

Assignment 2

Luis Dale Gascon

Computer Science

Towson University

lgascon1@students.towson.edu

I. RELATED WORKS

A. LightFM

LightFM is a hybrid approach for recommendation systems that aims to solve the problem of “cold-start,” where the recommendation system cannot draw any inferences for new users or items due to a lack of information. [1] touts LightFM’s ability to perform as well as a pure collaborative matrix factorization model when interaction data is dense.

B. Collaborative Filtering Recommender Systems

A research paper written by [2] provides a deep dive into collaborative filtering systems. The majority of the paper discusses the different algorithms of collaborative filtering, highlighting its limitations and challenges. The challenges with collaborative filtering systems are privacy and security. Collaborative filtering systems perform better when there is more information about users. In the event that user information is hosted on a centralized server, a single breach could expose a lot of user information. Even with the clever implementation of distributed systems, where user data is encrypted and decrypted by the CF system through keys, users with unusual tastes may still be exploited to reveal their personal information, as stated by [3]. Another challenge with collaborative filtering systems is “review bombing,” where a group of users will negatively rate an item with malicious intent.

II. PROBLEM DEFINITION

The recent surge in the popularity and production of anime has led to an unprecedented variety of shows, many of which are short in length and easily consumed in a single sitting. With a seemingly endless array of anime series accessible through online streaming services, users are faced with a dilemma: the paradox of abundance. There is an abundance of different anime shows and their titles could be misleading as to what the show is about.

This overwhelming choice makes it difficult for viewers to efficiently discover new anime shows that match their tastes, especially when searching for similar short series.

III. APPROACH DESCRIPTION

A. Dataset

The dataset is taken from the MyAnimeList website. It consists of two CSV files: Anime.csv lists the anime shows and their metadata, such as genre, number of episodes,

rating, and the population of the anime’s “group.” This dataset contains 12,294 unique anime shows.

Ratings.csv contains user-generated content. It contains the anime_id (which matches a row in Anime.csv), the respective rating by a user, and the user_id. This dataset contains more than 6 million rows.

B. Approach

For our baseline recommendation system, given that our dataset is dense, we will be using collaborative filtering with a memory based approach via cosine similarity.

For our improved recommendation system, we will be using LightFM.

C. Data Preprocessing

We imported the datasets as Pandas dataframes. We merged the two dataframes on anime_id. Users who haven’t rated an anime have a rating of -1. We removed those entries by replacing -1 with NaN and dropping any rows with that value for rating.

We then split the dataset into training and test sets (70/30).

Both cosine similarity and LightFM expect an input of a sparse matrix, so we retrieved the pivot table of the merged dataframe and passed that value to scipy’s csr_matrix function to generate a sparse matrix.

D. Collaborative Filtering

For our collaborative filtering approach, we initialized sklearn’s NearestNeighbors model with the following arguments: metric="cosine" and algorithm="brute". We then trained the model with the training sparse matrix.

E. LightFM

LightFM’s build system is broken with an issue recently reported: <https://github.com/lyst/lightfm/issues/725>

IV. EXPERIMENTATION

During our experimentation, we encountered multiple errors, including LightFM’s build system failing when importing the package and difficulties implementing functions to evaluate the performance of our memory-based collaborative filtering system. Pivoting towards a different hybrid model was not possible due to time constraints.

REFERENCES

- [1] M. Kula, “Metadata Embeddings for User and Item Cold-start Recommendations.” [Online]. Available: <https://arxiv.org/abs/1507.08439>

- [2] B. Schafer *et al.*, “Collaborative Filtering Recommender Systems,” 2007, p. .
- [3] B. A. Ojokoh *et al.*, “Privacy and Security in Recommenders: An Analytical Review,” *Artificial Intelligence Review*, vol. 58, no. 11, p. 351, Aug. 2025, doi: 10.1007/s10462-025-11333-4.