Beer Recommendation Model

Based on beer reviews from BeerAdvocate

By Tomer Danon
Data Science Capstone
University of Denver

Introduction

It is hard to avoid reviews and ratings in today's world. We are constantly asked to rate the establishments we visit and the products we purchase, and often we look to these reviews to decide on whether to visit somewhere or purchase something. Throughout the last two decades, online review platforms and user feedback requests have made it to nearly every industry. It could be valuable and insightful for a company to learn about it's users. With this information the company could adjust their products and services accordingly, in order to increase profits and retain high rates of customer satisfaction. Given large amounts of data on user preferences and growing computing power of the last two decades, collaborative filtering algorithms have been used to find groups or clusters of users with shared preferences. This allows the company to explore the probabilities of a user preferring a product or establishment based on their shared preferences with other users, and what the other users also prefer. Some examples of this are movie or music recommendations from Netflix or Spotify, product recommendations on Amazon, and restaurant recommendations on Yelp! and Google. This project explores collaborative filtering for a beer recommendation model based on user beers reviews from the online beer community, BeerAdvocate.

Dataset

BeerAdvocate was created in 1996 and is one of the oldest and largest online communities of beer fans and industry professionals dedicated to supporting, promoting, and discussing beer. The dataset is a collection of user beer reviews from roughly ten years ago (2012). There are five rating attributes provided for each beer review: aroma, appearance, palate, taste, and the average of these. They are numbers in the range of 0 to 5 that include decimals. Associated with these ratings are the beer name and style, the brewery name, and the user who reviewed it. There are two ID columns for the beer ID and the brewery ID as well.

The dataset consists of:

- 1,586,614 reviews
- 33,388 of users

- 66,055 beers in 104 styles
- Brewed by 5,743 breweries

The dataset is large and holds an abundance of buried information that can be extracted with a series of data manipulation techniques. In order to make recommendations based on user preferences, it is vital to organize the user data in a clearer fashion.

Data Cleaning & Preparation

The data of interest is the beer and styles the user has reviewed, and the total number of reviews they submitted. To compute this count, a grouping of the main dataframe is performed on the user profile name. This is converted to dataframe with the username as the index and a column for the number of reviews they submitted. Two other columns are created, one for the beer styles reviewed by the user, and one for the beers reviewed by the user. To achieve this, two functions are created. The first filters the dataframe for every username, groups by beer style, and averages the overall rating column. This is populated into a dictionary with the beer style as the keys and average rating the user gave for the style as the values. The second function filters the dataframe for every username, groups by beer, and averages the overall rating column. This is populated into a dictionary with the beer name as the key and brewery name, beer style, and average rating as the values. This process effectively makes an instance in the dataframe a "profile" for the user. This dataframe is used in the recommendation model algorithms.

Recommendation Models

Great beer recommendations start in one of two ways: beer(s) of preference, or style(s) of preference. For this reason, two algorithms were written for this project to handle each of these scenarios.

The first, recStyle, recommends beer to a user based on their beer style(s) of preference. The function has three inputs: the list/set of beer styles, the rating standard, and a value representing the number of recommendations for each style. The default value for the rating standard is 4.25. This instructs the the algorithm to find a cluster of users who have rated the input style(s) of preference with at least 4.25 for the overall beer rating. It can be adjusted by the

user when calling the function. When this function is called, it takes the list of style preferences and filters the user dataframe to the users who have reviewed all the styles with a rating equal to or greater than the standard rating. Next, the algorithm will inspect the users within the cluster and aggregate all the beers in the preferred style that were rated with the standard rating or greater. This yields a dataframe of beers to recommend, but it has duplicate beer entries, one for each user who reviewed it. Thus, the following step is the group this dataframe by the beer and average the overall score. This provides the user with a recommendation list with average rating form all the users in the cluster. Depending on the styles provided, this list can be very long. Therefore, the user can choose to limit the list or use the default value of 10. The function returns a dataframe that consists of the top 10 beers from each of the styles in the input style list.

The second function, recBeer, recommends beers based on specific beer preferences of a user. This function has three inputs: the list/set of beer names, the rating standard, and a value representing the number of recommendations for each style. The rating standard and number of recommendations has the same definitions and defaults as the recStyle function. When recBeer is called, it takes the list of preferred beers and filters the user dataframe to the users who have reviewed them all with an overall rating of the standard rating or greater. The rest of the steps are the same as recStyle. As a results, the function returns a dataframe with a default of 10 beers for each style of the beers provided to the function.

Results & Conclusions

There are a couple of things to consider when evaluating a recommendation model. For one, the age of the data. Given that the beer industry is constantly evolving, with new breweries and beers popping up in many cities of every state, recommendations based on outdated data could yield inaccurate evaluations. In addition, the designer of a great recommendation model with no users may never know how great the model actually is. In the case of this project, it is impossible to evaluate the recommendations without user feedback. This, however, opens the door to a future project of a more complex, machine learning algorithm that incorporates the user feedback in the recommendation model.

Sources

- BeerAdvocate: https://www.beeradvocate.com/
- Kaggle: https://www.kaggle.com/datasets/rdoume/beerreviews

Appendix

Figure 1 - Attribute Distributions

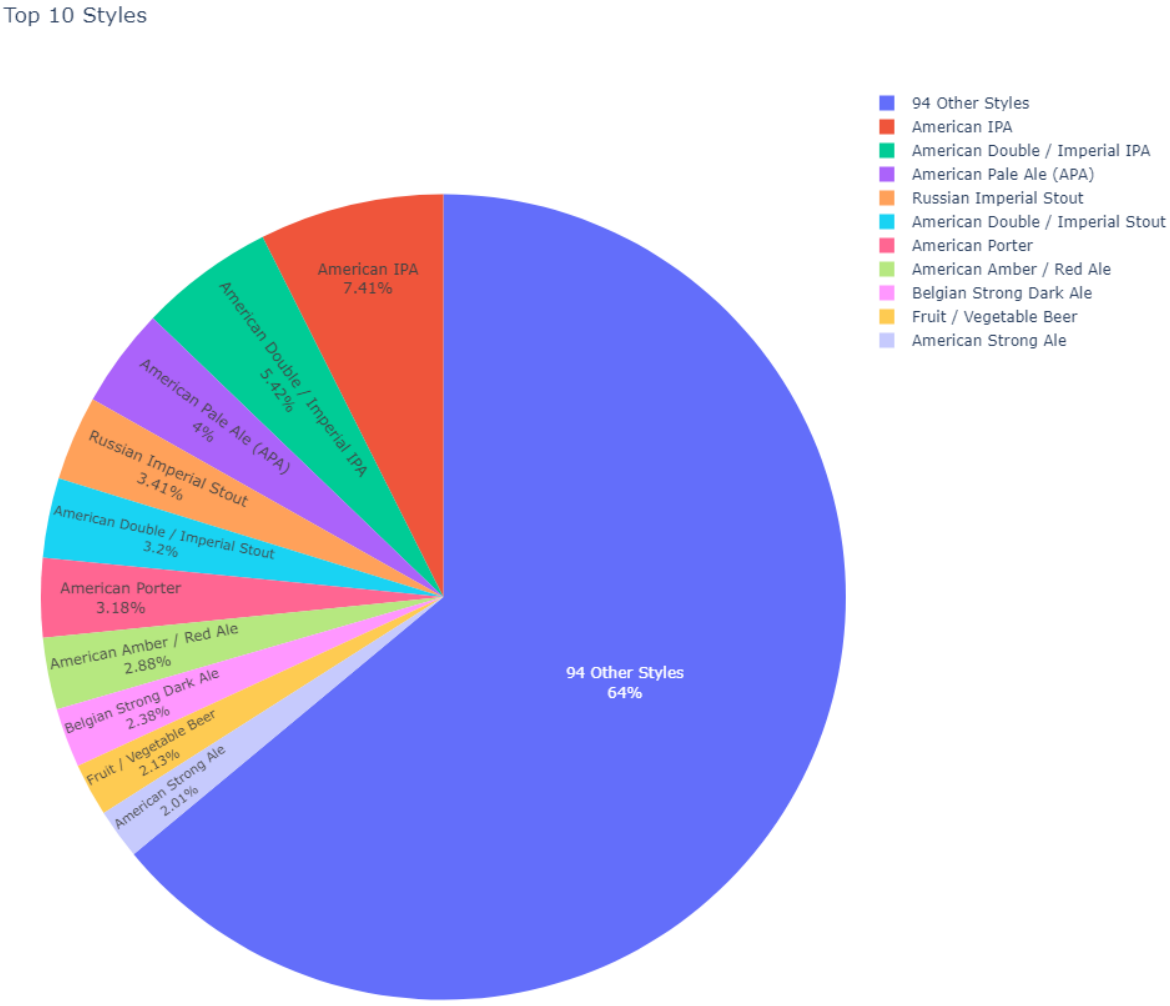

Figure 2 - Top 10 Styles