# Music recommendation system

Gayatri Yendamury, MTech (AI)(gayatriy@iisc.ac.in)
Bagi Shirisha, MTech (AI), bagishirisha@iisc.ac.in
Priyabrata Samantaray, MTech (ECE), priyabratas@iisc.ac.in
Vineet Kumar Tripathi, MTech (ECE), vineetkt@iisc.ac.in

3 December 2024

## 1 Problem Statement

Music streaming platforms offer millions of tracks, but users often struggle to find songs that match their choice. Existing recommendation systems are limited, often suggesting irrelevant content, leading to user disappointment and disengagement.Providing accurate song recommendations is crucial for keeping users engaged with the platform. Better recommendations mean happier users, who are more likely to listen music longer ,explore more and remain loyal to the service. This, in turn, increases user retention and revenue for the platform.

A smarter, data-driven recommendation system is needed to enhance personalization and improve user experience by offering more accurate music suggestions.

## 2 Dataset description

### 2.1 Dataset

We have used the public Spotify dataset available in Kaggle (`spotify Dataset`).

The datasets used are `data.csv` and `data_by_genre.csv`.

**Dataset columns description** :
The dataset has below mentioned feature columns:

- **valence**: Indicates the musical positiveness of a track, where high values represent happier and more positive sounds, and low values indicate sadder or negative tones.
- **year**: The release year of the track.It is useful for analyzing trends over time.
- **acousticness**: Measures how acoustic a track sounds. High values suggest the track is more acoustic, while low values imply more electronic or synthetic sounds.
- **artists**: Names of the artists involved in the track.It helps to group songs by same artist.
- **danceability**: Reflects how suitable a track is for dancing. High values denote easier dance rhythms, while low values indicate more complex or irregular beats.
- **duration_ms**: Duration of the track in milliseconds.The longer durations values are associated with extended compositions or live recordings.
- **energy**: It Represents the intensity and activity level of a track. Higher values suggest energetic tracks, while lower values indicate calm or mellow tracks.
- **explicit**: Indicates if the track has explicit lyrics (1 : yes, 0 : no).
- **id**: Unique Spotify ID for the track.
- **instrumentalness**: Predicts the likelihood of a track being instrumental. High values (close to

1.0) suggest little or no vocals, while low values indicate vocal presence.

- **key**: The musical key of the track, represented by numbers (0 to 11).
- **liveness**: It estimates the presence of a live audience. High values suggest live performance , while low values indicate studio recordings.
- **loudness**: Overall loudness of the track in decibels. Higher values mean louder tracks, while lower values are quieter.
- **mode**: Indicates modality (1 = major, 0 = minor); major often sounds happier, while minor is generally more somber.
- **name**: The title of the track.
- **popularity**: Popularity score on Spotify (0 to 100). High values indicate popular tracks, while low values indicate less popular tracks.
- **release_date**: The date the track was released.It helps to analyze time-based trends.
- **speechiness**: Measures the presence of spoken words in the track. High values indicate more speech-like content (e.g., podcasts), while low values suggest minimal or no speech.
- **tempo**: Tempo of the track in beats per minute (BPM). High values indicate faster tracks, while low values indicate slower songs.

This dataset provides a range of audio features that describe the musical characteristics and popularity metrics for each track, essential for building a recommendation system.

# 3 Data Collection and Pre-Processing

The dataset was read into a pandas dataframe for further analysis.

## 3.1 Assessing Missing Data and checking for duplicate values

The dataset is well prepared .There were no missing values found in any of the columns.Also , no duplicate rows were present.

## 3.2 Descriptive statistics of features

Descriptive statistics (like mean, standard deviation, minimum, maximum, and quartiles) provide insights into data distributions.It helps in identifying central tendency, spread, and outliers across multiple datasets in one go. Shown below is a snippet of output statistics using Pandas library command describe.

Summary Statistics for Data:

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| valence | 170653.0 | 0.528587 | 0.263171 | 0.0 | 0.3170 | 0.540000 | 0.7470 | 1.000 |
| year | 170653.0 | 1976.787241 | 25.917853 | 1921.0 | 1956.0000 | 1977.000000 | 1999.0000 | 2020.000 |
| acousticness | 170653.0 | 0.502115 | 0.376032 | 0.0 | 0.1020 | 0.516000 | 0.8930 | 0.996 |
| danceability | 170653.0 | 0.537396 | 0.176138 | 0.0 | 0.4150 | 0.548000 | 0.6680 | 0.988 |
| duration_ms | 170653.0 | 230948.310666 | 126118.414668 | 5108.0 | 169827.0000 | 207467.000000 | 262400.0000 | 5403500.000 |
| energy | 170653.0 | 0.482389 | 0.267646 | 0.0 | 0.2550 | 0.471000 | 0.7030 | 1.000 |
| explicit | 170653.0 | 0.084575 | 0.278249 | 0.0 | 0.0000 | 0.000000 | 0.0000 | 1.000 |
| instrumentalness | 170653.0 | 0.167010 | 0.313475 | 0.0 | 0.0000 | 0.000216 | 0.1020 | 1.000 |
| key | 170653.0 | 5.199844 | 3.515094 | 0.0 | 2.0000 | 5.000000 | 8.0000 | 11.000 |
| liveness | 170653.0 | 0.205839 | 0.174805 | 0.0 | 0.0988 | 0.136000 | 0.2610 | 1.000 |
| loudness | 170653.0 | -11.467990 | 5.697943 | -60.0 | -14.6150 | -10.580000 | -7.1830 | 3.855 |
| mode | 170653.0 | 0.706902 | 0.455184 | 0.0 | 0.0000 | 1.000000 | 1.0000 | 1.000 |
| popularity | 170653.0 | 31.431794 | 21.826615 | 0.0 | 11.0000 | 33.000000 | 48.0000 | 100.000 |
| speechiness | 170653.0 | 0.098393 | 0.162740 | 0.0 | 0.0349 | 0.045000 | 0.0756 | 0.970 |
| tempo | 170653.0 | 116.861590 | 30.708533 | 0.0 | 93.4210 | 114.729000 | 135.5370 | 243.507 |

Figure 1: descriptive statistics for numerical feature columns of dataset

Below are some observations from descriptive statistics:

1. **Balanced Features:** Balanced Features: Valence, Acousticness, Danceability, and Energy show a diverse range of musical characteristics, suggesting varied tracks, while Instrumentalness and Speechiness indicate a vocal-dominant dataset, with most tracks being music-focused rather than speech-heavy.
2. **Popularity:** With 75% of the tracks having a popularity score below 48, there is a notable scarcity of highly popular tracks. This highlights an opportunity for the recommendation system

to explore and promote lesser-known but potentially valuable tracks.

3. **Year :** We have data samples from 1921 to 2020 in this dataset.

4. **loudness :** The loudness feature ranges from -60.0 to 3.855, encompassing both positive and negative values. To facilitate better analysis and ensure that all features are on a similar scale, it would be beneficial to normalize this feature.

## 3.3 Visualizing Outliers

Outliers are data samples that falls on the far left or right side of the ordered data. Generally, the outliers fall more than the specified distance from the first and third quartile (IQR: Interquartile Range i.e. outliers are greater than Q3 + (1.5 * IQR) or less than Q1 - (1.5 * IQR).

Shown below are the boxplots for different numerical features of the dataset(figure 2):

Below are the conclusions drawn from the boxplot analysis:

- **duration_ms and tempo**: Songs with very high durations (like live performances) or unique tempos (extremely slow or fast) are often valid data points and may be retained depending on analysis needs.

- **instrumentalness, liveness and speechiness** : It is expected that the dataset will show a greater concentration of values near 0 for instrumentalness, liveness, and speechiness. This aligns with the intention to recommend tracks that are primarily music-focused and feature significant vocal elements.

- **loudness**: Extremely low loudness values may represent specific song genres or recording qualities and may need further investigation to decide if they should be capped or retained.

- **explicit** : This is a categorical feature and it has only few data samples with explicit feature value as 1.However, it might help in personalised song recommendations.Hence, its better to retain this feature outliers.

- **danceability** : There are 143 outliers in Data and 11 outliers in Genre Data, with low danceability values.These outliers can be capped.

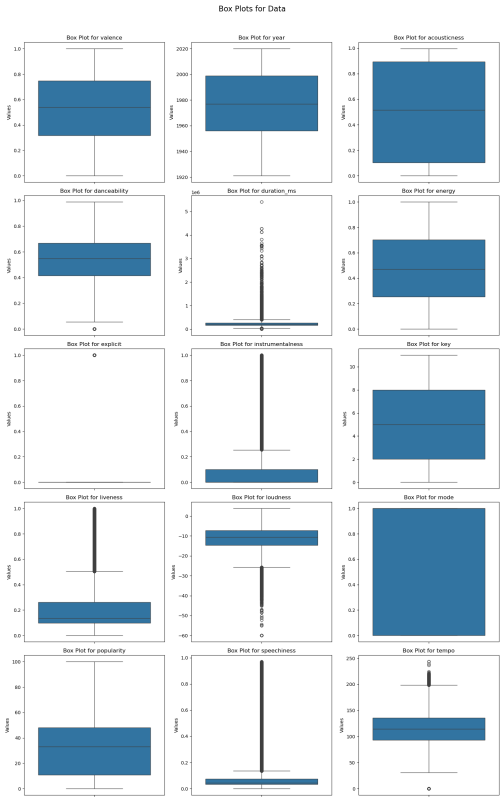


Figure 2: boxplot of numerical features

## 3.4 Outlier Handling

For outlier handling , Outlier Capping is chosen for its simplicity, effectiveness, and ability to retain data integrity. This method minimizes the impact of extreme values while preserving the dataset for accurate, interpretable analysis.

The column danceability had 143 outliers in Data and 11 outliers in Genre Data. After outlier capping , the column danceability has no outliers as plotted in below boxplot(figure 3). All other features are retained with outliers , as they have too many outliers

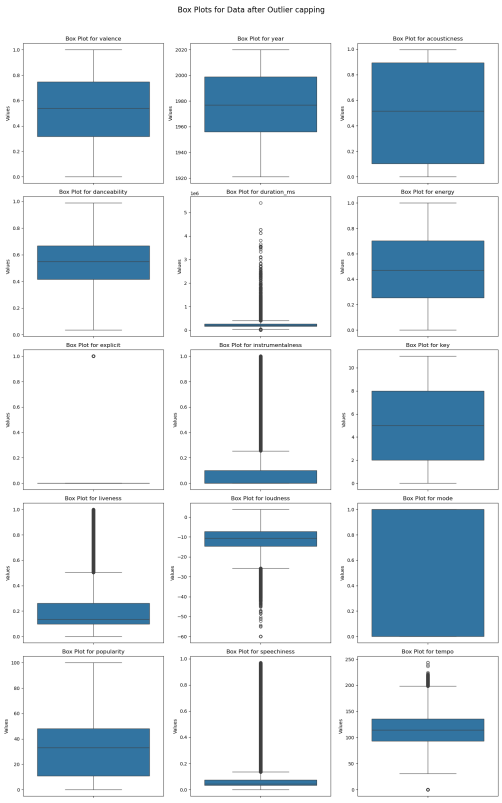


Figure 3: boxplot after outlier capping of numerical features

and they can provide important information for music recommendation system.

## 3.5 Feature Engineering

To enhance interpretability and enable more effective analysis, we created new categorical features and transformed existing ones. These engineered features will make it easier to identify trends and improve compatibility with models that benefit from normalized or categorical data.

### 3.5.1 Add new music feature categorical columns

Columns valence_category, acousticness_category, danceability_category, energy_category, liveness_category, instrumentalness_category, speechiness_category, tempo_category,loudness_category, popularity_category, key_category, duration_ms_category have been added to provide a clearer understanding of the numerical values, based on their relative positions within the overall range (e.g. less than 25%, 50%, 75%, or 100%).

### 3.5.2 Convert or Normalize existing columns

- To address the negative values in the 'loudness' column, we created a new column, 'loudness_scaled', which has been normalized to a range of 0 to 1.
- The column 'duration_min' is added after converting 'duration_ms' to minutes and duration_ms is discarded.
- The column 'release_decade' is added after converting 'year' to corresponding decade.
- Category Columns 'mode_category', 'explicit_category' are added which helps in mapping numerical values i.e. 0 and 1 to the corresponding meaning.This will help in further data analysis.

### 3.5.3 Removing unecessary columns

Column 'release_date' is inconsistent and thus discarded as the release year information is already present in 'year' column.

### 3.5.4 Adding columns based on artists

New columns 'artist_count','artist_category' are added to the dataframe.This will help in data analysis based on artist count. This might also support in music recommendation by capturing user preferences for solo versus group performances.

Finally , the pre-processed dataset is stored as `data_cleaned.csv` and `genre_data_cleaned.csv`

# 4 Data Exploration

## 4.1 Observations based on popularity

### 4.1.1 Least popular songs and the similarity in their features

The least popular songs (having popularity score between 0 to 10) have common features like very low energy,low danceability , low valence- emotionally negative or neutrals, low tempo , low key-typically quieter tonal center,moderate live presence,mostly instrumental and from release decade 1940's.

### 4.1.2 Most popular songs and the similarity in their features

The song 'Dakiti' is the most popular song with popularity score of 100.It has high energy , neutral emotions and high danceability. The most popular songs (having popularity score between 90 to 100) have common features like moderate energy,high danceability , low valence- emotionally negative or neutrals, high tempo,very low live presence,no instrumental elements, primarily vocal driven and from release decade 2020's.

### 4.1.3 Duration of the Songs in most popular Genres

The genre 'south african house' has highest song duration and 'alberta hip hop' has lowest song duration , as shown in below bar graph.

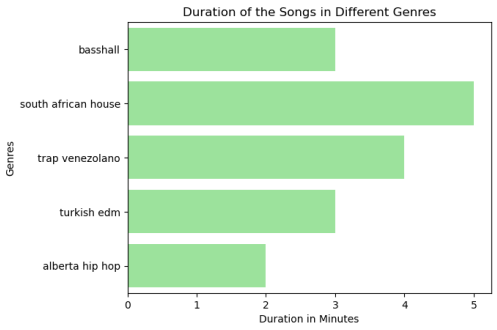


Figure 4: song duration across multiple genres

## 4.2 Univariate Analysis

### 4.2.1 Distribution of Numerical Features

The feature distributions give insights into the characteristics of the dataset.Shown below are the distribution of numerical features of the dataset.

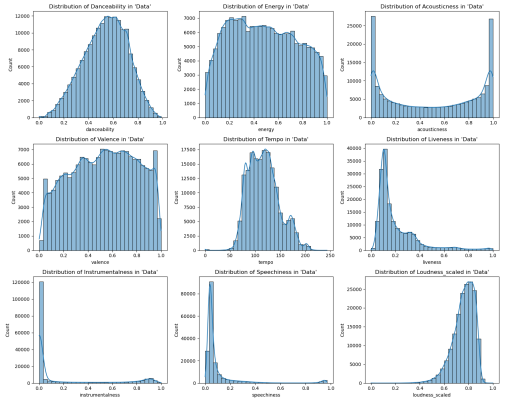

Figure 5: distribution of numerical features

From above observations we can say that the dataset represent a variety of musical genres with an emphasis on moderately energetic, moderate tempo , mostly high loudness , studio-produced tracks with some vocal content .

### 4.2.2 Distribution of Categorical Features

We have plotted the distribution of categorial features of newly added columns such as valence category, acousticness category, danceability category, energy category, etc.. in order to understand the spread of the dataset with respect to each music feature. The series of bar charts below (figure 6) provides a breakdown of various categorized features in the dataset, displaying the distribution of songs across different features.

These distributions suggest that the dataset is diverse across multiple musical features, allowing for a broad range of recommendation possibilities. The dataset is slightly skewed towards recommending songs which are vocal and studio recorded songs , and with no explicit content and high mode. However,

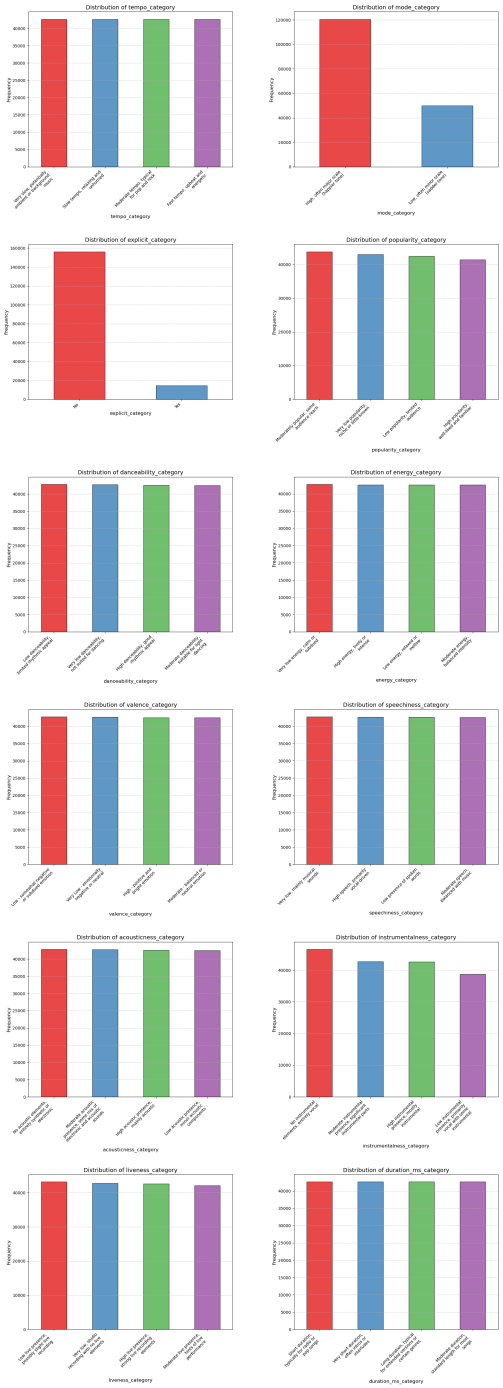


Figure 6: distribution of categorical features

this aligns with the objective of music recommendation which focuses more on recommending studio recorded songs suitable for wide range of audience.

Below shown pie chart shows the distribution of songs across two emotional categories(major and minor scale) based on their mode.

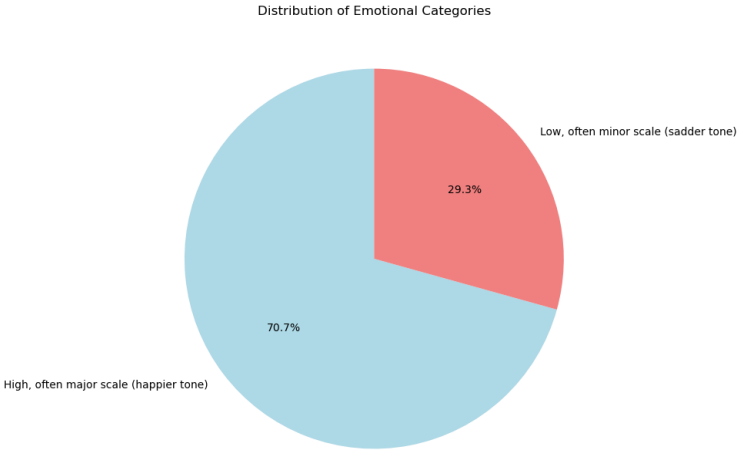


Figure 7: pie chart of emotional categories

From above pie chart , we can say that category 'High, often major scale(happier tone)' represents 70.7% of the songs. The major scale is typically associated with a happier or more uplifting mood, so a large portion of the dataset may be perceived as positive or cheerful.
The category 'Low, often minor scale (sadder tone)' represents 29.3% of the songs. The minor scale is often linked to a sadder or more melancholic tone, so this smaller portion of the dataset might have a more somber or introspective mood. This distribution indicates that the dataset is weighted towards songs with a "happier" emotional tone, which may reflect a preference for major key compositions in the dataset.

## 4.3 Bivariate Analysis - Numerical vs. Numerical Relationships

### 4.3.1 Correlation Analysis - Heatmap of Numerical Feature Correlations

The correlation heat map is generated for all numerical features as shown below (figure 8):

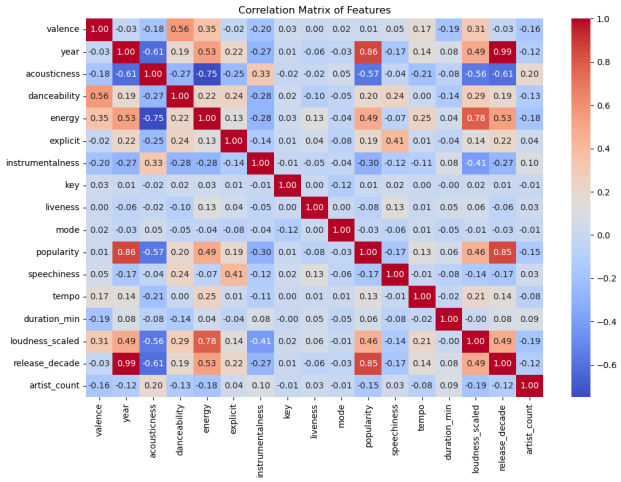


Figure 8: correlation heat map of all numerical features

We can see below observations from this correlation heatmap:

- **Valence and Danceability**: There is a moderate positive correlation between valence and danceability (0.56). This suggests that songs with a higher valence (happier or more positive mood) tend to also be more danceable. This makes sense, as upbeat, positive songs are often more suitable for dancing.
- **Year and Popularity**: There is a strong correlation between year and popularit (0.86) indicating that newer songs are much more likely to be popular in this dataset.
- **Energy and Loudness_scaled**: Energy and loudness have a strong positive correlation (0.78), which is expected as louder tracks often convey a higher energy level.
- **Acousticness and Energy**: Acousticness is negatively correlated with energy (-0.75), indicating that acoustic songs tend to have lower energy levels.
- **Acousticness and Year**: There is a moderate negative correlation between acousticness and year (-0.61). This suggests that newer songs tend to be less acoustic, possibly due to the increased use of electronic production techniques in recent music.

### 4.3.2 Heatmap - Audio Features by Popularity Category

This heatmap shows the average values of various audio features across different popularity categories as shown below(figure 9).

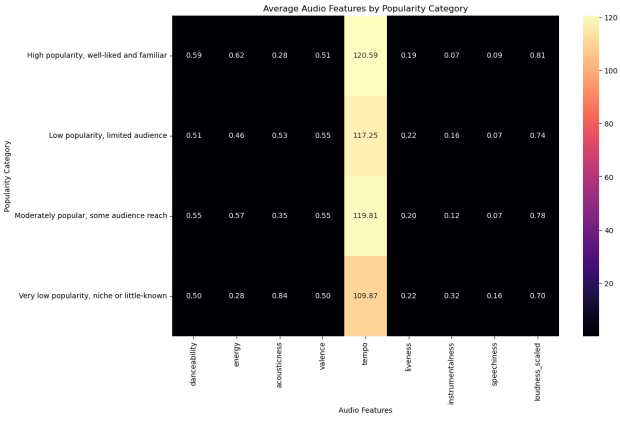


Figure 9: heat map of average audio features by popularity

We can see below observations from this heatmap:

- **Danceability and Energy:** : Popular tracks have higher danceability (0.59 vs. 0.50) and energy (0.62 vs. 0.28), indicating that upbeat and energetic songs are more likely to be popular.
- **Acousticness and Tempo:** : Popular songs are less acoustic (0.28 vs. 0.84) and tend to have a slightly faster tempo (120.59 BPM vs. 109.87 BPM), suggesting a preference for digitally enhanced, fast-paced tracks.

- **Loudness**: Higher loudness values (0.81 vs. 0.70) in popular songs indicate that dynamic and engaging tracks resonate better with listeners.

### 4.3.3 Violin Plot - Distribution of Popularity by Tempo Category

The violin plot displays the distribution of popularity across different tempo categories as shown below.

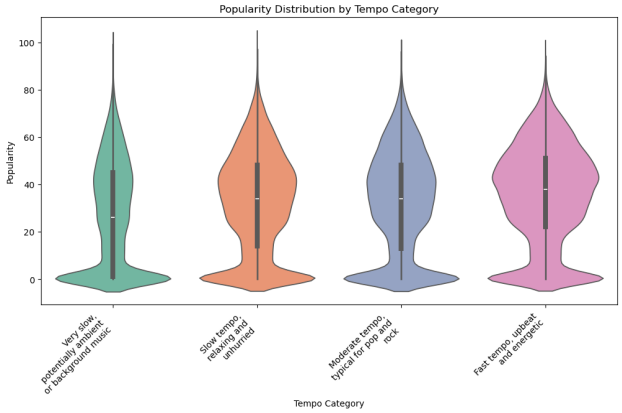


Figure 10: Violin plot of popularity across different tempo

Fast-tempo songs have a wide spread in popularity, with a slight concentration around moderate popularity levels. This shows that while fast, energetic music can reach high popularity, it also varies widely in its appeal. Overall, moderate and fast-tempo categories appear to align more with popular music, while slower tempos may have more specialized or selective appeal.

## 4.4 Trend analysis

Below shown graphs were plotted to observe music trend over the years.

- **Year vs Duration** : There was an increase in Duration in 1940s-1970s which reflects the influence of rock, album-oriented rock, and experimental genres that allowed for longer tracks.Later there was a decrease in Duration

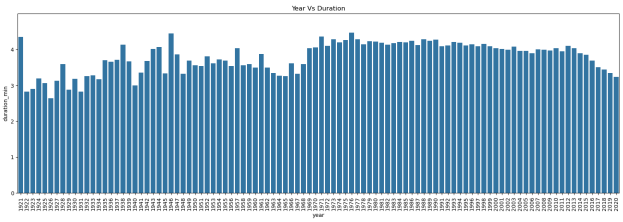


Figure 11: song duration across the years

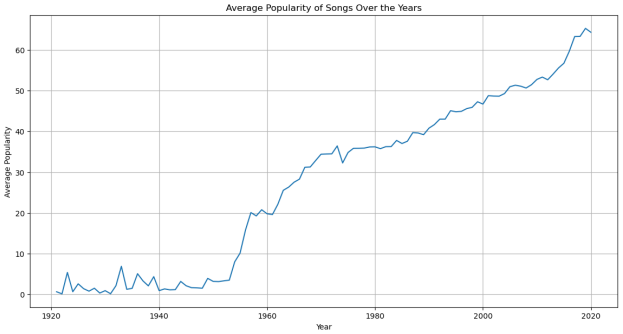


Figure 12: song popularity across the years

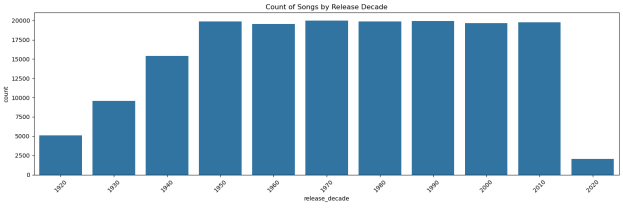


Figure 13: number of songs released across the years

1990s-2020s likely tied to the shift towards pop and streaming, where shorter tracks are more commercially viable.

- **Average Popularity of Songs Over the Years** : Song popularity has seen steady growth over time, with significant boosts during periods of technological advancement in music distribution and accessibility.i.e. between 1980s to 2020s.

- **Count of Songs by Release Decade** : The music industry has sustained high levels of song production from the 1950s onwards, with significant growth in earlier years as recording technology and distribution developed.In 2020s there is a noticeable drop in song count, likely due to incomplete data for this decade or the limited number of years covered within it so far.

## 4.5 Analysis based on artists

### 4.5.1 Distribution of artist category

Below distribution graph of artists category was plotted (figure 14).We observe that the dataset contains highest number of data samples from Solo tracks followed by duet, small group and choir.This indicates that audience is more aligned towards solo tracks.

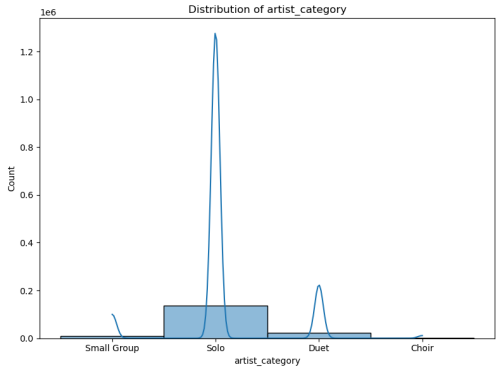


Figure 14: distribution of artist category

### 4.5.2 Average audio features by artist category

The average music features for different artist types was plotted as shown below(figure 15)

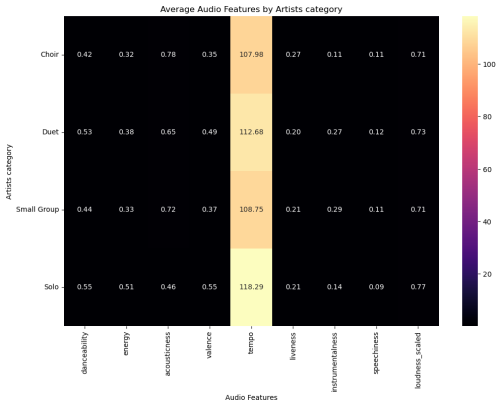


Figure 15: distribution of artist category

- **danceability and loudness_scaled** : The average danceability and loudness_scaled are almost the same across all artist categories.

- **energy and tempo** : The average energy and tempo are highest for solo tracks followed by duet.

- **acousticness and instrumentalness**:The average acousticness and instrumentalness is the highest in choir and small group.This is as expected as group songs usually have more acoustics present , unlike solo tracks.

# 5 Modeling

## 5.1 Optimal Cluster Detection Using the Elbow Method

Elbow Method is an effective technique used to determine the optimal number of clusters (K) using K-Means clustering algorithm. It examines the relationship between the number of clusters and the within-cluster sum of squares (WCSS) also known as

distortion score.

**U**sing this technique, we systematically experimented with different numbers of clusters (K) ranging from 1 to 100. With each K value, the distortion score is computed and plotted against K, the resulting graph resembles an elbow. The optimal number of clusters are determined for non scaled and scaled Genre data and data. Sharp decline in distortion score is observed until the elbow point followed by a gradual flattening.

Table 1: Optimal K Value

| **Type** | **Non Scaled** | **Scaled** |
|---|---|---|
| Genre Data.csv | 13 | 20 |
| Data.csv | 11 | 14 |

## 5.2 Train Test Split

In order to evaluate and compare the performances between different recommendation models, the train test split is applied to the data set. For all users, 80% of the songs they listened to are kept as a train set while the other 20% are used as a test set. Hence, we assume that these songs in the test set are not listened to by those users when training our model.

## 5.3 Clustering Algorithms

Clustering is a technique in unsupervised learning where data points are grouped together on the basis of their similarities, aiming to discover inherent patterns or structures within the data. Details of the types of clustering algorithms are provided in table 2:

Table 2: Types of Clustering algorithm

| **Type** | **Key Characteristics** | **Example Algorithms** |
|---|---|---|
| Partition-Based | Divides data into fixed clusters | K-Means, K-Medoids |
| Hierarchical | Builds a hierarchy of clusters (tree-like structure). | Agglomerative, Divisive |
| Density-Based | Groups dense regions, identifies outliers as noise. | Agglomerative, Divisive |
| Model-Based | Fits data to probabilistic models | GMM, Bayesian Clustering |

**I**n this work, we have implemented KMeans,KMediods,Agglomerative,DBSCAN and Gaussian mixture to analyze model selection.

**a. KMeans:** A distance-based algorithm that partitions the data into 'k' clusters. Divide the data into k groups by assigning points to the nearest cluster center and then updating the centers based on the mean of the points in each group. Repeat this process until the cluster assignments stop changing.

**b. KMedoids:** A clustering algorithm that minimizes the sum of dissimilarities between data points and a set of representative points (medoids), making it robust to noise and outliers.

**c. Agglomerative Clustering:** A hierarchical approach that merges data points iteratively. Group data by starting with each point as its own group and gradually merging the closest pairs into larger groups until everything is combined. It uses measures similarity to decide which points or groups to merge. The process creates a tree-like diagram called a dendrogram that shows how the clusters were formed.

**d. DBSCAN:** A density-based clustering method that identifies noise and irregular cluster shapes. Groups points based on their density, forming clusters if there are enough nearby points. It can handle noise (outliers) and detects clusters of different shapes.

**e. Gaussian Mixture Models:** Gaussian Mixture models data as a combination of multiple Gaussian (bell-shaped) distributions, with each representing a cluster. GMM assigns points to clusters probabilistically, making it ideal for overlapping clusters. It adjusts the shapes and sizes of clusters to fit the data.

## 5.4 Evaluation Metrics

The output in unsupervised learning is probabilistic and hence it is crucial to evaluate the model. In this work , we have used the following metrics to evaluate the performance of a model.

**a. Silhouette Score:** Measures how similar data points are within a cluster vs. other clusters (higher is better).[**?**]

**b. Davies-Bouldin Score:** Evaluates the compactness and separation of clusters (lower is better).

**c. Calinski-Harabasz Score:** Measures the ratio of between-cluster to within-cluster dispersion (higher is better). Additionally , Average Rank is calculated across the metrics to determine the overall best-performing algorithm.

## 5.5 Observations

Graphs have been plotted based on the performance metrics for implemented machine learning algorithms with Genre data and data.


Figure 16: Sihouette Score - Scaled Data

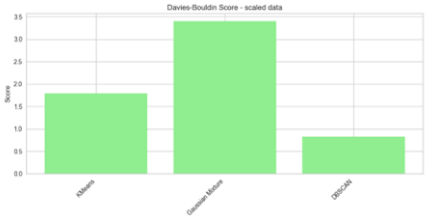


Figure 17: Genre data based Davies-Bouldin- Scaled Data

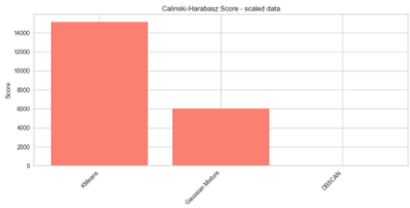


Figure 18: Genre data based Calinski-Harabasz Score - Scaled Data

KMeans is the best overall clustering algorithm, offering superior clustering quality and efficiency. Agglomerative Clustering serves as a strong alternative


Figure 19: Genre data based Sihouette Score -Non Scaled Data

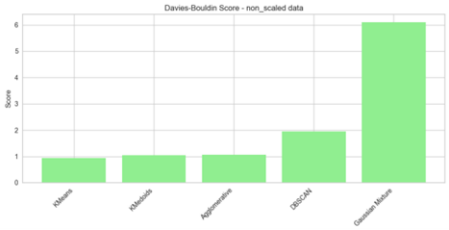


Figure 20: Genre data based Davies-Bouldin- Non Scaled Data

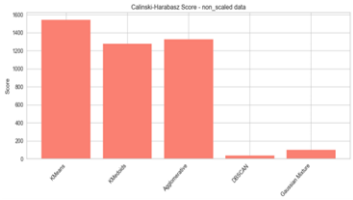


Figure 21: Genre data based Calinski-Harabasz Score - Non Scaled Data

with good performance and moderate execution time. While DBSCAN excels in handling highly separable clusters, it struggles with compactness and small datasets. KMedoids delivers consistent but moderate performance, though it is slower than KMeans. Gaussian Mixture is not recommended due to its poor metrics and high computational cost. It can also be observed that Non scaled data is performing better than scaled data.

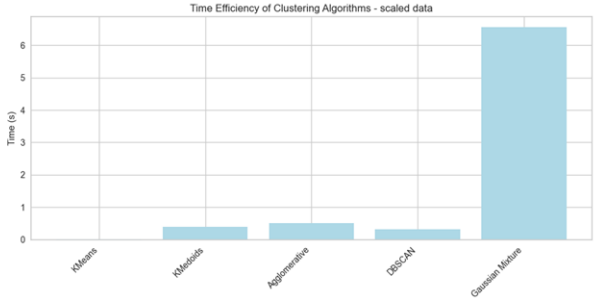


Figure 22: Genre data based time efficiency - Scaled Data


Figure 23: Genre data based time efficiency - Non Scaled Data

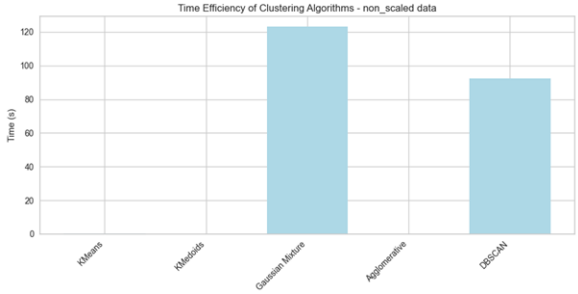


Figure 24: Data based time efficiency - Non Scaled Data

### 5.6 Hyperparameter Tuning Using Cross-validation

In this work, GridSearch with k-fold cross validation is implemented for each machine learning algorithm implemented. The best hyper-parameters or the approximate range in which they lie can be obtained using this technique. Value of "k" we considered is 3. When we pass training data to grid-search it implicitly implements k-fold cross validation. In k-fold cross-validation we divide the entire training set into k parts. Out of which (k-1) parts of the data are used for training and remaining one part is used for validation. At every iteration different (k-1) parts of the data are considered for training and the remaining one part is considered for validation. The best hyper parameter obatined are provided in table 3. The performance metric is Silhouette score,Davies-Bouldin Index and Calinski-Harabasz for K-Means clustering algorithm , KL Divergence loss for t-SNE and reconstruction error for PCA.

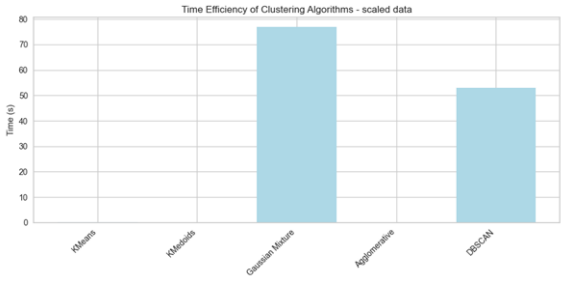


Figure 25: Data based time efficiency - Scaled Data

Table 3: Best parameters

| K-Means-Genre | K-Means-Data | t-SNE- Genre | PCA-Data |
|---|---|---|---|
| Init: k-means++ | Init: k-means++ | Early exaggeration:24 | PCA n components:2 % |
| Max iter:300 | Max iter:300 | Learning rate:200 | PCA svd solver:'auto' % |
| $N_clusters$ : 13 | N clusters:11 | Max iter:2000 | PCA tol:0.0001 % |
| N init:20 | N init:10 | Perplexity:50 | PCA whiten:True % |
| Tol:0.001 | Tol:0.0001 | | |

## 6 Model Training

Clustering is a fundamental technique in unsupervised machine learning, used to identify patterns within data by grouping similar data points together. The core objective of a clustering algorithm is to locate data points that share common characteristics, thus assigning them to the same cluster. Based on previous observation , Kmeans outperforms other algorithms. Therefore, KMeans algorithm is further

used for clustering by Genre and clustering by songs.

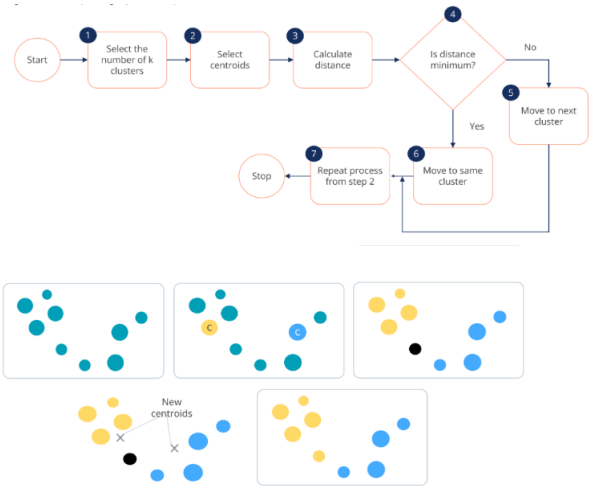


Figure 26: K-Mean Clustering Algorithm

## 6.1 Clustering by Genre

**K**-means clustering algorithm is used to divide the genres in this dataset into 13 clusters based on the numerical audio features of each genre.

t-SNE stands for T-Distributed Stochastic Neighborhood Embedding). It is an unsupervised non-linear dimensionality reduction technique for data exploration and visualizing high-dimensional data.This visualization helps to visually depict the clusters and patterns in data.The aggregation of data points in the graph indicates that they are close to each other in a high-dimensional space. The colour of each cluster represents a group of genres with similar characteristics. For example, it can be observed that a particular music genre is often located in a particular cluster, which represents similar musical characteristics or user preferences.

Based on the provided table, we can observe the following:

- Highly Populated Clusters are Clusters 4, 5, 8 and 9 dominate with the most genres, indicating these clusters represent the most common groupings or genres with significant overlap.

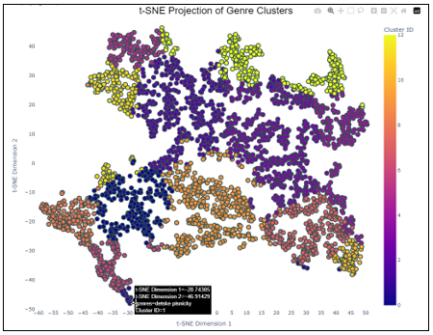


Figure 27: t-SNE Projection of Genre Clusters

- Sparse Clusters are Clusters 6 and 12 are outliers, with very few genres. These may represent niche or unique categories.

The uneven distribution across clusters highlights varying densities and potential differences in genre similarity.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| cluster | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| total_genres | 179 | 356 | 359 | 610 | 284 | 211 | 55 | 21 | 272 | 124 | 6 | 321 | 175 |

Figure 28: Genre Clustering

## 6.2 Clustering by Song

**P**CA stands for Principal Component Analysis (PCA). It is a technique for dimensionality reduction that transforms the data into a lower-dimensional space while retaining as much variance as possible. Reconstruction error (Mean squared error) is the metric used for evaluation. K-means algorithm for clustering divides the data into 11 clusters. The code fits the data, each song is assigned a clustering label, and the song is categorized into one of 11 clusters. The visualization depicts the data points after dimensionality reduction with PCA.Various Coloured dots represent songs assigned to different clusters.

A concrete example is provided in the image: a song called "Take a step back", located at coordinates (1.57,0.75)is grouped into cluster 5. In this scatter plot, the horizontal (x) and vertical (y) axes represent the two principal components transformed by PCA, which are the most important sources of variance in the original data.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| cluster_label | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| total_songs | 39151 | 26041 | 21076 | 8177 | 3166 | 11735 | 16333 | 21289 | 4958 | 17141 | 1586 |

Figure 29: Song Clustering

Based on the provided table, we can observe the following:

- Highly populated clusters are clusters 0, 1, and 2 dominate with the highest number of songs, suggesting that these clusters represent more general or widely distributed categories.
- Sparse clusters are clusters 4 and 10 which have very few songs, with cluster 10 being particularly sparse. These may represent niche or less common song categories.
- The varying song counts across clusters indicate diverse groupings, with some clusters being more populated, possibly reflecting more common genres or songs with overlapping features. Others are smaller, suggesting that these could represent unique or specialized song categories.

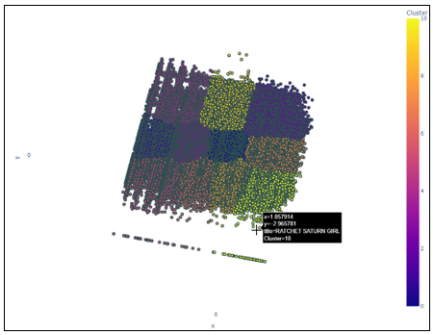

Figure 30: PCA Projection of Song Clusters

# 7 Model Deployment

## 7.1 Content based filtering

**I**n this work, Content based filtering technique with KMeans clustering algorithm is implemented to identify the inherent patterns within non scaled numeric features of playlist tracks .Content based filtering technique works under the assumption that users will like songs that have similar characteristics.

## 7.2 Cosine similarity:

**A** cosine similarity is a value that is bound by a constrained range of 0 and 1. The closer the value is to 0 means that the two vectors are orthogonal or perpendicular to each other. When the value is closer to one, it means the angle is smaller and the songs are more similar.

## 7.3 Recommendation System

**I**n this work to start with, K-means is performed on the dataset to identify distinct clusters of songs/genre based on various features like danceability, tempo, duration etc. Then for each user in the train set, top 2 clusters are identified that contain the most number of songs they have liked before, and selected songs closest to the cluster center as the recommendation choices. Then cosine distances is calculated between the selected song's features and all other songs' features. It identifies selected songs closest to the cluster center as the recommendation choices. It also maps the cluster labels of the recommended songs to their corresponding genres.

Recommendation function is designed to provide users with personalized song recommendations based on their input. It seamlessly utilizes numerical data to create a solid foundation for generating accurate suggestions. Users can interact with drop-downs to select songs, artists, or genres for receiving recommendations .Three drop-downs are created for the user to choose and receive recommendation:

- All Songs: Displays a list of all songs.

- By Artist: Displays songs filtered by the artist.
- By Genre: Displays songs filtered by genre.

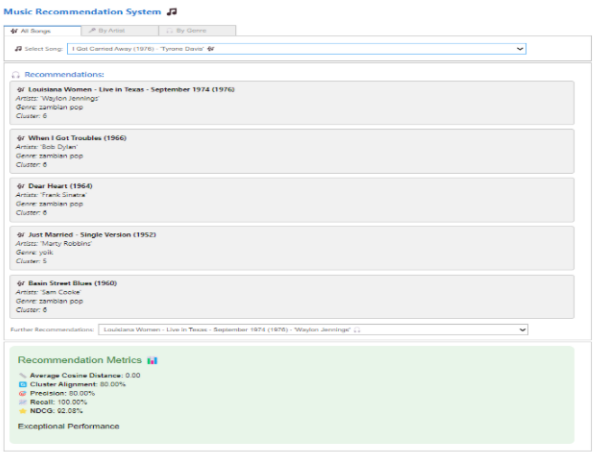


Figure 31: Recommendation System

### 7.4 Evaluation metrics

- **Average Cosine Distance:** Average Cosine Distance: Measures the average cosine similarity between items in a dataset, with a value of 0 indicating perfect similarity
- **Cluster Alignment:** Indicates the percentage of data points that are correctly grouped into their respective clusters as per the model's predictions.
- **Precision:** The proportion of positive identifications that were correct.
- **Recall:** The proportion of actual positives that were correctly identified.
- **NDCG (Normalized Discounted Cumulative Gain):** A measure of ranking quality, assessing the effectiveness of a recommendation system by weighing the ranks of relevant items.

## 8 Conclusion

In this work, Firstly, we introduced music recommendation background and rigorously performed EDA on spotify dataset[1] [2] [3] [4] [5] [6] Next we explored application of various machine learning techniques in the music recommendation system. Anlyzed 5 different unsupervised clustering algorithms for modeling. Opted K means algorithm to cluster songs into different groups with common musical characteristics and recommending them to users.Built effective music recommendation system using content based filtering and cosine similarity. t-SNE and PCA are the dimensionality reduction algorithms used for visualization.

## 9 Future Scope

- Real Time Processing - Build user profiles, integrate real-time feedback (likes/dislikes) and apply collaborative filtering for more accurate, personalized suggestions.
- Advanced Models - Implement Hybrid models combining collaborative, content-based and context-aware filtering for better recommendations.
- Enhanced Features - Incorporate audio features, sentiment analysis of lyrics and temporal trends better recommendations.

## References

[1] Shashank Bangera, Vaishnavi Nagaonkar, Aditya Tiwari, Saud Ansari, and Kanchan Talekar. Spotify recommendation system. In *International Research Journal of Modernization in Engineering Technology and Science*, 02 2024.

[2] Xinyue Li. Analysis of machine learning-based music recommendation system using spotify datasets. *Applied and Computational Engineering*, 77:49–55, 07 2024.

[3] Tie-min Ma, Xue Wang, Fu-cai Zhou, and Shuang Wang. Research on diversity and accuracy of the recommendation system based on multi-objective optimization. *Neural Computing and Applications*, 35(7):5155–5163, 2023.

[4] Shari Trewin. Knowledge-based recommender systems. *Encyclopedia of library and information science*, 69(Supplement 32):180, 2000.

[5] Makarand Velankar and Parag Kulkarni. *Music Recommendation Systems: Overview and Challenges*, pages 51–69. Springer International Publishing, Cham, 2023.

[6] De-Jia Zhang. An optimized item-based collaborative filtering recommendation algorithm based on item genre prediction. In *PIAGENG 2009: Intelligent Information, Control, and Communication Technology for Agricultural Engineering*, volume 7490, pages 342–346. SPIE, 2009.