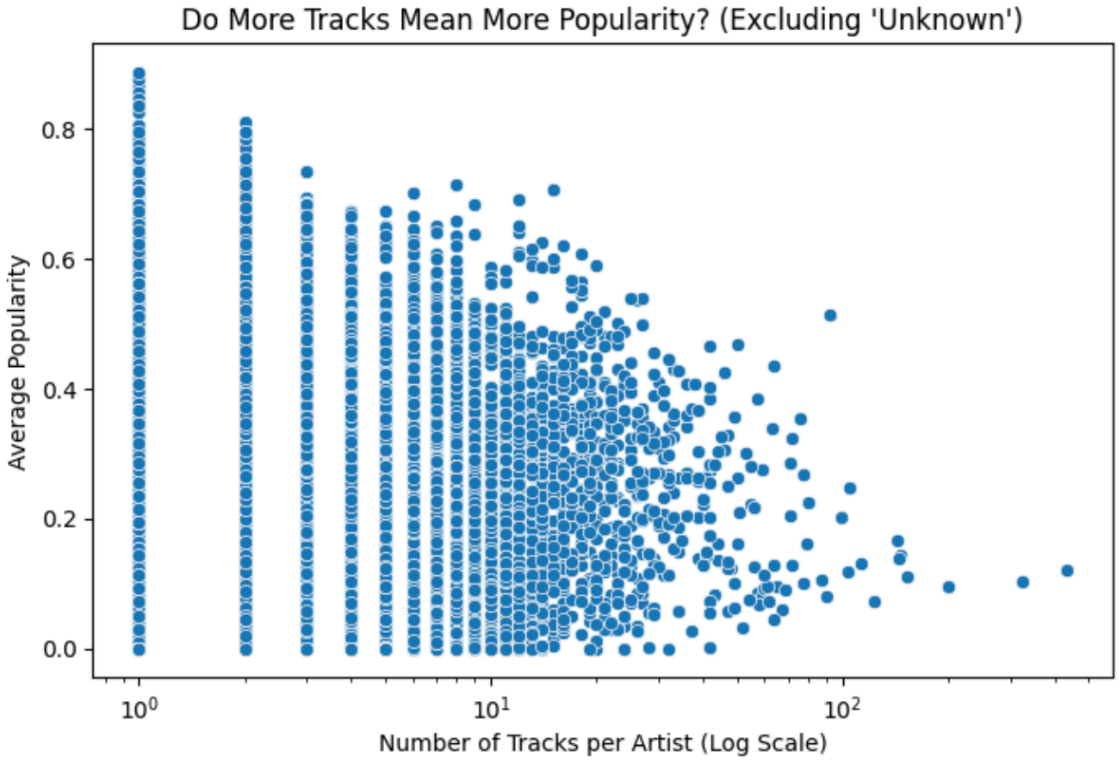


Exploring Relationship

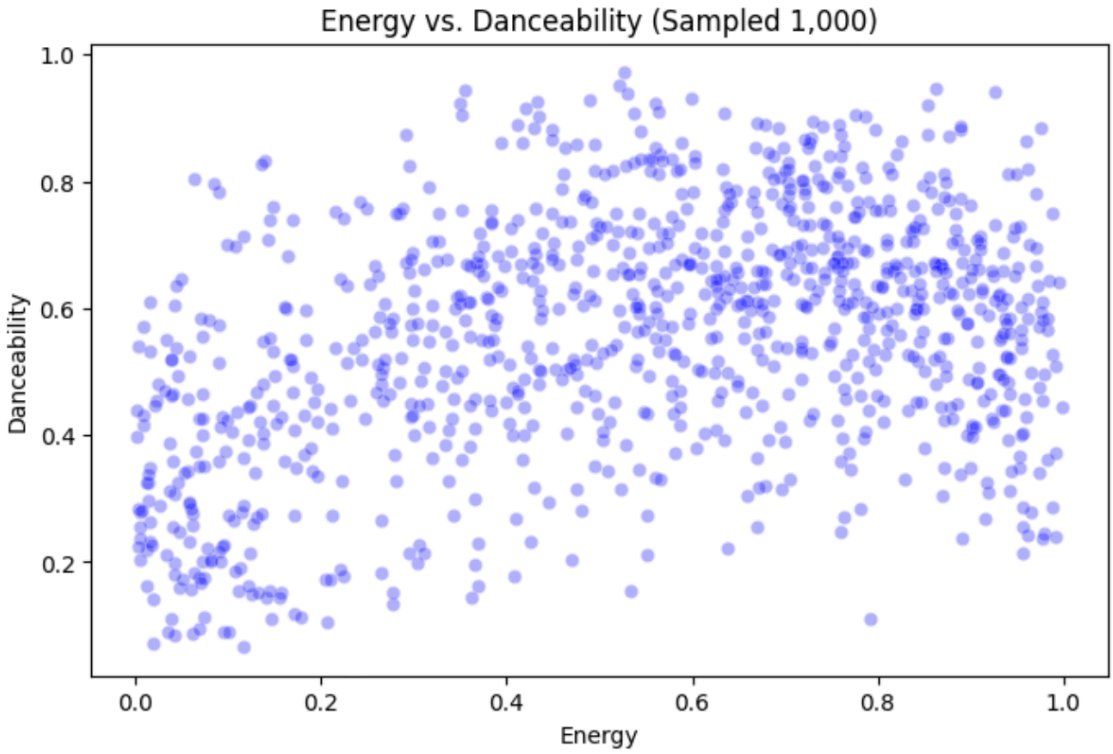
Next, we explored the connection between the number of followers an artist has and the popularity of their tracks to see if a bigger fanbase is linked to more popular songs. To investigate this, we created a scatter plot comparing Artist Followers to Track Popularity. The visualization showed a modest positive correlation, suggesting that artists with a larger following often produce tracks with greater popularity scores. Nevertheless, the connection wasn't very robust—numerous tracks possess no popularity irrespective of the artist's number of followers. This implied that although a substantial fanbase may help a track succeed, it was not an absolute indicator, and other elements probably had important influences on a song's popularity.



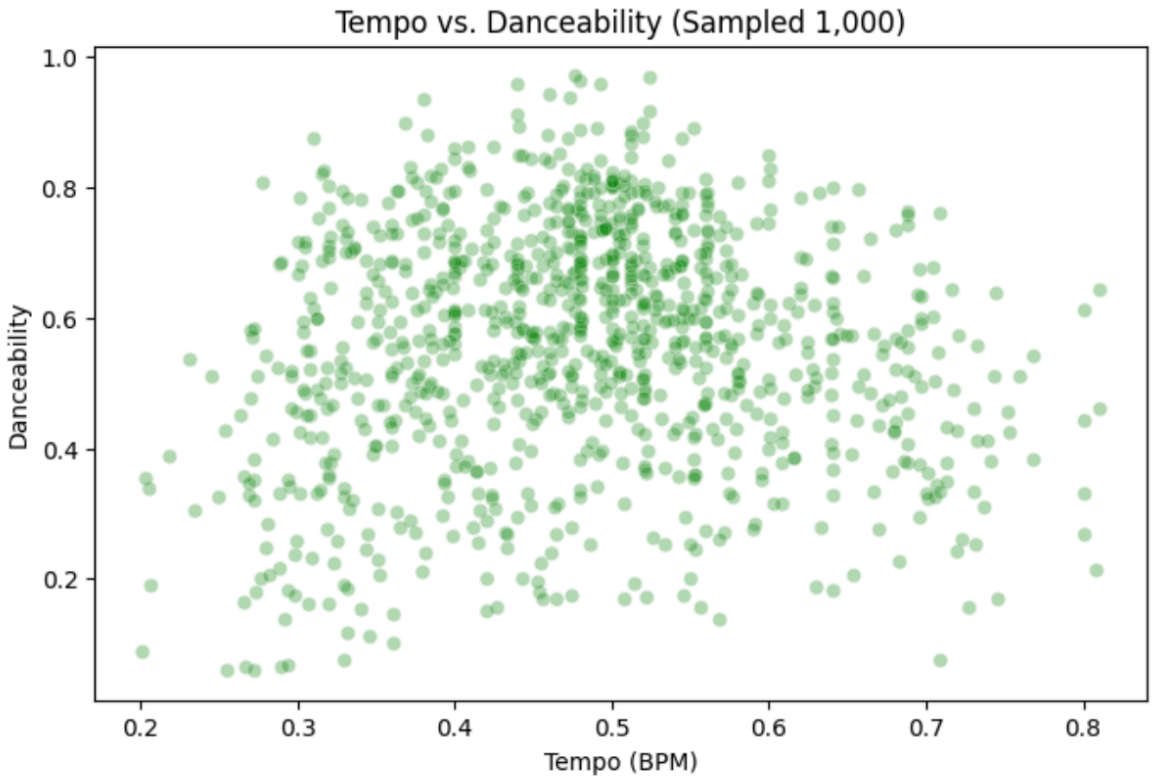
In addition, we investigated the question, "Do Increased Tracks Result in Greater Popularity?" To explore this, we created a scatter plot that looks at the correlation between the total tracks released by an artist and their average track popularity—omitting any entries marked as "unknown" for clarity. The findings indicated that the data points are extensively dispersed, suggesting that adding more tracks does not inherently relate to increased popularity. Indeed, although certain artists with extensive catalogs showed low average popularity, others with just a handful of songs attain significant popularity. This implied that merely boosting the number of releases does not ensure success, and other elements like track quality, marketing, and audience interaction are probably more crucial in influencing a song's popularity.



Next, we explored the relationship between danceability and other musical attributes, focusing on two questions: "Do high-energy tracks tend to have higher danceability?" and "Are higher tempo tracks more danceable?" For the first question, we generated a scatter plot for a sample of 1,000 tracks examining the relationship between energy and danceability. The plot reveals a clear positive correlation—tracks with higher energy levels tend to exhibit greater danceability. This finding suggests that the intrinsic energy of a track is a strong indicator of its potential to inspire movement and rhythm. While the analysis of tempo versus danceability remains to be explored further, these initial insights underscore the importance of energy as a key driver of danceability in music.



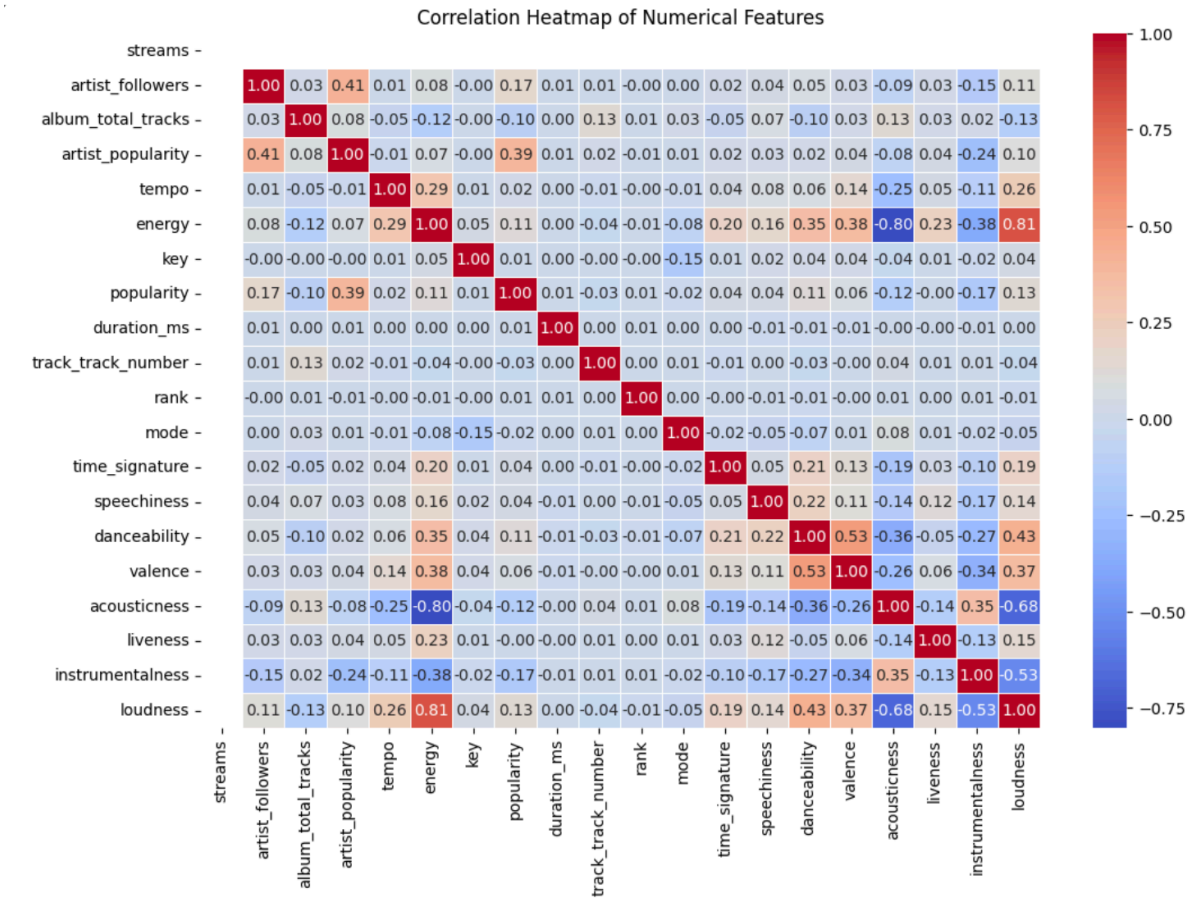
From the scatter plot of Tempo and Danceability for a random sample of 1,000 tracks, we can see that tempo (BPM) and danceability have a weak or unclear relationship. While there wasis a concentration of tracks with moderate tempo values (0.4 to 0.6) and varying danceability, no strong trend is visible. Since this wasis only a small sample, calculating correlation might help to see whether there exists a stronger connection.



Identify patterns, correlations, and anomalies

Correlation plot

The correlation heatmap allows us to intepretate the relationships between each numerical variables in this dataset. The color intensity represents the strength and direction of correlation, where:

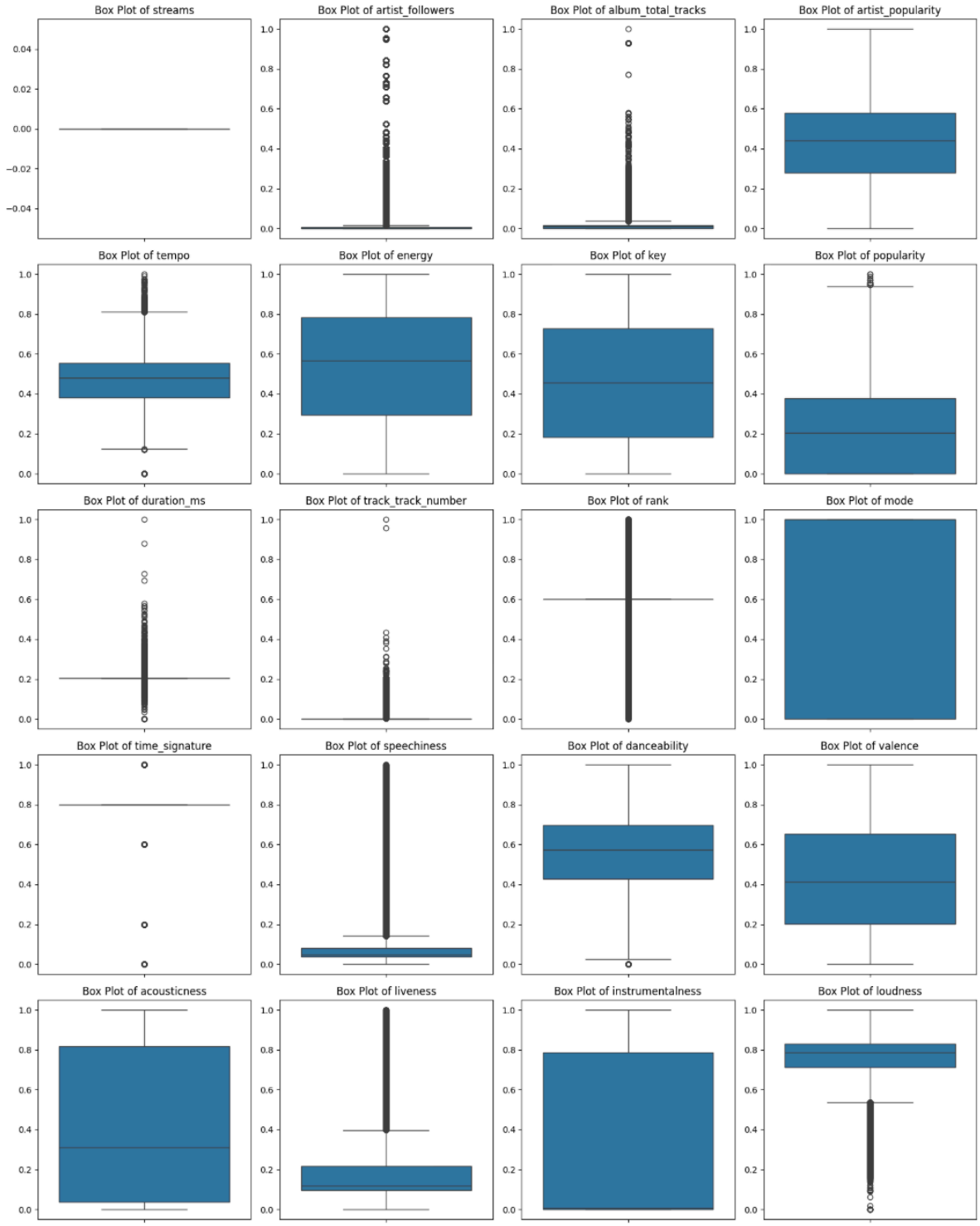


Red tones indicate positive correlations (closer to +1); Blue tones indicate negative correlations (closer to -1); Neutral colors represent weak or no correlation.

From the figure, we discover some strong positive correlations: Loudness and Energy (0.81): This strong correlation indicates that tracks with higher energy levels tend to be louder; Valence and Danceability (0.53): Tracks with higher valence also tend to have higher danceability, suggesting that upbeat and happy songs are generally easier to dance to.

Anomalies detection

Based on the box plots, lots of numerical variables exhibited significant outliers, such as artist_followers, album_total_tracks, and streams, which indicates the presence of a few highly dominant artists or albums that skew the dataset. Features like speechiness and liveness also displayed a heavy concentration near the lower end, with a few extreme cases. On the other hand, energy and danceability had a more even distribution with fewer extreme values. The box plots revealed that many numerical features contain outliers, and we should be carefully examined before modeling.


Box Plot of streams
Box Plot of artist_followers
Box Plot of album_total_tracks
Box Plot of artist_popularity
Box Plot of tempo
Box Plot of energy
Box Plot of key
Box Plot of popularity
Box Plot of duration_ms
Box Plot of track_track_number
Box Plot of rank
Box Plot of mode
Box Plot of time_signature
Box Plot of speechiness
Box Plot of danceability
Box Plot of valence
Box Plot of acousticness
Box Plot of liveness
Box Plot of instrumentalness
Box Plot of loudness

Provide initial interpretations of data insights

From the exploratory data analysis, here are some key observations:

1. Track popularity is highly skewed, with a significant portion (25.44%) having zero popularity, suggesting most songs from the dataset fail to gain traction.
2. Energy and loudness are strongly correlated, while danceability shows a moderate correlation with energy and tempo, indicating that higher-energy songs tend to be more danceable, but tempo alone does not correlate strongly with danceability.
3. Artist followers have a weak but noticeable relationship with track popularity, though outliers suggest that some songs gain popularity despite low follower counts, possibly due to virality or playlist placements.
4. The regional distribution of tracks is uneven, with certain countries producing significantly more music. However, this might mainly be due to the Unknown region dominating the whole dataset.
5. In addition, anomalies in artist followers, album track counts, and song popularity suggest the presence of extreme outliers, which may need further filtering.

Feature engineering process and justification

Normalize/standardize numerical variables

In this step, we are aiming at normalizing and standardizing numerical features in a dataset.

First, we identifyidentifying numeric columns using and handle the skewed numeric columns. We applyapplies the Box-Cox transformation to specific columns ('annual_income' and 'purchase_amount_total'), ensuring all values are positive by adjusting the minimum value before transformation. The Box-Cox method helps stabilize variance and improve normality, making the data more suitable for future use.

Next, we applyapplies StandardScaler to standardize all numerical columns, transforming them into a normal distribution with a mean of zero and a standard deviation of one.

Finally, we using MinMaxScaler re-scales the standardized values into the [0,1] range, ensuring all features have a consistent scale.

Our transformations in this step improves the dataset's quality by reducing skewness, standardizing distributions, and ensuring uniform feature scaling, which enhances model performance.

Encode categorical variables

This step handleshandling categorical variables effectively, ensuring our categorical data is machine-readable, making it suitable for predictive modeling.

Firstly, we identifyidentifies columns with object data type, listing them as potential categorical features. To refine the selection, we excludeexcludes specific columns ("track_id", "genres", "available_markets", "added_at", "name") that are unsuitable for encoding, ensuring only relevant features are transformed. We also addressaddresses rare categories within the "occupation" column by collapsing infrequent values (occurring in less than 5% of the dataset) into a common "other" category, preventing excessive feature fragmentation, which could negatively impact model performance. Finally, we applyapplies one-hot encoding to the remaining categorical columns, converting them into binary variables while dropping the first category to avoid multicollinearity.

Create meaningful new features

Firstly, we calculatecalculates the average stream count per artist (artist_stream_avg) and mergemerges this metric back into the dataset, providing context on an artist's prominence. This helpshelp us assess an artist's general popularity based on their historical streaming performance. Next, Genre Features are extracted by counting the number of genres associated with each track (genre_count) that provides insights into the diversity of a track's genre classification.

Secondly, for audio Profile Features, we create an interaction term (energy_dance_ratio) to capture the relationship between a song's energy and danceability, ensuring the denominator is not zero that might be a good topic to explore in the future since it captures the relationship between energy and danceability, which can be a key factor in hit song prediction.

Thirdly, in Market Penetration Features, we want to calculate market coverage metrics. We evaluateevaluates a track's availability by counting the number of markets (market_coverage) and flagging whether it is accessible in at least 20 markets (top_20_market). This might helphelps evaluate a track's global reach and availability and be usefuland useful in determining the impact of market exposure on streaming success.

Fourthly, for Track Version Features, where it identifies special versions of tracks (e.g., remixes, live versions) using regular expressions. This might be useful for identifying trends in the success of different track versions and help in market segmentation for targeted marketing campaigns.

Fifthly, to capture Trend Response Features, we encodeencodes whether a track is a new entry in trending charts (is_new_entry). It helpshelp to track how often new entries appear in trending lists, providing insight into market dynamics.

Lastly, Collaboration Features are included by detecting whether multiple artists collaborated on a track (is_collaboration), identified through the presence of a comma in the artist field. This can helpHelps determine whether collaborations lead to higher streaming numbers.

The engineered features above can provideproviding additional context on artist influence, genre diversity, song characteristics, market reach, and trend dynamics, which significantly improve model performance.

Final Data Checks

Finally, we check the data to see whether this is dataa data with good quality and are readyreadiness for further analysis or modeling. We check the missing value, data types, data shape and sample and the results confirming that all transformations, encoding and feature engineering, have been applied correctly.

Summary of key findings

In this project, we faced significant data challenges, including missing values, inconsistent formatting, and outliers. We are looking for the

Exploratory Data Analysis (EDA)

- Energy and loudness are strongly correlated, while danceability shows a moderate correlation with energy and tempo, indicating that higher-energy songs tend to be more danceable, but tempo alone does not correlate strongly with danceability.
- Artist followers have a weak but noticeable relationship with track popularity, though outliers suggest that some songs gain popularity despite low follower counts, possibly due to vitality or playlist placements.

Feature engineering process and justification

- We create six meaningful new features that can be further use in the research.

We Artist Influence: Average stream count per artist.

Audio Characteristics: Energy-danceability ratio to study hit song trends.

Market Reach: Number of available markets and top 20 market presence.

Track Versions: Identifying remixes, live versions, and special releases.

Trend Analysis: Marking new chart entries.

Collaboration Impact: Detecting multi-artist tracks

Challenges faced and future recommendations

Data Cleaning and Handling Inconsistencies:

Challenge Faced:

1. **Inconsistent and Mixed Data Types**: There are several columns that were stored as string instead of numeric or boolean, causing several parsing errors. Some date columns used different formats (DD/MM/YYYY vs. ISO timestamps).
2. **High Volume of Missing Values**: Multiple columns had NaNs, including numeric, categorical, and date fields. Deciding between dropping rows, median imputation, or placeholder values involved trade-offs and more careful consideration.
3. **List-Like Columns**:Columns such as genres and available_markets stored lists of labels per row, which caused issues with standard one-hot encoding and led to errors ("unhashable type: list")
4. **Outlier Influence**: Certain numeric columns like "streams" contained extreme values that risked skewing analyses. Balancing data integrity with the need to cap or worsen them required careful consideration.
5. **Large Dataset Size**:with hundreds of thousands of rows, operations like checking for duplicates, and converting data types were time-and memory-intensive, it needs more efficient data-handling techniques.

Future recommendations:

Advanced Imputation: beyond median or placeholder fills, maybe consider using KNN or Iterative imputation to better leverage relationships among features for missing data.

1. **Multi-Label Encoding**:For columns that contain lists of labels (e.g., genres), explore **c**ustom multi-hot or text-based encoding. This can provide richer insights than ignoringthanignoring those columns or treating them as free text.
2. **Domain-Specific Outlier Analysis**:Instead of generic winsorization, if we could have a better understanding of datasets, for example, it is better that we collaborate with domain experts (e.g., music industry) to identify if high streams or unusual duration_ms are authentic phenomena or data errors.Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA)

- Challenge Faced:

During the initial exploration process, we apply one-hot encoding techniques to handle categorical variables. However, we chose to retain some visualizations using the original categorical data to preserve interpretability. Since one-hot encoding can significantly increase dimensionality and reduce model interpretability, keeping both versions allows for a more balanced analysis.

For Boolean variables, the "chart_top200" and "chart_viral50" columns were appropriately coded as Boolean features. However, we observed a significant class imbalance, where False values far outnumbered True values. This imbalance could introduce bias in future modeling if these variables are used as predictive features. A similar issue was found in most categorical variables, where the dominant category was often "unknown", potentially skewing insights and limiting predictive power.

- Future recommendations:

We plan to incorporate time series analysis to explore how key variables, such as "popularity", "artist_followers", and "artist_popularity" evolve over time. Understanding trends and seasonality could provide deeper insights into the factors influencing a track's success and help improve predictive modeling.

Feature engineering process and justification

- Challenge Faced:

Initially, we aimed to create a‘days_since_release_to_added‘ feature to measure market responsiveness by calculating the time interval between album release and chart entry. However,implementation revealed critical issues: over 50% of "added_at" values were missing. Using default dates (e.g., 1970-01-01) to fill gaps resulted in extreme negative deltas (e.g., -18,000 days), distorting data distributions and risking model misinterpretation of artificial patterns (e.g., "tracks from 1970 added in 2023"). Even with missing flags, these anomalies polluted feature space.

- Future recommendations:

Abandoned the delta calculation to prioritize reliability. We retained static temporal features (release year/month) andencoded ‘added_at‘ missingness as a binary flag. While sacrificing some temporal granularity, this approach ensured cleaner data and preserved model interpretability—no longer forcing explanations for nonsensical"time-travel" signals. The trade-off strengthened trust in the feature set while addressing core data quality issues.

Each member's contribution to the project

Task 1: Anxin Yi, Anqi Wu, Yuantingyu Li, Xiaoying Wang

Task 2: Anxin Yi

Task 3: Anqi Wu

Task 4: Yuantingyu Li

Task 5: Xiaoying Wang, Anqi Wu