Creating a Recommender System for Beers

Team 37: Timothy Alland, Brandon Butler, Phuc Nguyen, and Aidan Lorenz
GitHub: https://github.com/b-butler/beer-recommender-erdos-fall-2023

Overview

Stakeholders: Beer consumers, Online beer sellers, Craft breweries
KPIs: Weighted normalized recall at k
Baseline: Recommend most popular beers from breweries or beer styles a reviewer has positive reviews for

We created a beer recommender that will suggest beers for a user based on an inputted list of beers that they like. To build the underlying model, we used a Beer Advocate dataset from data.world. The data contains 33,365 unique users, 65,395 unique beers, and a median of 3 reviews (1-5 stars) per user. In the dataset, we have multiple beer features, but no user features. Recommendation systems provide benefit on both the consumer and producer/distributor side. For producers/distributors, insights into what beers a person might also enjoy provides new revenue opportunities. For consumers, finding new recommendations may lead to greater overall satisfaction and discovering new favorites.

Approach

Recommendation systems generally follow one of two paradigms: content-based or collaborative filtering. The lack of user features limits collaborative filtering approaches to matrix factorization. In this project, we limit our focus to matrix factorization (MF).
We test 3 different flavors of MF:

1. Standard MF with Frobenius loss excluding non-interactions
2. MF using the logistic loss (implemented in LightFM)
3. MF using the warp-KOS loss (implemented in LightFM)

To use MF, we need an interaction matrix. For each of the flavors of MF described above, we use the following means, respectively, to create an interaction matrix suited toward that specific flavor of MF.

1. A 0-5 matrix where 0's represent non-interactions and values 1-5 correspond to actual ratings given
2. A -1, 0, 1 matrix where -1's are negative reviews, 1's are positive reviews, and 0's are non-interactions.
3. A 0, 1 matrix with negative reviews and non-interactions given by 0's and positive reviews given by 1's.

For each respective MF method, we performed a grid search over the following hyperparameters:

1. Latent space dimension
2. Latent space dimension, matrix positive/negative review threshold
3. Latent space dimension, k (hyperparameter in warp-KOS loss), matrix positive/negative threshold

To test model performance and optimize hyperparameters, we used a modified 5-fold cross validation with the normalized recall at $k = 10$. In the modified validation sets, we never take interactions from reviewers with less than 5 reviews, we take 1 per set for each reviewer with 5-10 reviews, and 10% of reviews from all other reviewers. The weighted normalized recall at k, is simply recall at k where the denominator is the minimum of the number of reviews in the validation set or k. This modification of standard recall at k is necessary to not penalize users with many reviews. Individual reviewer's recalls are weighted according to the number of reviews in the validation set so that the model will prioritize reviewers with more than 1 or 2 reviews in the test set.

Results

We found that the warp-KOS loss with $k = 2$ at a latent space dimension of 70 with a threshold of 2.5 performed best in the cross validation with an average score of 0.256 on the validation sets and a final score of 0.635 when trained and tested on the entire dataset. The model significantly outperforms our baseline models which had scores of less than 0.05 and 0.39 on the validation sets and entire dataset, respectively, and succeeds in recommending beers not previously seen. Our GUI implements a cold-start solution to add new users and a list of their reviews to the model, overcoming a common issue amongst MF models.

Future Work

Incorporating other recommendation systems such as content-based approaches to complement our model would provide more robust recommendations specifically for new users to the dataset.