Beer Recommender System

Written by:

Deshmukh Rohit (11304014)

Kumar Saahil (11288768)

Patel Yash (11288771)

MATH60629A: Machine Learning I: Large Scale Data Analysis and Decision Making

Professor Laurent Charlin

HEC Montreal

December 15th, 2022

1. Background

Beer is the oldest and most widely consumed alcoholic beverage and is the third most popular drink after water and tea (Wikipedia, 2022). There exists a plethora of choices when it comes to beers, and it can get confusing. We usually take recommendations from friends on which beers to try next, but we don’t always like those beers. As a team of 3 beer enthusiasts, we wanted to leverage machine learning and the available data from BeerAdvocate and Kaggle to get beer recommendations based on our favourite beers and tastes.

Use-Cases of a Beer Recommendation System:

A. **Personalized Recommendations:** Improves the user experience by suggesting dynamic recommendations for different types of audiences, like how Netflix does.
B. **Better Product search experience**: It helps in categorizing beers based on their features. For example, alcohol content, IBU, etc.

2. Objectives

The following are the objectives of this project:

1. To build a Beer Recommender System applying content-based and collaborative filtering models using user reviews, ratings, and beer characteristic datasets.
2. Evaluate the performance of each model and choose the best model.
3. Perform qualitative analysis of the beer recommendations suggested by the models.

3. Introduction

With the growth of the internet and e-commerce systems there is a abundant information availability and hence makes it difficult to deal with. Therefore, personalized recommendation systems provide us with the important data by employing information filtering systems. It comprises of two methods – content based filtering and collaborative filtering.

3.1. Content Based Recommender system

Based on the previous ratings given by the user, the model learns to make recommendations by evaluating the feature similarity among the items. For example, based upon the ratings given by a user for different items, the model will learn to recommend the category of item which is positively rated by the user. Based on the items the user has previously rated, a content-based recommendation system creates a user profile. A user's interests are shown on their profile, which can also change to reflect new interests. The basic recommendation procedure involves comparing the user profile to the characteristics of the content object. An information access method will be more useful with a highly accurate user interest profile. (Mohanty et al., 2020)

3.2. Collaborative Filtering Recommender system

This technique uses user behaviour for making recommendations. This approach does not use any features corresponding to users or items. It uses a utility matrix and is preferred recommendation system given that it is independent from any additional information.

Drawbacks of content-based recommendation system can be overcome by the collaborative filtering approach for example, it can make predictions for those items for which content is not available. It uses the rating given by other users for making recommendations. This system assesses the quality of an item based on peer review. It can also suggest products with varied variety as long as other users have shown the interest in the content (Mohanty et al., 2020).

Collaborative filtering has demonstrated that it is one of the most effective techniques given its simplicity(Breese et al., 2013). Many researchers have used collaborative filtering techniques to make quality recommendations. All of them make recommendations based on user-item matrix consisting of users and items and their ratings. There are two methods in collaborative filtering(Sarwar et al., 2001):

- User based collaborative filtering

  This technique assumes that a good way to find a certain user's desired item is to find other users who have similar interest. So, it finds user's neighbours based on user similarities and then combine the neighbours' rating and express them by similarity weighted averaging.

- Item based collaborative filtering

  This technique investigates a set of items that the user has already rated and computes how similar they are to the target item under recommendation. It also combines the user's previous preference based on the item similarities.

Jeong in his work used an item based collaborative filtering in his movie recommendation system (Jeong, 2021). The prediction algorithm we developed for collaborative filtering was inspired from the approach adopted in this article where user ratings are predicted by the weighted average of similarity between the movies rated by users. But in his approach, he does not mention anything about evaluating the performance of the recommender system.

Novelty in our approach compared to existing work – 1. Evaluation of content-based model for the quality of recommendations 2. Comparison between content based and collaborative filtering model based on the variety of recommendations. Of all the material we saw for previous works in the field, we found that nobody has divided their data into train and test matrices to evaluate the performance. With our project we wanted to have a better intuition of how recommender systems work and hence, we divided our data into test and train to compare the performance of recommender systems on various models we implemented. Qualitative and Quantitative analysis was done for both the models – Variety comparison and calculation of precision and recall for both the models.

4. Methodology

To develop a recommender system for beers, we have used two recommendation techniques: Content-based filtering and Collaborative filtering. Initially, data preprocessing and cleaning are done to get the data where all the features are comparable. In the content-based approach, we used the beer characteristics data to recommend relevant beers to the users. In the collaborative filtering approach, the user rating data for specific beers is used for recommendations. Within collaborative filtering, we developed the recommender system for two of its classes; item-based approach and user-based approach.

4.1. Data Preprocessing

This project is built around two datasets. The first dataset consists of the beer-name, user-name and ratings. The second dataset consists of beer profiles and its characteristics such as chemical composition and tasting profiles. The beer characteristics are obtained through a bag of words encoding and counting the number of words for each characteristic from the user review raw data. These characteristics are then scaled or normalized with a min-max scalar. The two datasets are merged on beer names and the final data is used for modelling purposes.

user	beer_name	rating	Style	ABV	Min IBU	Max IBU	Astringency	Body	Alcohol	Bitter	Sweet	Sour	Salty	Fruits	Hoppy	Spices	Malty
bloberglawp	Leffe Blonde	3.5	Blonde Ale - Belgian	0.114783	0.230769	0.3	0.411765	0.676471	0.352941	0.382353	1.0	0.323529	0.0	0.823529	0.470588	0.647059	0.941176
MrHungryMonkey	Leffe Blonde	3.5	Blonde Ale - Belgian	0.114783	0.230769	0.3	0.411765	0.676471	0.352941	0.382353	1.0	0.323529	0.0	0.823529	0.470588	0.647059	0.941176
irishkyle21	Leffe Blonde	3.5	Blonde Ale - Belgian	0.114783	0.230769	0.3	0.411765	0.676471	0.352941	0.382353	1.0	0.323529	0.0	0.823529	0.470588	0.647059	0.941176
jjjeremy	Leffe Blonde	4.0	Blonde Ale - Belgian	0.114783	0.230769	0.3	0.411765	0.676471	0.352941	0.382353	1.0	0.323529	0.0	0.823529	0.470588	0.647059	0.941176
tr4nc3d	Leffe Blonde	4.0	Blonde Ale - Belgian	0.114783	0.230769	0.3	0.411765	0.676471	0.352941	0.382353	1.0	0.323529	0.0	0.823529	0.470588	0.647059	0.941176

Figure 1: Merged Data

The data has 1258 unique beers and 22293 unique users.

4.2. Content-Based Filtering

The data used in content-based filtering is username, beer name and beer characteristics. User ratings are not considered in making recommendations. In content-based filtering, we recommended beers in two ways. First, to the users according to their purchasing history, and second according to the beer name. The data is filtered such that only beers whose average rating over all the users is more than or equal to 3.5. This filtering is done so that we only recommend highly rated beers according to its content.

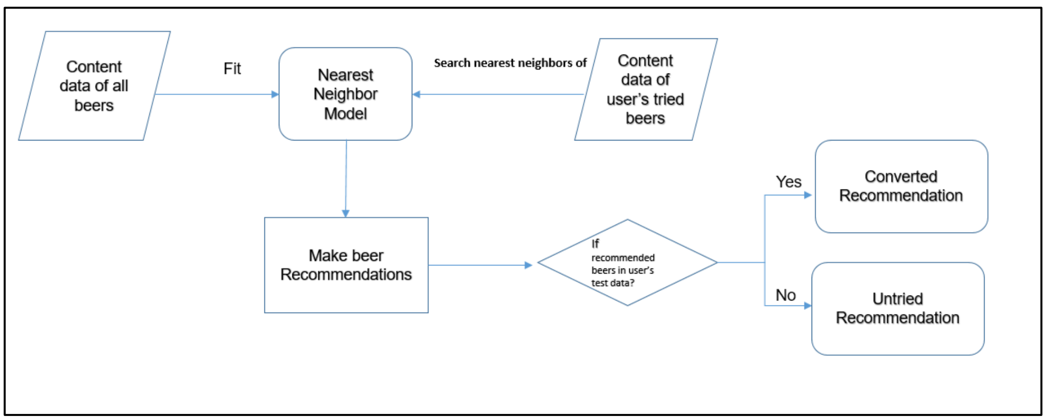
4.2.1. According to user data

a. Test-Train split

The user's purchasing history is divided into training and testing (80-20 split) data. This is done for each user separately, to ensure there is uniformity in test and train data. For example: If user A and user B have tried 100 beers each, then the test data will have 20 beers from user A and 20 beers from user B, while the rest of the data will be in training data. The test dataset will be used further for performance evaluation.

b. Nearest Neighbors

The Nearest neighbour model is fitted on all the beer's content data, which consists of the beer name and its characteristics (tasting and chemical compositions). The nearest neighbours are calculated with the cosine distance metric. Suppose a recommendation has to be done for a particular user, the training data of that user is fed to the nearest neighbour's algorithm. The model will search for the nearest neighbours of the user's purchasing history. If we want to make the top 10 recommendations for user A, the algorithm will search for the 10 nearest neighbours based on



the content of all the beers that user A has already tried and make the recommendations.

Figure 2: Algorithm for content-based recommendation

c. Performance evaluation

After the recommendations were made by the content-based model, the performance of the model was evaluated on the test data. In general, the intuition is that the user should purchase the recommended beers. As shown in figure 2, all the beers that are recommended to the user and are in the user's test data are classified as "converted beer". The more converted beers, the better the recommendations.

4.2.2. According to the beer name

In this type of recommendation, beers are recommended when a beer name(s) is given as input. This type of approach can be used to solve the cold start problem, as new users can be recommended the like-for-like items when they give a new beer as input. When a beer name is given as an input, the content data of that particular beer is fed to the algorithm. The model searches for the nearest neighbours of that particular beer and provides the top recommendations.

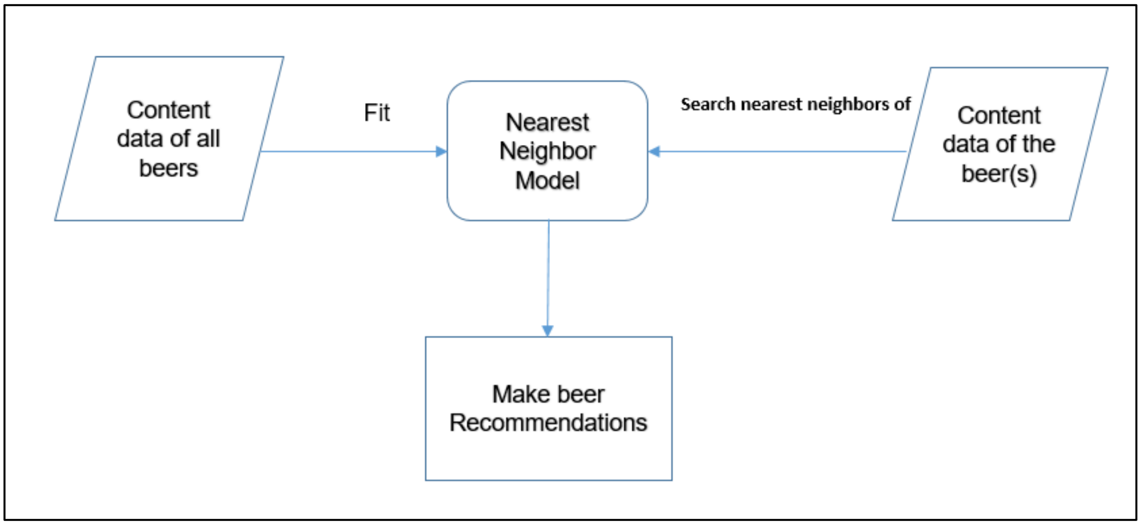


Figure 3: Algorithm for content-based recommendation

a. New User Recommendation (Cold-Start Problem)

If a new user wants to try out our recommender system. The user's beer preferences according to the taste (Malty, salty, sweet etc.) and chemical composition (IBU, alcohol % etc.) of beer are requested from the user. The content-based algorithm will recommend the closest beers according to the user's taste.

b. Performance Evaluation

The performance of this model can only be evaluated on the basis of domain knowledge. We observed that the model generally recommends beer, which has very similar characteristics to the input beer name.

4.3. Collaborative Filtering

The data used in collaborative filtering is username, beer name and user ratings. Beer tasting profile and chemical composition are not considered while making recommendations. We developed a recommender system for two approaches: User-based collaborative filtering and Item-based collaborative filtering. In collaborative filtering, we try to predict the ratings of the beers, which are not rated by the user. Once the predictions are done, the performance is evaluated by comparing the predictions with test data.

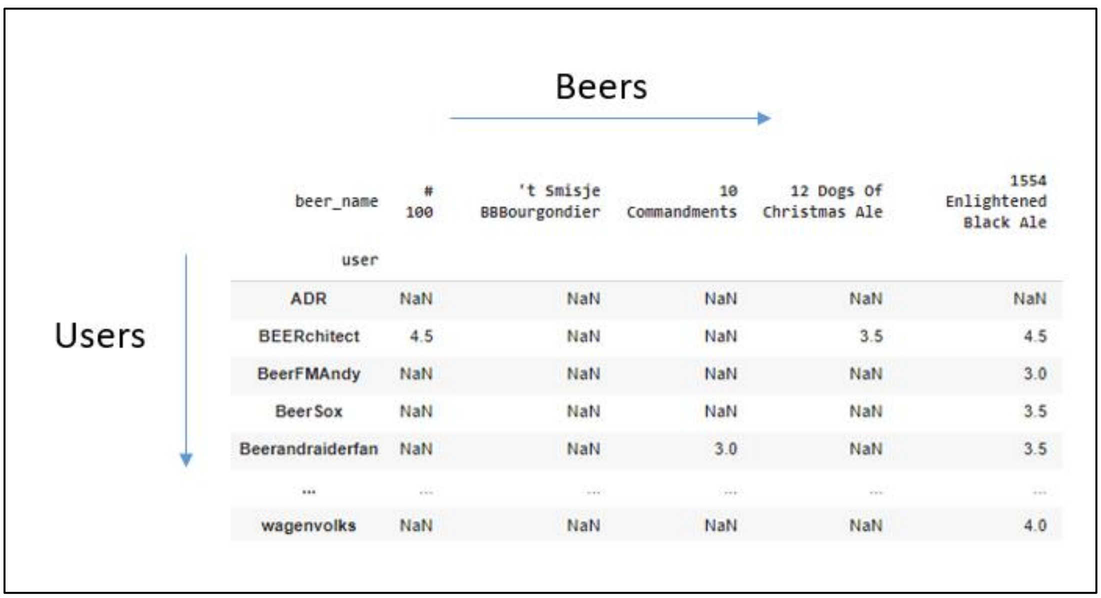
4.3.1. Test-Train-Split

Like the Content-based model, for every user the data is divided into train and test data. For each user, 80% of the data is in the training dataset and 20% of the data is in the testing dataset. The rating predictions are then done based on the training data.

4.3.2. User Based Filtering

In this technique, we predict the beers that a user might like based on the ratings given to that beer by other users who have similar tastes as of target user. In this approach, first, the similarity between the users is calculated. To calculate the similarity between users, a correlation matrix was developed. There are many ways to compute the similarity matrix- we tried cosine similarity and centred-cosine similarity (Pearson-Correlation). The rating predictions are done using both these approaches and the results are compared in the result section.

user-based filtering, we pivot the data, so that the rows represent users, and the columns represent the items. The training matrix will be of dimension u x b where u is the number of users and b is the number of beers. Following is the snapshot of what the data looks like:



beer_name / user	# 100	't Smisje BBBourgondier	10 Commandments	12 Dogs Of Christmas Ale	1554 Enlightened Black Ale
ADR	NaN	NaN	NaN	NaN	NaN
BEERchitect	4.5	NaN	NaN	3.5	4.5
BeerFMAndy	NaN	NaN	NaN	NaN	3.0
BeerSox	NaN	NaN	NaN	NaN	3.5
Beerandraiderfan	NaN	NaN	3.0	NaN	3.5
...	...	...	...	...	...
wagenvolks	NaN	NaN	NaN	NaN	4.0

Figure 4: User-based CF training data (small sample)

The predictions are done on the training data. In the above figure, the ratings are predicted for all the cells with NaN values. A prediction algorithm is developed that calculates the unknown ratings based on user similarity. The ratings are predicted using the following formula:

$$R(u,b) = \frac{\sum_i S_{u,u_i} \times R_{u_i,b}}{\sum_i S_{u,u_i}}$$

R- Rating, **u** – user, **b** – beer**, S-** Similarity

The correlation matrix, we discussed above gives us the values of similarities between user u and user u_i. This formula tells us that the predicted rating for user u and beer b is the weighted average of ratings for beer b by similar users. This essentially means that the weight for each rating given by user u_i for beer b becomes greater if user u and user u_i are closer.

The prediction matrix will be of the same dimensions (u x b) as of train and test matrix. Predicted ratings are filled in the prediction matrix for all the cells having NaN values in train data. The beer ratings by the user in train data will appear as NaN values in the Prediction matrix.

4.3.3. Item Based Filtering

We follow a similar approach as above for predicting the user ratings. However, in this technique, we predict the ratings based on the similarity of beers rated by the users. Unlike the previous approach, the similarity matrix is calculated between the beers. The correlation matrix is developed using two ways- cosine similarity and centred cosine similarity. The recommendations are made using both ways and the comparison between the two approaches is carried out in the results section.

In this technique, we keep the beer names as rows and usernames as columns to develop the prediction algorithms. The predictions are done on training data. Following is the snapshot of what the training data looks like:

Users

Beers

user	ADR	BEERchitect	BeerFMAndy	BeerSox
beer_name				
# 100	NaN	4.5	NaN	NaN
't Smisje BBBourgondier	NaN	NaN	NaN	NaN
10 Commandments	NaN	NaN	NaN	NaN
12 Dogs Of Christmas Ale	NaN	3.5	NaN	NaN
554 Enlightened Black Ale	NaN	4.5	3.0	3.5
...	...	...	...	...
ZÔN	3.5	3.5	NaN	NaN

Figure 5: Training Data-CF Item Based (small sample)

The ratings are predicted for all the NaN values in the matrix. The prediction formula for item-based collaborative filtering is as follows-

$$R(b,u) = \frac{\sum_i S_{b,b_i} \times R_{b_i,u}}{\sum_i S_{b,b_i}}$$

R- Rating, **u –** user, **b –** beer**, S-** Similarity

In this case, the correlation matrix gives values of similarity between beer b and beer b_i. The formula suggests that the predicted rating for beer b given by user u is equivalent to the weighted average of ratings for similar beers by user u. This also signifies that if beer b and beer b_i are close to each other, then their weight for each rating becomes greater. Here the basic assumption is that a user rates similar items in a similar way.

In this algorithm, we fix the user and iterate over the beers to fill the prediction matrix column by column. Similar to user-based filtering; the predicted ratings are filled in the prediction matrix for all the cells having NaN values in train data.

4.3.4. Performance Evaluation

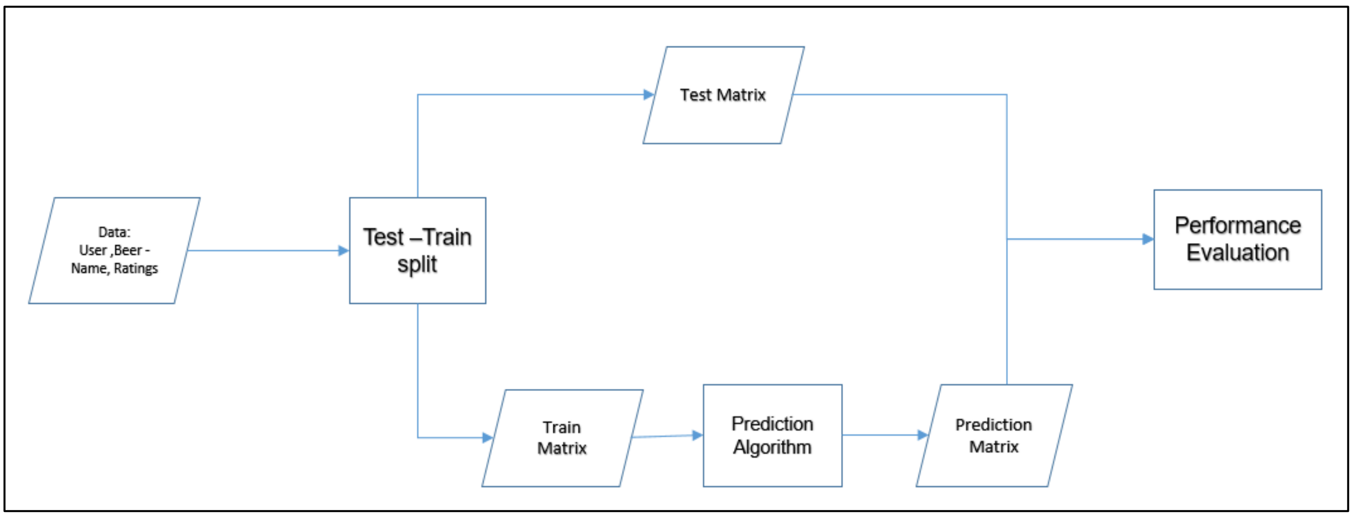


Figure 6: Performance evaluation pipeline

Once the test, train and prediction matrices for each of the approach is obtained, the performance of the predicted ratings is evaluated. We did a quantitative and qualitative analysis, to understand how good these recommendations are.

4.3.5. Quantitative Analysis:

Prediction and test matrices are used to evaluate the performance of the model.

Test Data:

beer_name user	# 100	't Smisje BBBourgondier	10 Commandments
ADR	NaN	NaN	NaN
BEERchitect	NaN	NaN	NaN
BeerFMAndy	NaN	NaN	NaN
BeerSox	4.0	NaN	NaN
Beerandraiderfan	NaN	NaN	NaN

Figure 7: Test Matrix(Small sample)

Prediction Data:

Unnamed: 0	# 100	't Smisje BBBourgondier	10 Commandments
ADR	4.150219	3.898131	4.105672
BEERchitect	NaN	3.872590	4.038190
BeerFMAndy	4.101651	3.891283	3.974072
BeerSox	4.105113	3.879341	4.057878
Beerandraiderfan	4.103471	3.930669	NaN

Figure 8: Prediction Matrix(Small sample)

Consider the small samples of test and prediction matrices. The predicted ratings in the prediction matrices for which there is no data in the test matrix will be ignored while calculating error. For instance, to calculate error (RMSE) in the given sample, we will only consider the user "BeerSox" rating of "# 100" beer. The error calculations are done over all the beers and all users to get the total RMSE of the recommender system.

4.3.6. Qualitative Analysis

The prediction matrix is used to recommend beers to a particular user. If we want to recommend k beers to the user, then the top k beers having the highest predicted ratings will be recommended to the user. Intuition tells us that if the recommended beer is already tried by the user i.e., the beer is in the user's test data, then the recommendation is good. The number of recommendations made to each user is based on the user's purchasing history. In our analysis, we recommended 4 times the number of beers in the user's test data and analyzed various performance measures such as precision and recall. The flowchart below sum's up our qualitative analysis.

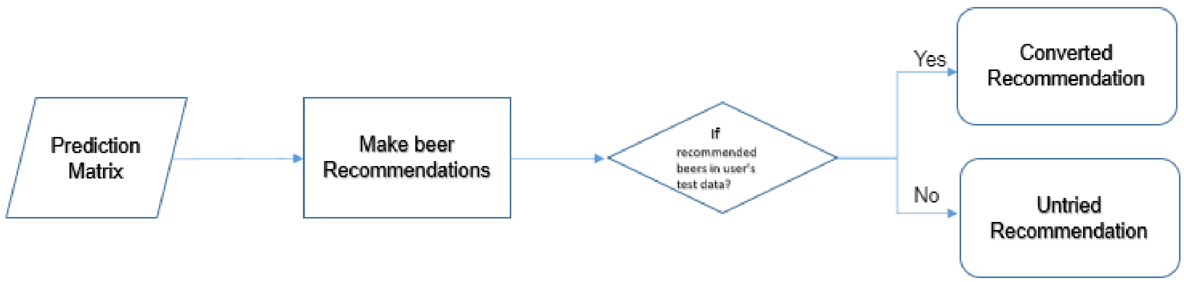


Figure 9: Qualitative Analysis CF Flowchart

5. Result and Implications

5.1. Content-Based Model

The top 180 users with a maximum number of ratings were selected for the analysis. The average Recall of the Content-Based model is 26.53 % and the average Precision is 6.63%. Over 1 in every 5 beers that the users are drinking is from our Content-Based recommender system!

	user	number_recommended_beers	number_test_beers	number_converted_beers	Recall	Precision
0	Hibernator	244.0	61.0	12.0	19.672131	4.918033
1	kmpitz2	312.0	78.0	22.0	28.205128	7.051282
2	Crosling	268.0	67.0	17.0	25.373134	6.343284
3	blitheringidiot	248.0	62.0	15.0	24.193548	6.048387
4	smcolw	400.0	100.0	30.0	30.000000	7.500000
...	...	...	...	...	...	...
175	tpd975	304.0	76.0	18.0	23.684211	5.921053
176	mothman	328.0	82.0	23.0	28.048780	7.012195
177	bashiba	272.0	68.0	9.0	13.235294	3.308824
178	johnmichaelsen	288.0	72.0	24.0	33.333333	8.333333
179	Bierguy5	248.0	62.0	12.0	19.354839	4.838710

180 rows × 6 columns

Figure 10: Snapshot of Precision and Recall values for the Content-based Model

5.2. Collaborative Filtering Model

Following are the RMSE values for the Collaborative Filtering Models (User-Based & Item-Based) with both Pearson-Correlation and Cosine-Similarity:

	Pearson-Correlation	**Cosine-Similarity**
User-Based	**0.5757**	**0.6531**
Item-Based	**0.6134**	**0.6872**

Table 1: RMSE Values of Collaborative Filtering Model

As evident from the results, both models achieved better performance with Pearson-Correlation as it handles the "tough rates" and "easy raters" to remove the user bias in ratings by centering them around zero which also in a certain way normalizes the ratings. The User-Based model performed slightly better than the Item-based model which can be explained by comparatively higher user-similarity than item-similarity.

Following are the Precision and Recall results for the User-Based Collaborative Filtering Model:

	User	recommeded_beers	converted_beers	precision	recall
0	ADR	280.0	18.0	6.428571	25.714286
1	BEERchitect	552.0	113.0	20.471014	81.884058
2	BeerFMAndy	336.0	39.0	11.607143	46.428571
3	BeerSox	284.0	17.0	5.985915	23.943662
4	Beerandraiderfan	328.0	36.0	10.975610	43.902439
...	...	...	...	...	...
175	wagenvolks	256.0	12.0	4.687500	18.750000
176	weeare138	324.0	25.0	7.716049	30.864198
177	womencantsail	416.0	37.0	8.894231	35.576923
178	woodychandler	284.0	16.0	5.633803	22.535211
179	zeff80	520.0	43.0	8.269231	33.076923

180 rows × 5 columns

Figure 10: Snapshot of Precision and Recall values for the CF Model

The average Recall of the Collaborative filtering model is 31.58 % and the average Precision is 7.89%.

5.3. Comparison: Collaborative Filtering Model and Content-Based Model

5.3.1. Qualitative Comparison

Both models are compared on the variety of recommendations given by them. As expected, the content-based model had on average only 25% variety in recommendations (unique style of beers) whereas the collaborative filtering model had 75 % unique beer styles in its recommendation.

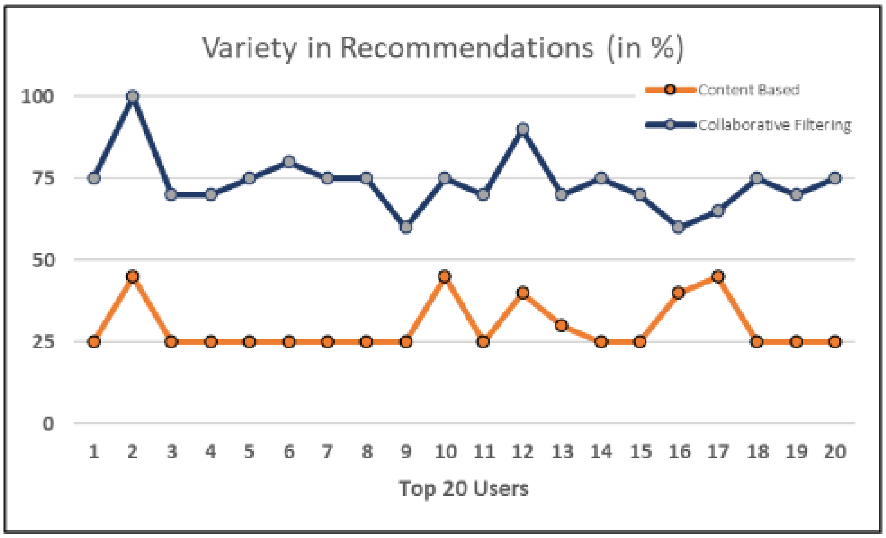


Figure 11: Variety in Recommendations (%)

5.3.2. Quantitative Comparison

Following are the Precision and Recall values for the Collaborative Filtering (User-based) and Content-Based filtering models from the analysis done over the top 180 users with maximum number of ratings:

	Precision	**Recall**
Collaborative Filtering	**7.896955 %**	**31.5878 %**
Content-Based Filtering	**6.6334 %**	**26.533746 %**

Table 2: Precision and Recall values for CF and CB model

The Content-based Filtering model has better Precision and Recall values.

6. Conclusion

The User-based collaborative filtering model has performed better than the Content-based filtering model in both qualitative and quantitative comparisons. The User-based Collaborative filtering model with Pearson Correlation performed the best. Our model gives good intuition of how the recommender system works, how ratings are predicted and finally how the performance is evaluated for different types of recommender systems.

Our recommendation is to use the Content-based filtering model for new users in order to tackle the cold start problem encountered with the collaborative filtering model. And Collaborative filtering model will be used for old users with available data of ratings.

6.1. Further Scope

Our aim with this project was to gain a better understanding of how recommender systems work, hence we split the data in train and test matrices and did the performance evaluation. More complex models like latent factors - SVD can be implemented to get better performance. However, in implementing SVD data can't be exclusively split in test and train, hence it is not very intuitive as we wanted to compare the performance of all of our models on same train and test data. However, SVD can solve all the potential drawbacks of the collaborative filtering model namely data sparsity, scalability, and the cold start problem. Hybrid Models can also be used to achieve better performance and address some of the shortcomings of the CF model.

Another important factor for beer recommender system is the geographical data. All beers will not be available everywhere. Hence, it is important to provide the user with beer recommendations from his respective location (Country, State, etc.). Geographical data of users and beers was not available in the data. Access to geographical data of users and beers would greatly improve the practicality of our model.

Link to the notebook!

Cheers!

7. Bibliography

Breese, J. S., Heckerman, D., & Kadie, C. (2013). *Empirical Analysis of Predictive Algorithms for Collaborative Filtering* (arXiv:1301.7363). arXiv. http://arxiv.org/abs/1301.7363

Jeong, Y. (2021). *Item-Based Collaborative Filtering in Python*. https://towardsdatascience.com/item-based-collaborative-filtering-in-python-91f747200fab

Mohanty, S. N., Chatterjee, J. M., Jain, S., Elngar, A. A., & Gupta, P. (Eds.). (2020). *Recommender system with machine learning and artificial intelligence: Practical tools and applications in medical, agricultural and other industries*. Wiley-Scrivener.

Sarwar, B., Karypis, G., Konstan, J., & Reidl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the Tenth International Conference on World Wide Web - WWW '01*, 285–295. https://doi.org/10.1145/371920.372071

Wikipedia. (2022). Beer. In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Beer&oldid=1125947211