Predicting Beer Ratings From User Reviews

Radu Manea
rmanea@ucsd.edu

Tyler Houchin
thouchin@ucsd.edu

Nima Yazdani
niyazdan@ucsd.edu

Abstract: Recommender systems offer a solution to the information overload problem by utilizing rating prediction approaches to generate personalized recommendations. These systems predict the ratings that a user will give to particular items and create lists of items sorted according to the user's preferences. As internet users become increasingly diverse in their personalization drifts and behaviors, these systems are increasingly effective in helping to filter information. In this study we use beer ratings data to feature engineer a model to predict ratings users give to a beer. The final iteration of our model is a logistic regression algorithm that utilizes a bag-of-words model in conjunction with features surrounding the qualities of each beer.

1 DATASET AND ANALYSIS

This dataset[1] contains ~3 million beer reviews curated from 2000 to 2011. Each entry in the dataset is partitioned into information about the beer review. The beer information contains details such as the name, ABV, and style. The review information contains a user's profile name, review time, as well as ratings for the appearance, aroma, palate, taste and an overall rating. In order to efficiently analyze this data and build a predictor with the computation power available, we randomly subsetted the dataset into 200,000 records from 2000 to 2011. There are a total of 9,510 users and 8,202 unique beers in the dataset.

Beer Properties

Beer Analysis

First, we wanted to analyze which beers were most popular in the dataset, both through the number of reviews the beer received, as well as the average rating it received. The most reviewed beer received 3056 reviews, while beers received on average 24 reviews, with a standard deviation of 101. 50% of the beers in our dataset were only reviewed 4 or fewer times. Beers received an average rating of 12.63, with a standard deviation of 2.30.

ABV Analysis

In exploring the distribution of beers in the data set, we discovered the following interesting takeaways. The average ABV across all beers was 4.54%, with a standard deviation of 3.04. The minimum ABV was 0.0% (Alcohol-Free) and the maximum was an ABV of 25.4%.

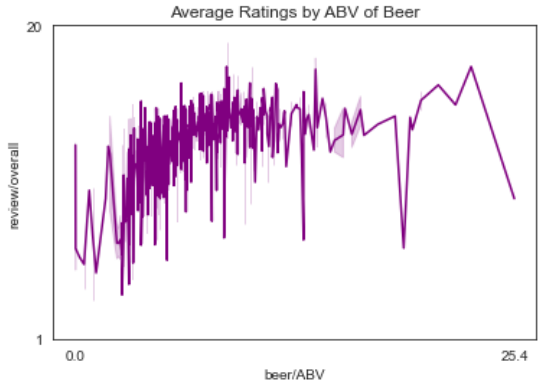

Figure 1.1: average ratings observed by ABV levels of beers

In order to understand how ABV plays a role in rating the beer, we analyzed the average rating by ABV of a beer. As seen in Figure 1.1, strongest positive correlation appears to be between the rating and beers with ABVs of roughly 4-10%, with the correlation overall being 0.36.

Brewery Analysis

The average rating across all breweries was 12.36, with a standard deviation of 1.94. The minimum average rating for a brewery was 2.29, and the highest average rating was 19.5. The average amount of reviews a brewery received was 349, however the most reviewed brewery consisted of 21,613 reviews.

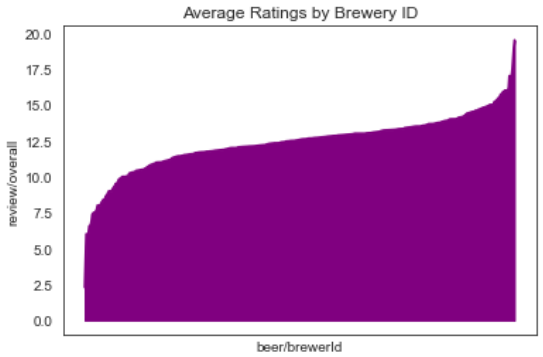

Figure 1.2: sorted distribution of brewery average ratings

We also looked at the top 10 breweries in the dataset, with their average ratings shown in the figure below (1.3). The top 10 breweries received 88,393 reviews of our dataset. This may be of concern as there are a total of 573 breweries in the dataset. Keeping in mind that some breweries have many different products, this becomes less concerning as the dataset just contains products from a limited number of breweries.

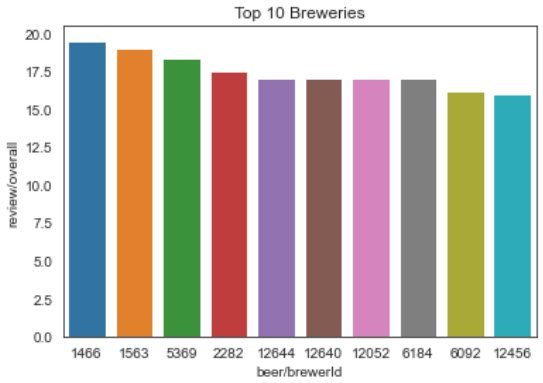

Figure 1.3: average ratings of the top 10 breweries (IDs)

User Properties

User Analysis

When looking at the users who make up the dataset, we first wanted to understand how many users have given more than one review in the dataset. There are 3,272 users in the dataset who only have given one review and 6,238 users with more than one review. That means roughly ~35% of users in the dataset have only one review given that is associated with their profile. The maximum reviews a user in the dataset produced was 1197, the average being 21 with a standard deviation of 56.

Review Properties

The global average overall rating across all beers was 13.51/20. The ratings had a standard deviation of 2.94. The maximum rating for a beer was 20/20, and the minimum rating was a 1/20.

Subratings Analysis

We wanted to understand if certain aspects of a beer's ratings, such as the appearance, aroma, palate and taste ratings, had different effects on the overall rating a user gave the respective beer.

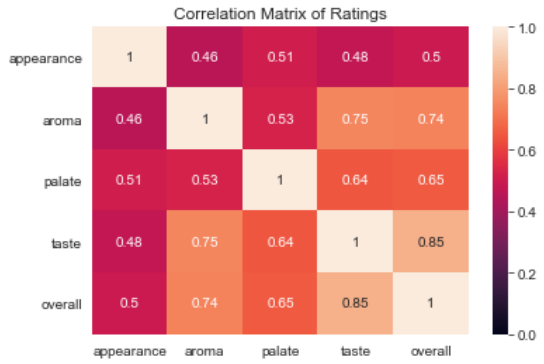

Figure 1.4: correlation heat map of rating categories

In Figure 1.4, we notice a few interesting aspects. Firstly, this matrix confirms that all subratings have a positive effect on the overall rating. The most positively correlated aspect is the taste of the beer, with aroma, palate, and appearance following in that order.

Text Analysis

To gain intuition as to how the rating text plays a role in the prediction of the overall rating score, we did some analysis on the review/text portion of the dataset. In Figure 1.4, we see the 25 most frequent words that appear within the text portion of a review. Words such as 'head,' 'aroma,' and 'hops' are expected as they are definitely correlated with beer; however, as we are trying to predict the rating, we are not sure how influential these words are on how high or low a user will rate them.

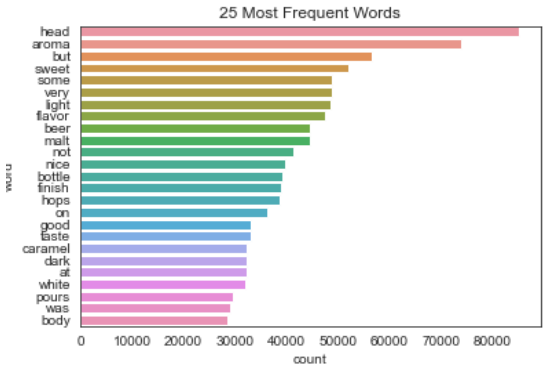

Figure 1.5: words sorted by frequency in review/text

With the goal of determining how influential a word is with respect to the rating, we build a bag-of-words feature vector out of the 2,000 most common words. We then use this to train a linear regression model and extract words whose presence have a large impact on how a user rates the respective beer. This metric of influence is given to us by the theta values where words that increased the overall rating have positive theta values and words that decreased the overall rating have negative theta values.

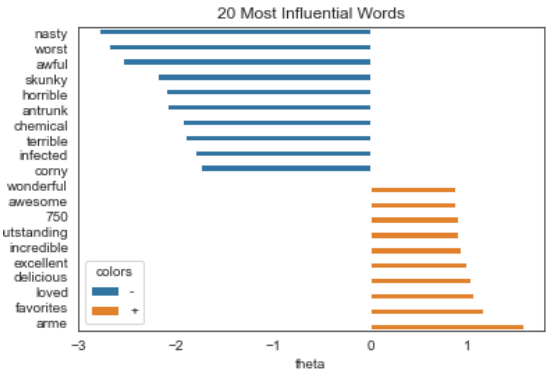

Figure 1.6: words sorted by influence on review/text

2 PREDICTIVE TASK

We aim to accurately predict a beer's overall rating (review/overall) from the user, beer, and review data as well as information gleaned from the entire dataset. To do this, we partitioned our dataset into a training and testing set with a 80/20 split ratio. In order to quantify how well our model predicts ratings, we will be using the Mean Squared Error (MSE) function which measures the error as the squared difference between predicted and actual values. A greater MSE represents poor model performance, in the context of our study. Thus, our goal is to minimize the MSE while preventing overfitting to the training set.

Baseline Model

The most trivial metric to use in order to predict the rating of a beer would be to simply predict the global average for ratings over all beers. Doing this over the training set obtained a MSE of 8.90. A slightly more engineered predictor will predict the rating based on the average of all ratings that a user has given as well as all ratings that a beer has received. To do this, we constructed dictionaries *ratingsPerUser* and *ratingsPerItem* which mapped ratings to their respective user or beer. Using these, we were easily able to compute the averages. This baseline obtained a MSE of 4.64 which is much better than the trivial predictor. For our model, we attained a larger feature space covering more aspects of the data at hand.

Data Processing and Feature Engineering

The feature space for our model includes review text data, beer attribute data, sub-rating data and rating count data by beer and user, as shown in the table below. The feature vector is composed of data from review/text, beer/beerId, review/profileName, beer/ABV, review/appearance, review/aroma, review/palate, and review/taste. The subrating data and ABV data was already included in the original dataset, however we had to engineer the other features. The training set's average ratings were computed by taking the average overall rating given by each user and for each beer in the training set reviews. We grouped the data by user and took the average over all of their respective overall ratings. We then grouped the data by item (beer), and took the average over all overall ratings for each beer. Our text data was created by scraping through the review text of each review in our dataset, implementing a bag-of-words model. Doing this involves taking the count of each word and recording the top 2000 most frequently included words, not

including syncategorematic words[1]. We remove punctuation from the review/text and create a list of words from the text. We then create a list of zeros corresponding to the number of words in the review. If a word in the review is present in the list of words, then the corresponding element of the list is incremented by one.

Feature Space
Text Data - 2000 Most Frequent ("Popular") Words In Reviews, Not Including Syncategorematic Words
Beer Attribute Data - ABV% of respective beer
Sub-Rating Data - Appearance Rating - Aroma Rating - Palate Rating - Taste Rating
Training Set Average Ratings - Average Rating Per User - Average Rating Per Item (Beer)

Table 2.1: Feature Space for Regression Model

Although sub-rating data and ABV data was provided, the records had to be cleaned so that the features could be used. The ABV column contained blanks, recorded as '-', for beers with ABV percentages of 0.00 (alcohol-free). We replaced these values with 0.0 and converted the values in the column to float. The different sub-rating columns have string values with ratings out of either 5 or 10. For example, in the Aroma Rating column, a value appears as a string as follows: '8/10'. With these columns, we removed the latter portion of the value and converted the rating to a float value, keeping only the rating given. In the context of the previous example, the new value would be 8.0. Our feature vector begins with our binary indicators for each of the 2000 most frequent words. Next we append the user averages and item averages from the data, followed by the cleaned data from the beer/ABV, review/appearance, review/aroma, review/palate, and review/taste columns.

3 MODEL

We chose to use a logistic regression model to predict the beer's rating using a feature vector. The decision to include features compiled from a bag-of-words model in addition to additional features gathered from each review was based on the MSEs computed when these were run exclusively. When we trained the model with a feature vector of just the 2000 most frequent words, we got a MSE of 5.38 on the test set. Further, when we trained the model using a feature vector with just the cleaned data from beer/ABV, review/appearance, review/aroma, review/palate, and review/taste columns, we obtained a MSE of 4.40 on the test set. Finally, a model trained using a feature vector that uses all of this information provided us a MSE of 3.32 on the test set and we therefore conclude that this was the most optimal feature vector for our application of predicting the overall rating. The intuition behind all of this follows quite simply from our exploratory data analysis. The most frequent words must have an impact on the overall rating as we previously noted that words such as 'nasty' and 'delicious' had a large impact on the overall rating during our initial analysis. Additionally, the inclusion of additional review data such as the user and beer's average rating have an influence on the individual rating as a beer's average rating tells us how that beer has been rated historically and similar logic follows for the user's average rating. The decision to include sub-ratings in our feature vector was simple as we found that all subratings have a positive correlation with the overall rating and therefore give us a good intuition as to whether the beer will be rated highly or not. We attempted to include the average ratings of the top 10 most similar beers using a Jaccard-based similarity function; however, our computational power was not able to process feature vectors in a time reasonable enough to make tweaks and determine if this was helping much or not.

[1] Syncategorematic words that were excluded: 'a' , 'and', 'the', 'with', 'of', 'is', 'to', 'in', 'this', 'I', 'it', 'that', 'of', 'can', 'get', 'me'

4 LITERATURE REVIEW

Relevant Research

The dataset that we used in this model was provided by Professor McAuley at the University of California, San Diego. It is a 388 mb file that consists of ~3 million reviews where 200,000 were sampled to split into testing and training data. The training data was used to build a logistic regression model to predict the ratings of the beer based on features such as: beer/name, beer/beerId, beer/style, and review/text. Though not many beer prediction models have been built in the past, prediction tasks have been conducted on a plethora of datasets ranging from CT scans, to hotel reviews.

Recommender systems are a way to generate personalized recommendations for users by predicting ratings they will give a particular item. These systems are effective but still have challenges such as accuracy, scalability, cold-start, and data sparsity. Currently the state-of-the-art technique for making rating predictions found in other publications [2] utilize Restricted Boltzmann Machines (RBMs). RBMs are a special type of Boltzmann Machines (BMs) that are used in rating prediction and collaborative filtering. RBMs are capable of inferring latent features and modeling user preferences and item ratings by forming relationships between the ratings and the items. RBMs are preferred over BMs as they have effective parametrization and scalability features, and are able to handle large datasets. RBMs may be used in hybrid models with visible layers connected to hidden layers for user and item correlations separately. They are also able to integrate heterogeneous data from different sources. Various techniques have been proposed to employ RBMs for collaborative filtering, such as context-boosted RBMs, explainable RBM-CF, item based RBM-CF, NC-RBM, and CRBM-IR. These techniques have improved accuracy in collaborative filtering, and CRBM-IR is currently the state-of-the-art technique. Though we have found a logistic regression based model that performs the best when using a feature vector consisting of a bag-of-words with additional review data, a more accurate rating predictor could be obtained using neural networks as found in other papers [3] and deep learning models which has been similarly concluded in other articles on recommendation systems.

5 MODEL EVALUATION

Results and Conclusions

In conclusion, after iterating through different feature vectors and determining their performance using MSE's, our final model produced an optimal result relative to the baseline model. Obtaining a mean squared error of 3.32 represents the ability to predict the rating given a user and beer within 16 percent of the true rating as all ratings are out of a total of 20. Representing the counts of common words, including each individual sub-rating as its own value, and keeping ABV as a float representation were optimal feature representations that contributed to the success of the model. Though this is the case, approaching this problem using a logistic regression model has some shortcomings compared to using RBMs as they suffer from different challenges such as accuracy, scalability, cold-start, and data sparsity.

6 References

[1] Project, UCSD CSE Research. "Recommender Systems and Personalization Datasets." Recommender Systems Datasets, https://cseweb.ucsd.edu/~jmcauley/datasets.html#multi_aspect.

[2] Khan, Zahid Younas, et al. "Deep Learning Techniques for Rating Prediction: A Survey of the State-of-the-Art." *Artificial Intelligence Review*, vol. 54, no. 1, 2020

[3] Julian McAuley, Jure Leskovec, Dan Jurafsky International Conference on Data Mining (ICDM), 2012