Music Recommendation System

Mid-Project Review

Isha Sehrawat
2019046

Srishti Singh
2019114

Bhavesh Sood
2019355

PROBLEM STATEMENT

With the vast amount of songs in the streaming services database, it's overwhelming for users to be up to date with the newest releases and the songs matching their mood. While many music recommendation approaches are available, they rarely take into account the user's emotions at the current time. They have underutilized the lyrics of the songs that the users have listened to in the past. We are planning to create a novel music recommendation system that takes emotions, user's playing history as well as preferences and the lyrics of the music to recommend the most relevant songs and achieve user satisfaction.

LITERATURE REVIEW

Music recommender systems based on dataset features have been the focus of significant research in recent years. A wide variety of features are used from the dataset ranging from audio features, user and artist metadata, and emotions associated with songs to train such systems.

Oord et al. [1] proposed a deep neural network model for music recommendation. They used raw audio data to extract more relevant information, like timbre, melody, rhythm, and harmony, in a compact and meaningful way. The focus was on getting latent characteristics from different song audio. To extract these representative features from songs, two approaches were used. The bag of words approach extracts MFCCs from audio and a deep learning approach that uses CNNs consisting of alternating feature extraction and pooling layers. The paper showed how audio data features though not sufficient alone, but provide sensible recommendations.

Oramas et al. [2], in their research, presented a multimodal approach to recommend songs. One model is trained to learn feature embeddings for the artists based on their biographies. The second model learns the song feature characteristics from the audio data files. These two models are then combined to get the final track features based upon which suggestions are made. The qualitative analysis of the MSD showed improvement, which suggests a new approach to use artists' metadata.

Wang et al. [3] have pointed out in their recent work how the performance of the current recommender system is bounded due to limited contextual information. Thus, they proposed a new music recommender system that integrates content and contextual information. Their Hybrid music recommendation system utilizes graph embedding techniques to generate dense feature representations of songs in a low-dimensional space, allowing for more effective similarity comparisons between songs. Recommendations are then generated based on cosine similarity between users, and song embeddings learned from graphs.

La Gatta et al. [4] propose using a hypergraph data model. The model consists of three parts: hypergraph, embedding generator, and recommendation generator. The hypergraph data model is used to represent complex interactions between users and songs, while the embedding generator encodes context-based information in low-dimensional vectors. The recommendation generator then computes the similarity score between the embeddings to rank the results, generating the final recommendation

Wang Pengfei et al. [5] proposed yet another new approach that exploits the sequential nature of the user's music listening history. Apart from using user metadata, it makes use of transactions too.

X. Luo et al. [7] presents a novel approach to collaborative filtering for recommender systems, which is based on non-negative matrix factorization (NMF). The authors argue that this approach has several advantages over traditional collaborative filtering methods, including improved accuracy, scalability, and interpretability. The paper involves decomposing the user-item rating matrix into two non-negative matrices - a user latent factor matrix and an item latent factor matrix. The latent factors(LFM) capture the underlying features that explain the observed ratings.

Model	**Dataset Features**
Deep Content Based [1]	Audio
Deep Multimodel [2]	Audio, Artists Biography
CAME [3]	Content and Context
HEMR [4]	Song Metadata
HRM [5]	Sequence
LFM [7]	Simple ratings/click data.

Ours	Lyric features, Sentiment

PROPOSED SOLUTION AND METHODOLOGY

Most of the papers we came across used features like audio, timestamp, user history, and lyrics content to perform recommendations. Lyrics are another crucial feature of songs that can be used to extract more information, like emotions associated with it. We plan on utilizing some new features to improve upon the recommendation quality. We will make use of sentiment pertaining to the lyrics and use CNNs to get a more representative embedding for the lyrics. We will also try adding song popularity ranking to make predictions.

DATASET

For our project, we are using either subset of or data spawned from The Million Song Dataset. The two datasets we are using for our project are Song_data.csv and 10000.txt.

song_data.csv - This is a csv file that contains song_id, title, release(album name), artist_name and year(of release).

10000.txt - This dataset contains count play of 10,000 songs, including the user ID, song ID, and the number of times each song was played by the user.

LIBRARIES IMPORT

The code begins by importing various Python libraries such as numpy, torch, argparse, cv2, os, PIL, sklearn, matplotlib, seaborn, and tqdm, among others. These libraries are used to manipulate data, create machine learning models, visualize data, and perform other functions.

PREPROCESSING

Merging the two datasets

We merged the two datasets by song ID for easy access. Merging will enable us to work with song names instead of song IDs. Our code groups the song_data.csv data by song ID and selects the maximum values for each group. This effectively removes any duplicate song entries. The resulting dataframe is then merged with the 10000.txt data based on song ID. After the merge, the play_count column is renamed to listen_count, and the song_id column is dropped.

The resulting dataframe after removing duplicate entries -

```
Number of rows after unique song ID treatment: 999056
Number of unique songs: 999056
```

	song_id	title	release	artist_name	year
0	SOAAABI12A8C13615F	Afro Jazziac	To Birdland And Hurry	Herbie Mann	2000
1	SOAAABT12AC46860F0	Herre Gud Ditt Dyre Namn Og Ære	Som Den Gyldne Sol Frembryter	Bergen Big Band	0
2	SOAAABX12A8C13FEB2	N.Y.C. Remix	Paris Can´t Wait	Guardner	0
3	SOAAACR12A58A79456	Irresistible	Wowie Zowie	Superchumbo	2002
4	SOAAACY12A58A79663	Untitled 1	Pine Cone Temples	Thuja	0

The 10000 songs dataframe -

	user	song	play_count
0	b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOAKIMP12A8C130995	1
1	b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOBBMDR12A8C13253B	2
2	b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOBXHDL12A81C204C0	1
3	b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOBYHAJ12A6701BF1D	1
4	b80344d063b5ccb3212f76538f3d9e43d87dca9e	SODACBL12A8C13C273	1

Merged dataframe -

	user	song	listen_count	title	release	artist_name	year
0	b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOAKIMP12A8C130995	1	The Cove	Thicker Than Water	Jack Johnson	0
1	b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOBBMDR12A8C13253B	2	Entre Dos Aguas	Flamenco Para Niños	Paco De Lucia	1976
2	b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOBXHDL12A81C204C0	1	Stronger	Graduation	Kanye West	2007
3	b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOBYHAJ12A6701BF1D	1	Constellations	In Between Dreams	Jack Johnson	2005
4	b80344d063b5ccb3212f76538f3d9e43d87dca9e	SODACBL12A8C13C273	1	Learn To Fly	There Is Nothing Left To Lose	Foo Fighters	1999
...	...	...	...	...	...	...	...
1999995	d8bfd4ec88f0f3773a9e022e3c1a0f1d3b7b6a92	SOJEYPO12AAA8C6B0E	2	Ignorance (Album Version)	Ignorance	Paramore	0
1999996	d8bfd4ec88f0f3773a9e022e3c1a0f1d3b7b6a92	SOJJYDE12AF729FC16	4	Two Is Better Than One	Love Drunk	Boys Like Girls featuring Taylor Swift	2009
1999997	d8bfd4ec88f0f3773a9e022e3c1a0f1d3b7b6a92	SOJKQSF12A6D4F5EE9	3	What I've Done (Album Version)	What I've Done	Linkin Park	2007
1999998	d8bfd4ec88f0f3773a9e022e3c1a0f1d3b7b6a92	SOJUXGA12AC961885C	1	Up	My Worlds	Justin Bieber	2010
1999999	d8bfd4ec88f0f3773a9e022e3c1a0f1d3b7b6a92	SOJYOLS12A8C13C06F	1	Soil_ Soil (Album Version)	The Con	Tegan And Sara	2007

2000000 rows × 7 columns

SENTIMENT ANALYSIS ON SONG TITLE

Sentiment analysis is a technique used to extract the emotional tone of a piece of text, such as a song title, by analyzing its words and context. In the case of a song title, it can help determine the emotional theme or message that the title is conveying to its listeners.

For example, if the title of a song is "Broken Heart," we may conclude that the emotional tone of the song is sad or melancholic, as the title suggests that the song is about heartbreak. On the other hand, if the title is "Happy Days," it may indicate that the emotional tone of the song is positive and upbeat.

Sentiment analysis can be performed manually or using automated tools, such as machine learning algorithms. These algorithms can be trained to analyze various factors that contribute to the emotional tone of a song title, such as word choice, language style, and cultural context.

However, it's important to note that sentiment analysis may not always accurately reflect the emotional tone of a song title. For example, a song titled "Tears of Joy" may initially suggest sadness or grief, but upon listening to the song, it may become clear that the title is actually ironic or sarcastic, conveying a message that is positive or humorous.

In conclusion, sentiment analysis can be a valuable tool for understanding the emotional tone of a song title, but it's important to consider the context and overall message of the song before making any definitive conclusions.

We have used Vader Sentiment library to extract sentiment scores related to different song titles present in our dataset. We in the future experiments, intend to extract more sentiment oriented features from the lyrics of these songs.

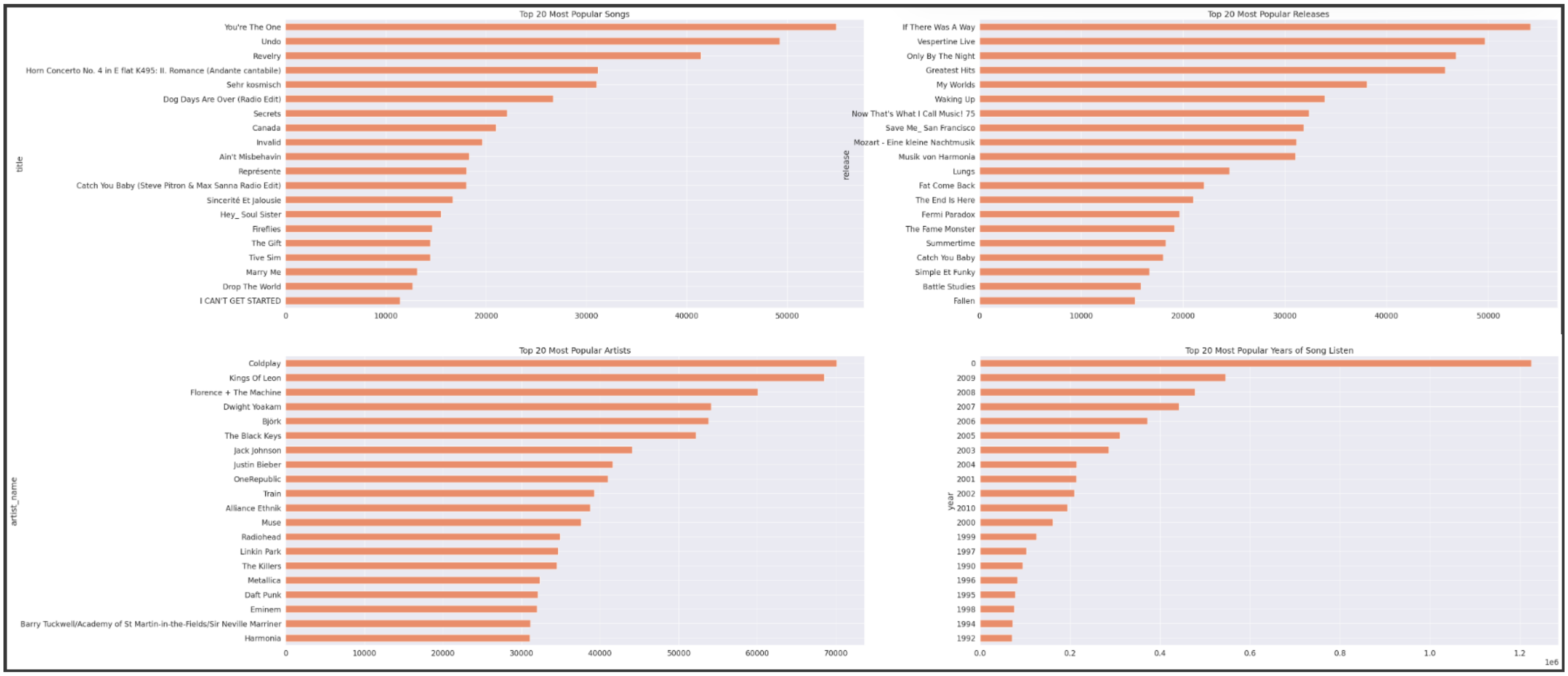
EXPLORATORY DATA ANALYSIS (EDA)

After preprocessing, we perform exploratory data analysis (EDA) to understand the dataset's characteristics.

It begins by generating a summary of the 'listen_count' column, which shows that a song was played 2,213 times by a single user.

```
count    2.000000e+06
mean     3.045485e+00
std      6.579720e+00
min      1.000000e+00
25%      1.000000e+00
50%      1.000000e+00
75%      3.000000e+00
max      2.213000e+03
Name: listen_count, dtype: float64
```

Next, the code creates four subplots to display the top 20 most popular songs, releases, artists, and years from user play data. It groups the data by title, release, artist_name, and year, respectively, and then sorts them in descending order based on the number of times they were played. It then creates horizontal bar charts for each of the top 20 lists and adds them to the subplots. Below is the screenshot of the bar plots of the top 20 in each different feature of a song(song title, album(release), artist and year).



DIFFERENT RECOMMENDATION ENGINES

Recommendation engines filter out products a user might be interested in based on their previous history. But if the previous history of a user is not available, then different methods are used.

As part of our investigation into music recommendation systems, we examined two different approaches to generating recommendations in the absence of a user's listening history. Our aim was to establish a baseline for comparison with our proposed recommendation engine that we will be doing as part of our project.

Popularity Based Recommendation Engine

The first method we examined was a popularity based recommendation engine, which recommends the songs or artists based on their global popularity among all the users in our dataset.

Fig: Recommendations on the basis of popular songs -

	title	score	rank
6837	Sehr kosmisch	8277	1.0
8726	Undo	7032	2.0
1965	Dog Days Are Over (Radio Edit)	6949	3.0
9497	You're The One	6729	4.0
6499	Revelry	6145	5.0
6826	Secrets	5841	6.0
3438	Horn Concerto No. 4 in E flat K495: II. Romanc...	5385	7.0
2596	Fireflies	4795	8.0
3323	Hey_ Soul Sister	4758	9.0
8495	Tive Sim	4548	10.0
8781	Use Somebody	3976	11.0
5721	OMG	3947	12.0
2120	Drop The World	3879	13.0
5000	Marry Me	3578	14.0
1265	Canada	3526	15.0

Fig: Recommendations on the basis of popular artists -

	artist_name	score	rank
649	Coldplay	29422	1.0
2850	The Black Keys	19862	2.0
1651	Kings Of Leon	18747	3.0
1107	Florence + The Machine	18112	4.0
1370	Jack Johnson	17801	5.0
2946	The Killers	16063	6.0
2374	Radiohead	14890	7.0
736	Daft Punk	14715	8.0
2073	Muse	14005	9.0
1554	Justin Bieber	13959	10.0

While this approach is straightforward and easy to implement, it has some disadvantages. Firstly, it does not take into account the unique preferences, listening history and many other factors of individual users which can limit the user experience and the overall engagement of the user. Secondly, it is solely based on how often a song or artist has been played, which might be biased. In other words, the recommendations generated using this technique may not reflect a user's personal taste or current mood. However, this method is useful when a user's listening history is not available or insufficient.

Item Similarity based Recommendation Engine

The second method we evaluated was the item-based recommendation engine. This method focuses on the similarity between the user's list of songs (listening history, their playlists, etc.) and song data for other users that are present in the training data. This method calculates the similarity between the user list of songs and songs in our dataset using the Jaccard index based on common users of songs. Two songs can be considered similar to each other if a significant portion of the same users listens to both out of the total number of listeners.

This method provides more personalized recommendations based on the user's listening history and preferences, thus improving the overall user experience and engagement.

Fig: Recommendation list -

```
                                    user_id                          song  \
0  b80344d063b5ccb3212f76538f3d9e43d87dca9e                  Quiet Houses
1  b80344d063b5ccb3212f76538f3d9e43d87dca9e                   Meadowlarks
2  b80344d063b5ccb3212f76538f3d9e43d87dca9e           Heard Them Stirring
3  b80344d063b5ccb3212f76538f3d9e43d87dca9e  Tiger Mountain Peasant Song
4  b80344d063b5ccb3212f76538f3d9e43d87dca9e                  Sun It Rises
5  b80344d063b5ccb3212f76538f3d9e43d87dca9e                Your Protector
6  b80344d063b5ccb3212f76538f3d9e43d87dca9e                  Oliver James
7  b80344d063b5ccb3212f76538f3d9e43d87dca9e                 Great Indoors
8  b80344d063b5ccb3212f76538f3d9e43d87dca9e           White Winter Hymnal
9  b80344d063b5ccb3212f76538f3d9e43d87dca9e                    If I Could

      score rank
0  0.044710    1
1  0.043836    2
2  0.042740    3
3  0.041485    4
4  0.040973    5
5  0.039942    6
6  0.039287    7
7  0.036765    8
8  0.036345    9
9  0.034576   10
```

We compared the results by listening to the top song in the recommendation list and the song most listened to by the user.

Fig: Top 2 songs of Users from their most listened songs lists. The above recommended songs are quite similar to the user's most listened to song, indicating that the recommendation system was successful in providing personalized and relevant suggestions.

	user	song	listen_count	title	release	artist_name	year
16	b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOMGIYR12AB0187973	6	Behind The Sea [Live In Chicago]	Live In Chicago	Panic At The Disco	0
43	b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOYHEPA12A8C13097F	8	Moonshine	Thicker Than Water	Jack Johnson	2003

However, even the item based recommendation system may not capture the diversity and novelty of user preferences, especially for users who have eclectic or changing tastes. Item-based recommendation relies on finding similar items based on ratings or interactions from other users, but this may not reflect the actual content or attributes of the songs, such as genre, mood, tempo, etc. Therefore, some users may find item-based recommendations too repetitive or predictable [6]. Moreover, it suffers from the cold start problem when dealing with new or unpopular items.

Latent Factor Model based Recomendation (Matrix Factorisation):

Matrix factorization is a powerful technique used in machine learning to discover latent features between two different kinds of entities. It has been successfully applied to a wide range of domains, including recommendation systems, image processing, and natural language processing.

Matrix factorization can also be used to discover latent features of songs, such as beats, tempo, and other musical features. Once these features have been defined, they can be used to find matches for a user based on some similarity criteria, such as recommending songs that have similar beats or tempo to songs that the user has previously liked.

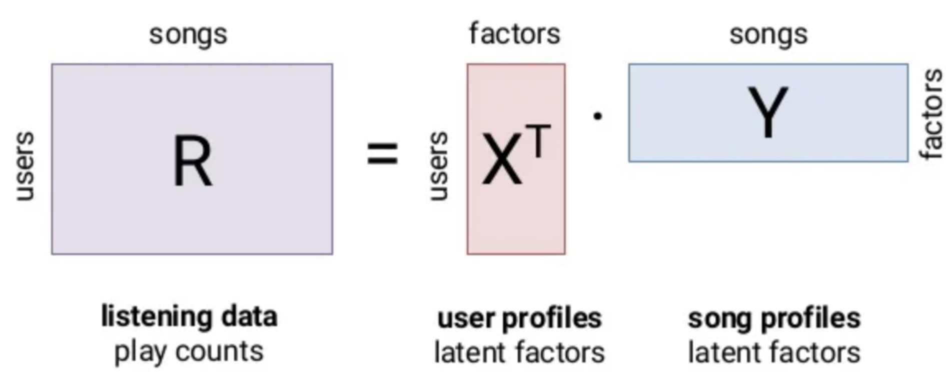
The algorithm aims to minimize the reconstruction error between the observed listen counts and the predicted listen counts obtained by the product of the user and songs latent factor matrices.

Assuming the process helps us identify latent factors/features, meaning as K, our aim is to find two matrices X and Y such that their product (matrix multiplication) approximates R.

X = |U| x K matrix (A matrix with dimensions of num_users * factors)

Y = |P| x K matrix (A matrix with dimensions of factors * num_songs)

There are multiple algorithms available for determining factorization of any matrix. We use one of the simplest algorithms, which is the singular value decomposition or SVD.



Since, we don't have a 'rating' in our database, we rely on play counts. We replace the play count with a fractional play count to measure the 'likeness' of a song within the range of [
0, 1].

Fig: Table with fractional play counts of songs for user 'd6589314c0a9bcbca4fee0c93b14bc402363afea'

	user	song	listen_count	fractional_play_count
299	d6589314c0a9bcbca4fee0c93b14bc402363afea	SOADQPP12A67020C82	12	0.038961
300	d6589314c0a9bcbca4fee0c93b14bc402363afea	SOAFTRR12AF72A8D4D	1	0.003247
301	d6589314c0a9bcbca4fee0c93b14bc402363afea	SOANQFY12AB0183239	1	0.003247
302	d6589314c0a9bcbca4fee0c93b14bc402363afea	SOAYATB12A6701FD50	1	0.003247
303	d6589314c0a9bcbca4fee0c93b14bc402363afea	SOBOAFP12A8C131F36	7	0.022727

We tried to find 10 recommendations for user 4.

Fig: Recommendations for user 4

```
Recommendation for user id 4
The number 1 recommended song is 911 BY Wyclef Jean / Mary J. Blige
The number 2 recommended song is If I Ain't Got You BY Alicia Keys
The number 3 recommended song is Sample Track 2 BY Simon Harris
The number 4 recommended song is Kennedy Rag BY Suzy Thompson
The number 5 recommended song is Clocks BY Coldplay
The number 6 recommended song is Yellow BY Coldplay
The number 7 recommended song is Hey_ Soul Sister BY Train
The number 8 recommended song is The Scientist BY Coldplay
The number 9 recommended song is Eco BY Jorge Drexler
The number 10 recommended song is Kryptonite BY 3 Doors Down
```

CONCLUSION

In conclusion, our investigation into the music recommendation system highlighted the importance of providing personalized and diverse recommendations to the users based on their unique preferences. While the popularity-based recommendation method can be useful in the absence of user history, the item-based similarity recommendation method provides more personalized recommendations based on user listening history. However, both methods have their limitations and more advanced recommendation techniques are necessary.

To create a more personalized recommendation, it is important to take into account the features of the songs, such as genre, mood, tempo, and lyrics, and use them to find similarities and differences between songs.

In addition to above, we plan on incorporating lyrics sentiment analysis into our music recommendation system. This technique involves analyzing the emotional states and expressions conveyed in the lyrics of a user's listening history. By doing so, we can gain insights into the user's emotional state and preferences, which can help us find songs that match or contrast those emotions.

We believe that this approach will help users discover new and exciting songs that they may have missed and improve their overall engagement with the music recommendation system.

REFERENCES

[1] Van den Oord, A., Dieleman, S. and Schrauwen, B., 2013. Deep content-based music recommendation. *Advances in neural information processing systems*, *26*.

[2] Oramas, S., Nieto, O., Sordo, M. and Serra, X., 2017, August. A deep multimodal approach for cold-start music recommendation. In *Proceedings of the 2nd workshop on deep learning for recommender systems* (pp. 32-37).

[3] Wang, D., Zhang, X., Yu, D., Xu, G. and Deng, S., 2020. Came: Content-and context-aware music embedding for recommendation. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(3), pp.1375-1388.

[4] La Gatta, V., Moscato, V., Pennone, M., Postiglione, M., & Sperlí, G. (2022a). Music Recommendation via Hypergraph Embedding. IEEE Transactions on Neural Networks and Learning Systems, 1–13. https://doi.org/10.1109/TNNLS.2022.3146968.

[5] Wang, P., Guo, J., Lan, Y., Xu, J., Wan, S. and Cheng, X., 2015, August. Learning hierarchical representation model for nextbasket recommendation. In *Proceedings of the 38th International ACM SIGIR conference on Research and Development in Information Retrieval* (pp. 403-412).

[6] Sunitha, M., Adilakshmi, T., Ali, M.Z. (2021). Enhancing Item-Based Collaborative Filtering for Music Recommendation System. In: Satapathy, S.C., Bhateja, V., Favorskaya, M.N., Adilakshmi, T. (eds) Smart Computing Techniques and Applications. Smart Innovation, Systems and Technologies, vol 224. Springer, Singapore. https://doi.org/10.1007/978-981-16-1502-3_28.

[7] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrixfactorization-based approach to collaborative filtering for recommender systems," IEEE Trans. Ind. Informat., vol. 10, no. 2, pp. 1273–1284,

May 2014.