AI Music Recommendation

Evan Griswold
emgriw20@earlham.edu
Earlham College
Cincinnati, Ohio, USA

ABSTRACT

This project focuses on developing an AI-powered music recommendation system by harnessing diverse music data sources with the primary dataset Million Song Dataset (MSD) and having a sub-dataset, The Echo Nest. These contain lyrics, user data, genre labels, and Track info. Employing content-based filtering and deep learning algorithms together will achieve the best results, providing users with personalized and diverse music recommendations. Using the platform Xcode, I created an App using Swift UI code and made it personable to the user.

KEYWORDS

Machine Learning, Data Collection, Music Recommendation System, Evaluation Metrics, Matrix Factorization, Music Discovery

ACM Reference Format:
Evan Griswold. 2024. AI Music Recommendation. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Music consumption and discovery have been dramatically transformed in recent years, owing to the advent of digital streaming platforms and the proliferation of music databases. Amid this evolving landscape, the application of AI to music recommendation systems has emerged as a pivotal endeavor. The promise of enhancing user experiences and facilitating the exploration of a vast and diverse repertoire excites music enthusiasts. This proposal introduces what an AI-driven music recommendation system is through data gathering, preprocessing, model construction, evaluation, and user-centric considerations [2]. These considerations will include the user's involvement, experience, and overall system usability. After collecting the data from MSD, the steps for preprocessing will need to be usable. The preprocessing includes checking for missing values and creating features based on the song's tempo, loudness, etc [14].

- Sound Cloud (for audio feature extraction)
- Scikit-Learn (for machine learning)
- Pandas (for data manipulation)
- GitLab (for code examples and projects)

The Million Song Dataset encompasses diverse music-related information, involving cover songs, lyrics, user data, and song-level tags and similarity.

The efficacy of the music recommendation system hinges on extracting relevant features from the dataset, creating an easy and fast working system for the user. Features of interest include genres, tempo, artist popularity, user listening history, and more. The subsequent construction of the recommendation model involves a multifaceted approach, drawing upon a few techniques such as using content-based filtering and deep learning. These techniques are tools to discern patterns within the music data, enabling the system to provide personalized recommendations.

Deep learning and content based filtering will be used together to improve accuracy, have a better understanding of user preferences, and a more scalable solution. However, careful consideration and experimentation will be necessary to strike the right balance and achieve optimal performance.

2 DATASET

This section introduces one dataset that will be used in the research. The choice of dataset depends on your specific project goals, data availability, and the type of music recommendation system you intend to build. Combining multiple datasets is common to create a more comprehensive and accurate model [1]. With the Million Song Dataset, I will be using sub-datasets to include in my work to improve my recommender system [5].

- The Echonest - All Meta Data
- Sound Cloud - Audio Files

2.1 Million Song Dataset

The Million Song Dataset is a freely available collection full of audio features and metadata for a million contemporary popular music tracks. The core of the dataset is the feature analysis and metadata for one million songs, provided by The Echo Nest. The dataset does not include any audio, only the derived features. Note, however, that sample audio can be fetched from services like Sound CLoud, I plan to use the code Million Song Dataset provides. The purposes of the dataset are:

- To encourage research on algorithms that scale to commercial sizes
- To provide a reference dataset for evaluating research
- As a shortcut alternative to creating a large dataset with APIs (e.g. The Echo Nest's)
- To help new researchers get started in the MIR field [5].

3 METHODS

There are numerous methods to achieve the task at hand, however, the proposal will only focus on Content Based Filtering and Deep Learning. These methods will be essential for the work at hand

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

to produce the AI music recommendation system. The choice of method depends on the specific goals of the music recommendation system and the nature of the available data. These methods will achieve the best results, providing users with personalized and diverse music recommendations.

3.1 Content Based Filtering

Content based filtering recommends music based on the content attributes of songs and the user's historical preferences. It focuses on matching user profiles with music features like genre, tempo, and lyrics [12]. The item description and a profile of the users orientation play an important role in Content-based filtering. Content based filtering algorithms try to recommend items based on similarity count [15].

- Pros: Personalization, transparency, less data dependency, and no cold start.
- Cons: Profile narrowness, over-specialization, and staleness.

Like all recommender systems, content based filtering has both pros and cons. These Pros and Cons come from user experiences. It is good that there are no cold start problems because it can cause issues for the filtering algorithm not to make recommendations, as it relies on interactions. A cold start is when items or songs are added to a catalog with either none or very few interactions [10].

One way to avoid the cold start problem is to combine different types of recommender systems, such as collaborative filtering, content-based filtering, and knowledge-based filtering. Hybrid models can leverage each approach's strengths and compensate for others' weaknesses [11].

3.2 How Content Based Filtering Works

Creating a content-based profile of users is done with the help of a weighted vector of item features. The weights denote the importance of each feature to the user. It can be calculated from individually rated content vectors using various proficiencies. Below shows CBF mechanism, which includes the following steps:

(1) Educe the attributes of items for recommendation.
(2) Compare the attributes of items with the preferences of the active user.
(3) Recommend items according to features that fulfill the users interests.

When the attributes of the items and the user profiles are known, the key role of CBF is to determine whether a user will like a specific item. This task is traditionally answered by using heuristic methods [3] or classification algorithms, such as: rule induction, nearest neighbors methods, Rocchio's algorithm, linear classifiers and probabilistic methods [7]. I plan to use classification algorithms for the recommender system.

3.3 Deep Learning

Deep learning is using a neural network with several layers. These include deep recommendation models, which are used to model complex patterns in user behavior and music features for more accurate recommendations [2] and [13].

The difference between sparse features and dense features lies in the distribution of their values in a dataset.

- Sparse features have very few non-zero values
- Dense features have many non-zero values.

This difference in the distribution has implications for machine learning algorithms, as algorithms may perform differently on sparse features compared to dense features [9].

Deep learning, a branch of machine learning, is an algorithm that attempts to perform high-level abstraction of data using multiple processing layers that contain complex structures or consist of multiple nonlinear transformations. The uniqueness of deep learning is that it allows multiple processing layers to be composed while the processing layer can be a traditional neural network or algorithms in other fields. Such a computing model can not only be extended but also learned [16].

- Pros: Highly effective, scalability, state-of-the-art performance, and continuous improvement.
- Cons: Data dependency, complexity, and limited data efficiency.

Deep learning techniques can be employed to learn complex patterns and representations from this sparse user rating matrix for making recommendations. Deep learning models for recommendation can be categorized into two main types:

- Matrix Factorization Models: These models directly factorize the user-item rating matrix into low-rank matrices, capturing latent factors.
- Neural Collaborative Filtering Models: These models use neural networks to learn non-linear relationships between users and items, often providing more expressive representations.

These approaches leverage the power of deep learning to capture intricate patterns and relationships in user preferences and item characteristics, enabling more accurate and personalized recommendations [6].

A deep belief network (DBN) is a multi-layer learning architecture that uses a stack of RBMs to extract a deep hierarchical representation of the training data. In such design, the hidden layer of each sub-network serves as the visible layer for the upcoming sub-network [8]. When learning through a DBN, firstly the RBM in the bottom layer is trained by inputting the original data into the visible units. Then, the parameters are fixed up, and the hidden units of the RBM are used as the input into the RBM in the second layer. The learning process continues until reaching the top of the stacked sub-networks, and finally, a suitable model is obtained to extract features from the input. Since the learning process is unsupervised, it is common to add a new network of supervised learning to the end of the DBN to use it in a supervised learning task such as classification or regression. [4]

4 METHODOLOGY

Define the problem statement and objectives of the AI music recommendation system. The system aims to provide personalized and diverse music recommendations based on user preferences and content attributes.

Gather a comprehensive dataset of music tracks with associated attributes such as genre, tempo, lyrics, etc. Preprocess the dataset to clean and standardize the attributes, handling missing values and outliers appropriately.

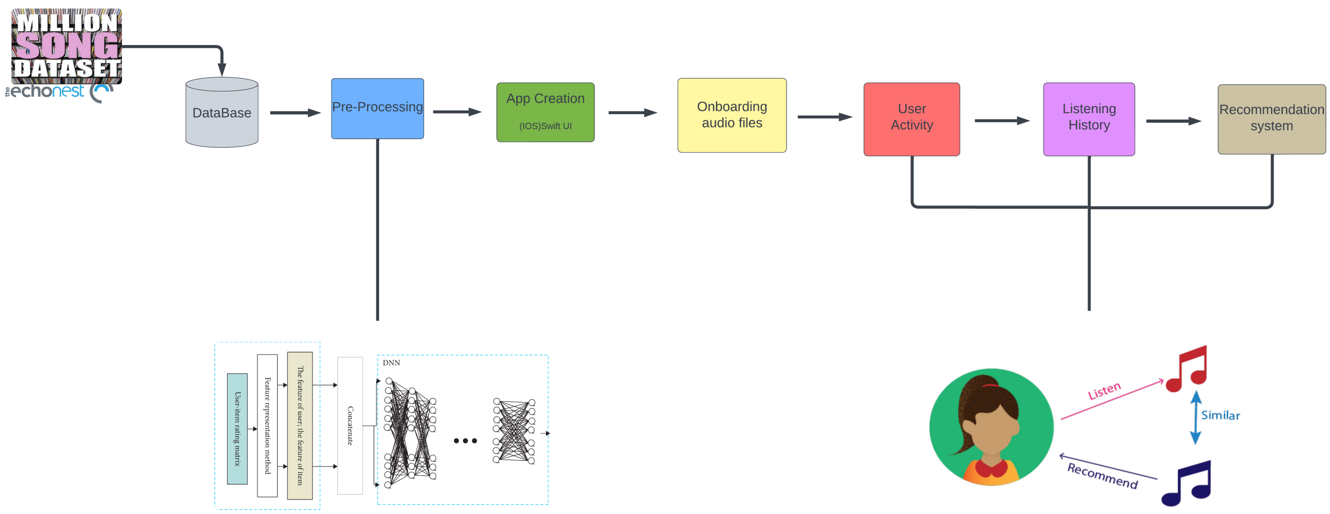

Figure 1: Data Architecture Diagram

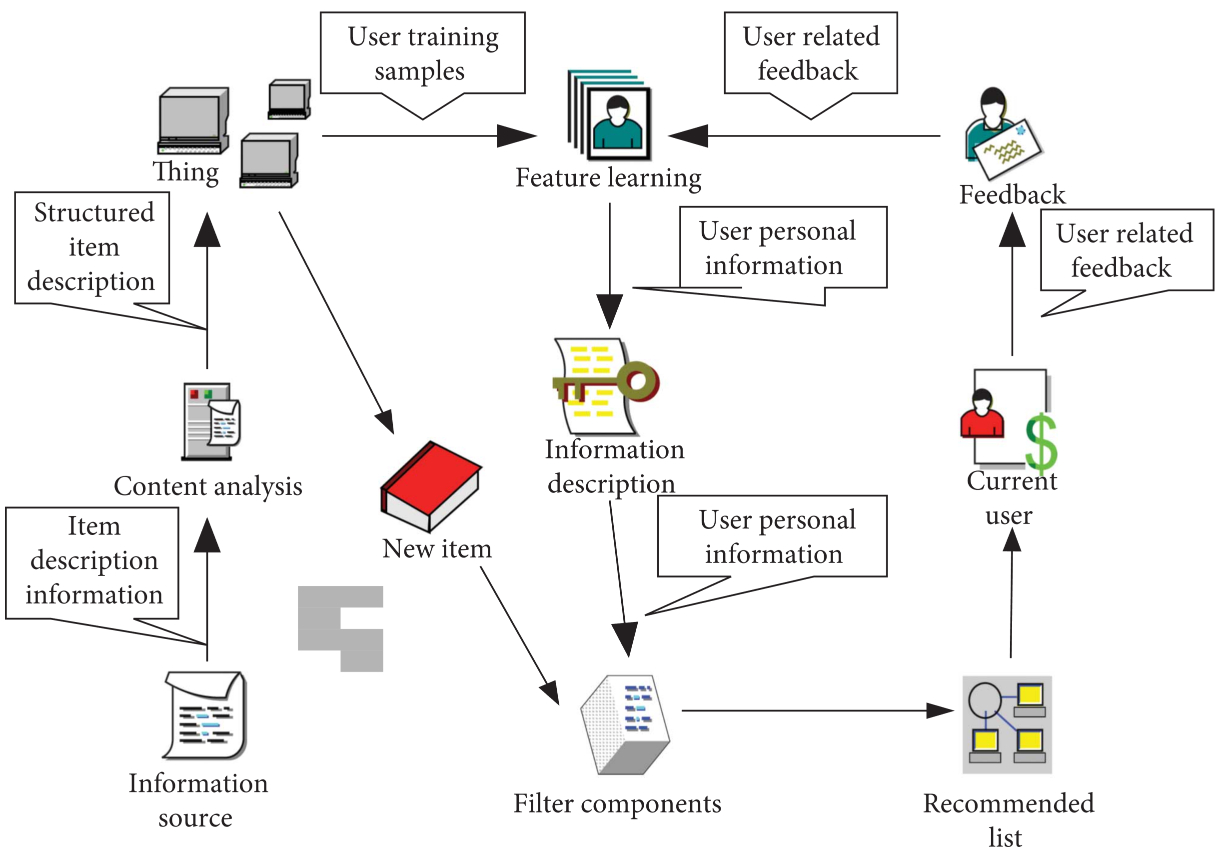

Figure 2: Process diagram of content based algorithm

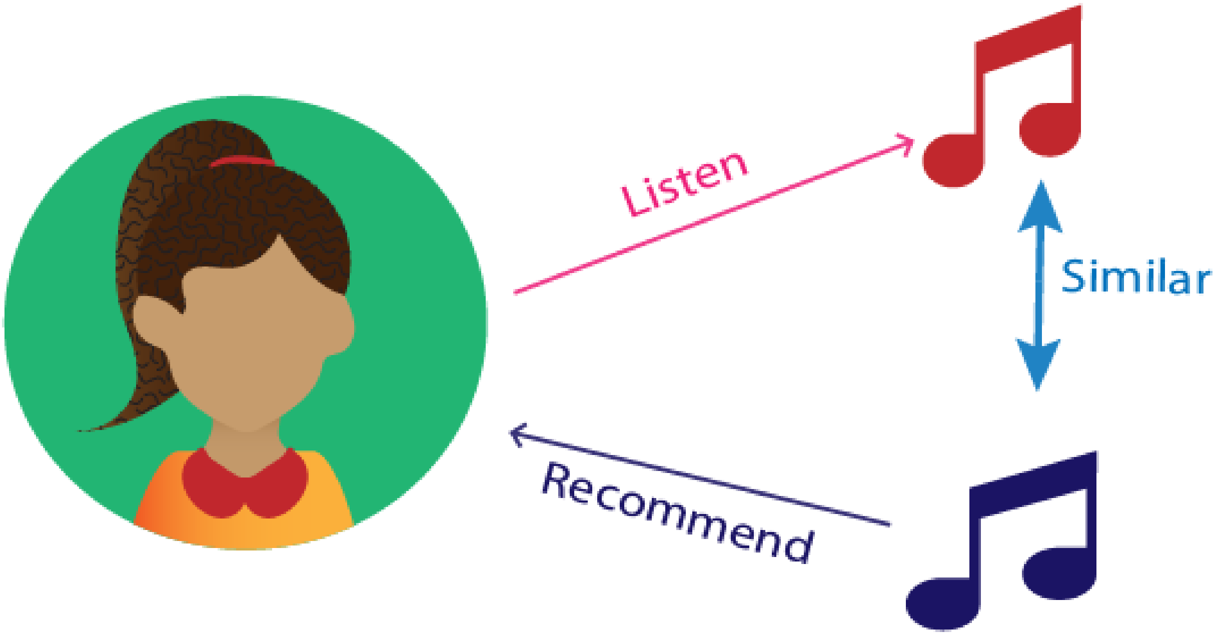

Figure 3: Content Based Filtering

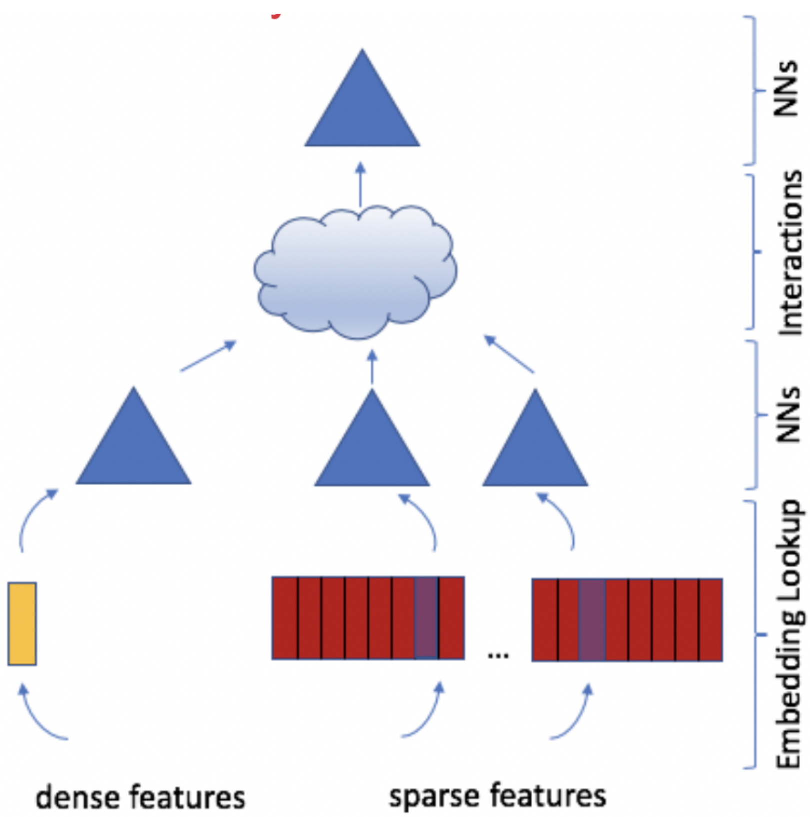

Figure 4: A deep learning recommendation model

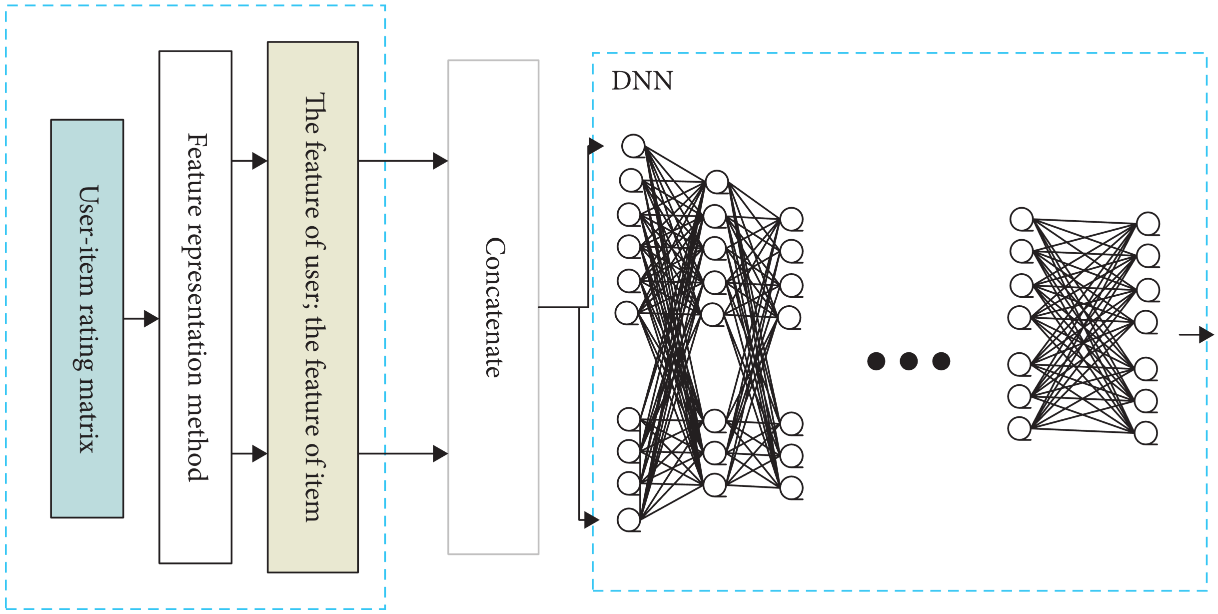

Figure 5: Deep learning neural network model

Implement Content-Based Filtering (CBF) algorithms to recommend music based on content attributes and user preferences. Create content-based user profiles using weighted vectors of item features. Utilize classification algorithms such as rule induction, nearest neighbors methods, or neural networks to predict user preferences and recommend music items.

Develop Deep Learning models to learn complex patterns in user behavior and music features for more accurate recommendations. Explore Deep Learning techniques such as Deep Belief Networks (DBNs) to extract hierarchical representations of music data. Train Matrix Factorization Models or Neural Collaborative Filtering Models to capture latent factors and non-linear relationships between users and items.

Evaluate the performance of the developed recommendation models using appropriate metrics such as accuracy, precision, recall, or F1-score. Use cross-validation techniques to ensure the robustness and generalizability of the models. Compare the effectiveness of Content-Based Filtering and Deep Learning approaches in providing personalized music recommendations.

Explore the possibility of integrating hybrid recommendation models that combine Content-Based Filtering and Deep Learning techniques. Leverage the strengths of each approach to compensate for weaknesses and enhance recommendation quality. Evaluate the performance of the hybrid model against individual methods and select the most effective approach for deployment.

Deploy the developed AI music recommendation system in a real-world environment or a controlled user study. Gather feedback from users to assess the system's usability, satisfaction, and effectiveness in providing personalized music recommendations. Iterate on the system based on user feedback and continuously improve recommendation algorithms.

Document the methodology, implementation details, and experimental results for future reference. Prepare a comprehensive report summarizing the development process, key findings, and recommendations for future enhancements. Disseminate the research findings through publications, presentations, or academic forums.

5 EVALUATION

The evaluation plan should define its goal as precisely as possible. The plan is to use content based filtering and deep learning together to perform the desired tasks within the recommender system. The way these will be evaluated will be through trial and error and from the feedback collection system that the user will provide. Necessary changes will be accounted for and improved in the system.

By combining content-based filtering and deep learning, the recommendation system can leverage both explicit features and intricate patterns in the data, resulting in more accurate and personalized music recommendations for users. This hybrid approach is used to provide a better user experience. Hybrid recommender systems utilize multiple approaches together, and they overcome the disadvantages of certain approaches by exploiting the compensations of the other [4].

6 RESULTS

Significant progress has been made in developing the application by utilizing Xcode and Swift UI. The project comprises separate files and incorporates a functional simulation compatible with all iOS platforms. The application interface is user-friendly and responsive. Upon launching the app, users are presented with an extensive list of songs drawn from the dataset. Clicking on a song triggers a new page to display essential details such as the title, artist, and interactive buttons for play song and get recommendations. The "Get Recommendations" button successfully generates three similar songs from the dataset for user to listen to. However, while the recommendation feature is operational, the functionality of the play button is currently under development. The integration of audio files from sound cloud into the application's dataset is ongoing to enable seamless playback functionality. Overall, the project demonstrates substantial progress towards its completion, with key features implemented and being operational.

7 FUTURE WORK

- Get App completely functional - Currently developing and at the stage where I need to focus on completing it and ensuring that all features are fully functional. This involves addressing any remaining bugs or issues, implementing necessary enhancements, and thoroughly testing the app to ensure a seamless user experience. The goal is to have a polished and reliable product ready for deployment.
- Enhanced Personalization - Tailoring experiences, services, or content to individual preferences using advanced algorithms and AI. It aims to boost engagement and satisfaction by delivering customized interactions.
- Multi-Modal Recommendation - Suggesting content through various channels like text, images, or voice commands. These systems provide more accurate and engaging recommendations by considering different user preferences and interaction patterns.
- Localized Recommendation - Offering personalized suggestions based on the user's location and cultural context. This includes local events, weather, and language, enhancing relevance and user experience.
- Voice Assistant Integration - Incorporating voice-controlled assistants like Alexa or Siri into apps or devices. Users can interact hands-free, enabling seamless tasks like information retrieval and home automation.
- Cross-Platform Compatibility - Ensuring software works smoothly across different devices and platforms. It allows users to access the same features regardless of their device, enhancing consistency and user experience.

REFERENCES

[1] Md Arshad Ahammed. 2022. What are the benefits of merging multiple datasets into one dataset? *Quora* (2022). https://www.quora.com/What-are-the-benefits-of-merging-multiple-datasets-into-one-dataset

[2] R. Anand, RS. Sabeenian, Deepika Gurang, R. Kirthika, and Shaik Rubeena. 2021. AI based music recommendation system using deep learning algorithms. In *IOP conference series: earth and environmental science*, Vol. 785. IOP Publishing, 012013.

[3] Chumki Basu, Haym Hirsh, and William W. Cohen. 1998. Recommendation as Classification: Using Social and Content-Based Information in Recommendation. (1998). https://api.semanticscholar.org/CorpusID:14176567

[4] Zeynep Batmaz, Ali Yurekli, Alper Bilge, and Cihan Kaleli. 2019. A review on deep learning for recommender systems: challenges and remedies. *Artificial Intelligence Review* 52 (2019), 1–37.

[5] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. 2011. The Million Song Dataset. (2011).

[6] Denise Chen. July 8, 2020. Recommender System — Matrix Factorization. *Medium* (July 8, 2020). https://towardsdatascience.com/recommendation-system-matrix-factorization-d61978660b4b

[7] Marco de Gemmis, Pasquale Lops, Giovanni Semeraro, and Pierpaolo Basile. 2008. Integrating Tags in a Semantic Content-Based Recommender. (2008), 163–170. https://doi.org/10.1145/1454008.1454036

[8] G. E. Hinton. 2009. Deep belief networks. *Scholarpedia* 4, 5 (2009), 5947. https://doi.org/10.4249/scholarpedia.5947 revision #91189.

[9] Induraj. Feb 23. what are Sparse features and Dense features? *Medium* (Feb 23). https://induraj2020.medium.com/what-are-sparse-features-and-dense-features-8d1746a77035

[10] Blerina Lika, Kostas Kolomvatsos, and Stathes Hadjiefthymiades. 2014. Facing the cold start problem in recommender systems. *Expert Systems with Applications* 41, 4, Part 2 (2014), 2065–2073. https://doi.org/10.1016/j.eswa.2013.09.005

[11] Linkedin. 2023. How can you avoid the cold start problem in recommender systems? *Linkedin* (2023). https://www.linkedin.com/advice/0/how-can-you-avoid-cold-start-problem-recommender-r75tc#:~:text=One%20way%20to%20avoid%20the,for%20the%20weaknesses%20of%20others.

[12] Ankita Mahadik, Shambhavi Milgir, Janvi Patel, Vijaya Bharathi Jagan, and Vaishali Kavathekar. 2021. Mood based music recommendation system. *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume* 10 (2021).

[13] Batta Mahesh. 2020. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]* 9, 1 (2020), 381–386.

[14] Ebuka (Gaus Octavio). Jan 15. Music Recommendation System Built With Python And Machine Learning. *Medium* (Jan 15).

[15] Fernando Ortega, José-Luis Sánchez, Jesús Bobadilla, and Abraham Gutiérrez. 2013. Improving collaborative filtering-based recommender systems results using Pareto dominance. *Information Sciences* 239 (2013), 50–61. https://doi.org/10.1016/j.ins.2013.03.011

[16] Yezi Zhang and Chia-Huei Wu. 2022. Music Recommendation System and Recommendation Model Based on Convolutional Neural Network. *Hindawi* (2022). https://www.hindawi.com/journals/misy/2022/3387598/