Lucerne University of
Applied Sciences and Arts

HOCHSCHULE
LUZERN

Business

Recommender Systems

Submitted by

Tejesh Reddy Koki

Matthias De Paolis

Submitted to
Dr. Guang Lu

This paper is submitting as part of module requirement at Lucerne University of Applied Sciences at Arts on 07.03.2023.

Contents

Table of Figures 3
Chapter-1: Introduction 4
1.1 Brief about Deezer 4
1.2 Comparison with competitors 4
Chapter-2: Developing Algorithm 5
2.1 Content-based music recommendation: 7
2.2 Collaborative filtering: 7
Chapter-3: Recommendation system proposal 7
3.1 How LightFM works: 8
3.2 Why LightFM: 8
3.3 Implementation of LightFM: 8
Chapter-4: Actual vs Proposal 11
Chapter-5: Conclusion 13
References 14
Appendix 15
Reflection Paper – Recommender Systems 16

Table of Figures

Figure 1: Global streaming music subscription market, Q2 2022 6
Figure 2: Correlation Matrix from the Dataset 9
Figure 3: Gender Distribution 9
Figure 4: "is_listened" variable distribution 10
Figure 5: Model Evaluation 10
Figure 5: Content-Based Filtering Method 11
Figure 6: Collaborative Filtering Method 12
Figure 7: Matrix Factorization Method (General) (Lara-Cabrera et al., 2020) 12
Figure 8: LightFM Model 13

Chapter-1: Introduction

1.1 Brief about Deezer

Deezer is a music streaming service that presents a comprehensive assortment of music and podcast content to its users. Established in 2007, Deezer has evolved into a leading music streaming platform globally, catering to over 16 million monthly active users across 180 countries. The platform's expansive library of music comprises over 73 million songs, encompassing content from popular and independent artists spanning multiple genres, including pop, rock, hip-hop, electronic, and classical music, among others. As such, Deezer provides a platform that caters to the diverse musical preferences of its users. Deezer's personalized playlist and music recommendation system serves as a noteworthy feature of the platform, enabling users to explore new artists and songs that align with their music taste based on their listening history, mood, and preferences. Another feature that distinguishes Deezer from its peers is the Deezer HiFi service, providing users with a lossless, CD-quality audio streaming experience. In addition, Deezer provides other features such as the option to create personalized playlists, access to song lyrics, and social media sharing options. Deezer also encompasses a vast collection of podcasts that cover a range of topics, such as news, sports, and lifestyle, among others. This vast collection of audio content caters to users across all subscription tiers, providing a one-stop-shop for audio content.

The platform is accessible across various devices, including Android, iOS, Windows devices, and the web. Deezer offers both a free and premium subscription tier, with the latter offering additional features such as ad-free streaming and offline listening. In summary, Deezer is a robust music streaming platform that caters to the diverse musical preferences of its users through a vast collection of music and podcasts. With its music discovery capabilities and other features, Deezer presents an excellent option for users who seek to explore new music and enjoy their favourite content in high quality. Deezer is a popular music streaming platform that offers its users access to a vast music library (Deezer, n.d.).

One of the key features that sets Deezer apart from other music streaming platforms is its recommendation algorithm, which uses machine learning techniques to analyse the listening habits of users and suggest new music based on their preferences (Bontempelli T. et al., 2022). This personalized approach to music discovery ensures that users have a unique listening experience that matches their individual musical tastes. Deezer's recommendation algorithm uses a collaborative filtering approach, where it analyses the listening habits of users and compares them to those of other users who have similar tastes (Papadopoulos, Manolopoulos, & Theodoridis, 2016). It then suggests new tracks that are popular among these similar users, increasing the chances that users will enjoy the recommended tracks. The algorithm also takes into account the user's history of skipped tracks, likes, and dislikes to further refine its recommendations.

Deezer's recommendation algorithm is also enhanced by other features, including Flow, Mixes, and Smart Playlists. Flow is a unique feature that allows users to create a customized radio station that plays a mix of their favourite tracks and new recommendations. Mixes are playlists that are created based on the user's listening habits and preferences, while Smart Playlists are automatically generated based on the user's mood or activity (Deezer, n.d.). In addition to its recommendation algorithm, Deezer focuses on supporting emerging artists and providing access to local music. The platform has a feature called Deezer NEXT, which supports emerging artists by providing them with exposure and promotional opportunities (Deezer, n.d.). Deezer also offers local music content in over 180 countries, ensuring that users have access to a wide range of music from different cultures and regions.

1.2 Comparison with competitors

Music streaming has become an increasingly popular way to consume music, with various music streaming services available to choose from. One of the significant aspects of music streaming is the music recommendation system, which provides users with personalized content based on their listening history and preferences. In this subchapter we want to compare Deezer with other music streaming providers, including Spotify, Apple Music, and Tidal, based on their music recommendation systems.

Deezer's recommendation system provides users with personalized content through its Flow feature, which creates a playlist based on the user's listening history, mood, and preferences. This personalized playlist is updated regularly, providing users with a continually evolving playlist that aligns with their music taste (Deezer, n.d.). In comparison, Spotify's recommendation system provides users with personalized playlists, including its popular "Discover Weekly" and "Daily Mix" playlists (Spotify, n.d.). Apple Music's recommendation system provides similar features, such as the "For You" section, which provides tailored recommendations (Apple, n.d.). Tidal, on the other hand, provides playlists curated by music experts (Tidal, n.d). Deezer's recommendation system presents an advantage in terms of its personalized and evolving playlist feature.
Additionally, Deezer's recommendation system also includes a "Flow to the Lyrics" feature, which provides users with lyrics to songs as they listen to them. This feature enhances the overall listening experience, allowing users to engage with their favourite music on a deeper level (Deezer, n.d.). Spotify's recommendation system also includes a similar feature, providing users with lyrics to songs they are listening to (Spotify, n.d.). Apple Music and Tidal do not provide lyrics feature as part of their recommendation system (Apple, n.d.).

Another aspect to consider is the ability to discover new artists and music through the recommendation system. Deezer's recommendation system provides users with new music recommendations based on their listening history and preferences (Deezer, n.d.). In comparison, Spotify's recommendation system provides new music recommendations through its "Release Radar" feature, which presents new releases from artists that the user has shown interest in (Spotify, n.d.). Apple Music provides new music recommendations through its "New Music Mix" feature, while Tidal's music discovery capabilities are more limited. Deezer and Spotify present similar music discovery capabilities, with Apple Music offering slightly more personalized content. Moreover, the quality of the audio streaming experience is another critical aspect to consider. Deezer's HiFi service delivers a lossless, CD-quality audio streaming experience, providing users with the best possible listening experience. In comparison, Spotify and Apple Music provide high-quality audio streaming, but not at the same level as Deezer's HiFi service. Tidal, however, delivers high-fidelity streaming that is comparable to Deezer's HiFi service, presenting an advantage in terms of audio quality.

Deezer's recommendation system presents a compelling offering to users, with its personalized and evolving playlist feature and "Flow to the Lyrics" feature. In comparison to its peers, Deezer provides similar music discovery capabilities and pricing, with only minor differences in audio quality (Deezer, n.d.). The importance of the recommendation system in music streaming cannot be overstated, and Deezer's offering presents an excellent option for users who value personalized content and discovering new music.

Chapter-2: Developing Algorithm

In recent years, the music industry has experienced an unprecedented level of internationalization. It is anticipated that the global music market will grow at a CAGR of 8.5% over the next five years (Music Market Landscape, 2023). It is foreseen that due to functionalities like song recommendations, playlist customization, offline availability and hassle-free connectivity on apps and browsers, more and more people are using the music applications. Surprisingly, in spite of the world's economic downturn, the music streaming market remains strong(Mulligan, 2022). For music industries it is really challenging to maintain the hype in the industry even during the recession periods. The Music streaming services are using various technologies to be more near to the users providing the best services ever they can offer and competing their competitors.

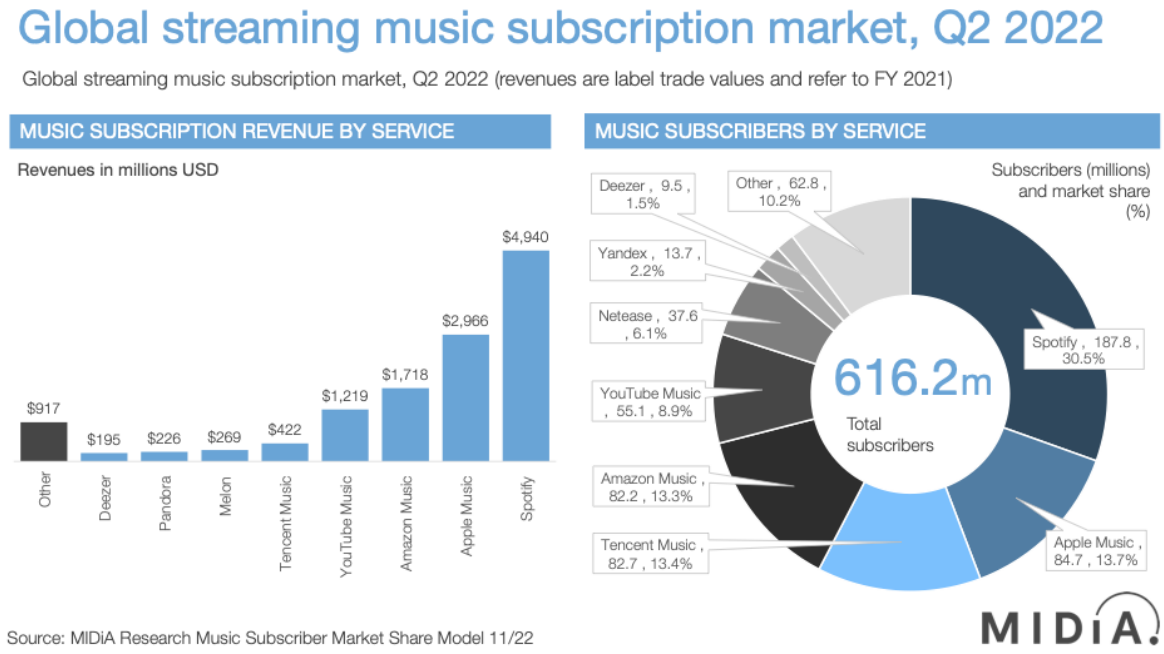

Figure 1: Global streaming music subscription market, Q2 2022

The popular music streaming service is Spotify with 30.5 % of subscribers, where Apple, Tencent and Amazon music holding 13% each. Overall, various other companies like YouTube music, NetEase, Yandex and Deezer counted approximately 20% of the subscribers. For a streaming service like Deezer it would be a big challenge to win their competitors and stand among the competitors.

In general, besides a good recommendation algorithm the factors that make streaming company successful are as follows,

User experience: The platform should make sure to provide a seamless and intuitive experience to the users.
Content: The steaming company should focus on its music library and provide latest music and songs by adding new artists regularly. Additionally, should be able to add exclusive content to attract the new users.
Personalization: The users should be able to personalize the application where they should be delivered with personalized recommendation and playlists based on their listening history and preferences.
Platform Compatibility: The application should be compatible with a wide range of devices and operating systems, including smart phones, tablets, laptops, and smart speakers.
Branding and Marketing: The company should focus on building a strong brand image and invest in marketing to increase the brand awareness and reach to potential users.
Customer Service: Despite of all other factors, the platform should make sure that the customer service is responsive and helpful. They should provide support services through various channels like live chat, email and queries.

By utilizing advanced machine learning algorithms and data analytics techniques, many music companies develop recommendations systems to compete by providing users with personalized music recommendations. Some key steps that these companies follow to develop the recommendation system are as follows:

i. Collect Data: Music companies collect a large amount of data about their customers. For instance, they track listening history, music preferences, and other behaviors of their customers.

ii. Data Preprocessing: The collected data is preprocessed and transformed to remove any inconsistencies, inaccuracies, and irrelevant information that may hinder the recommendation process.

iii. Algorithm Selection: A recommendation system is built using a variety of machine learning algorithms, including collaborative filtering, content-based filtering, and hybrid approaches.

iv. Training and Testing: Data preprocessing is used to train the algorithm. Once trained, the algorithm provides relevant recommendations based on the data.

v. User Feedback: To improve accuracy and relevance, the recommendation system is continuously updated based on user feedback.

vi. Integration: Music companies integrate recommendation systems into their platforms to provide personalized recommendations, such as online stores or streaming services.

Most of the recommend systems indeed depend on user preferences, usage patterns, the combination of songs that users have listened, how the items related to each other, the ratings to songs the users given. The approach to this recommends system is called collaborative filtering approach. The approach to predict user preferences from content and metadata is called content-based approach.

2.1 Content-based music recommendation:

In this model, the music is recommended based on the available metadata information such as artist, album, year of release etc., which is known. The drawback of this approach is the recommendation are predictable. For example, to recommend songs to the user from the artist which already known is not so useful. But using this approach the similar songs can be recommended based on the history of the songs that user previously listened by measuring the audio signals (Jan Schluter 2011). To implement this approach, one should know the suitable similarity metric.

2.2 Collaborative filtering:

In the collaborative filtering model, the method relies on the similarity measure between users and items. The recommendations are given based on the songs that have listened by the other user with similar preferences, or similar songs that user have already listened. This method can be neighbourhood-based or model-based (Francesco,2011). The model-based algorithm relies on the latent characteristics of the users and the songs, which can be represented as vectors of latent factors.

To conclude this chapter, to win the competition the music industry like Deezer should not only focus on the algorithms but the other factors like user experience, content they are providing, personalizing experience to the users, providing the application to be compatible in various devices, offering the service at reasonable values with on demand customer service. Besides this instead of using purely content-based algorithm or collaborative filtering they should use hybrid algorithms to recommend music to the users.

Chapter-3: Recommendation system proposal

In this chapter, a recommender system that is proposed to Deezer will be discussed to improve the efficiency of the recommendation to the user. To make Deezer better with the competitors and win the competition, LightFM recommender system can be proposed for music recommendation. LightFM is a recommender system that uses both content-based and collaborative filtering techniques to recommend items to the users. As in the previous chapter we already knew that the Content-based recommender system recommend items based on the user preferences and collaborative filtering systems recommend items based on the preferences of the other users. Whereas the LightFM hybrid model uses both approaches to improve the accuracy and relevance of the recommendation to the users. LightFM is also known as hybrid matrix factorisation model where it represents users and items as linear combination of the content features latent factors (Niyamatalmass,2019).

3.1 How LightFM works:

LightFM model learn the embeddings in such a way that encodes user preferences over items. It is done by creating user-item interaction matrix. This matrix is used to train the model to learn the relationship between users, items, and their features. After than the model use this information to make predictions about how likely a user is to interact with the particular item. Besides this it also considers the features of the item and the preferences of the users to make more accurate recommendations.

3.2 Why LightFM:

When compared with other models, LightFM performs at least pure content-based models, eventually surpassing them when either collaborative information available in the training set or user features are included in the model. The embeddings made by the LightFM can encode the important semantic information about features and can be used for related recommendation.

The key difference between other music recommender systems and the LightFM is its ability to handle the sparse data. In general, the sparse data in the recommendation systems can lead to poor recommendations. Despite of size of data, LightFM works very well with the sparse data by using both collaborative filtering and content-based techniques.

In terms of music recommendation, LightFM can recommend songs based on the user's listening history and preferences. Besides it can also take into accounts features such as genre, artist, mood, platform, album, year of release to make more personalized recommendations. Additionally, it can also recommend songs based on the preferences of other users with similar tastes. Due to this adaptiveness the recommendations to the users are more relevant and efficient than the other recommendation systems.

3.3 Implementation of LightFM:

To provide music recommendation to the users of Deezer, the freely available data from Kaggle is taken and fit to the LightFM model to train and test the algorithm in order to provide the personalized music recommendation. A data-driven approach is used to apply LightFM to the Deezer (See figure-1)

***Data pre-processing*:** We started the project by exploring the Deezer dataset and conducting some data pre-processing to clean and transform the data for use in the lightFM framework.

Exploratory Data Analysis (EDA):
The Deezer dataset is a rich source of user interactions with music content. Exploratory data analysis of the dataset could reveal patterns and insights that could inform music recommendation systems.

<u>Correlation between the features</u>
It is clearly visible that some features have a strong correlation with each other than with other features.
So, for example, has the feature "context_type" a positiv correlation with the features "platform_name" and "platform_family".

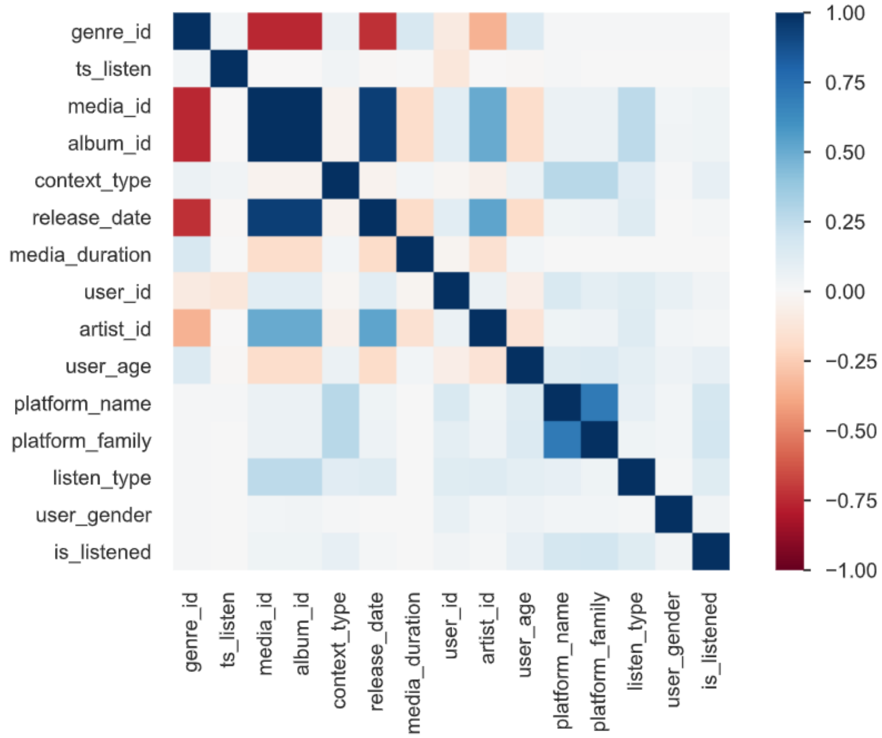

Figure 2: Correlation Matrix from the Dataset

User Gender
If we take a look at the distribution of the user gender we can see that gender 0 (0 = male) is far more represented that gender 1 (1 = female).

Figure 3: Gender Distribution

The visualization of the variable "is_listened" shows that almost half of the total songs (aprox. 350) haven't been listened for more than 30 seconds.

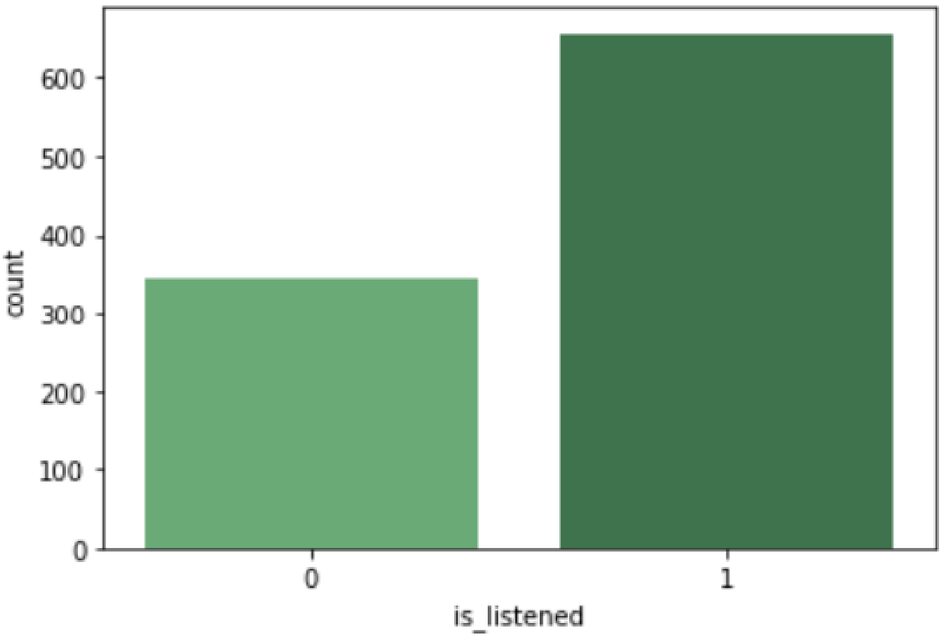

Figure 4: "is_listened" variable distribution

Model training and testing:

The code performs the modeling and testing of two models, pure CF and Hybrid, using the LightFM library. The models are trained on the "train" dataset, where pure CF is trained with only the ratings matrix, and the hybrid model is trained with both user and item features, along with the ratings matrix. Both models use the weighted approximate-rank pairwise (WARP) loss function, with 160 components, a small item regularization of 1e-7, a learning rate of 0.02, and 50 maximum samples per gradient update. The trained models are then evaluated using two evaluation metrics, AUC and Precision@10, on both the train and test datasets. The evaluation results are saved in a dataframe, where each row corresponds to a model and metric combination.

Evaluation of model:

There are totally 4 metrics in lightFM API, the most popular choices are AUC and Precision@k. Both AUC and Precision@K are effective evaluation metrics. AUC is the area under the precision-recall curve, and precision@k is the fraction of known positives in the first k positions of the ranked list of results.

The primary objective for most recommendation scenarios is to keep users engaged on the website. While users are unlikely to leave the website just because they encounter something they dislike, they are more likely to continue browsing if they come across something of interest. Therefore, providers should prioritize recommending items that are likely to pique user interest, rather than focusing too much on avoiding bad recommendations. While avoiding bad recommendations is also important, precision is more crucial than recall in this context. As a result, Precision@k should be considered a more important metric than AUC. It is recommended to evaluate the recommender engine using both Precision@k and AUC.

Method	Evaluation Metric	Train	Test
Pure CF	AUC	0.999888	0.95447135
Pure CF	Precision@10	0.33306533	0.1297538
Hybrid model	AUC	0.97431844	0.9486087
Hybrid model	Precision@10	0.33306533	0.1297538

Figure 5: Model Evaluation

By increasing the no_components parameter from 80 to 160 when creating the LightFM instance, the model's capacity to learn from each feature was enhanced. As a result, the performance of the Hybrid model improved. With the updated parameter, both the CF and Hybrid models exhibited similar performance on the test dataset.

Conclusion:

*Exploratory data analysis revealed interesting patterns and insights that could be useful in informing such systems. Correlation analysis showed that some features were more strongly correlated with each other than with others. Additionally, the gender distributi*on of users was skewed towards males, and about half of the songs in the dataset had not been listened to. The LightFM library was used to train and test two models, pure CF and Hybrid, with the latter model performing better on the Precision@k metric. The importance of precision in recommendation scenarios was emphasized, and the no_components parameter was adjusted to enhance model performance.

Chapter-4: Actual vs Proposal

In this chapter, we will discuss the differences between various recommender system with the proposed recommender system LightFM. We will compare various features and functioning with LightFM and the other recommender system.

In the content-based filtering the recommendation is given based on the characteristics of the items in this case of Deezers, characteristics of the music that user listening to. Before fitting the model, user profile is created where features such as genre, albums, artists, year of release are taken as user preferences. These preferences are then checked if the user in the past liked the songs from the genre, from album, is listened or skipped and then song will be recommended based on the interest from the past history of the user.

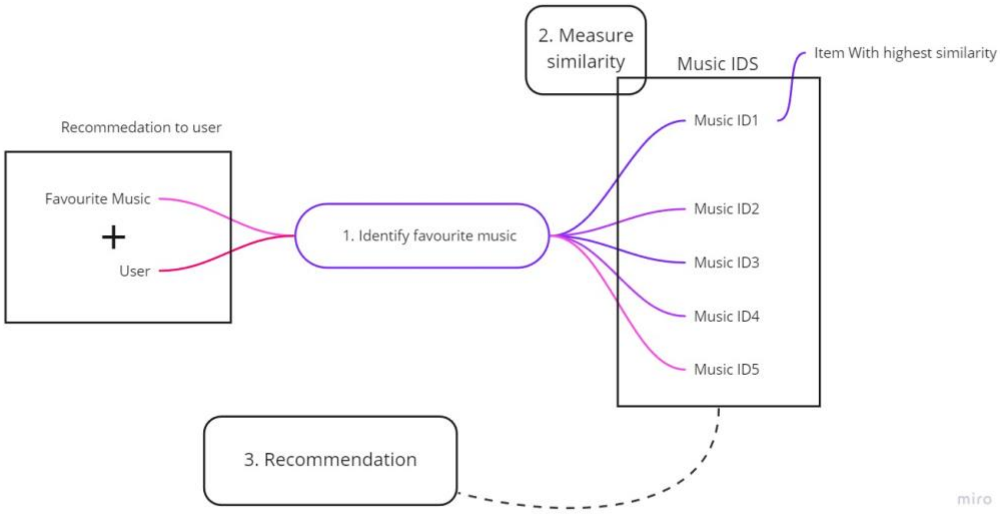

Figure 56: Content-Based Filtering Method

Whereas in the Collaborative Filtering the recommendations are based on the preferences of similar users, in the context of Deezers, firstly the set of songs listened by set of users taken in to consideration implicitly or explicitly. Secondly the users with similar preferences to the other users are found and the songs that they have listened most of the times will be given with high similarity score. So if the similarity score between two users is high then the song listened by one user will be recommended to the other user.

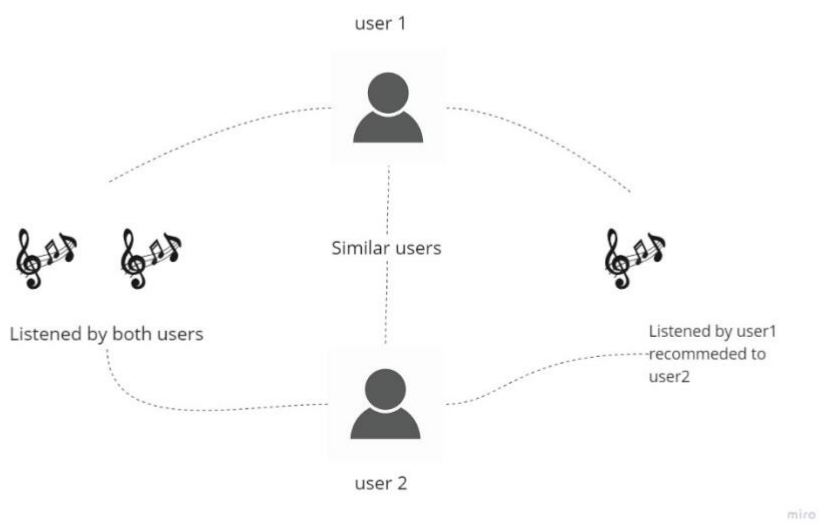

Figure 67: Collaborative Filtering Method

Matrix Factorization works by decomposing the user-item interactions matrix into two lower dimensional matrices representing users and items respectively. This technique reduces the dimensionality of the data while preserving the essential information needed for the recommendation. To implement this method on Deezer dataset firstly the user-item interaction should be prepared. This helps capturing the interactions between the users and songs such as the number of times a user listened to particular song. Secondly the matrix is decomposed to two lower dimensional matrices using Matrix Factorization techniques such as singular value decomposition (SVD) or alternating least squares (ALS). After decomposing the model is trained to learn the latent features that represent user preferences and item attributes. The latent features are used to make personalized recommendation to users based on the listening history and preferences.

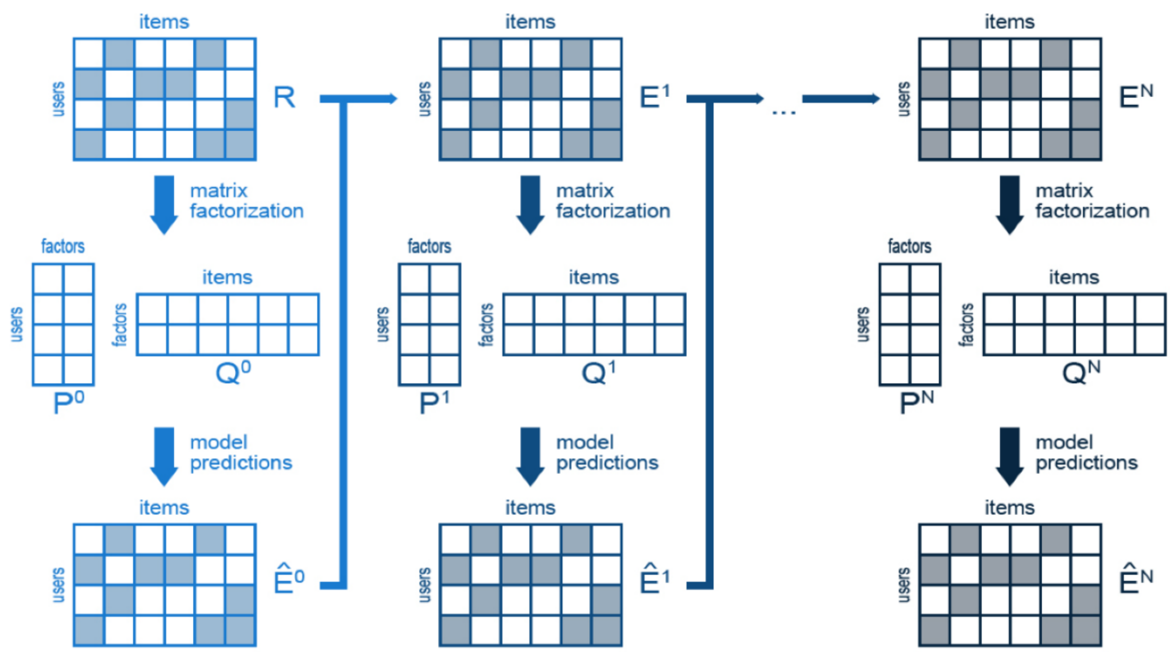

Figure 78: Matrix Factorization Method (General) *(Lara-Cabrera et al., 2020)*

LightFM is the mix of content-based and collaborative filtering system, this works same as content-based and Collaborative filtering. In the terms of Deezer, from the available data the features such as genre, artists id, album and year of release are taken and the interactions such as ts_listened and is_listened are prepared to form the user-item interactions matrix. This matrix capture the interactions between users and songs and the model is trained to learn the relationship between users, songs and their attributes. Thus model can be used to make personalized recommendation to the users based on their listening history and preferences also by recommendation based on the preferences of other users with similar tastes.

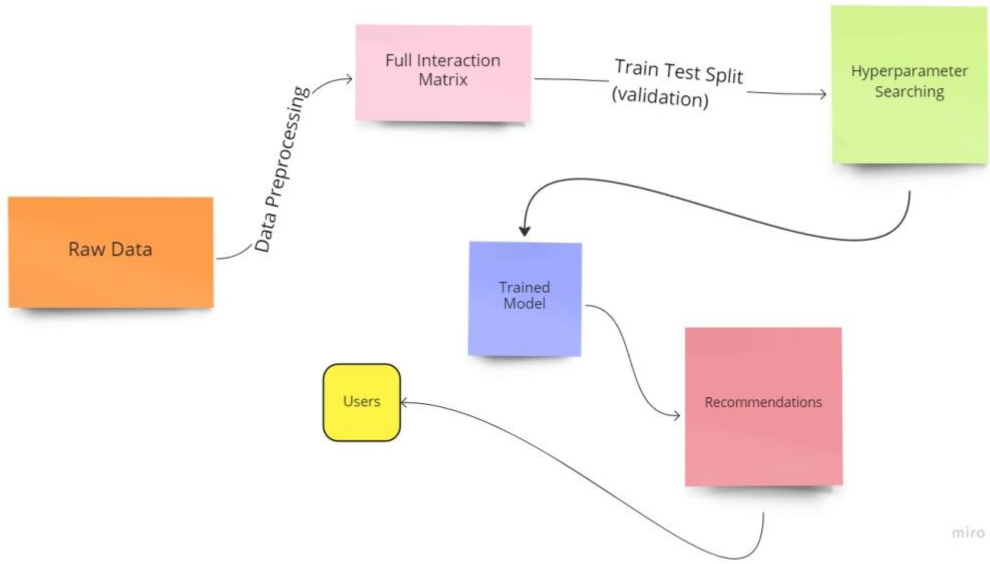

Figure 89: LightFM Model

LightFM model is similar to all the three models Content-based filtering and Collaborative Filtering and Matrix Factorization methods. Thus LightFM can be better choice for Deezer in order to improve their recommendations to the users and also to win the competition with their competito

Chapter-5: Conclusion

The implementation of a recommender system is crucial for music streaming services such as Deezer to provide personalized and relevant recommendations to their users. The proposed solution of using LightFM, a hybrid matrix factorization model, can be an effective approach for Deezer to improve their recommendation system. LightFM combines the benefits of both content-based and collaborative filtering techniques to improve the accuracy and efficiency of the recommendation system.

The implementation process of LightFM involves data pre-processing, exploratory data analysis, model training, and testing. The data pre-processing involves cleaning and transforming the data to fit into the LightFM framework. The exploratory data analysis revealed patterns and insights that can inform the recommendation system. The model training and testing involves creating two models, pure CF and Hybrid, using the LightFM library and evaluating them using metrics such as AUC and Precision@k.

LightFM can handle sparse data effectively, which is a significant problem in recommendation systems. LightFM can also consider the preferences of other users with similar tastes to provide relevant recommendations. The LightFM model can be customized and optimized based on the data and the preferences of the users, which makes it a flexible and adaptable solution for Deezer.

References

1. Apple. (n.d.). Apple. Retrieved 2.2023, from https://www.apple.com/
2. Bontempelli, T., Chapus, B., Rigaud, F., Morlon, M., Lorant, M., & Salha-Galvan, G. (2022, September). Flow Moods: Recommending Music by Moods on Deezer. In Proceedings of the 16th ACM Conference on Recommender Systems (pp. 452-455).
3. Deezer. (n.d.). Deezer. Retrieved 2.2023, from https://www.deezer.com/en/
4. Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors. Recommender Systems Handbook. Springer, 2011.
5. Jan Schluter and Christian Osendorfer. Music Similarity Estimation with the Mean-Covariance Restricted ¨ Boltzmann Machine. In Proceedings of the 10th International Conference on Machine Learning and Applications (ICMLA), 2011.
6. Malcolm Slaney, Kilian Q. Weinberger, and William White. Learning a metric for music similarity. In Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR), 2008.
7. Mulligan, M. (2022, December 7). *Music subscriber market shares 2022*. Music Industry Blog; Music Industry Blog. https://musicindustryblog.wordpress.com/2022/12/07/music-subscriber-market-shares-2022/
8. Music Market Landscape - Growth, Trends, COVID-19 Impact, and Forecasts (2023 - 2028). (2023). Mordorintelligence.com. https://www.mordorintelligence.com/industry-reports/music-market-landscape
9. Niyamatalmass. (2019, April 23). LightFM Hybrid Recommendation system. Kaggle.com; Kaggle. https://www.kaggle.com/code/niyamatalmass/lightfm-hybrid-recommendation-system
10. Lara-Cabrera, R., González-Prieto, Á., & Ortega, F. (2020). Deep Matrix Factorization Approach for Collaborative Filtering Recommender Systems. Applied Sciences, 10(14), 4926. https://doi.org/10.3390/app10144926
11. Spotify. (n.d.). Spotify. Retrieved 2.2023, from https://www.spotify.com/
12. Tidal. (n.d.). Tidal. Retrieved 2.2023, from https://tidal.com/

Appendix

Data Description

The dataset contains 1048576 observations of music listening events, with 16 features for each observation. The features in the dataset are as follows:

genre_id: an integer identifier for the genre of the song listened to
ts_listen: a Unix timestamp for the time the song was listened to
media_id: an integer identifier for the song listened to
album_id: an integer identifier for the album the song belongs to
context_type: an integer identifier for the context in which the song was listened to (e.g. playlist, album ...)
release_date: a date representing the release date of the song
platform_name: an integer identifier for the type of OS on which the song was listened to
platform_family: an integer identifier for the type of device on which the song was listened to
media_duration: the duration of the song in seconds
listen_type: an integer identifier for the type of listen event if the songs were listened in a flow or not
user_gender: an integer identifier for the gender of the user listening to the song
user_id: an integer identifier for the user listening to the song
artist_id: an integer identifier for the artist who created the song
user_age: the age of the user listening to the song
is_listened: a binary flag indicating whether the user listened to the song in its entirety (1) or not (0)

Reflection Paper – Recommender Systems

Lecturer : Dr. Guang Lu **Opinion of: Tejesh Reddy Koki**

Through this reflection paper, I will share my thoughts and experiences of me on the module Recommender systems that took place from January 30th till February 3rd at Lucerne University of Applied Sciences and Arts under Dr. Guang Lu.

The block course has been very informative and enlightening experience. Through this course my enthusiasm and keen towards the recommender algorithms have been increased. With this course, I have gained a deeper understanding of the various algorithms, evaluation of the models, application of the recommender systems.

Summary of the course content: The course covered a variety of topics of recommender systems, including Modern frameworks of the recommender systems, Collaborative Filtering and Content-Based techniques, Factorization methods, Various models of the recommender systems like deep-generative models, knowledge graphs, hybrid systems, recommendation engines and frameworks. The exercises related to the concepts are also well structured and markdown for ease of understanding code.

Reflection on Learning Experiences: The course content and exercises challenged me to think critically about the recommender systems. However, I found various topics interesting to deep diver like Matrix Factorization, Neural Matrix Factorization and in particular the deep learning where with neural collaborative filtering the movies are recommended to the users.

Evaluation: I found this course content comprehensive, the only problem I feel was to many presentation slides for one day instead of one main master PowerPoint for that day. Also the time is not enough to have proper hands on experience with the exercises and to get more deeper thoughts and ideas on the code. I feel five days are short for this course to develop the mindset of implementing the algorithms for the project work. I would rather have this module in the course instead of block week so I can learn better the recommender systems with challenges from course work and try to incorporate the solutions in the project work. However, I will have much appreciated if we have more time for hands on exercises.

Conclusion: In conclusion, I would highly recommend the recommender system course to anyone interested. The course provides a diverse materials and content on the application of recommender systems and the materials are very organized. Overall, the course has provided me with a valuable and enriching learning experience, to expand my knowledge and understanding of recommender system.

Lecturer : Dr. Guang Lu **Opinion of: Matthias De Paolis**

I have gained a significant amount of knowledge and insight. The course provided an opportunity to deepen my understanding of different types of recommender systems, including their applications and limitations.

Over the course of the module, we explored a variety of topics related to modern frameworks of recommender systems, including Collaborative Filtering and Content-Based techniques, Factorization methods, and Various models of the recommender systems like deep-generative models, knowledge graphs, hybrid systems, recommendation engines, and frameworks. Through coding examples and the final project, I was able to apply my skills and gained experience in data pre-processing and cleaning, which are essential steps in the implementation of recommender systems.

The literature study we conducted in the course also provided me with valuable insights into the underlying principles and techniques used in recommender systems, as well as the challenges involved in their implementation. One aspect of the study that stands out was the implementation of the lightFM framework on a Deezer dataset together with Tejesh.

While the course content was comprehensive, I feel that more time is needed for hands-on exercises to gain a deeper understanding of the concepts and ideas presented in the code. I also found the presentation slides overwhelming, with too many slides in one day. To maximize the learning experience, a longer course duration or inclusion of the module in the coursework instead of a block week would provide more opportunities for challenging coursework and incorporating solutions into project work.

Overall, the Recommender Systems module at Lucerne University of Applied Sciences and Arts has been an enriching experience that has broadened my knowledge and skills in various ways. The course provides diverse materials and content on the application of recommender systems, and the materials are well-organized. I am grateful for the opportunity to have gained a deeper understanding of these systems and to have developed my coding skills, and I would highly recommend this course to anyone interested in the field of recommender systems.