Modeling a Music Recommendation System Using Spotify Web API

Siyeon Park, Isabella Francesconi

Abstract

Society has witnessed exponential growth in the popularity of music streaming services throughout the past decade, making music more accessible and integral to daily life. Navigating through the extensive libraries of such platforms can be time-consuming and overwhelming, which introduces a need for effective music recommendation systems to help users discover content more simply. Our project addresses this need by using Spotify's Web API data to develop a content-based music recommendation system. Focused on musical attributes, our model explores the potential of utilizing songs' technical music features for personalized recommendations. Using Spotify's WEB API, we collected music metadata, emphasizing attributes such as danceability, energy, tempo, and volume. Machine learning techniques such as minmaxscaler and cosine similarity were employed to analyze the musical features of an input song and generate a list of song recommendations. Evaluation of our model through scatter plot analysis demonstrated our prediction system's effectiveness in recommending songs with similar musical features as the song input. Further evaluation of our model through satisfaction surveys from 15 users indicated positive responses to our model's recommendations, demonstrating the potential for increased satisfaction with additional user feedback trials. Our project demonstrates the viability of music technical content-based recommendation systems, with areas for improvement regarding personal user data constraints and the quantity of music data incorporated into our model.

Introduction

Given the wide variety of options available on streaming services, music has become an increasingly important part of our daily lives in the twenty-first century. The task of helping consumers find songs that fit their tastes becomes more and more important in this large musical landscape. Our project focuses on comprehending and utilizing music attributes to assist in song choices, addressing the shortcomings and complications seen in user-centric music recommendation systems. Opportunities and challenges arise from the availability of musical options and the expansion of music streaming platforms.

Figuring out the user preferences in this abundance music industry is significant.By investigating the effectiveness of using musical characteristics in recommendation algorithms while taking into account the limitations of obtaining individualized user data, our project seeks to recommend songs based on these abundant amount of music datas.

Our goal is to develop a powerful recommendation system that utilizes the qualities of music. With limited access to extensive user data, we utilized Spotify's API to gather music metadata, emphasizing attributes such as danceability, key, and volume. Even though retrieving individual user preferences is difficult, our method aims to improve the user experience by providing personalized song recommendations.

To identify song similarities, we analyzed music features using machine learning techniques like minmaxscaler and cosine similarity. Using a song that the user submits, our system finds ten suggestions based on similar musical qualities. We also performed scatter plot studies, particularly the connection between energy and danceability in the input and output songs.

Using machine learning approaches, we developed an effective music recommendation system based on musical qualities. When we enter the name of a song into our music recommendation system, a list of 10 recommended songs appears in the results. We compared density and energy with four pairings of one input and twenty outputs to visualize the link between music qualities based on our music recommendation system. We discovered that danceability and energy matched input and output, indicating that the algorithm for recommending music is operating correctly based on the characteristics of the song. Lastly, we gathered input from 15 users, which showed a wide range of songs they enjoyed from the suggestions, and we used that information to perform the baseline prediction.

Our research underscores the viability of music property-based recommendation systems in aiding song discovery. However, we recognized larger and more diverse datasets of the music tracks will enhance the accuracy of the system. Despite limitations, our approach showcases potential avenues for refining music recommendation systems, especially in contexts where personalized user data is constrained.

Methods

Data was collected in two different ways, by requesting our user account data from Spotify and getting track data from the playlist using Spotify Web API. Figure 1 is the workflow of our whole project.

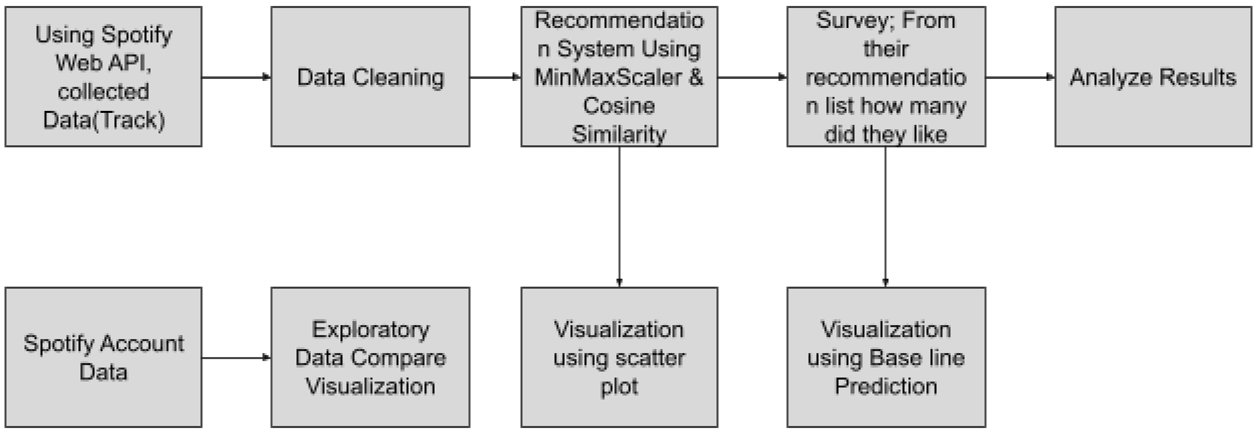


Figure 1: Project Workflow

Spotify user account data is accessible from Spotify involving users directly accessing user account settings within the Spotify platform's Privacy settings section, requesting our data by agreeing to the terms of the conditions. Then, it takes up to 5 days for Spotify to send our account data through our email. Once we download the Spotify Account Data, we can access our account data such as "Playlist", "Searching Queries", "Streaming History", and "YourLibrary" in JSON files. After downloading our user data, we were able to conduct an exploration of our user data. We created a visualization of a comparison of the streaming history of our group members and compared the top artist list and how similarities that we have.

We collected approximately 4500 track data using Spotify Web API. To access the Spotify Web API, we first had to log in to the Spotify Web API page and create a development app for Client ID and Client Secret. Since we did not have a clear idea about the music recommendation system and what kind of machine learning is used in the recommendation system, we searched about the music recommendation system and figured out how music recommendation systems are usually modeled. Filtering methods are used for music recommendation systems, mainly divided into content-based filtering and collaborative filtering. Content-based filtering music recommendation system analyzes the attributes of tracks such as tempo, artist, lyrics, or musical elements and then recommends tracks that share similarities with users previously liked or listened to music(Dey, Victor). Collaborative filtering music recommendation identifies users with similar tastes and recommends tracks based on user behavior or interactions within a system(Dey, Victor). We decided to create a music recommendation system based on content-based filtering. However, since the user data we could get was our data, user data wasn't sufficient to train the model to give the recommendation based on the user's history. Instead, we decided to create a recommendation function that returns the songs that are similar to the input song based on the music properties. Before we created a function that generates the recommendation list, we first had to collect the track information using Spotify Web API.

Spotify Web API allows to get diverse information related to Spotify, including user's top songs, playlists, etc. Since we need loads of data for the music attributes we decided to get track information by getting a playlist. To get the track information from the playlist, a playlist ID is required. Playlist IDs were able to be accessed from the Spotify playlist URL. For example, Spotify playlist URL is "https://open.spotify.com/playlist/5VcubOjfAeoDcSZtU3Ypq?si=Rcf_1AnEQ068AHbYRn5Ftw&pi=u-8zT0Q2CMQQa4". Then, the playlist id will be 5VcubOjfAeoDcSZtU3Ypq, from after playlist/ and before ?si=. However, only limited access was able with Spotify Web API, regarding getting track information using playlists, we were only able to get 100 track information from one playlist no matter how many songs were in the playlist. Some playlists that we tried to get data from had more than 100 songs, so I had to create a copy of the playlist modify the playlist into 100 songs, and create another playlist to input the songs that are not included in 100 songs. We extracted relevant information and stored it in a list of dictionaries and created a pandas database from the list of dictionaries. Since one playlist could get up to 100 tracks, we repeated the process of getting the track information. At some point, the error happened with a notation that max retries reached. So, we first combined the data into one data frame and saved it to a CSV file. The solution for the max retries error was resolved when we created a new Spotify web document with a new client ID and client secret. Since we knew tracking data would be better if it was larger, we continued the process and ended up creating 4 Python notebook files of data collection.

After we finished the data collection, we first opened every CSV file downloaded from the data collection Python notebook, and combined it into one data frame. We first dropped unnecessary columns such as “Popularity” and “Explicit URLs”. For popularity, popularity had values originally when we got the playlist data using Spotify web API but changed into null in the process of combining it into one whole data frame. Therefore, we decided to drop the “Popularity” column. We also dropped duplicates of the track since our track data frame is a combination of multiple playlists and some playlists could have the same track. There also existed some tracks where the artist and the track name were the same but since track ID and album ID were different so treated as different songs. We created the “Artists_Song” column to drop the duplicates. Figure below shows the duplication drops for the case above.

In [147]:
```
df[df['Track Name']== "Agora Hills"]
```
Out[147]:

	Track Name	Artists	Album Name	Album Id	Track ID	Release Date	Duration (ms)	Danceability	Energy	Key	Loudness	Mode	Speechin
94	Agora Hills	Doja Cat	Scarlet	6DmPNcfpkXBVRJsElJY9tl	7dJYggqjKo71Kl9sLzqCs8	2023-09-22	265360	0.750	0.674	8	-6.128	0	0.(
815	Agora Hills	Doja Cat	Scarlet	1bBez9PNvkJPW08bU7NYta	5PyDJG7SQRgWXefgexqlge	2023-09-20	265360	0.755	0.687	8	-6.247	0	0.
2968	Agora Hills	Doja Cat	Scarlet	6DmPNcfpkXBVRJsElJY9tl	7dJYggqjKo71Kl9sLzqCs8	2023-09-22	265360	0.750	0.674	8	-6.128	0	0.(

In [148]:
```
df['Artists_Song'] = df.apply(lambda row: row['Artists']+' '+row['Track Name'],axis = 1)
```
In [149]:
```
df.drop_duplicates('Artists_Song',inplace = True)
```
In [228]:
```
df[df['Track Name']== "Agora Hills"]
```
Out[228]:

	Track Name	Artists	Album Name	Album Id	Track ID	Release Date	Duration (ms)	Danceability	Energy	Key	Loudness	Mode	Speechiness
94	Agora Hills	Doja Cat	Scarlet	6DmPNcfpkXBVRJsElJY9tl	7dJYggqjKo71Kl9sLzqCs8	2023-09-22	265360	0.75	0.674	8	-6.128	0	0.097

Figure 2: Dropping duplicates where Albumn ID and Track ID different but same song

We used minmaxscaler to rescale the music properties to have values between a specified range(“Sklearn.Preprocessing.MinMaxScaler.”). Then, we used cosinesimilarity to compare the similarity between the features of different songs(Karabiber, Fatih). By representing songs as feature vectors and calculating the cosine similarity between these vectors, we could find songs that are more alike in terms of their musical attributes. This similarity score helps in identifying songs that might be recommended to users who enjoy similar kinds of music. Using this machine-learning technique we were able to create a model that prints out the recommendations based on the input songs.

Results and Discussion

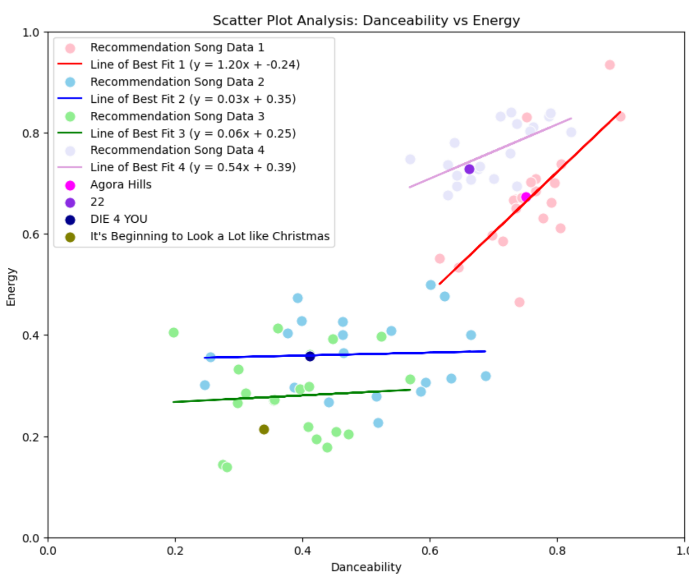
Two main methods were employed to assess the effectiveness of our recommendation system: musical features correlation and user satisfaction. Figure 3 below shows a sample output from our recommendation system. Note the musical features including tempo, danceability, energy, and loudness associated with each provided track.

```
Out[277]:
Content based recommended songs for 'Rema, Charm':
```

	Track Name	Artists	Album Name	Release Date	Tempo	Danceability	Energy	Loudness
1	Distant	B Young	Differences	2021-07-30	104.995	0.693	0.508	-8.727
2	PAMI (feat. Wizkid, Adekunle Gold & Omah Lay)	DJ Tunez, Wizkid, Adekunle Gold, Omah Lay	PAMI (feat. Wizkid, Adekunle Gold & Omah Lay)	2020-08-13	99.889	0.755	0.645	-6.933
3	BESO	ROSALÍA, Rauw Alejandro	RR	2023-03-24	95.050	0.768	0.644	-6.671
4	HIBIKI	Bad Bunny, Mora	nadie sabe lo que va a pasar mañana	2023-10-13	119.935	0.801	0.645	-5.605
5	You	Belly, Kehlani	Another Day In Paradise	2016-05-27	96.011	0.658	0.700	-1.974
6	A Night To Remember	beabadoobee, Laufey	A Night To Remember	2023-10-20	114.990	0.726	0.624	-7.799
7	soso	Omah Lay	Boy Alone	2022-07-14	105.016	0.711	0.725	-8.315
8	Cool With You	NewJeans	NewJeans 2nd EP 'Get Up'	2023-07-21	134.993	0.752	0.650	-6.683
9	Lifting You	Jungle	Loving In Stereo	2021-08-13	107.025	0.674	0.671	-8.352
10	Si Tu Novio Te Deja Sola	J Balvin, Bad Bunny	Si Tu Novio Te Deja Sola	2017-03-03	121.927	0.718	0.460	-9.498

Figure 3: Ten content based song recommendations including musical features for each track from our music recommendation system.

Linear regression analysis showed positive correlation between musical features of danceability and energy for various songs input into our model and their respective output song recommendations



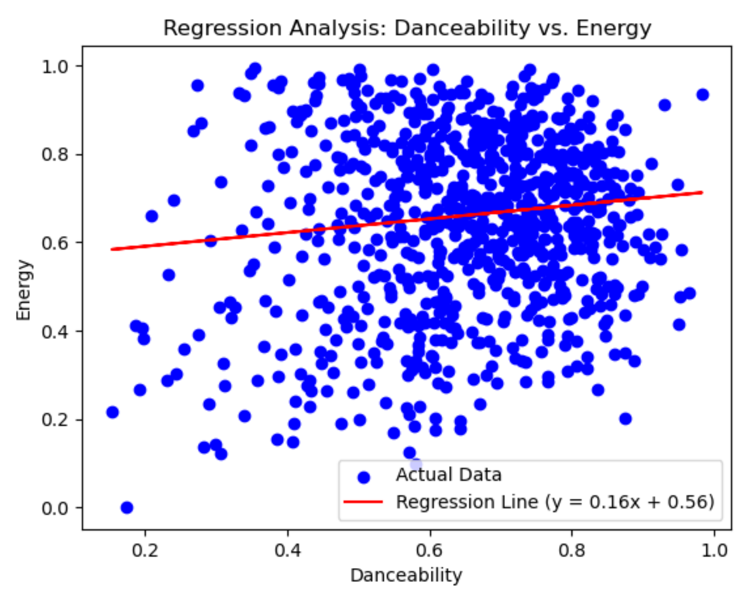


Figure 4: (Left) Linear fit analysis of danceability vs. energy scores of four song inputs and their respective output recommendations from our model. (Right) Regression analysis of danceability vs. energy scores of every song collected from Spotify API and fed into our model.

The left plot in Figure 4 displays four scatter plots (one for each song) of the association between each input song and its ten song recommendations provided by our model. While the strength of each association varied for the four songs assessed, danceability and energy scores were found to be positively correlated in each case. To ensure these findings were consistent with expectations, that danceability and energy scores were anticipated to have a positive association for songs in general—we then performed a linear regression analysis for these musical features on all of the songs in our model's dataset. The right plot in Figure 4 shows the regression output for our dataset of over 4000 songs' danceability and energy scores, demonstrating that danceability and energy scores for an individual song are positively associated on a larger scale. Thus, asserting that the recommended songs provided by our model are effective in terms of musical features, in particular danceability and energy.

After analyzing the correlations between musical features for our model's inputs and recommendations, we further assessed our model's effectiveness through user satisfaction. Using Google Forms, we collected 15 responses indicating the proportion of songs each person enjoyed from their 10 provided recommendations. Figure 4 below shows a baseline predictive model for the expected number of liked songs from our system's recommendations based on our user feedback trials.

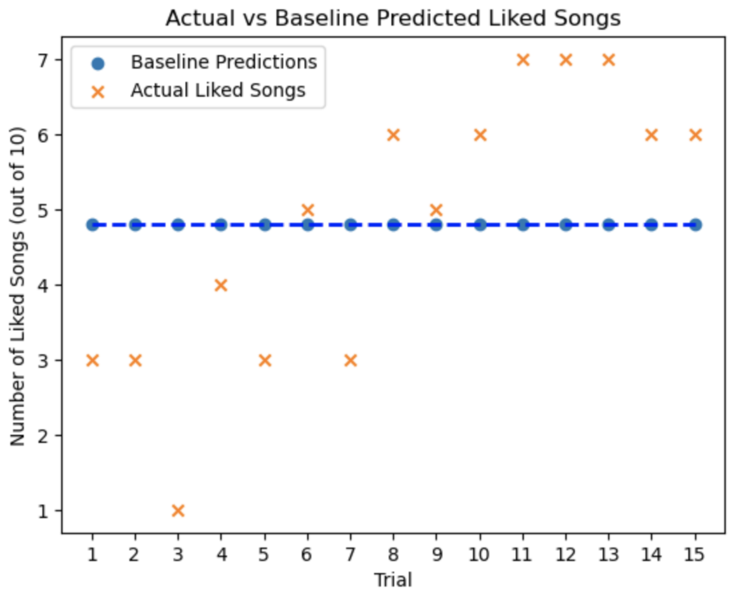


Figure 5: Plot displaying number of liked songs for each trial conducted (yellow x's) and baseline prediction for expected number of liked songs (blue line).

The yellow 'x' markings represent the actual number of songs each person liked from the 10 recommended for 15 total users surveyed. The baseline model, indicated by the blue line of data points, predicts the number of songs users will like from our model's recommendations for future trials. The baseline prediction was generated using average user satisfaction from our 15 surveys which gave an approximate number of 5 liked songs from 10 recommendations, implying that our model currently has about 50% satisfaction. Performing a much larger number of user survey trials would likely result in a smaller variation in a number of liked songs, potentially increasing current user satisfaction—but would certainly provide more consistency in a number of liked songs due to having more data for computing the baseline prediction.

Although our project demonstrated the music recommendation system based on the Spotify track data, there exist several limitations to our project. First of all, the data we collected for the track data wasn't enough though we collected over 4000 tracks. As the music industry is active, the amount of songs released on Spotify is abundant. Over 4000 tracks is a huge amount of data itself, but compared to the whole massive big data that Spotify Music data has, it is not a sufficient amount of data collected. Thus, the more data we collect the more accuracy will be driven on our model(Ray, Sunil). Another limitation is that we weren't able to interpret user's preferences in the model. We solely used music properties for the recommendation system, if we interpreted user preferences and supervised it in machine learning, we could have been able to get better accuracy on our survey itself.

Conclusion

Overall, our project successfully demonstrates the effectiveness of a content-based music recommendation system leveraging Spotify's Web API data. The positive scatter plot associations we observed between input and output songs' musical features, along with the positive preliminary user satisfaction feedback asserts our system's potential for personalized music discovery. To build upon our current model and results, future efforts could incorporate music genre data for a more comprehensive analysis of song similarities that may improve the accuracy of our model's predictions. Expanding the dataset to include a more extensive song library would also help refine our recommendation model. Additionally, additional testing on a much larger user sample would allow for additional methods of model evaluation and offer insight into potential adaptations to be made for our system. Finally, expanding our model's capability to directly process user streaming data, rather than relying on manual song input, could enhance our system's practicality and user experience. These potential directions for improvement of our model emphasize the continuous evolution of personalized music recommendation systems in the dynamic music streaming industry.

Roles

Isabella: creating Spotify web application, obtaining and visualizing user data, evaluating model accuracy, Report: abstract, results/discussion, conclusion

Siyeon: creating Spotify web API application, data collection & data preprocessing, building music recommendation model, visualizations of data, Report: Introduction, Method

References

Dey, Victor. "Collaborative Filtering vs Content-Based Filtering for Recommender Systems." *Analytics India Magazine*, 30 May 2022, analyticsindiamag.com/collaborative-filtering-vs-content-based-filtering-for-recommender-systems/.

Karabiber, Fatih, et al. "Cosine Similarity." *Learn Data Science - Tutorials, Books, Courses, and More*, www.learndatasci.com/glossary/cosine-similarity/. Accessed 19 Dec. 2023.

Ray, Sunil. "8 Ways to Improve Accuracy of Machine Learning Models (Updated 2023)." *Analytics Vidhya*, 25 Sept. 2023, www.analyticsvidhya.com/blog/2015/12/improve-machine-learning-results/#:~:text=Having%20more%20data%20is%20always,more%20accurate%20machine%2Dlearning%20models.

"Sklearn.Preprocessing.MinMaxScaler." *Scikit*, scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html. Accessed 19 Dec. 2023.