How Streaming Services Use AI and Algorithms to Personalize Your Music Experience

Mykyta Nosenko

November 1, 2024

Abstract

Imagine you're walking through a park, and everywhere you look, people are wearing headphones. Daily, you can see hundreds or even thousands of people rap in their own worlds of sound. What are they listening to? Some may be into podcasts, some into audiobooks or radio, but most are likely enjoying music. In today's digital age, powerful streaming services like Spotify make accessing our favorite songs effortless. In this article, we will dive into how their recommendation systems work.

We're about to explore the world of algorithms and artificial intelligence that services use to provide specifically tailored music to our tastes. In Spotify's recommendation engine, you can meet such algorithms: collaborative filtering and content-based filtering. For example, collaborative filtering uses matrix factorization to identify hidden user preferences based on listening habits, while content-based filtering employs techniques like TF-IDF to analyze song lyrics and extract significant themes. Moreover, Spotify leverages an AI system called Marlin, which combines these algorithms to enhance the user experience and provide personalized music recommendations[1].

How do these systems understand our preferences? Who develops these complex technologies, and what exactly goes into building them? There are so many questions, and much remains unknown about how AI enhances our music experience, constantly adapting to our moods and listening habits. Let's take a closer look at how it all works.

Contents

1 Recommendation Systems An Overview

Greetings to everyone! I hope your mood is great. Before we begin, try to clear your mind of distractions and fully immerse yourself in the topic. We have gathered information from articles,podcasts and interviews with company executives and developers to better understand how these systems are built and what technologies power them. Let's dive in!

To begin with, modern music streaming services offer users not only access to extensive music catalogs but also ready-made playlists with personalized music combinasions. Systems based on artificial intelligence and machine learning, provides tracks that perfectly match each user's preferences. These Systems are playing a crucial role in discovering new music. According to data from the web-site "Music Tomorrow" , around 62% Schema 1 of users identified Streaming services and YouTube as the main sources for finding new songs. Further details are provided in the diagram below [1][2].

Today, we will explore how recommendation systems work in practice, analyzing the biggest music platforms experience — Yandex Music, Spotify, Apple Music, and Deezer. Each of them is unique and holds a leading position on its continent.

- **Yandex Music** — the leading music service in Russia and the CIS countries, offers access to a vast collection of tracks and uses sophisticated algorithms to analyze your musical preferences and moods.
- **Spotify**, a Swedish platform popular in Europe and America, is renowned for its smart playlists and recommendations based on your tastes, time of day, and even your mood.
- **Apple Music** — a global giant popular in the USA and many other countries, is tightly integrated into the Apple ecosystem. It uses your listening habits and interactions with music to create a personalized experience.
- **Deezer**, a French leader, has gained popularity in Europe thanks to localized recommendations and unique algorithms that adapt music selections to each user as much as possible.

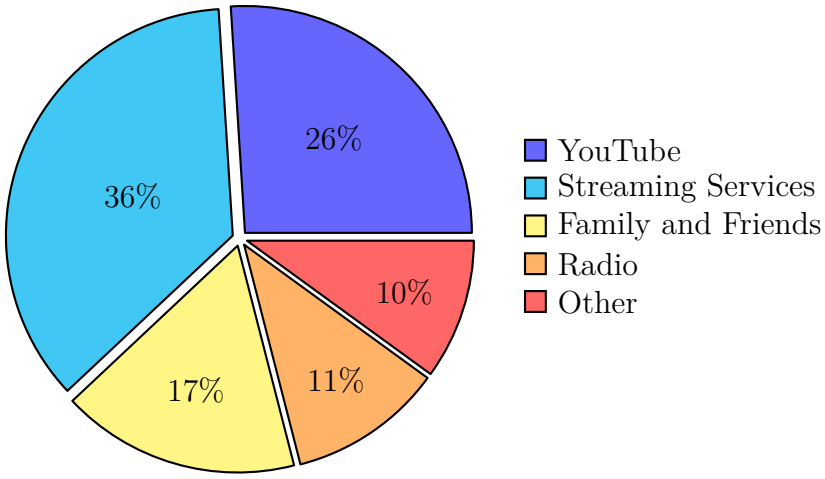

Figure 1: Resourses from which people recieves new tracks[3].

2 Algorithms Behind the Recommendation Systems

Spotify, with its vast library of over 82 million songs and 406 million subscribers, has transformed how listeners discover and engage with music. Spotify uses advanced machine learning (ML) algorithms to recommend new tracks, but also to enhance the overall listening experience. Here, I will try to hilighight, explain key algorithms that is used by Spotify[4]. For a clearer understanding of how the system works, I created a diagram that displays the algorithms, starting from the simplest to the more complex ones Schema 2.

- **Collaborative Filtering (CF)**

  Below we will talk about that one method more precisely. Here the point is to understand that collaborative filtering is a key function of Spotify's recommendation system. That technique relies on analyzing the collective behavior and preferences of users to predict what an individual might enjoy based on the actions of others with similar tastes. Millions of users provide a rich dataset for Spotify.

  Analyzing how users interact with their playlists enables Spotify to provide customized music suggestions based on preferences and listening habits. If User A adds a song by Eminem, the algorithm increases the likelihood of recommending Eminem tracks to User B, who has similar listening habits[4].

- **Natural Language Processing (NLP)**

  Natural Language Processing uses Spotify's song text databases to analyze hidden preferences of the user. It understands the theme of the song and the type it refers to. In the beginning, the algorithm scans hole texts, extracting keywords and assigning them weights based on emotional context, then categorizes songs into the groups. Songs can be classified by the emotions they evoke, such as happiness, sadness, or nostalgia, making it easier for the platform to group similar tracks together.

  NLP helps Spotify's algorithms figure out which songs can be used in new playlists based on the language used in them. Spotify understands which speech is common to you and which languages you like (agresive, monotonous). Function not only improves the accuracy of recommendations but also allows for the creation of emotionally resonant playlists[4].

- **Reinforcement Learning (RL)**

  Reinforcement learning takes Spotify's recommendation system to the next level; a clue of it is to reach maximum score while interacting with the user.

  The great question will be. How it works? At first, Spotify's RL system was made for advising something new, combining with long-time favorite tracks. Tracks that Spotify RL may be completely different from what you are listening to because it analyzes a lot of factors like likes, time of the day, place, your mood by what are you listening to now. Moving on, algorithms observe how users engage with newly recommended songs. If a user listens to a song repeatedly, adds it to playlist, sends it to someone, it signals to the algorithm that the recommendation was successful, and RL will recieve point. Conversely, if the user skips the song, it indicates a mismatch with the user's preferences, and RL will receive a negative amount of points.[5]

  Maximizing Long-Term Engagement: The goal of RL is to optimize long-term user satisfaction. As Spotify's Vice President of Personalization, Oskar Stål, explains, RL aims to push users towards a diverse and fulfilling listening experience rather than a monotonous one. This encourages exploration of new genres and artists, ultimately broadening the user's musical palate[6].

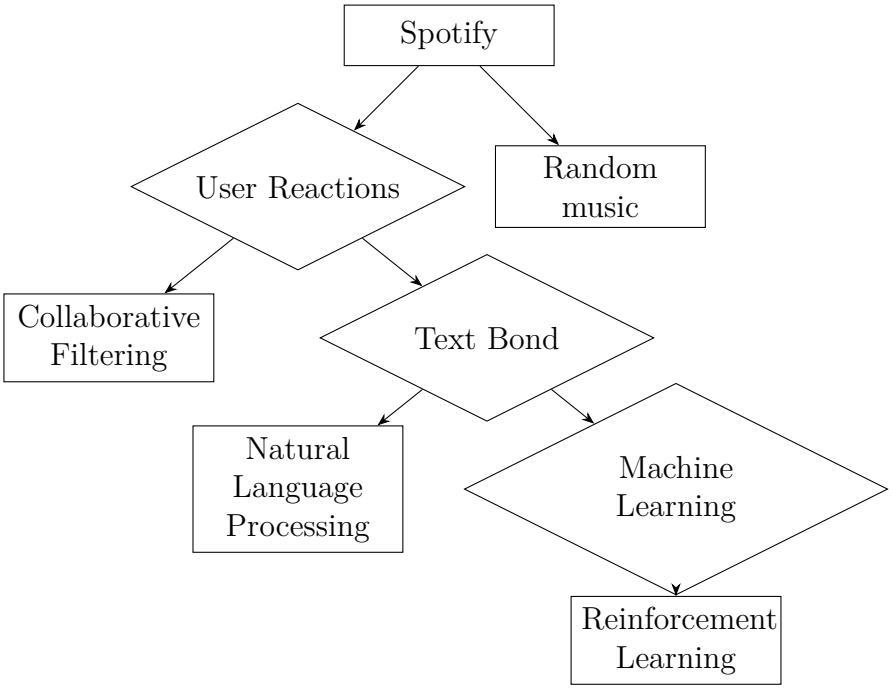

Figure 2: Hierarchy of spotify's recomendation system

3 An Inside Look: Interview with the Head of Yandex's Recommendation Systems, Danil Burlakov

We analyzed an interview with Danil Burlakov, where he explained how the recommendation system of Yandex Music is structured and what technologies are at its core. Our guest has been working in the recomendation systems for about five years, during which time it went through several leadership changes before Danil himself became the head of the project. He was graduated from the Faculty of Mechanics and Mathematics at Moscow State University, Danil now leads the development of recommendation systems for Yandex media services, including Yandex Music[7].

3.1 Programming Language Choice

One of the key points in the interview was the explanation of why Java is used for the recommendation systems at Yandex Music. The recommendation system processes massive data arrays, such as matrices of 10,000 by 1,000, where

each number contains information about user plays, likes, and preferences. Java ensures high performance and efficient handling of multithreaded tasks, which is critical for real-time request processing[8].

While Python is not suitble? It is popular in machine learning, but it is less efficient in handling high loads and large numbers of threads, making it less suitable for tasks that involve processing such large data arrays. Moreower, in Yandex for particularly resource-intensive calculations, such as gradient boosting, is used C++ to achieve maximum speed when performing complex operations.

3.2 Collaborative Filtering and Vector Models

Collaborative filtering is the primary mechanism behind Yandex Music's recommendation system, works like that. Both users and tracks are represented as vectors. The vector $\vec{A}$ for a user is based on their activity, including likes, skips, plays, and genre preferences. Each of activity is evaluated as number and transformed to vector. The vector $\vec{B}$ is based on another people expirience by the same way [9]. After we will count coeficient of similarity between some vectors. If the coficient is high, we can advies for user $\vec{A}$ tracks that was liked by user $\vec{B}$.

For example, if a user frequently listens to electronic and rock music, the vector will be closer to tracks in those genres. When a simillar ganers appears in another users taste, the system will suggest other preferences of second user to user numder one.

Matrix factorization is used to improve the efficiency of finding relevant tracks. This reduces the dimensionality of the original "users-tracks" matrix, speeding up the process of identifying hidden preferences.

I also developed my own collaborative filtering model in C, which helps optimize the recommendation process and improve system performance (For better understanding we made our model of Colaborative systems, refer to paragraph 5) [1].

3.3 Offline and Online Playlists

The recommendation system is divided into offline and online playlists. Offline playlists are formed based on long-term user data and are processed in batches, reducing the system's load. Online playlists are updated instantly, using the most recent data about user plays, likes, and preferences, allowing the system to reflect changes in the user's music taste almost in real-time. In

[1]For better understanding, refer to paragraph 5.

Yandex Music, daily playlists are updated every time user opens the app, allowing the system to consider the user's latest preferences. It distributes the load from similtanious to permonent. It figured out the problem of adapting to time zones changes. For example with daily playlist updating, if a person flies from the USA to Europe, he will not recieve playlist in time. In addition, if the user doesn't open the app for a long time, the massive of songs remains the same, but after a week of inactivity, the data becomes outdated, and the playlist will be updated the next time the user logs in.[10][11].

3.4 Data Transfer Protocol and Playlist Updates

Protocol Buffers (Protobuf) is used to transfer data between services, which is more efficient than JSON. Protobuf reduces data size and speeds up processing, which is crucial when handling millions of users. Yandex aimed to make one of the fastest servise in the world.

Thus, the interview with Danil Burlakov provides a clear understanding of how Yandex Music combines technology and algorithms to create accurate, personalized recommendations.

4 Comparison

Criterium	**Apple Music**	**Yandex Music**	**Spotify**	**Deezer**
1 Level	Collaborative Filtering	Collaborative Filtering	Collaborative Filtering	Collaborative Filtering
2 Level	Natural Language Processing	Matrix Factorization	Natural Language Processing	Natural Language Processing
3 Level	Content-Based Filtering	Reinforcement Learning	Reinforcement Learning	Reinforcement Learning
4 Level	Reinforcement Learning	Content-Based Filtering	Audio Feature Analysis	Content-Based Filtering

Table 1: Comparison[12]

Similarities: All four platforms — Apple Music, Yandex Music, Spotify, and Deezer — use collaborative filtering at the first level. This method is based on analyzing user data, such as listening habits, to recommend new tracks by identifying similarities in user preferences. Additionally, for all platforms except Apple Music, reinforcement learning is applied at the third

level, allowing them to improve recommendations in real-time based on user interactions with the system (for example, when users skip or replay songs). Table 1

Differences: After collaborative filtering, Apple Music applies Natural Language Processing (NLP) to analyze song lyrics or descriptions, then moves to content-based filtering, and concludes the process by using reinforcement learning at the final level.

Yandex Music, after collaborative filtering, uses matrix factorization, which differs from other platforms. This method helps reveal users' hidden preferences. In the following levels, Yandex Music applies reinforcement learning and content-based filtering.

Spotify, after collaborative filtering, also uses Natural Language Processing (NLP) and then applies reinforcement learning. However, at the final stage, Spotify uses audio feature analysis, which includes examining the musical characteristics of tracks, such as tempo, mood, and rhythm.

Deezer, like Spotify, uses Natural Language Processing and reinforcement learning after collaborative filtering, but at the final level, it prefers content-based filtering, similar to Apple Music.

Thus, collaborative filtering and reinforcement learning are key methods used in the early and advanced stages of recommendations for most platforms. However, each platform has unique differences: Yandex Music stands out with its use of matrix factorization, Spotify applies audio feature analysis, while Apple Music and Deezer use content-based filtering at the final level of recommendations. Table 1

5 Colaborative filtering project

Based on the information I studied, I created a first-level recommendation system using collaborative filtering. To improve accuracy, I surveyed around ten people with different music tastes, who rated tracks on a scale from 0 to 5. Let's break down how the program and cosine similarity work.

Program Steps:

- **Data Collection:** We have user data and their track ratings (from 0 to 5), saved in the file `UserMarks.txt`. The new user also rates tracks on the same scale.
- **Similarity Calculation:** Using cosine similarity, we calculate the similarity coefficient between the new user and each of the users in the database. This coefficient is used to weight the ratings.

$$\cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

Cosine Similarity Formula: Cosine similarity (the formula above) measures how close the preference vectors of two users are. The closer the value is to 1, the more similar the users' preferences are.

```
double calculate_coefficient(int* user_ratings,
   int* song_ratings, int size) {
    double dot_product = 0.0, length_user = 0.0,
       length_songs = 0.0;
    int valid_count = 0;

    for (int i = 0; i < size; i++) {
        if (user_ratings[i] != -1 &&
           song_ratings[i] != -1) {
            dot_product += (double)user_ratings[i]
               * song_ratings[i];
            length_user += (double)user_ratings[i]
               * user_ratings[i];
            length_songs += (double)song_ratings[i]
               * song_ratings[i];
            valid_count++;
        }
    }

    if (valid_count == 0 || length_songs == 0.0)
       return 0.0;

    return dot_product / (sqrt(length_user) *
       sqrt(length_songs));
}
```

In this formula, we find the dot product of the users' ratings, then divide it by the product of the vector lengths (`length_user` and `length_songs`), normalizing them and producing a coefficient between -1 and 1. This determines how close the preferences are. How it's done in my code you can see here Listing 5.

- **Calculating the Final Rating:** To recommend a track the new user hasn't rated, the system uses the following formula:

$$\text{Rating}_{\text{track}} = \frac{\sum_{i=1}^{n}(\text{UserRating}_i \times \text{SimilarityCoefficient}_i)}{\sum_{i=1}^{n} \text{SimilarityCoefficient}_i}$$

This formula considers the ratings of all users in the database, weighted by their similarity to the new user.

- **Sorting Recommendations:** We sort tracks by their final ratings, from most to least suitable.In use, this formula is easier than you think Listing 5.

```
void sort_recommendations(double* ratings, char
   recommendations[MAX_SONGS][255], int size) {
    for (int i = 0; i < size - 1; i++) {
        for (int j = i + 1; j < size; j++) {
            if (ratings[i] < ratings[j]) {
                double temp_rating = ratings[i];
                ratings[i] = ratings[j];
                ratings[j] = temp_rating;
                char temp_song[255];
                strcpy(temp_song,
                   recommendations[i]);
                strcpy(recommendations[i],
                   recommendations[j]);
                strcpy(recommendations[j],
                   temp_song);
            }
        }
    }
}
```

- **Outputting Recommendations:** Finally, the program outputs tracks in order from most interesting to least, based on calculated final ratings.

This is how collaborative filtering works — by recommending tracks that similar users enjoyed.

6 Conclusion

In today's world of music, where millions of tracks are available at the click of a button, recommendation systems play a key role in shaping our musical experience. Streaming platforms like Yandex Music, Spotify, Apple Music,

and Deezer use complex algorithms based on machine learning to tailor music recommendations to individual user tastes.

We explored the main methods used in recommendation systems, including collaborative filtering, natural language processing, and reinforcement learning. Each of these methods offers unique approaches to analyzing listener preferences and providing personalized recommendations. This not only enhances user interaction with the platforms but also fosters the discovery of new music, broadening their musical horizons.

In an interview with Danil Burlakov, we learned about the challenges and future prospects of developing recommendation systems in Yandex Music, as well as how technology and innovation continue to transform the music industry.

Thus, recommendation systems not only simplify the search for music but also actively shape our musical preferences, creating a unique and engaging experience for every listener. In the future, we can expect further refinement of these systems, making them even more accurate and intuitive, while also opening new horizons in the music world.

7 References

References

[1] Dmitry Pastukhov. Inside spotify's recommender system: A complete guide to spotify recommendation algorithms. 2022. https://www.music-tomorrow.com/blog/how-spotify-recommendation-system-works-a-complete-guide-2022.

[2] Julie Knibbe. Understanding music discovery algorithms - how to amplify an artist's visibility across streaming platforms. 2020. https://www.music-tomorrow.com/blog/understanding-music-discovery-algorithms-how-to-amplify-an-artists-visibilit

[3] Glenn Peoples. Consumers now favor streaming services for music discovery over all other sources. 2020. https://www.billboard.com/pro/consumers-streaming-music-discovery-music-360/.

[4] Mage. How does spotify use machine learning? 2022. https://dev.to/mage_ai/how-does-spotify-use-machine-learning-4pdg.

[5] Surya Kanoria, Joseph Cauteruccio, Federico Tomasi, Kamil Ciosek, Matteo Rinaldi, and Zhenwen Dai. Simulated spotify listening experiences for reinforcement learning with tensorflow and tf-agents. 2023. https://blog.tensorflow.org/2023/10/simulated-spotify-listening-experiences-reinforcement-learning-tensorflow-tf
html.

[6] Federico Tomasi, Joseph Cauteruccio, Surya Kanoria, Kamil Ciosek, Matteo Rinaldi, and Zhenwen Dai. Automatic music playlist generation via simulation-based reinforcement learning. 2023. https://research.atspotify.com/2023/07/automatic-music-playlist-generation-via-simulation-based-reinforcement-learn

[7] Anna Ganja. You will like it. how recommendation algorithms work. 2023. https://www.setters.media/post/tebe-ponravitsya-kak-rabotayut-rekomendatelnye-algoritmy.

[8] Daniil Burlakov. Music recommendation algorithms / yandex.music under the hood / interview with daniil burlakov. 2022. https://www.youtube.com/watch?v=vkFcmHfIpto.

[9] Ilya Garkusha. Yandex.music recommendation system. 2021. https://i-m-i.ru/post/how-does-yandex-music-work.

[10] Sam Partee, Tyler Hutcherson, and Nathan Stephens. Offline to online: Feature storage for real-time recommendation systems. 2023. https://developer.nvidia.com/blog/offline-to-online-feature-storage-for-real-time-recommendation-systems-with-

[11] Jane S. Consumers now favor streaming services for music discovery over all other sources. 2018. https://d3.harvard.edu/platform-rctom/submission/spotify-machine-learning-as-recommendation-engine-and-musical-composer/.

[12] Alois Gruson, Praveen Chandar, Christophe Charbuillet, James McInerney, Samantha Hansen, Damien Tardieu, and Ben Carterette. Offline evaluation to make decisions about playlist recommendation algorithms. 2019. https://pchandar.github.io/files/papers/Gruson2019.pdf.