Title: Deep Dive into Music Recommendation Systems

Author: Daniella Prado

Mentor: Professor Daniel Lopresti

Abstract:

This paper explores the evolution of music recommendation systems, shedding light on their past, present, and future implications. Drawing from personal experiences as both a user and creator of recommendation systems, alongside information from articles, academic papers, and textbooks, it offers a comprehensive exploration of the technological advancements and challenges shaping music recommendations. By examining the methodologies and algorithms employed by leading platforms such as Spotify, and Netflix, as well as insights from academic literature and my own project, this paper shows the relationship between user data, technological innovation, and personalized content delivery. Also, it discusses the potential for future improvements in music recommendation systems, emphasizing the importance of continual innovation for media consumption.

Introduction:

One of the struggles of someone who pays for a movie or music subscription service is having a plethora of content in a streaming service and not knowing what to choose. Users can become overwhelmed at this amount of options, which is why these services had to provide a solution for them: make recommendations. Still, streaming services had to acknowledge that this could not be an impersonal strategy. A user that exclusively enjoys techno and jazz music would not be the target audience for a new GooGoo Dolls song, and vice versa. Similarly, an audience member who never watches romantic comedies would not be the target variable to watch the latest Nicholas Sparks movie adaptation.

Recommendation systems became the new question to be answered to support the consumerism that this market is currently experiencing. As this consumerism evolved, so did the recommendation systems that supported it. Being a Spotify subscriber since 2012, I have noticed the exponential improvement in their algorithm for music recommendation. During this time, I distinctly remember getting a recommended playlist from Spotify with songs that I had never heard before. However, all these songs were in Hindi or Mandarin, which caused confusion. Since I don't speak either language, it would make sense why I never would have heard these songs. Now that Spotify's algorithm has improved, the songs that are recommended to me are more similar to the artists, genres, and sentiments that I frequently circulate.

Main text:

I. Background:

A. Movies and Music Recommender Systems

Platforms such as Netflix and Spotify are known for their ability to recommend content to their subscribers. They seem to have mastered treading the fine line of recommending content simply because it's new to recommending content that is new and it correlates to what the user is already consuming. Although these are both forms of media, there are still some significant differences in how these recommendations are made to users. Different factors must be taken into account to make these personalized guesses, although the same technology could be shared. In the textbook "Advanced Analytics with PySpark", it is explored how to use this data to create personalized recommendation systems for users. Chapter 3, "Recommending Music and the AudioScribbler Dataset", explains how to create a music recommendation system through PySpark. However, what caught my attention in this section was the matrix factorization used, which is a way to display data. It is shown in the image below.

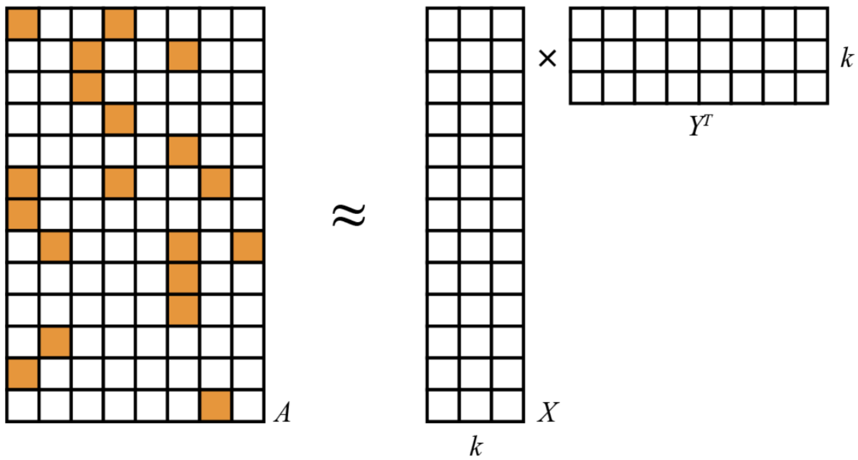
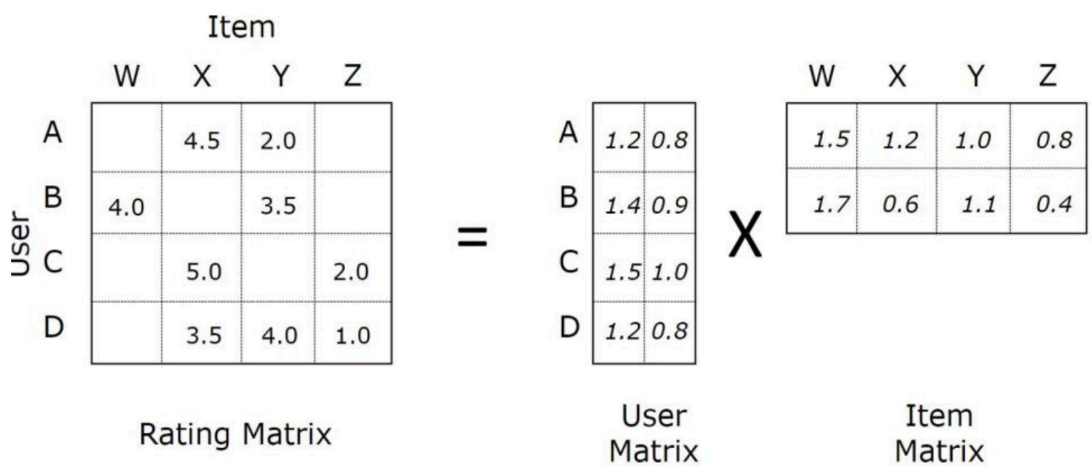


Figure 3-1. Matrix factorization

This image was shown in the textbook to explain user data and a product as matrix A. If rows are i, and columns are j, what this matrix hopes to showcase is whether or not a user has consumed a product or not through a score of 0 and 1. A score of 1 indicates that they have consumed that product and a score of 0 shows they have not. This matrix could be applied to music in the sense that it could check whether a user has listened to a specific artist or not. Similarly, this same matrix could apply to movie recommendation systems and showcase if an audience member has seen a movie by a certain director or a specific movie star as the lead (Tandon, Akash).

Alternating Least Squares is another method that could be used to recommend media, and it is one that Netflix has used in the past. It uses the same concept as matrix factorization, and simply manipulates these matrices to make predictions. Another technique used by Netflix was collaborative filtering, which tries to tackle the problem of spareness and scalability in user profiles (Hu, Yifan). How Netflix collects user information on whether they liked a movie or show is by directly asking the user. After watching a movie or show, Netflix allows users to state whether they liked or disliked what they had just seen. This way, Netflix's algorithm learns the type of content you want to watch. Similarly, when a movie or show is not available on the platform, movies from the same genre are offered. This further proves that Netflix's categorical system of movie genres is what heavily dictates what is recommended to its users.

In contrast, Spotify uses an AI algorithm called "Bandits for Recommendations as Treatments", or BaRT. Similarly, they use a confusion matrix to evaluate these statistics, such as the figure shown below (Mohan, Arya).



As BaRT has been described, its two main policies are: exploit and explore. The algorithm is present on Spotify's home page as users open the app. Instantly, new music is recommended to the user, some because it is new to the algorithm and some because it has been in the database for a long time and the user has not yet explored that content. Exploit focuses on tracking the user's personalized data to make recommendations based on that. During exploring, BaRT looks at what is popular throughout the world, and what other users similar to you are listening to. Another interesting aspect about BaRT is how long it takes to decide if a music recommendation it made was positive or not. If a song was listened to for 30 seconds or more, it is taken as a positive remark (Balaganur, Sameer).

B. A Simple View Into Music

Based on this research, it has been concluded the most naive recommender system for music would be built based simply on connecting a user to a genre that they generally enjoy. For instance, a user that listens to a lot of Bad Bunny music would be prone to enjoy Reggaeton and Spanish Trap music. However, the beats per second and sentiment of the music would also have to play a factor into these recommendations. If a user only likes Bad Bunny Reggaeton that has a sad sentiment, this should alter the recommendations for other artists and songs they receive. Still, this information should be enough to make a semi-educated prediction on a song a user would enjoy, but this prediction would be extremely broad as the only factors being taken into consideration are genre and sentiment. Gradually, this recommendation system would evolve to turn into what Spotify has achieved today.

II. Personal Project:

For my personal project, I tried to build my own music recommendation system using my own Spotify data. The purpose of this was for me to try to get some hands-on experience on the technologies that I had previously researched this and last semester. Additionally, I wanted to take advantage of the opportunity of having a project that I could call my own.

The first step for this project was to request data from my music streaming service - Spotify. This data can be requested on Spotify's web page by going to Home > Safety & Privacy. In this page, the user can then log into their account. Once this is done, one must scroll down until they reach the "Download your data" section. Then, the user must confirm their request on another page and specify the email address they would like to receive this information in. An email from Spotify will be sent to this account confirming their request has been received, and that they should receive their data in no more than 30 days. In my experience, this data took about 10 days to be received from Spotify.

Once the data has been received, Spotify sends additional documentation explaining what this data means, since when it has been collected, and how to use it or understand it. The data itself is in separate JSON files and the ones focused on for this project were the files containing Streaming History, Streaming History, and Playlist1. After analyzing these datasets, the three files containing Streaming History were the ones that were used to build a model. This data came in the format of:


```
{
  "endTime" : "2023-03-06 13:38",
  "artistName" : "Taylor Swift",
  "trackName" : "Daylight - Live From Paris",
  "msPlayed" : 426944
},
```


Using Pycaret and JupyterNotebook, all this data was then read and minimally processed to get an idea on how to be able to use this data to make predictions on music recommendations. The statistics retrieved from these datasets were: most streamed artist, most streamed song, least streamed song, and the amount of time spent listening to the most streamed artist. The results were as follows:

```
Most listened to song file 1:
Artist: Taylor Swift
Track: happiness
Milliseconds played: 1893848
```

```
Most listened to song file 2:
Artist: Taylor Swift
Track: Lover - Live From Paris
Milliseconds played: 679653
```

```
Most listened to song file 3:
Artist: Minnz Piano
Track: Everything Has Changed (Extended Wedding Version)
Milliseconds played: 760058
```

```
Least listened to artist: Miley Cyrus
Least listened to song: Flowers
Ms heard: 0
```

```
Most listened to artist: Taylor Swift
Most listened to song: happiness
```

```
Total time listened to Taylor Swift: 1000071593 miliseconds
  That is : 16667 minutes
  Or :  277 hours
  Equivalent to : 11 days!
```

The three files containing user data were then combined into a singular file named "combined_streaming_history.csv". After this, a Spotify dataset was downloaded from Kaggle called dataset.csv. This dataset contained track_id, artists, album_name, track_name, popularity, duration_ms, explicit, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valance, tempo, time_signature, and track_genre. This dataset was intended to be used to study the user's information and then make a prediction on a song they would like to listen to. To do this, a target variable "liked_or_not" was added to combined_user_data with a value of either 0 or 1. A score of 1 meaning the user liked the song if it was streamed for over 1000 ms and 0 if otherwise. Then, the combined user data was preprocessed to match the column names of the dataset downloaded.

After data cleaning, the user data was then split into testing and validation groups. A Naive Bayes model was created to make an initial prediction, and it was trained on this dataset. However, when the model tried to make a prediction an error occurred as the Kaggle dataset and the user data seemed to be incompatible. Unfortunately, there was not enough time left in the semester for this error to be further revised, but in the future either the online dataset will be changed or necessary changes will be made to the dataset acquired.

Discussion:

Learning to recommend music and movies has become an intricate and important task to benefit media companies and consumers. This paper investigated how movie and music recommender systems have similarities and differences, and how these can be applicable to each. Then, an explanation of my personal project's challenges and findings was presented. Finally, the AI techniques that could be used to further improve this technology were discussed.

I. Media Recommender Systems Findings

The discussion of recommendation systems employed by platforms like Netflix and Spotify reveals intriguing insights into their respective methodologies and the underlying technologies driving personalized content suggestions. While both platforms excel in leveraging user data to enhance the content discovery experience, they adopt distinct approaches for their success.

Netflix's recommendation engine relies on sophisticated algorithms such as Alternating Least Squares and collaborative filtering, which analyze user behavior to predict preferences. In contrast, Spotify employs an AI algorithm known as "Bandits for Recommendations as Treatments" (BaRT), which operates on the principles of exploit and exploration. BaRT dynamically balances personalized recommendations based on individual user data with broader trends and global popularity metrics. Spotify uses a real-time adaptation to user behavior, taking into account how long a user consumed certain media.

Notably, Netflix and Spotify have different target variables that they try to gather information for in order to tailor media to their users. The technologies behind these recommendation systems, such as matrix factorization and confusion matrices, offer deeper insights into how user data is processed and utilized. Matrix factorization, as demonstrated in the textbook "Advanced Analytics with PySpark," serves as a fundamental framework for modeling user-product interactions, applicable across various domains from music to movies.

Still, it is arguable how much more advanced one of these technologies is to another. Music is a dynamic form of media that accompanies users in their everyday lives. This is something Spotify considers, even with their newest feature the “daylist”, which provides a different playlist for a user based on the mood the algorithm predicts the user will be in at different times of the day. Still, movie recommendations on Netflix do not change during the time of day, but rather on the holidays, month, or newest releases.

Though both platforms have deployed technologies that deeply study their users, music seems to be the most complicated one as it is the one that is the most present in people’s lives. Especially during this time in college, most students use music to walk to class, study, shower, drive, and complete other tasks. A movie or show is something that students would generally consume when they have time to relax or need a break, but it is not a form of entertainment that is implicitly on-the-go. In the future, I predict movies and music will have equally sophisticated recommendation systems, but that will have to depend on how media consumerism evolves in the future.

II. Personal Project Reflection

The purpose of a personal project on building a music recommendation system using Spotify data was to gain hands-on experience with technologies previously researched and take a proactive approach to learning. However, there are several areas for improvement.

The process of requesting and receiving data from Spotify was an extremely practical way to obtain personalized user data. However, there was a lack of information in this data. It would have been more beneficial to this project if the album name, genre, or the explicitness of a song would have been provided. This would have improved the robustness of the data, and might have made it easier to use and understand.

The utilization of Pycaret and JupyterNotebook was based on previous experiences with the technologies. At first, I believed this would be the easiest API to manipulate the data in until I ran into errors I could not identify. After placing the csv files in another API, Weka, I realized there was a problem with this data. As such, I realized the Kaggle dataset presents a significant challenge. While this setback limited the project's progress, it also highlights the importance of thorough data validation and compatibility checks before model training. In future iterations of the project, allocating sufficient time for data exploration and troubleshooting could mitigate such issues. I learned that I should try to manipulate data downloaded from the internet before trying to base a project off of it. If I had caught these discrepancies with the Kaggle data from the beginning, I believe I would have obtained very different results in my model.

Despite encountering challenges, the project provided me with multiple takeaways for working with data and creating recommendation systems. Using personalized data made me further understand how everything I can do in a platform can be used to personalize and improve my experience. Moving forward, prioritizing thorough data preprocessing, model validation, and iteration cycles can contribute to the development of a robust and effective music recommendation system. In the future, I plan to continue using personalized music data to further understand how these recommender systems work, and how these new technologies study humans through our searches, clicks, and time consuming media.

Conclusion:

In conclusion, the study of movie and music recommender systems offers valuable insights into the intersection of technology and media consumption in the digital age. The analysis of platforms like Netflix and Spotify showcases different approaches and methodologies employed to deliver personalized content recommendations to users. While both platforms have succeeded in leveraging user data to enhance recommendations, they adopt different strategies for their respective media outlets. The personal project on building a music recommendation system using Spotify data highlights practical challenges and opportunities in working with personalized user data. Moving forward, prioritizing technological innovation, user-centric design, and continuous iteration will be essential in advancing the capabilities and effectiveness of recommendation systems across various media platforms. As media consumerism continues to evolve, more sophisticated and personalized recommendation systems will arise, creating advancements in AI and data analytics.

References:

Balaganur, Sameer. "How Spotify’s Algorithm Manages To Find Your Inner Groove." Analytics India Magazine, January 6, 2020, https://analyticsindiamag.com/how-spotifys-algorithm-manages-to-find-your-inner-groove/

Hu, Yifan, et al. "Collaborative Filtering for Implicit Feedback Datasets." AT&T Labs – Research, Florham Park, NJ 07932, and Yahoo! Research, Haifa 31905, Israel, and AT&T Labs – Research, Florham Park, NJ 07932.

Mohan, Arya. "How Does Spotify's Recommendation System Work?" Univ.ai Blog, January 5, 2023, https://www.univ.ai/blog/how-does-spotifys-recommendation-system-work#:~:text=Every%20Monday%2C%20Spotify%20gives%20its,To%20Find%20Your%20Inner%20Groove.

Tandon, Akash, et al. "Chapter 3. Recommending Music and the Audioscrobbler Dataset." Advanced Analytics with PySpark, O'Reilly Media, Inc., 2019.