Music Artist Recommendation System

University of California, San Diego - Laura Zhou

1. DATASET

1.1 Introduction

Last.fm, an established music website that started in the United Kingdom, serves as a platform for creating comprehensive user profiles that capture individual musical preferences by logging track details listened to by the user.

Additionally, it incorporates social networking functionalities for inter-user interaction. The dataset under examination comprises data encompassing social networking activities, tagging behaviors, and music artist listening patterns drawn from a sample of 2,000 users subscribed to the Last.fm online music system. This dataset includes the files

1. *artists.dat* - information about music artists listened to and tagged by the users.
2. *tags.dat* - the set of tags available in the dataset.
3. user_artists.dat - artists listened by each user along with a listening count for each [user, artist] pair.
4. user_taggedartists.dat - the tag assignments of artists provided by each particular user.
5. user_friends.dat - the friend relations between users in the database.

1.2 Data Cleaning

In the initial phase, data refinement began by dropping any duplicate or missing entries. Furthermore, we remove values outside the percentile range [0.05, 0.95] by year, as there are significantly sparser values in earlier years of our dataset. Moreover, a subsequent stage involved enhancing dataset relevance through feature selection, specifically eliminating columns deemed irrelevant for the analysis, such as [‘url’, ‘pictureURL’] within the *artists.dat* dataset.

Furthermore, as part of data refinement for streamlined analysis, the consolidation of datasets was performed. Specifically, the datasets ‘artists.dat’ and ‘user_artists.dat’ were combined into a unified dataframe referred to as ‘tag_user_merge’. Similarly, ‘tags.dat’ and ‘user_tagged_artists.dat’ were integrated into a combined dataset denoted as ‘artist_user_merge’ to improve computation convenience.

	tagID	tagValue	userID	artistID	day	month	year
0	1	metal	4	918	1	5	2008
1	1	metal	12	181	1	5	2010
2	1	metal	12	198	1	2	2010
3	1	metal	12	500	1	2	2010
4	1	metal	12	503	1	3	2010

Figure 1.1 - Dataset #1 tag_user_merge

	artistID	name	userID	weight
0	1	MALICE MIZER	34	212
1	1	MALICE MIZER	274	483
2	1	MALICE MIZER	785	76
3	2	Diary of Dreams	135	1021
4	2	Diary of Dreams	257	152

Figure 1.2 - Dataset #2 artist_user_merge

1.3 Exploratory Data Analysis

Some introductory statistics about the dataset:

- The dataset comprises 1,892 users, 17,632 artists, and 12,717 bi-directional user friend relations, averaging 13.443 relations per user.
- There are 92,834 user-listened artist relations, with an average of 49.067 artists listened to per user and 5.265 users per artist.
- The dataset contains 11,946 unique tags and 186,479 tag assignments. On average, users have 98.562 tag assignments, artists have 14.891 tags, users employ 18.930 distinct tags, and artists are associated with 8.764 distinct tags.

Based on this dataset, several possible insights can be drawn about user-artist interactions, user genre or tagging interests, and possible social network structures. For instance, if we visualize the most popular artists by listening frequency, we see that

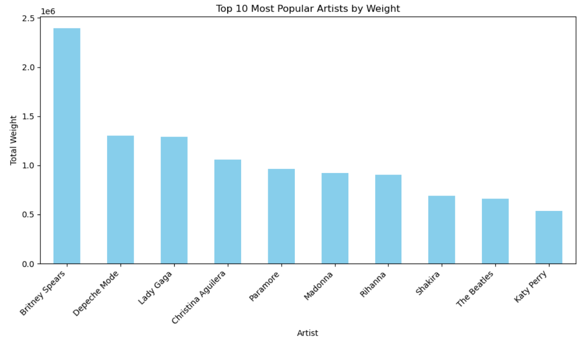

Figure 1.3 - Popular Artists by Weight

prominent figures like ['Britney Spears', 'Depeche Mode', 'Lady Gaga', 'Christina Aguilera', 'Paramore'] and others in the dataset. However, if we group artists by tagging frequency it reveals that a different set of artists as the most popular:
['Britney Spears', 'Lady Gaga', 'Christina Aguilera', 'Madonna', 'Depeche Mode', 'Muse', 'The Beatles', 'Radiohead', 'Rihanna', 'Paramore']. Interestingly, rock tagged artists like Radiohead and Muse are more commonly tagged, but not listened to as much in comparison.

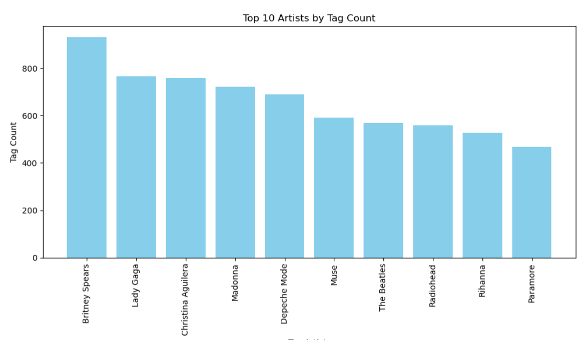

Figure 1.4.1 - Popular Artists by Tag Count

	tagValue	count
tagID		
1	metal	1729
2	alternative metal	212
3	goth rock	22
4	black metal	301
5	death metal	582

Figure 1.4.2 - Popular Genres by Tags

Similarly, prevalent music genres tagged in our dataset encompass 'rock', 'pop', 'alternative', 'electronic', and 'indie'. An interesting thing to note is that while 'rock' was the most popular tag, the majority of the top artists by weight are tagged with 'pop' and 'alternative'.

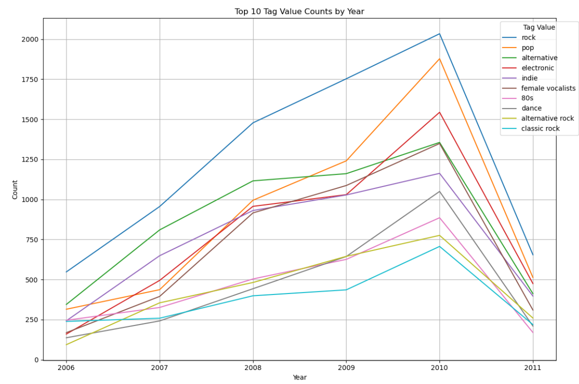

Figure 1.5 - Top Genre Tag Counts Over Time

Finally, the analysis of genre tag frequencies grouped by year indicates a consistent prevalence of tagged rock music across the years. Notably, there's also a discernible upward trend in the popularity of pop music over the last two decades.

This data proves pivotal for our analysis, offering significant insights into listening patterns and trends. Such insights serve as foundational benchmarks for our predictive models, enhancing their relevance and accuracy.

2. PREDICTIVE TASK

2.1 Introduction

Our objective is to predict which music artists a user might listen to based on their past listening history, ratings, or interactions. This task is important because this information can be used to generate new music recommendations for users for music streaming services.

2.2 Baselines & Evaluation

Relevant baselines that can be used for comparison include popularity based recommenders or random recommenders.

A popularity based recommendation would simply sample our dataset for the most popular artists in terms of listening count or weight. This is an appropriate baseline, as users tend to gravitate toward popular items. This method assumes that recommending frequently listened-to artists or tracks will likely resonate with users due to their established popularity within the dataset. It's a straightforward baseline that sets an expectation for comparison, reflecting a common user behavior of engaging with what's already popular or widely favored. This approach doesn't account for personalization but serves as a benchmark against which more sophisticated recommendation models can be evaluated.

Random recommenders operate by randomly selecting items from the available pool of artists or tracks without considering any user preferences or item popularity. This baseline model disregards any inherent patterns in user behavior or item popularity. It serves as a simplistic comparison point to assess whether more complex recommendation systems perform better than purely random selections. Random recommenders assume that users' tastes are unpredictable and that suggesting items at random could potentially yield similar outcomes to personalized or popularity-based approaches. However, they generally lack the ability to offer meaningful, tailored recommendations based on user preferences or item characteristics. Our model should outperform these baselines.

The evaluation criterion employed to assess the model performances centered on accuracy. It involved computing the proportion of accurately identified artists for each user relative to the total number of artist identifications made. An accurate identification was determined based on whether the artists users engaged with in our dataset appeared among the top recommended similar artists generated by our model.

2.3 Features & Data Processing

Reiterating from our data cleaning, we did some data processing on our dataset to improve readability, by creating new merged data frames 'artist_user_merge' and 'tag_user_merge' for increased data readability.

To build a collaborative recommender system, our primary focus centered on leveraging the 'artist_user_merge' dataset. This file encapsulates information regarding user interactions with artists. By constructing a sparse matrix with 'userID' columns and artists' rows, we captured the listening patterns of each user across various artists. This mapping facilitated the assessment of artist similarities derived from user behavior, particularly their 'plays' or 'weight'. In this context, our emphasis revolved around utilizing the 'userID' and 'artistID' features, employing the 'listening counts' or 'weight' to populate our sparse matrix.

For the collaborative recommendation system, we also standardized the listening weights using standard scaling, which involves transforming the data to possess a mean of 0 and a standard deviation of 1. This process centers the data around zero and adjusts its scale based on its standard deviation. The primary goal behind this approach was to enable our models to converge more efficiently and avoid potential uneven penalization when employing regularization techniques.

In contrast, our approach for the content-based recommender system primarily relied on the 'tag_user_merge' dataset. Here, our objective was to discern similarities between artists based on tags associated with them.

3. MODEL

3.1 Final Model

The final model utilizes a predictor that is a hybrid recommendation system, leveraging both collaborative and content-based filtering techniques to generate artist recommendations for users on music listening platforms.

The predictor employs collaborative filtering by analyzing users' listening behavior, generating recommendations based on similarities between users in the user-item matrix. This method recommends items based on users' historical interactions. We used cosine similarities to achieve this similarity calculation.

It also utilizes content-based filtering by considering features of artists (tag counts) to create artist profiles. It recommends artists similar to those the user has already liked, based on the characteristics and counts of those artists.

3.2 Attempted Models

To begin, I first implemented my baseline models to gauge the accuracy of my models and form a baseline accuracy to beat. The two baseline models I implemented were a random recommender and a popularity based recommender.

For the random recommender, I simply generated a list of random artistIDs for each userID based on the artists that they had not interacted with. I did this by implementing a user interaction matrix, where I stored the artist interactions for each user and used it to recommend random artistIDs. For the popularity based recommender, I simply recommended the top 10 artists for every user. If they had interacted with the artist in the original dataset, this was a correct prediction and otherwise it was an incorrect prediction. This was not a true recommender system, as it recommends the same values for each user, but its accuracy speaks into the prevalence of popular artists in interaction data and importance of popularity as a crucial feature for accurate content recommendation.

Even with this however, as expected, both baselines had lackluster results, resulting in accuracies of ~0.388 & ~0.541 for each of the models.

My next attempt was to approach collaborative recommendation based on user-user interactions and using cosine similarity as a measure to quantify the similarity between different users.

$$S_C(A,B) := \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\cdot\sqrt{\sum\limits_{i=1}^{n} B_i^2}},$$

Figure 3.1 - Cosine Similarity Equation

By employing cosine similarity, I aimed to gauge the likeness in preferences or behaviors between users. This metric measures the cosine of the angle between two user vectors in a high-dimensional space, indicating how similar their tastes are based on their interactions with items or content. My accuracy on this approach was ~0.683, which was an improvement from our baseline. I wanted to further see if content based filtering would improve this value so I implemented that as well.

To approach content-based recommendations, I did this using tag values to create artist profiles based on their genre tags which would later be vectorized and used to generate similar artist profiles. TF-IDF Vectorizer is a measure of originality of a word by comparing the number of times a word appears in a document with the number of documents the word appears in.

The code aggregates and combines the tags associated with each artist. This amalgamation results in comprehensive artist profiles that encapsulate the tags related to each artist to make artist profiles. Employing the TF-IDF (Term Frequency-Inverse Document Frequency) vectorization technique, the code transforms the collected artist tag data into numerical vectors.

Finally, utilizing cosine similarity, the model computes the similarity between the user profiles (previously established based on interactions) and the artist profiles constructed from tags. This similarity estimation quantifies the likeness or affinity between user preferences and artist characteristics expressed through tags.

Some unsuccessful model attempts I had included attempting to implement matrix factorization methods like Implicit ALS (Alternating Least Squares) [1] for collaborative filtering. We say that this uses 'implicit' ALS, as we can only assume that a user likes an artist if they listen to them. Therefore, to account for this, the approach learns by using different confidence levels on binary preferences: unseen artists are treated as negative with a low confidence, whereas present artists are treated as positive with a much higher confidence [5].

$$loss = \sum_u \sum_i C_{ui}(P_{ui} - X_u Y_i)^2 + \lambda(\|X_u\|^2 + \|Y_i\|^2)$$

Figure 3.2 - Implicit ALS Loss Equation

Although the model did work, it needed some fine-tuning to replace my final model as it was extremely time and memory consuming and inefficient. The model was also not very accurate (~ 0.581 accuracy) for unknown reasons, but this might have been due to the hyperparameter tuning and maybe in the future I would try to parallelize the computation to make it run faster.

The strength of using user-based collaborative filtering is that it was generally pretty simple to implement and accurate. However, in terms of scalability possibly using item-based filtering would be a better choice in the future, as it would be very computationally expensive for large datasets due to similarity calculations. The factorization model is typically better for capturing intricate patterns that our model cannot, but however it requires more computational resources and training time and may struggle with cold-start problems and data sparsity.

My final hybridized model was a combination of the two content-based and collaborative models, where I combined the weighted similarity scores to produce a final recommendation list.

4. LITERATURE

In terms of literature, I had an initial desire to create a music based recommender. I chose the LastFM dataset from a github repository that provides quality datasets for recommender systems [2]. This dataset was most likely gathered from existing articles, recommender sites and academic experiments, as LastFM has an API that allows you to gather user and artist data.

When it came to model and approach decisions, I had read many recommender system implementations that also used music tagging data and social interaction data. One source I examined also pointed out the benefits of matrix factorization using ALS and attempts on improvement by using Logistic Matrix Factorization. It states that cutting-edge recommendation systems such as Spotify's use this to generate their lists of related artists [3].

Similar to our approach, cipher813 also employs collaborative filtering to produce artist similarity rankings. However, they utilize libraries that may be of interest for further improvements upon my model such as sci-kit learn for K Means Clustering models, turicreate to evaluate various recommender algorithms, and fuzzywuzzy for fuzzy matching of artist names. They stated that they pass their dataset through a latent mapping algorithm, K-nearest neighbors, to determine cosine similarity amongst the user/artist relationships [4]. This helps to determine which artists are most similar as in shortest distance apart within this latent mapping. Overall, it seems to be a common consensus that collaborative filtering is one of the most accurate techniques for achieving this predictor. All sources used a different means of implementation but the same base concept for their individual models.

5. RESULTS & CONCLUSION

The music artist recommendation system constructed for this study utilized a hybrid approach combining collaborative and content-based filtering techniques. The evaluation involved comparing the model against baseline recommenders such as popularity-based and random recommenders, aiming to discern its performance and significance.

Our analysis commenced with exploratory data investigation, highlighting the dataset's characteristics, including user interactions, artist profiles, and tag assignments. Notably, the examination of popular artists and genres provided crucial insights into user preferences and music trends, serving as foundational benchmarks for our predictive models.

Two baseline models, focusing on popularity and randomness, set initial performance benchmarks. These rudimentary models exhibited limited accuracy, indicating the necessity for more sophisticated recommendation systems.

Collaborative filtering, leveraging user-user interactions through cosine similarity, presented promising results, significantly surpassing baseline accuracies. Content-based filtering, utilizing artist profiles constructed from tag data, although attempted, did not demonstrate substantial improvements in accuracy. Attempts to implement matrix factorization techniques like Implicit ALS proved computationally expensive and faced challenges in accurately predicting artists.

The final hybridized model, a fusion of collaborative and content-based approaches, combined weighted similarity scores to generate recommendations and resulted in a final accuracy of ~.715. Despite challenges faced with scalability and computational complexities, this model exhibited improved accuracy compared to baselines and individual techniques.

	name	accuracy
0	baseline	0.388
1	popularity baseline	0.541
2	collaborative-based	0.683
3	content-based	0.573
4	hybrid	0.715

Literature review and comparative analysis with other recommender system implementations supported the efficacy of collaborative filtering techniques for music recommendation systems. The consensus across various sources affirmed the significance of collaborative filtering while highlighting diverse approaches and tools employed in model construction.

In conclusion, the hybridized recommendation model, incorporating collaborative and content-based filtering techniques, presented improved accuracy compared to rudimentary baselines. Collaborative filtering emerged as a robust technique, while the hybrid model demonstrated the potential for enhanced recommendations, despite computational complexities. The proposed model succeeded over more complicated machine learning models due to its robustness and simplicity. Further enhancements could explore other matrix factorization techniques, optimizing computational efficiency and addressing data sparsity challenges for more scalable and accurate recommendations.

6. CITATIONS

[1] Caserec. "Caserec/Datasets-for-Recommender-Systems: This Is a Repository of Topic-Centric Public Data Sources in High Quality for Recommender Systems (RS)." GitHub, 2018, github.com/caserec/Datasets-for-Recommender-Systems.

[2] Frederickson, Ben. "Finding Similar Music Using Matrix Factorization." Ben Frederickson, www.benfrederickson.com/matrix-factorization/. Accessed 5 Dec. 2023.

[3] "Tutorial - Recommending Music with the Last.Fm 360k Dataset." Tutorial - Recommending Music with the Last.Fm 360K Dataset. - Implicit 0.6.1 Documentation, 2017, benfred.github.io/implicit/tutorial_lastfm.html.

[4] McMahon, Brian. "Cipher813/Recommender_system: A Music Recommender System Based on Data from Last.Fm." GitHub, 2018, github.com/cipher813/recommender_system.

[5] Fred, Ben. "Benfred/Implicit: Fast Python Collaborative Filtering for Implicit Feedback Datasets." *GitHub*, 2022, github.com/benfred/implicit/tree/main.