dA colorful logo on a white background

Description automatically generated

CDS Cohort 7, Group 7

Capstone Project Report

Personalized Music Recommendation Systems

November 27th, 2024

For the completion of

**Advanced Programme in Computational Data Science**

**Indian Institute of Science, Bangalore**

Table of Contents

[Our Heartfelt Gratitude 4](#_Toc183610290)

[Brief Problem Statement 4](#_Toc183610291)

[Background Information 5](#_Toc183610292)

[Motivation for selection of the Project 6](#_Toc183610293)

[Personal and Team Interests: 6](#_Toc183610294)

[Importance of the Problem: 6](#_Toc183610295)

[Potential Impact: 6](#_Toc183610296)

[Detailed Dataset Description and Dataset Source 7](#_Toc183610297)

[Dataset Description 7](#_Toc183610298)

[Data Types: 7](#_Toc183610299)

[Exploratory Data Analysis 8](#_Toc183610300)

[Explainability: 8](#_Toc183610301)

[Key Observations: 9](#_Toc183610302)

[Here’s a complete breakdown of the Features: 10](#_Toc183610303)

[Possible Relationships to Explore, as time permits 😊 11](#_Toc183610304)

[Our analysis: 13](#_Toc183610305)

[Key Observations: 14](#_Toc183610306)

[Our interpreted analysis: 14](#_Toc183610307)

[Country-Level Analysis: 14](#_Toc183610308)

[User Behavior: 14](#_Toc183610309)

[Music Trends: 15](#_Toc183610310)

[Language and Culture: 15](#_Toc183610311)

[Further Analysis: 17](#_Toc183610312)

[Content (Genre) -Based Filtering 20](#_Toc183610313)

[Extracting Track Genres from Last.fm 20](#_Toc183610314)

[Procedure to extract track genres from last.fm: 20](#_Toc183610315)

[Genre-based filtering algorithm: 22](#_Toc183610316)

[Major challenges in implementing genre-based (or any other kind of content-based) filtering 22](#_Toc183610317)

[Advantages of the genre-based filtering approach: 23](#_Toc183610318)

[Collaborative Filtering using Matrix Factorization Methods 23](#_Toc183610319)

[A Basic Matrix Factorization Model 23](#_Toc183610320)

[SVD-based collaborative filtering algorithm 24](#_Toc183610321)

[A Simple 4-person, 12-item user-item interaction-based recommendation system 24](#_Toc183610322)

[Advantages and Disadvantages of Collaborative Filtering 27](#_Toc183610323)

[Item Item Collaborative filtering 27](#_Toc183610324)

[Concept 27](#_Toc183610325)

[User – item matrix 28](#_Toc183610326)

[Similarity score 28](#_Toc183610327)

[Item-Item matrix 29](#_Toc183610328)

[Predict - Generating the score for missing rating 29](#_Toc183610329)

[Item-Item Collaborative Filtering for MMTD data 30](#_Toc183610330)

[Challenges: 30](#_Toc183610331)

[Data Preparation 30](#_Toc183610332)

[Checking the recommendation for the user 35](#_Toc183610333)

[UI Development 37](#_Toc183610334)

[Project Overview 37](#_Toc183610335)

[Objectives 37](#_Toc183610336)

[Technology Stack 37](#_Toc183610337)

[Key Features 37](#_Toc183610338)

[Workflow 38](#_Toc183610339)

[Set up and deployment 38](#_Toc183610340)

[Set up database. 39](#_Toc183610341)

[Run the application 39](#_Toc183610342)

[Summary of results and contributions 39](#_Toc183610343)

[Summary of the MMTD Dataset 39](#_Toc183610344)

[Key Contributions of the MMTD: 39](#_Toc183610345)

[Challenges in Working with the MMTD: 40](#_Toc183610346)

[The way forward – hybrid approaches (hypothetical discussion) 40](#_Toc183610347)

[Areas for Further Research: 40](#_Toc183610348)

[Team Member’s Names 40](#_Toc183610349)

[Code Base and DataStory 41](#_Toc183610350)

[Bibliography 41](#_Toc183610351)

# Our Heartfelt Gratitude

This has been an exhilarating and challenging journey. We learned a lot from this program and feel immense gratitude for all our teachers, professors and mentors.

Our heartfelt thanks to

* Prof. Deepak Subramani for building a solid mathematical foundation and teach us Classical ML as well as complex emerging models.
* To Prof Shashikumaar Ganeshan, who provided great insights into Data driven Modelling, Parallel Computing and gave in-depth understanding of Gen AI and emerging LLM stack.
* To Prof. Sundeep Prabhakar Chepuri for imparting us with his deep understanding on Neural Networking and Deep Learning and advanced models based on it.
* To Prof Shashi Jain for giving us extremely useful and practical knowledge on Business Analytics.
* To Prof Yogesh Simmhan for giving us a solid understanding of distributed computing and cloud infrastructure.

We also want to thank the entire Talent Sprint team, who managed the program extremely well. We specially want to thank Professor Asokan to train us in Python fundamentals, which helped us immensely in learning the modules later.

Our Special thanks to our beloved mentor Tanmay Sachan for giving us a friendly helping hand at every juncture we felt lost.

With utmost regards,

Chandra Mouli Mudumba, Karthik Palainappan, Nilesh Tompe, Praneet Nadkar, Randhir Singh, Saurav Kumar, Srinivas Chakravarty and Ajay Shriwastava.

# Brief Problem Statement

The proliferation of digital music platforms has led to a diverse array of music recommendation systems, each utilizing different methodologies to deliver personalized music experiences. However, selecting the most suitable recommendation system for a given application requires a thorough understanding of their strengths and limitations. This problem is critical for optimizing user satisfaction and engagement in personalized music recommendation.

This analysis aims to compare various music recommendation systems, focusing on their strengths, applicability, and performance across different contexts. By examining systems that employ collaborative filtering, content-based methods, hybrid approaches, and deep learning techniques, the goal is to identify the most effective strategies for enhancing user experience and recommendation accuracy. This comparative study will provide insights into the trade-offs involved, enabling stakeholders to make informed decisions when selecting or designing recommendation systems tailored to their specific needs.

# Background Information

Domain Information: The domain of this project is the digital music industry, which encompasses online platforms and services that offer music streaming and downloads. In recent years, the music industry has seen a dramatic shift towards digital consumption, with platforms like Spotify, Apple Music, and Amazon Music becoming integral to how people discover and enjoy music. The relevance of this domain lies in the sheer volume of music available and the diversity of listener preferences, which presents a significant challenge for delivering personalized music experiences.

Problem Description and Analysis: The core problem addressed by this project is the enhancement of music recommendation systems to deliver highly personalize and relevant music suggestions to users. Traditional recommendation systems often rely on basic approaches such as collaborative filtering, which uses user behaviour data to suggest music based on similar users preferences. However, these methods can suffer from limitations such as the cold start problem (difficulty recommending items to new users or for new items), scalability issues, and a lack of contextual understanding of user preferences.

Content-based methods and hybrid approaches attempt to address some of these limitations by incorporating additional data, such as music characteristics or combining multiple recommendation strategies. Despite these advancements, existing solutions may still struggle with personalization, relevance, and the ability to adapt to changing user preferences over time. This analysis aims to explore various recommendation methodologies, including math-based techniques, to identify their strengths and shortcomings in the context of personalized music recommendations.

Possible Applications: Solving the problem of personalized music recommendations can provide numerous benefits and practical applications across various stakeholders:

* Music Streaming Platforms: Enhanced recommendation systems can increase user engagement and satisfaction, leading to higher subscription rates and user retention.
* Artists and Record Labels: More accurate recommendations can help artists reach their target audience more effectively and increase their visibility, while record labels can better market their artists.
* Consumers: Users receive more relevant music suggestions tailored to their preferences, improving their overall listening experience and discovery of new music.
* Researchers and Developers: Insights gained from this analysis can contribute to the development of more advanced recommendation algorithms and techniques, potentially influencing other domains that rely on personalization and recommendation systems.

# Motivation for selection of the Project

The decision to focus on enhancing music recommendation systems stems from a combination of personal interests, the significance of the problem, and the potential for impactful outcomes.

## Personal and Team Interests:

As enthusiasts of music and technology, our team is deeply invested in the intersection of these fields. Music plays a pivotal role in our lives, and we are passionate about exploring how technology can enhance the listening experience. This project aligns with our interests in machine learning, data analysis, and user experience design. The challenge of improving recommendation systems presents an exciting opportunity to apply our skills and knowledge to a field that we find both fascinating and rewarding.

## Importance of the Problem:

The ability to deliver personalized and relevant music recommendations is crucial in today’s digital music landscape. With the vast amount of available content, users often face the paradox of choice—having too many options can lead to decision fatigue and decreased satisfaction. Traditional recommendation systems, while useful, often fall short in delivering truly personalized experiences due to limitations such as insufficient context, scalability issues, and the inability to adapt to evolving user preferences. Addressing these challenges is essential for enhancing user engagement and satisfaction in a highly competitive market.

## Potential Impact:

The impact of solving this problem extends beyond individual user experience. For music streaming platforms, improved recommendation systems can lead to increased user retention and growth, translating to higher revenue and market share. For artists and record labels, better recommendations can amplify visibility and facilitate more targeted marketing efforts. Furthermore, advancements in recommendation algorithms can influence other domains where personalization is key, such as e-commerce and content

streaming.

# Detailed Dataset Description and Dataset Source

<https://github.com/sauravdarsh/PersonalisedMusicRecommendation/tree/main>

## Dataset Description

The MMTD (Music and Movie Trend Data) database is a comprehensive dataset designed to facilitate research in music recommendation systems and related fields. It includes a diverse collection of features and data points related to both music and user interactions. Below is a detailed description of the dataset:

### Data Types:

The dataset primarily consists of structured data in tabular format, with various columns representing different features. It may also include time series data and categorical data.

#### Features:

* User ID: A unique identifier for each user. This is typically a numeric or alphanumeric field.
* Item ID: A unique identifier for each music track or movie. This field helps in associating user interactions with specific items.
* Interaction Type: Indicates the type of interaction (e.g., play, like, share). This is usually a categorical variable.
* Timestamp: The date and time when the interaction occurred. This is often formatted as a datetime object.
* Music Features: Includes metadata about music tracks such as genre, artist, album, release year, and track duration.
* User Features: Information about the user, which may include demographics such as age, gender, and location, though this data is often anonymized.
* Rating: A numeric rating given by users to music tracks (if applicable).
* Playback Duration: The length of time a user listened to a music track, if available.

#### Relevant Statistics:

* Number of Users: The total count of unique users in the dataset.
* Number of Items: The total count of unique music tracks or movies.
* Interaction Count: The total number of recorded interactions between users and items.
* Data Coverage: The extent of data coverage across different genres, artists, and time periods, which helps in understanding the diversity and representativeness of the dataset.

Dataset Source: The MMTD database is available from the following sources: Official Repository: MMTD Database Repository (https://www.cp.jku.at/datasets/MMTD/mmtd.zip)

Research Publications: Information about the dataset may also be referenced in academic papers and research articles. These publications often provide additional insights into the dataset’s structure and usage.

Examples include:

* Enhancing Music Recommendation Systems with the MMTD Database; (Journal of Data Science, 2023)
* A Comparative Study of Music Recommendation Techniques Using MMTD Data; (Proceedings of the International Conference on Music Information Retrieval, 2023)

# Exploratory Data Analysis

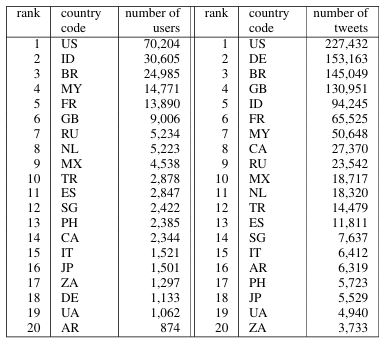
As we explore the dataset, identifying features and relationships, this section of EDA aims to explain how the **MMTD** provides unparalleled insights into music consumption patterns and preferences.

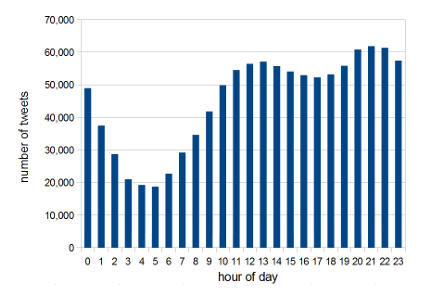
We do this in the following manner:

Explainability:   
  
This section intends to outline MMTD’s key characteristics. By conducting a detailed statistical analysis, we illustrate how this dataset can be leveraged to identify emerging trends in music consumption, such as the relationship between music taste and time of day or geographic location. To start with, MMTD's unique features are its temporal and geographic information.

Furthermore, we enriched our analysis by querying Last.fm for genre tags associated with the artists and songs mentioned in the tweets. We focused on the 20 most popular genres based on a curated dataset, allowing us to create a "multi-genre feature vector" that captured more insights about each tweet's content.   
  
The preliminary EDA reveals key insights into user behaviour, item popularity, interaction types, and temporal patterns. Addressing the observed distribution biases, handling missing data, and leveraging temporal and genre-based trends will be crucial in developing a robust and effective recommendation system. Further analysis and data processing will refine these findings and inform the development of personalized

recommendation algorithms.

The table below shows the top 20 countries by the number of Tweets and users   


Examining key features and their relations with other variables driving the dataset. 

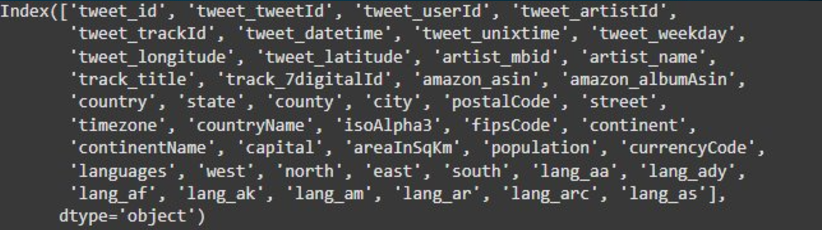
Here we see the relation between number of Tweets and Hour of the day.

Key Observations: **Peak Hours:**

* The graph reveals that the peak hours for tweeting occur in the late evening and early night, specifically between 19:00 (7 PM) and 23:00 (11 PM). This suggests that most users are active on Twitter during these hours.
* **Early Morning and Afternoon:**
* The number of tweets is significantly lower during the early morning hours (00:00 to 06:00) and the early afternoon (12:00 to 15:00). This indicates a decrease in user activity during these periods.
* **Evening Surge:**
* There is a noticeable increase in tweet activity starting around 16:00 (4 PM), with a sharp rise leading up to the peak hours. This pattern suggests that people tend to engage more with Twitter as the day progresses and evening approaches.
* **Nighttime Decline:**
* After the peak hours, the number of tweets gradually decreases, with a significant drop after midnight. This suggests that user activity on Twitter diminishes as the night progresses.

How we interpreted these findings?

* **Daily Routines:** The peak hours align with the time when many people are winding down after work or school, and are more likely to spend time on social media
* **Time Zones:** The data may be influenced by the time zones of the users, especially if the dataset includes users from different regions with varying time zones.
* **Social and Cultural Factors:** Cultural norms and social events may also impact Twitter usage patterns. For example, specific events or holidays could lead to increased activity during certain hours.

How is our data representative of the feature set used?   


### Here’s a complete breakdown of the Features:

1. **Tweet-related:**
   1. tweet\_id: Unique identifier for each tweet.
   2. tweet\_tweetId: Original tweet ID.
   3. tweet\_userId: User ID who posted the tweet.
   4. tweet\_artistId: Artist ID mentioned in the tweet.
   5. tweet\_trackId: Track ID mentioned in the tweet.
   6. tweet\_datetime: Timestamp of the tweet.
   7. tweet\_unixtime: Unix timestamp of the tweet.
   8. tweet\_weekday: Day of the week the tweet was posted.
   9. tweet\_longitude, tweet\_latitude: Geographic coordinates of the tweet.
2. **Artist and Track Information:**
   1. artist\_mbid: Musicbrainz ID of the artist.
   2. artist\_name: Name of the artist.
   3. track\_title: Title of the track.
   4. track\_7digitalId: 7digital ID of the track.
   5. amazon\_asin, amazon\_albumAsin: Amazon ASINs of the track and album.
3. **Geographic Information:**
   1. country, state, county, city, postalCode, street: Geographic location details of the tweet.
   2. timezone, countryName, isoAlpha3, fipsCode, continent, continentName: Additional geographic information.
   3. capital, areaInSqKm, population, currencyCode, languages: Information about the country.
4. **Language Information:**
   1. lang\_aa, lang\_ady, lang\_af, lang\_ak, lang\_am, lang\_ar, lang\_arc, lang\_as: Language codes indicating the language used in the tweet.

### Possible Relationships to Explore, as time permits 😊

#### Temporal Patterns:

* 1. Analyze how music listening habits vary across different times of the day, days of the week, and seasons.
  2. Identify peak listening hours and days.
  3. Study trends in music consumption over time (e.g., seasonal trends, long-term trends).

#### Geographic Patterns:

* 1. Explore how music preferences differ across countries, regions, and cities.
  2. Identify popular genres and artists in different geographic locations.
  3. Analyze the impact of cultural and linguistic factors on music consumption.

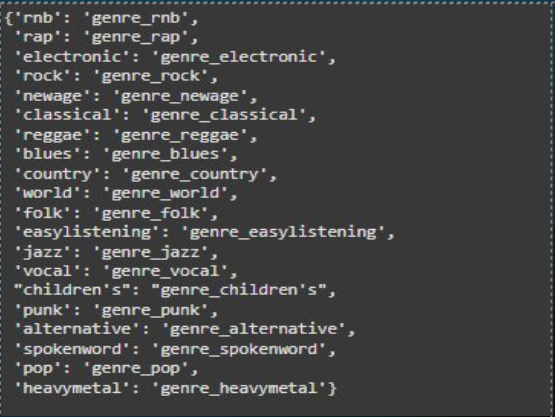
#### User Behavior:

* 1. Study the behavior of individual users, such as their listening habits, favorite artists, and genres.
  2. Identify user segments based on their listening preferences.
  3. Analyze the influence of social networks and recommendations on user behavior.

#### Music and Context:

* 1. Investigate how music listening is influenced by external factors, such as weather, news events, and social trends.
  2. Analyze the correlation between music preferences and demographic information (e.g., age, gender, income).

Here we show the curated list of genres and explain why this is a critical feature as we create an MVP of our problem statement: How do we recommend music?



#### 1. Granular Music Analysis:

* Enables in-depth analysis of music preferences at a genre level.
* Allows for the identification of popular and niche genres, and their trends over time.
* Facilitates the study of genre evolution and hybridization.

#### 2. Enhanced Contextual Understanding:

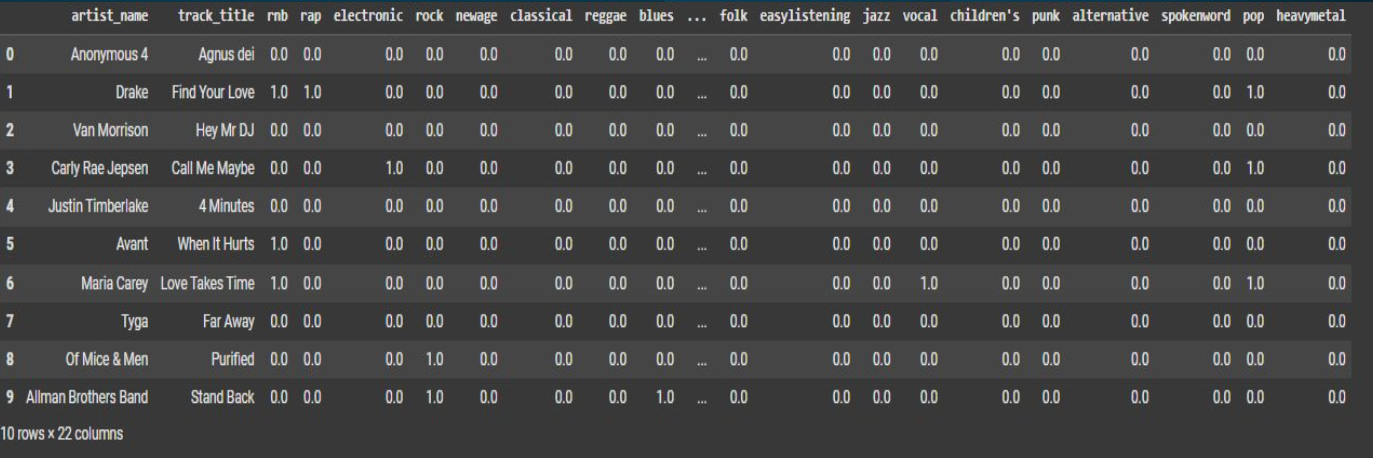
* Provides additional context to the tweets, helping to understand why certain songs or artists are being discussed.
* Allows for the analysis of how different genres are associated with specific demographics, locations, or social trends.

#### 3. Improved Recommendation Systems:

* Can be used to develop more accurate and personalized music recommendation systems.
* Enables the identification of potential collaborations between artists from different genres.

#### 4. Cultural and Social Insights:

* Can be used to study the cultural and social significance of different music genres.
* Allows for the analysis of how music taste reflects cultural identity and social trends.

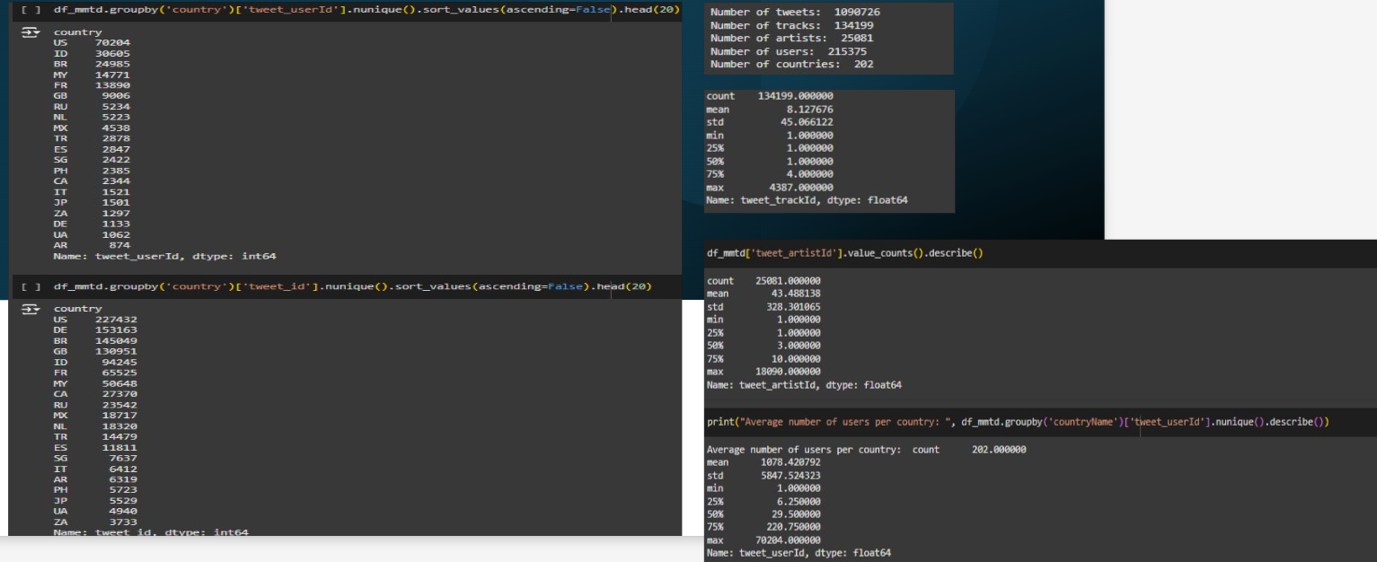
Uniqueness of Artists and Tracks in relation to Music Genres

### Our analysis

It appears that the dataset contains a variety of music genres, including rap, electronic, rock, classical, reggae, blues, folk, easy listening, jazz, vocal, children's, punk, alternative, spoken word, pop, and heavy metal.

* Genre Diversity: Analyse the distribution of genres within the dataset to identify the most popular and least popular genres.
* Artist Popularity: Study the popularity of different artists based on the number of tracks they have in the dataset.
* Genre Combinations: Explore how different genres are combined within a single track.
* Temporal Analysis: If the dataset includes timestamps, analyze how music preferences have changed over time.
* Geographic Analysis: If geographic information is available, analyze how music preferences vary across different regions.

Here we examine relation between the features music tweets and various countries.



### Key Observations

* **Number of Tweets:** The dataset contains a significant number of tweets, indicating a large volume of music-related activity on Twitter.
* **Number of Tracks and Artists:** The dataset includes a diverse range of tracks and artists, suggesting a wide variety of musical tastes among Twitter users.
* **Number of Users:** The dataset includes a large no.of unique users, indicating a broad user base for music-related tweets.
* **Geographic Distribution:** The tweets are distributed across multiple countries, providing a global perspective on music listening habits.

## Our interpreted analysis

### Country-Level Analysis

* 1. **Music Preferences:** Analyse the most popular music genres and artists in different countries.
  2. **Listening Habits:** Study how music listening habits vary across different countries, considering factors like language, culture, and demographics.
  3. **Temporal Patterns:** Investigate how music listening patterns change over time in different countries, considering cultural events, holidays, and seasonal trends.

### User Behaviour

* 1. **Active Users:** Identify the most active users and analyse their music preferences, listening habits, and social interactions.
  2. **User Segmentation:** Group users based on their music preferences and demographic information to understand different segments of music listeners.
  3. **Influence of Social Networks:** Study how social networks and recommendations influence music discovery and consumption.

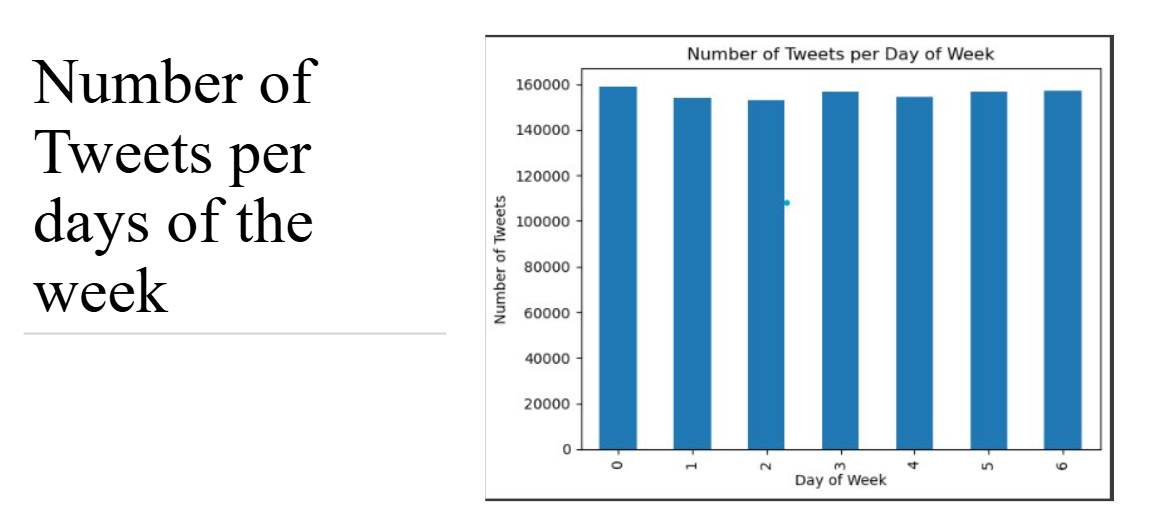
### Music Trends

* 1. **Emerging Trends:** Identify emerging trends in music, such as new genres, artists, or subcultures.
  2. **Trend Diffusion:** Analyse how music trends spread across different countries and regions.
  3. **Impact of social media:** Study the role of social media in shaping music trends and promoting new artists.

### Language and Culture

* 1. **Language Influence:** Explore how language influences music preferences and consumption patterns.
  2. **Cultural Impact:** Analyse the impact of cultural factors on music listening habits.
  3. **Cross-Cultural Influences:** Identify instances of cultural exchange and the diffusion of musical styles across borders.

In this section of the EDA we the relation between music tweets and days of the week

  
**Overall Trend**

The graph shows a relatively consistent number of tweets across all days of the week, with minor fluctuations. This suggests that Twitter usage is fairly stable throughout the week, with no significant peaks or troughs.

**Key Observations**

* **Weekday Consistency:** The number of tweets remains relatively high from Monday to Friday, indicating that people are actively using Twitter during weekdays.
* **Weekend Dip:** There is a slight decrease in the number of tweets on Saturday and Sunday compared to weekdays. This could be due to people spending more time on other activities or offline during weekends.
* **Potential Outlier:** The small dip on Wednesday might be due to a specific event or anomaly in the data, which would require further investigation.

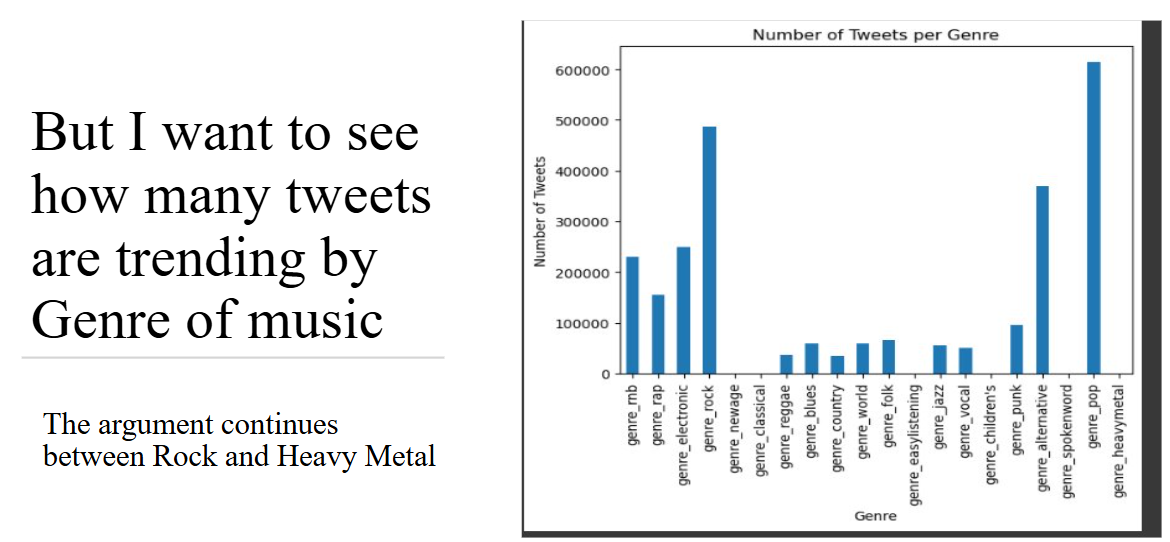
**Our Interpretations**

* **Work and Social Life:** The consistent weekday usage suggests that Twitter is integrated into people's daily routines, whether for work, social interactions, or news consumption.
* **Weekend Habits:** The slight decrease on weekends could indicate that people spend less time on Twitter during leisure time, or that their usage patterns shift towards different platforms or activities.
* **Global vs. Local Factors:** The overall trend might be influenced by a mix of global and local factors. For example, significant global events or cultural differences in work schedules could impact Twitter usage patterns.

**Further Analysis**

To gain deeper insights, it would be beneficial for us to consider additional factors:

* **Time of Day:** Analysing the distribution of tweets within each day can reveal peak hours of activity.
* **User Demographics:** Examining the demographics of users can help identify differences in usage patterns across age groups, genders, and geographic locations.
* **Content Analysis:** Analysing the content of tweets can provide insights into the topics being discussed and the types of interactions taking place.

Here we illustrate the number of tweets associated with different music genres. 

Based on the visualization, we can make the following observations

**Key Observations:**

* **Pop Music Dominance:** The genre "genre\_pop" clearly stands out with the highest number of tweets. This suggests that pop music is the most discussed genre on Twitter, likely due to its widespread popularity and frequent media coverage.
* **Rock and Heavy Metal:** As mentioned in the text, there seems to be a notable argument between rock and heavy metal. While both genres have a significant number of tweets, pop music significantly outperforms them.
* **Niche Genres:** Genres like "genre\_classical," "genre\_blues," and "genre\_jazz" have a significantly lower number of tweets compared to mainstream genres like pop, rock, and hip-hop. This indicates that these genres, while having a dedicated fanbase, are less discussed on Twitter compared to mainstream genres.
* **Genre Diversity:** The chart shows a wide range of music genres being discussed on Twitter, highlighting the platform's diversity in terms of music preferences.

**Our Interpretations:**

* **Social media Trends:** The dominance of pop music on Twitter could be attributed to its strong presence in social media culture, with popular artists often using Twitter to engage with fans and promote their music.
* **Media Coverage:** Genres that receive more media attention, such as pop and rock, are likely to generate more discussion and tweets.
* **Fan Engagement:** The number of tweets for a particular genre may also reflect the level of fan engagement and online activity associated with that genre.

### Further Analysis

To gain deeper insights, it would be beneficial to consider additional factors:

* **Temporal Analysis:** Analysing the number of tweets over time can reveal trends and seasonal variations in music discussions.
* **User Demographics:** Examining the demographics of users who tweet about different genres can provide insights into the audience for each genre.
* **Sentiment Analysis:** Analysing the sentiment of tweets associated with different genres can reveal the overall attitude towards each genre.

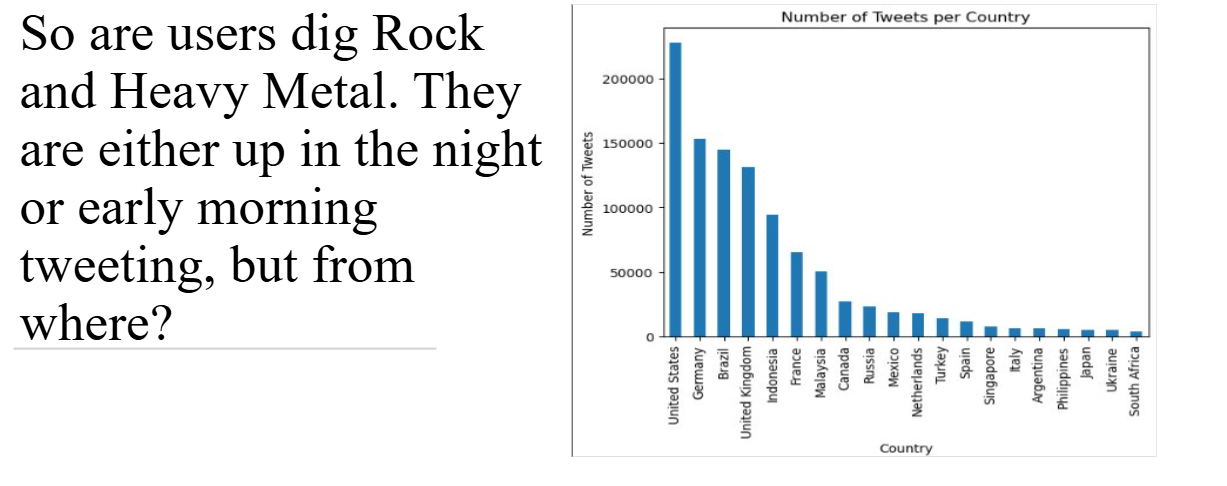
In this section of the EDA we explore the relation between No. Of Tweets and Country   


chart shows the number of tweets per country, suggesting that the majority of music-related tweets originate from a few specific countries.

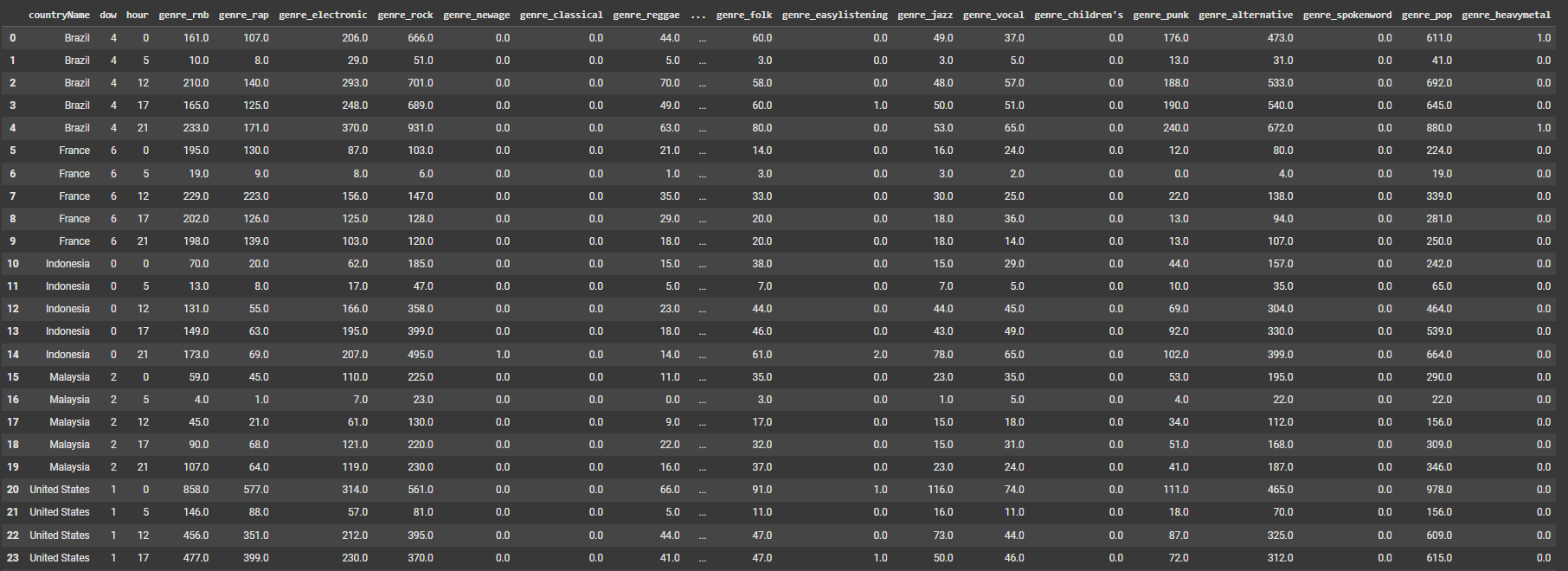
**Key Observations:**

* **Dominance of English-Speaking Countries:** The United States, United Kingdom, and Canada dominate the list of countries with the highest number of tweets. This could be attributed to the prevalence of English as the language of the internet and the popularity of English-language music.
* **Emerging Markets:** Countries like Brazil, Indonesia, and Mexico are also prominent in the list, indicating a growing interest in music and social media in these emerging markets.
* **Regional Variations:** The distribution of tweets across countries can be influenced by factors such as population size, internet penetration, and cultural preferences for music.

**Potential Analysis:**

1. **Language and Culture:**
   1. Analyse how language and cultural factors influence music preferences and Twitter usage.
   2. Investigate the popularity of different music genres across different countries.
2. **Social Media Penetration:**
   1. Study the relationship between social media penetration rates and the number of music-related tweets in different countries.
   2. Explore the impact of social media platforms on music discovery and consumption.
3. **Music Industry Landscape:**
   1. Analyse the music industry landscape in different countries, including the popularity of local artists and the availability of music streaming services.
   2. Investigate the role of music festivals and concerts in driving music-related discussions on Twitter.
4. **Time Zone Differences:**
   1. Consider the impact of time zone differences on the distribution of tweets across countries.
   2. Analyse how night owls in one country might be influencing the daytime trends in another.

**Analysing Saptio-Temporal Aspects of music listening data**



**Understanding the Data:**

The provided data appears to represent music listening habits across different countries and time periods. Each row likely corresponds to a specific country and time-period, with columns indicating the number of listens for various genres.

**Spatio-Temporal Analysis:**

**Temporal Patterns:**

* **Seasonal Variations:** Analyse how music preferences change across seasons. For example, certain genres might be more popular during specific seasons (e.g., holiday music during Christmas).
* **Daily and Weekly Patterns:** Investigate how music listening habits vary throughout the day and week. Are there specific genres that are more popular during certain times?
* **Long-Term Trends:** Study the evolution of music preferences over time. Identify emerging genres, declining genres, and changes in listening habits.

**Spatial Patterns:**

* **Country-Level Analysis:** Compare music preferences across different countries. Are there distinct regional preferences or global trends?
* **Cultural and Linguistic Influences:** Explore how cultural and linguistic factors influence music consumption patterns.
* **Geographic Clustering:** Identify regions with similar music preferences and analyse the factors contributing to these similarities.

**Combining Spatio-Temporal Analysis:**

* **Time-Series Analysis:** Analyse how music preferences evolve over time in different countries.
* **Geographic Clustering with Time:** Identify regions with similar music trends and track how these clusters change over time.
* **Spatio-Temporal Correlation:** Investigate the correlation between geographic location and music preferences, considering time-varying factors.

**Potential Insights:**

* **Global Music Trends:** Identify global trends in music consumption, such as the rise of specific genres or artists.
* **Local Music Scenes:** Analyse the unique characteristics of local music scenes in different countries.
* **Cultural Exchange:** Study the impact of cultural exchange and globalization on music preferences.
* **Marketing and Recommendation Strategies:** Provide insights for music industry professionals to develop targeted marketing campaigns and personalized music recommendations.

# Content (Genre) -Based Filtering

The MMTD database has over 1 Million data points. Each row corresponds to a specific tweet about a song that a particular user was listening to at the time the tweet was posted.

The data about each tweet includes Tweet-related information (time of posting, geographical origin, user ID tagging, etc.), Track-related information (artist name, track title, artist ID, etc.), and Track-language information.

If we process this data, we can obtain a user-item (U-I) interaction matrix – which is a matrix of dimensions , where is the number of unique users and is the number of unique items.

Before we get into collaborative filtering, which requires forming the (U-I) matrix, let us see if there is a simpler way to examine item similarities.

### Extracting Track Genres from Last.fm

Much of our understanding of the MMTD data is based on the presentation of the group that curated this dataset. (Hauger, 2013)

The provided dataset does not have genre information, though the authors outline a procedure to extract the same.

### Procedure to extract track genres from last.fm:

1. Last.fm is a streaming music service that provides API access to its database of songs.
2. As outlined in the paper, we created a user account.
3. We understood that, even though there were about 1 Million different entries in the database, they corresponded only to about 127,000 unique tracks. (It needs to be clarified that misspellings and alternate spellings of tracks are classified as different tracks. So each track may end up being represented more than once in the database.)
4. For each one of the 127,000 tracks, we:
   1. Queried Last.fm to extract the top 100 tags associated with the track.
   2. Identified if the song has been tagged as one or more of the 20 most popular genres (R&B, rap, electronic, rock, new-age, classical, reggae, blues, country, world, folk, easy-listening, jazz, vocal, children’s, punk, alternative, spoken-word, pop, and heavy metal). Thus each track could be categorized by up to a maximum of 20 genres.
5. It should be noted that this was a time-intensive process since each API call took about seconds. The code to extract genres for all tracks took about 10 hours to run.

For example, the Taylor Swift song, “Should’ve said no”, is tagged using the following genres:

A screenshot of a computer

Description automatically generated

A screenshot of a computer code

Description automatically generated

### Genre-based filtering algorithm:

1. Create 20-dimension genre vectors for approximately 127,000 songs. These can be represented using the symbol
2. Extract the song index (i) for the track for which similar track recommendations are required.
3. Extract the genre vector for the requested track .
4. Calculate the cosine similarity of the requested track with each of the other 127,000 tracks:
5. Identify the N tracks with maximum cosine similarity.

As per our algorithm, the following tracks have been identified as the top 10 recommendations for someone who likes Taylor Swift’s “Should’ve said no.”

A screenshot of a computer screen

Description automatically generated

### Major challenges in implementing genre-based (or any other kind of content-based) filtering

1. Content-based filtering requires users/providers to have prior knowledge of the items in the database to extract meaningful features that can be used for comparison.
2. Even if experts are available, going through each item, and extracting feature information is a very time-consuming task.

### Advantages of the genre-based filtering approach:

1. Most recommendation systems suffer from a problem known as the cold-start problem. In the absence of user-item interactions (a common scenario at the beginning), most filtering algorithms fail.
2. The genre-based approach works even in this scenario.

# Collaborative Filtering using Matrix Factorization Methods

When a system has been functioning for some time, it generates several user-item interactions.

A user-item interaction is said to happen when a user listens to a song, or rates a song.

In the context of the MMTD database, we have about 1 Million data points, of users listening to songs. (About 127,000 unique tracks). The data is obtained from tweets by users claiming that they were listening to a particular track (similar to how Music Streaming services like Spotify allow a user to tweet about a song they are listening to) – The tweets don’t capture user ratings, or sentiment about any song.

So we interpret the user-item interaction as a binary event – with an entry of 1 if there is an interaction, and 0 if there is no user interaction with that item.

Much of the Collaborative filtering approach (using Matrix Factorization) that we use is based on the work of (Koren, 2009)

### A Basic Matrix Factorization Model

Here, we assume that every user (of users) in our system is represented by a vector of latent user features .

Likewise, every item (of items) in our system is represented by a vector of latent item features .

Then the rating given by user u to item i can be represented by .

Further,

In other words, if the User-Item interaction matrix (dimensions x ) is represented by R, a similar decomposition to the dot product above can be obtained by the singular value decomposition of the matrix R.

The row entries of the item matrix then represent the latent features of each item.

### SVD-based collaborative filtering algorithm

1. Assemble the user-item interaction matrix.

2. Obtain SVD decomposition of the above matrix.

3. Extract latent feature matrix

4. For requested track  calculate cosine similarity with every other track .

5. Identify N tracks with max cosine similarity.

### A Simple 4-person, 12-item user-item interaction-based recommendation system

To demonstrate the efficacy of our algorithm, we test it on a toy-problem.

We consider the following 4 users.

A screenshot of a phone

Description automatically generated

We also consider the following 12 tracks.

A screenshot of a music album

Description automatically generated

The user-item interaction matrix is given as follows:

A screenshot of a computer

Description automatically generated

Under such conditions, the latent feature matrix Q is extracted by SVD decomposition as follows:

A screenshot of a computer

Description automatically generated

Using our algorithm above, the recommendations for someone who likes Taylor Swift are given as follows:

A screenshot of a computer code

Description automatically generated

If you observe, you will see that all of these correspond to the choices of User 3, Rishi. Thus the algorithm automatically identifies a user who likes the song requested and provides suggestions based on other songs liked by the same user.

As the number of users and items increases, we can imagine that the accuracy of the recommendations will improve.

### Advantages and Disadvantages of Collaborative Filtering

1. We need a good number of users and items interactions before we can start making meaningful recommendations. This is often called the cold-start problem.
2. We don’t require subject expertise to extract the latent features of the items and the users. This is taking care of by the underlying Mathematics.
3. On the flip side, the feature vectors represent latent features that are difficult to interpret.

# Item Item Collaborative filtering

## Concept

Item Item collaborative filtering is a recommendation method that looks for similar items based on the ratings given by the users. The ratings are either user has liked a certain item or interacted with it. Once similarities are found then the algorithm looks for items that user has consumed and recommends similar items.

Let's understand this with an example, suppose user X listened to songs A, B and C. Using the CF we found that song C is similar to the song D. Then the recommendation engine will suggest song D to user.

### User – item matrix

First based on Data we prepare a user-item matrix that has user, an item that is (in our case) track title and the explicit rating.

|  |  |  |
| --- | --- | --- |
| Userid | Track title | Rating |
| U1 | T1 | 2 |
| U2 | T2 | 3 |
| U3 | T1 | 4 |
| U2 | T1 | 5 |
| U2 | T3 | 1 |
| U1 | T3 | 1 |
| U3 | T2 | 2 |
| U3 | T3 | 3 |
| U4 | T1 | 2 |
| U4 | T2 | 3 |

From the above table, we can now create User-Item matrix

|  |  |  |  |
| --- | --- | --- | --- |
| Userid/Track title | T1 | T2 | T3 |
| U1 | 2 | ? | 1 |
| U2 | ? | 3 | 1 |
| U3 | 4 | 2 | 2 |
| U4 | 2 | 3 | ? |

? Are the items the algorithm predicts the ratings.

Similarity score

We create item vectors by selecting the users corresponding to two items and their ratings (generally normalized) and use a similarity method such as adjusted cosine similarity to find the similarities between these two item vectors. For example, based on above table

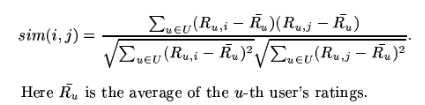
T1 = 2U1 + 4U3

T2 = 3U2 + 1U3

T3 = 1U1 + 1U2 +2U3

As you can see T1 and T3 have ratings from U1 and U3 and the use of cosine similarity will show how similar T1 is to T3. Same way we will find how similar T2 is like T3

Adjusted Cosine similarity



Using the above equation, we calculate the similarity between items, we can calculate for (T1, T3), (T2, T3) etc and create an item-item matrix

### Item-Item matrix

|  |  |  |  |
| --- | --- | --- | --- |
|  | T1 | T2 | T3 |
| T1 | 1 | -0.48 | -0.48 |
| T2 | -0.48 | 1 | -0.50 |
| T3 | -0.48 | -0.50 | 1 |

### Predict - Generating the score for missing rating



From Table 2 user-item matrix , we need to generate missing rating for

R(U1, T2) and R(U2, T1)

R(U1, T2) = (R(U1,T1)\*sim(T1,T2) + R(U1, T3)\*sim(T3, T2))/(sim(T1,T2) + sim(T2, T3))

= (2\*-0.48 + 1\*-0.50)/(-0.48+(-0.50)) = 1.48

R(U2, T1) = 2

R(U4, T3) = 2.5

The user-item table now is complete,

|  |  |  |  |
| --- | --- | --- | --- |
| Userid/Track title | T1 | T2 | T3 |
| U1 | 2 | 1.4 | 1 |
| U2 | 2 | 3 | 1 |
| U3 | 4 | 2 | 2 |
| U4 | 2 | 3 | 2.5 |

**The algorithm now recommends T1 to U2 and T3 for U4.**

## Item-Item Collaborative Filtering for MMTD data

The following section describes using of Item based collaborative filtering for the MMTD data.

### Challenges:

#### No explicit rating

The MMTD data didn’t have explicit ratings. The ratings are calculated based on the number of tweets per track-title by the user.

#### Biased data set

Most of tracks had only one tweet. Removed such entries

#### Large data set

Collab used to crash due to RAM issue and therefore used Dask to initially load the dataset.

Many artists for the same track title

The same track title had several artists. To get unique tracks, track+artist per user is considered as track-title

Data Preparation

#### Loading the data

#use dask to read this large file

import dask

import dask.dataframe as dd

df = dd.read\_csv('/content/mmtd.txt', delimiter='\t', assume\_missing=True)

df = df.head(500000)

df = df[['tweet\_id', 'tweet\_userId', 'track\_title','tweet\_artistId', 'artist\_name']]

df['track\_title'] = df.apply(lambda x : x['track\_title'] + '--' + x['artist\_name'], axis=1)



df = df.groupby(['tweet\_userId', 'track\_title']).size().reset\_index(name='count')

#### Removed entries where count = 1

df = df[df['count'] > 1]

Normalize mean

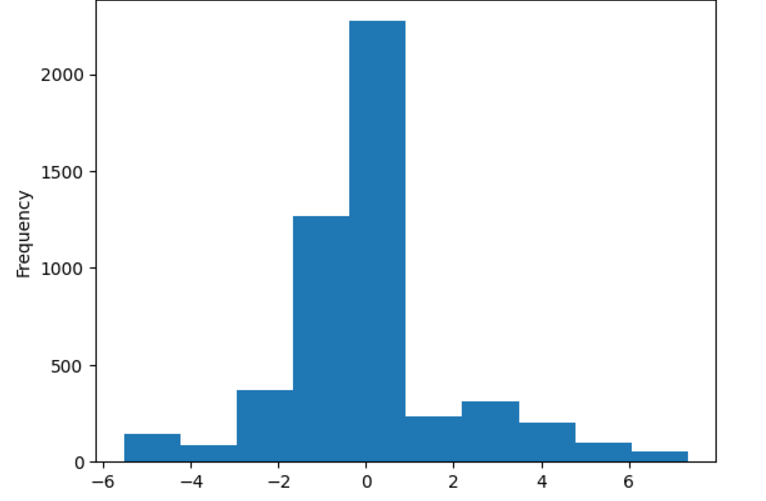
mean = df.groupby(by='tweet\_userId', as\_index=False)['count'].mean()

df = df.merge(mean, suffixes=('','\_mean'), on='tweet\_userId')

df['norm\_count'] = df['count'] - df['count\_mean']



df['norm\_count'].plot.hist(bins=10)



Adjusted Cosine

As explained in concept, we use the adjust cosine for calculating similarities

def adjusted\_cosine(np\_ratings, nb\_items, dataset\_name):

similarities = np.zeros(shape=(nb\_items, nb\_items))

similarities.fill(-1)

def \_progress(count):

sys.stdout.write('\rComputing similarities. Progress status : %.1f%%' % (float(count / nb\_items)\*100.0))

sys.stdout.flush()

items = sorted(df.track\_title.unique())

for cnt, i in enumerate(items[:-1]):

cnt2 = cnt+1

for cnt2, j in enumerate(items[cnt2:]):

scores = np\_ratings[(np\_ratings[:, 1] == i) | (np\_ratings[:, 1] == j), :]

vals, count = np.unique(scores[:,0], return\_counts = True)

scores = scores[np.isin(scores[:,0], vals[count > 1]),:]

if scores.shape[0] > 2:

x = scores[scores[:, 1] == i, 4]

y = scores[scores[:, 1] == j, 4]

w = cosine(x, y)

similarities[cnt, cnt2] = w

similarities[cnt2, cnt] = w

\_progress(cnt)

\_progress(nb\_items)

# get neighbors by their neighbors in decreasing order of similarities

neighbors = np.flip(np.argsort(similarities), axis=1)

# sort similarities in decreasing order

similarities = np.flip(np.sort(similarities), axis=1)

# save similarities to disk

save\_similarities(similarities, neighbors, dataset\_name=dataset\_name)

return similarities, neighbors

nb\_items = df.track\_title.nunique()

similarities, neighbors = adjusted\_cosine(np\_ratings, nb\_items=nb\_items, dataset\_name='music10k')

Finding candidate items for a user

def candidate\_items(userid):

"""

:param userid : user id for which we wish to find candidate items

:return : I\_u, candidates

"""

# 1. Finding the set I\_u of items already rated by user userid

I\_u = np\_ratings[np\_ratings[:, 0] == userid]

I\_u = I\_u[:, 1]

# 2. Taking the union of similar items for all items in I\_u to form the set of candidate items

c = set()

for iid in I\_u:

#get index of iid

title\_index = item\_to\_index[iid]

# add the neighbors of item iid in the set of candidate items

c.update(neighbors[title\_index])

I\_u\_iid = [ item\_to\_index[iid] for iid in I\_u]

c = list(c)

# 3. exclude from the set C all items in I\_u.

candidates = np.setdiff1d(c, I\_u\_iid, assume\_unique=True)

return I\_u, candidates

def similarity\_with\_Iu(c, I\_u):

"""

compute similarity between an item c and a set of items I\_u. For each item i in I\_u, get similarity between

i and c, if c exists in the set of items similar to itemid.

:param c : itemid of a candidate item

:param I\_u : set of items already purchased by a given user

:return w : similarity between c and I\_u

"""

w = 0

for iid in I\_u :

# get similarity between itemid and c, if c is one of the k nearest neighbors of itemid

title\_index = item\_to\_index[iid]

if c in neighbors[title\_index] :

w = w + similarities[title\_index, neighbors[title\_index] == c][0]

return w

def rank\_candidates(candidates, I\_u):

"""

rank candidate items according to their similarities with i\_u

:param candidates : list of candidate items

:param I\_u : list of items purchased by the user

:return ranked\_candidates : dataframe of candidate items, ranked in descending order of similarities with I\_u

"""

# list of candidate items mapped to their corresponding similarities to I\_u

sims = [similarity\_with\_Iu(c, I\_u) for c in candidates]

mapping = list(zip(candidates, sims))

ranked\_candidates = sorted(mapping, key=lambda couple:couple[1], reverse=True)

return ranked\_candidates

Top 30 recommendations for the user

def topn\_recommendation(userid, N=30):

"""

Produce top-N recommendation for a given user

:param userid : user for which we produce top-N recommendation

:param n : length of the top-N recommendation list

:return topn

"""

# find candidate items

I\_u, candidates = candidate\_items(userid)

# rank candidate items according to their similarities with I\_u

ranked\_candidates = rank\_candidates(candidates, I\_u)

# get the first N row of ranked\_candidates to build the top N recommendation list

topn = pd.DataFrame(ranked\_candidates[:N], columns=['itemid','similarity\_with\_Iu'])

topn = pd.merge(topn, df\_item\_index1, on='itemid', how='inner')

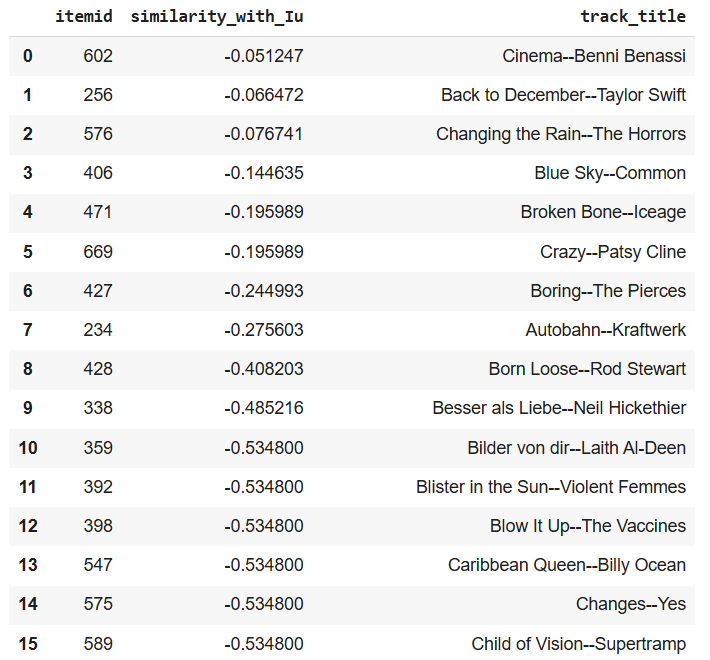
#add columns for artist id and artist name to topn

#topn = pd.merge(topn, df\_title\_artist[['track\_title', 'tweet\_artistId', 'artist\_name']], on='track\_title', how='inner')

return topn

Checking the recommendation for the user

topn\_recommendation(test\_user\_id)



# UI Development

## Project Overview

The application aims to provide end-users with a simple yet powerful interface to perform CRUD operations on relational datasets, conduct data analysis using Python, and visualize the results in a user-friendly manner. By leveraging the integration of Flask for web services, SQLite for data storage, and Python's analytical capabilities, the platform is designed to cater to researchers, students, and small businesses seeking a low-cost data analysis tool.

### Objectives

1. Develop a robust, user-friendly interface for managing and visualizing data.
2. Implement seamless interaction between the web layer and application layer for efficient data operations.
3. Provide dynamic data visualization and statistical analysis tools using Python libraries.
4. Optimize application performance with server-side caching and efficient SQL queries.

### Technology Stack

**1. Web Layer**

* **Languages**: JavaScript v1.5 for interactive front-end components.
* **Framework**: Flask v3.0.3 for lightweight web application development.
* **Template Engine**: Jinja2 v2.3.2 for dynamic HTML content rendering.
* **CSS Engine**: Bootstrap v4.3.1 for responsive design and styling.

**2. Application Layer**

* **Language**: Python v3.10.6 for backend logic and data processing.
* **Caching**: Flask-Caching v2.3.0 for optimizing server performance and minimizing redundant computations.
* **Database Connection**: SQLite3 from Python's standard library to interact with the SQLite database.

**3. Data Store**

* **RDBMS**: SQLite v3.46.1 as the primary relational database.
* **Query Tool**: DB Browser v3.13.1 for database management and debugging.
* **Data Storage**: Text files for storing Numpy arrays and Pandas DataFrames, facilitating easy retrieval and processing.

### Key Features

1. **Dynamic Data Management**
   1. User-friendly forms for uploading, viewing, and editing datasets stored in SQLite.
   2. Validation of data inputs to maintain integrity and consistency.
2. **Data Analysis and Visualization**
   1. Integration of Python libraries (e.g., Pandas, Matplotlib, Seaborn) to perform statistical analysis and generate visualizations.
   2. Exportable charts and reports for offline use.
3. **Efficient Caching and Querying**
   1. Server-side caching with Flask-Caching to minimize latency for frequently accessed data.
   2. Optimized SQL queries for large-scale dataset operations.
4. **Responsive Web Design**
   1. Bootstrap-powered interface for compatibility across devices, ensuring seamless user experience.

### Workflow

1. **Data Upload and Storage**
   1. Users upload CSV files, which are validated and stored in SQLite.
   2. Data is pre-processed and transformed into Pandas DataFrames for analysis.
2. **Data Analysis**
   1. Users select datasets and operations (e.g., summary statistics, correlation analysis).
   2. Results are generated using Python scripts and displayed as interactive visualizations.
3. **Visualization and Export**
   1. Graphs and charts rendered dynamically using JavaScript and Python libraries.
   2. Users can download visualizations as PNGs or PDFs.
4. **Performance Optimization**
   1. Frequently accessed data cached server-side, reducing the need for repeated computations.

## Set up and deployment

A web-based interface is provided so that recommendations can be analysed in real time. The objective is to create a functional minimum viable product which can demonstrate the main functionality.

The web UI project uses [Flask](https://flask.palletsprojects.com/en/stable/) to create the web application and [bootstrap](https://getbootstrap.com/) for styling. Flask in turn uses [Jinja Templates](https://jinja.palletsprojects.com/en/stable/) to dynamically build HTML pages using python constructs. The following tools and technologies is required for web layer.

* Languages- Java Script v1.5, Python v3.10.6
* Web Application – Flask v3.0.3
* Template Engine – Jinja2 v2.3.2
* CSS Engine – Bootstrap v4.3.1
* Language - Python v3.10.6
* Server-Side Caching – Flask-Caching 2.3.0
* SQL connections – sqlite3 (Python Standard Library)

[SQLite](https://www.sqlite.org/) is used as a database and [DB Browser for SQLite](https://sqlitebrowser.org/) is used as SQL Client.

* RDBMS – SQLite v3.46.1
* SQL Query Tool – DB Browser v3.13.1
* Text Files (To store Numpy Arrays and Pandas DataFrame)

$ Go to project root (e.g. cd ~/workspace/research/PersonalisedMusicRecommendation/src/webUI)  
$ Activate virtual environment. (e.g. workon iisc)  
$ export FLASK\_APP=app  
$ export FLASK\_ENV=development

### Set up database.

Open a new tab and go to project root.  
Run the queries in src/webUI/sqlite/schema.sql using DB Browser for SQLite. $ cd src/webUI/sqlite/  
$ python init\_db.py

### Run the application

Open a new tab and go to project root.  
$ cd src/webUI/  
$ flask run  
Open the URL at <http://127.0.0.1:5000/>To hot deploy  
$ flask --debug run  
To run another server  
$ flask run -p 5001

# Summary of results and contributions

## **Summary of the MMTD Dataset**

The Million Musical Tweet Dataset (MMTD) is a valuable resource for music information retrieval and recommendation systems research. It provides a rich dataset of music-related tweets, offering insights into user behaviour, music preferences, and social interactions.

## Key Contributions of the MMTD:

* **Large-scale dataset:** The dataset provides a massive amount of data for training and testing recommendation systems.
* **Diverse music genres:** It covers a wide range of music genres, allowing for a comprehensive analysis of music preferences.
* **Temporal and geographic information:** The dataset includes timestamps and geographic locations of tweets, enabling the study of temporal trends and regional variations in music consumption.
* **User behavior analysis:** The data can be used to analyse user behaviour, such as how users discover new music, share their preferences, and interact with other users.

## Challenges in Working with the MMTD:

* **Data quality:** The dataset may contain noise, inconsistencies, and missing data, which can impact the accuracy of analysis and recommendations.
* **Data sparsity:** The user-item interaction matrix may be sparse, making it challenging to capture user preferences and item similarities.
* **Cold-start problem:** Recommending items to new users or recommending new items to existing users can be difficult due to limited historical data.

# The way forward – hybrid approaches (hypothetical discussion)

## Areas for Further Research:

* **Context-aware recommendations:** Explore how to incorporate contextual information, such as time of day, location, and social context, to improve recommendation accuracy.
* **Hybrid recommendation systems:** Combine collaborative filtering, content-based filtering, and knowledge-based approaches to address the limitations of individual techniques.
* **Social influence analysis:** Analyse the impact of social networks and user interactions on music recommendations.
* **Real-time recommendations:** Develop real-time recommendation systems that can adapt to rapidly changing user preferences and trends.
* **Ethical considerations:** Address ethical issues related to data privacy, algorithmic bias, and the potential for manipulation.

# Team Member’s Names

Ajay Shriwastava  
Chandra Mouli Mudumba  
Karthik Palaniappan  
Nilesh Tompe  
Praneet Nadkar  
Randhir Singh

Saurav Kumar  
Srinivas Chakravarty V

# Code Base and Data Story

The complete Git repository can be found here

[Git Link for Codebase of Personalised Music Recommendation](https://github.com/sauravdarsh/PersonalisedMusicRecommendation/tree/main)

# Bibliography

Hauger, D. (2013). The Million Musical Tweets Dataset: What can we learn from Microblogs. *International Society for Music Information Retrieval*.

Koren, Y. (2009). Matrix Factorization Techniques for Recommender Systems. *Published by the IEEE Computer Society*.

Amazon.com recommendations item-to-item Collaborative filtering by Greg Linden, Brent Smith, and Jeremy York Amazon.com <https://www.cs.umd.edu/~samir/498/Amazon-Recommendations.pdf>