**Beer Recommendation Shiny App**

**Nilofer Rajpurkar, Jacob Collins, Mitch Lamm**

Teammate1@purdue.edu; [Teammate2@purdue.edu](mailto:Teammate2@purdue.edu); lammm@purdue.edu

**Abstract**

Our Shiny app takes a list of favored beers by the end-user, and recommends beers within that users comfort zone, as well as recommending beers with a style differing from what the user inputted. The craft beer industry was estimated to be worth $89.25 Billion in 2019, and is projected to be at 190B by 2027. Fans of craft beer love to explore and try new beers, and a beer recommendation app is a great way to quickly share possible brews that the user may also enjoy. Our app takes into consideration the beer name, style, various user ratings, and the profile who left a review, and uses this information to group the end-user into a group (cluster) of beers they make like. The app takes the profile names who also rated the same beer input by the end-user, and suggests similar beers that those users also liked. It then will also recommend beers of different styles, but also rated highly by those same users.

**Business Problem**

Beer fans like to find new beers, and going to a store or brewery can sometimes be overwhelming with so many selections. They have to spend time reading online to find beers that they may like. They need a tool that they can quickly use to find beers to try, based on other beers that they know they like.

**Analytics Problem**

Someone for shopping for a craft beer needs an easy way to create a list of beers that they like, and get a list of suggested beers based on their initial list. There are multiple beer review websites that currently exist, and utilizing that review data is one way to find other beers to recommend the viewer. The problem is that some of these datasets are over a million review, and users need a quick way to see a “top 5” list of beers to try based on what they already know they like.

**Data**

To make our app successful, we need a dataset that has beer name, brewery name, and various review data from users. Having the username of the reviewer is also helpful, so we can use that to determine if the app user has a similar flavor palate as other reviewers.

We found a dataset of Kaggle, which has 1.5 million beer reviews, which were published by BeerAdvocate, which is one of the most popular beer review sites. Our dataset was almost all populated, with the exception of the beer ABV column. To fill in this missing data, we used a prediction model using caret to populate the missing data, by predicting based on the beer style, as well as users reviews of the taste, aroma, appearance, and palate. We added in these review scores because some users may rate a taste and/or palate lower, if the beer has a higher ABV. Also, the aroma of a beer can be less pleasing if a beer has a higher ABV. We scaled these ratings using caret, and we also one-hot encoded the beer styles to get an accurate prediction of the beers who were missing their ABV data. Once we populated this, our dataset was ready to work with

**Methodology Selection**

We used predictive analytics to fill our missing ABV data with caret. We felt that this was better than using the standard average ABV of each beer style, because using an average just reinforces the average even further, for each beer style. Considering user review ratings into our prediction, this helped prevent this issue.

Once the data was cleaned and prepped, we use descriptive analytics to filter and manipulate the data to show the end-user a recommended beer list. We used clustering to group different reviewers into sub-groups, and based on the app-users inputted beer list, they were placed into one of the existing clusters. This cluster was considered their “comfort zone”, and a list of beers were recommend in this group. We also took beers from the same cluster of reviewers, but of a different beer style, and recommended those as an “out of the comfort zone” list.

**Model Building**

**Functionality**

**GUI Design and Functionality**

**Conclusions**

**References (if any)**

 If you use other people’s codes in your DSS, you should give them credit in your short

write-up, as well as references at the end of your video.