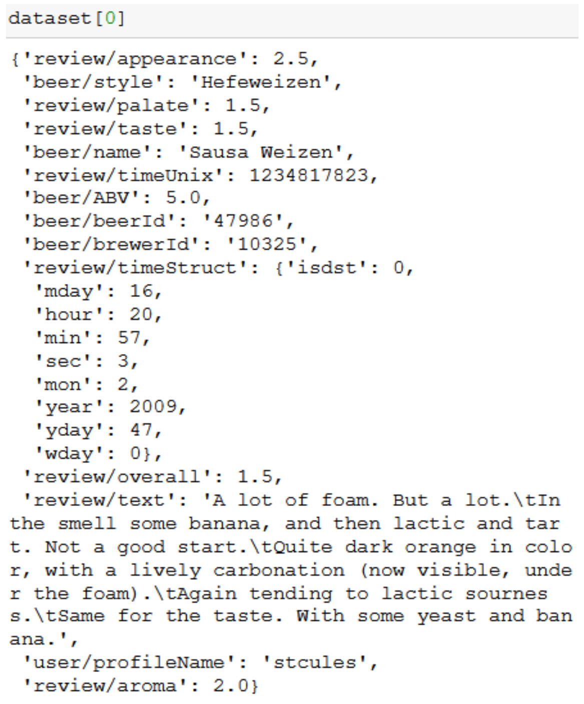
Beer Rating Prediction

1. Introduction and Exploratory Data Analysis

As avid fans of alcoholic beverages, especially beer, the dataset we chose was beer.json from Professor Mcauley's site http://jmcauley.ucsd.edu/data/beer/. We chose this dataset over the ratebeer and beeradvocate datasets because after quickly looking at all three, we found the beer.json dataset to be in a format most familiar and easy for us to work with. The dataset consists of roughly 1.5 million samples. After loading and prodding the dataset, we decided to cut down our working dataset to 500,000 samples as this was the highest value we could process efficiently. Each of these samples includes ratings and information regarding the following: beer appearance, beer style, palate rating, taste rating, beer name, date and time of review, alcohol by volume, beer ID, brewery ID, overall rating, text of review, reviewer's profile name, and aroma rating. Below is an example of a sample in our dataset.

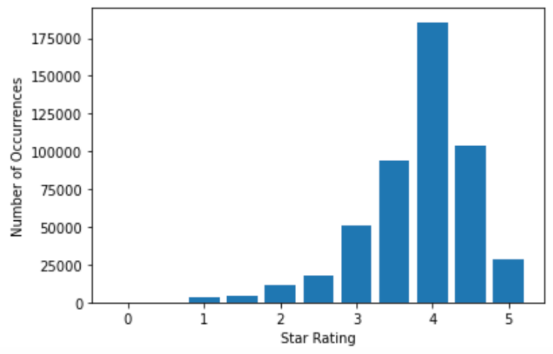


```
dataset[0]
```

```
{'review/appearance': 2.5,
 'beer/style': 'Hefeweizen',
 'review/palate': 1.5,
 'review/taste': 1.5,
 'beer/name': 'Sausa Weizen',
 'review/timeUnix': 1234817823,
 'beer/ABV': 5.0,
 'beer/beerId': '47986',
 'beer/brewerId': '10325',
 'review/timeStruct': {'isdst': 0,
  'mday': 16,
  'hour': 20,
  'min': 57,
  'sec': 3,
  'mon': 2,
  'year': 2009,
  'yday': 47,
  'wday': 0},
 'review/overall': 1.5,
 'review/text': 'A lot of foam. But a lot.\tIn
the smell some banana, and then lactic and tar
t. Not a good start.\tQuite dark orange in colo
r, with a lively carbonation (now visible, unde
r the foam).\tAgain tending to lactic sournes
s.\tSame for the taste. With some yeast and ban
ana.',
 'user/profileName': 'stcules',
 'review/aroma': 2.0}
```

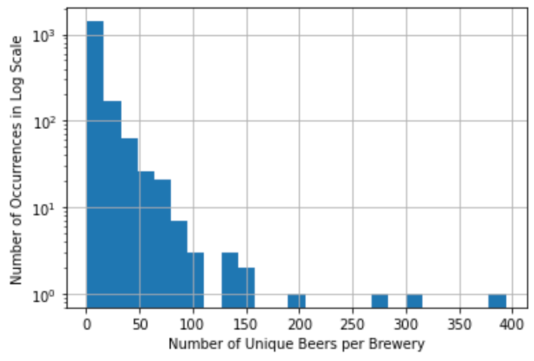
Looking at the information provided in this sample, we first identified a few features that could potentially be important in creating a predictive task and subsequent model. These features were the review text, overall rating, brewery ID, and reviewer's profile name. After looking at further data samples, we recognized that some features, such as reviewer's gender and reviewer's age were only included in some samples. Additionally, we noticed that some reviewer's profile names seemed unique, such as "msubulldog25" while others were very non-specific such as just "Jason". We took note that there could potentially be multiple users with the same profile name and to keep this in mind when performing further analysis. After recognizing that some features could have issues, we took a closer look at the features that we had identified as being potentially important.

To make sure that the review text feature could be used in a predictive task, we calculated the average number of words per review. With each review having an average of about 145 words, we decided the reviews' text were substantial enough to be used in the predictive task. Additionally, we took a look at the frequency of each reviews' overall rating. The distribution of these rating is as follows:



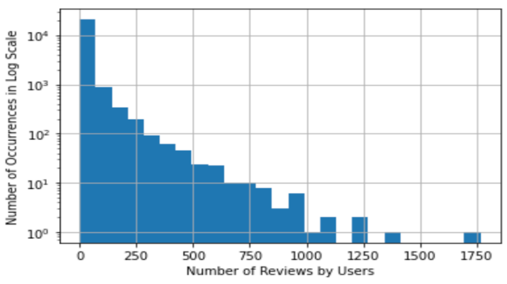
This indicated that ratings were not too skewed and could be used in the predictive task.

Next, to test the brewery ID feature's usefulness, we calculated several metrics. First, we calculated the average number of unique beers each brewery received a review for. We found that there were a total of 1710 breweries, each with an average of 11.3 unique beers and a median of 5 unique beers that were reviewed. This meant that only around half of the breweries had more than 5 beers that were reviewed. While this value was not very high, we also wanted to see how many reviews were done on breweries with more than 5 beers reviewed. This came out to 474,937 reviews. This showed us that brewery ID could definitely be used in our predictive task because the vast majority of reviews in our dataset were on breweries with more than 5 beers that we could get information from. Below is the distribution of the number of unique beers per brewery, as well as the 5 brewery ID's with the most unique beers reviewed and the number of unique beers they had.



```
16866      155
17981      201
5318       283
1177       312
1548       394
```

Finally, to determine the usefulness of the reviewer's profile name, we calculated several similar metrics. First, we calculated the number of unique profile names (users), which was 22375 users. We also found an average of 22 reviews per user and a median of 3 reviews per user. This showed us that there were a small percentage of users that submitted a very large number of reviews while most users only submitted a couple. Identifying this as a potential issue, we wanted to see how many users would have a substantial enough review history to potentially glean information from. We decided that 5 reviews submitted was a fair distinction to determine this. We found that there were only 8260 users who submitted more than 5 reviews, compared to the 14,115 users with 5 or less reviews. Although there are not many users with more than 5 reviews, we continued to see how many reviews were done by such users, which came out to be 473,031 reviews. This meant that the extreme majority of reviews came from users with a substantial review history. With this information, we decided that we could include profile name as a part of the predictive task. Below is a breakdown of the number of reviews by users, as well as the names of the 5 users with the most reviews and the amount of reviews done by those users.



```
                        ...
ChainGangGuy           1111
womencantsail          1202
BuckeyeNation          1224
mikesgroove            1352
northyorksammy         1765
```

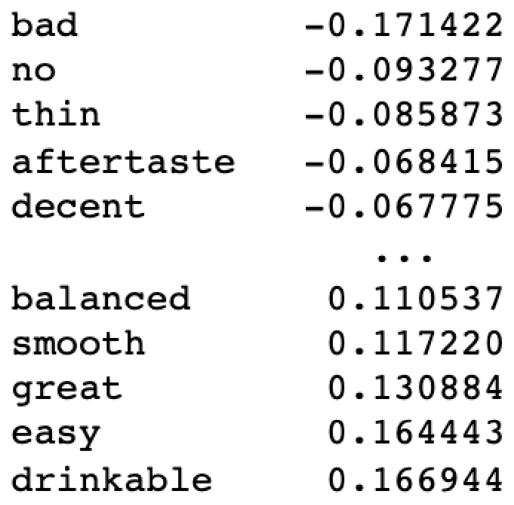
2. Predictive Task

Based on what we learned about the dataset and the different features within each data sample, we decided that our predictive task would be predicting the overall rating of a beer. We believe we can make this prediction using mostly review text, but by also adding in user history and brewery history. We will evaluate our prediction using mean absolute error. We chose mean absolute error as opposed to mean squared error because if we evaluate using mean squared error, we would receive results which seem to unrealistically reflect accuracy. This is because a large number of ratings are in the 3-5 range. As such, most predictions within this range will likely have errors that are less than 1. If the error is less than 1, then the squared error will be even less, making the prediction accuracy seem overly accurate.

The baseline model we will use for comparison will be a simple bag of words model. We felt that this would serve as a good baseline because review text was one of the most robust and descriptive features within the dataset and logically it makes sense that the review a user left a beer would have correlation with the user's rating of the beer. We will assess the validity of our model's prediction by comparing the mean absolute error of the baseline model and our model. Initially, our model will predict rating through sentiment analysis of the review text. However, we hope to improve this model by looking at how sentiment analysis of the review text can be combined with user and brewery review history. As such, the features used to predict overall rating will be those that we did more in-depth analysis on during our EDA: review/text, user/profileName, and brewerID. Since we already did a deeper analysis of the features and identified how they could be used, there were no additional steps for processing the data of these features that we needed to do.

3. Describe Model

We decided on using the TF-IDF model because we felt it would best capture the relationship between the text in the review and the overall rating. We felt that because TF-IDF would give higher weight to the rarer words and ignore the words that are overly common, we would get better results than the bag of words approach. The bag of words model may place larger value on words that did appear more times but also on common words that do not represent whether the user is leaving a positive or negative review. However, TF-IDF would capture this user sentiment. We ran into no issues setting up the baseline model or our initial TF-IDF model. To verify that the bag of words model was working as expected we took a look at the 5 most positive and negative words, as shown below.



```
bad          -0.171422
no           -0.093277
thin         -0.085873
aftertaste   -0.068415
decent       -0.067775
                ...
balanced      0.110537
smooth        0.117220
great         0.130884
easy          0.164443
drinkable     0.166944
```

We felt that these words demonstrated that our bag of words model was working properly as we have words like bad as being very negative while smooth and great were positive. We then continued on to calculate the mean absolute error. The mean absolute error of the baseline bag of words model using ridge regression was .4714. We played around with the lambda for optimization, which did show improved results. We were able to decrease our mean absolute error to about .4702 by increasing the lambda to 21,000. However, we felt that this was unnecessarily large for such marginal a change and kept our lamda and the default value. One problem we had with the bag of words was a memory and time constriction on the size of the feature vector. Because we were training our data with 400,000 samples of review text, we were limited on the size of our bag of words dictionary and could not go past a size of 200 words, even when using a sparse matrix.

Our TF-IDF model, similar to our bag of words model, was very slow to run, yet resulted in a much better mean absolute error of .3993. We tried both user unigrams and a combination of unigrams and bigrams and found no significant difference between the 2 versions. We thought we had found strong improvement and wanted to improve this model even more by incorporating user review history and brewery review history. To use our user review history, we looked at whether the user typically gives high reviews or low reviews. We did the same with breweries and looked at whether the brewery typically receive high or low reviews. The user history was calculated as the average difference between the users rating of a unique beer and the average rating of that beer across all users. So if a user gives a lower rating than average for most beers, then the rating we are trying to predict will be lower than if it were rated by an average user. The brewery history was modelled in a similar way, where we calculated the average difference between that brewery's beer ratings and the average rating for any beer, and we applied it in the same way as the user's history. By adding both the brewery history and user history to the prediction that the TF-IDF model made, we were able to lower the mean absolute error to .3972. Lastly we reduced any predictions above 5 stars, to 5 stars. This reduced the mean absolute error to .3965. As we worked through our model/models, we did consider using a few other models based on other features. We were thinking about potentially mixing the bag of words model with user and brewery history, but this did not beat our bag of words mean absolute error, let alone our TF-IDF model performance.

4. Literature

We got this beer.json.gz from Professor Mcauley's site http://jmcauley.ucsd.edu/data/beer/. We do not know exactly what this dataset is but it is very similar to the beeradvocate and ratebeer datasets. Both of these datasets were similar enough to ours that we decided to focus our search for literature and other studies on these datasets. We first looked at "From Amateurs to Connoisseurs: Modeling the Evolution of User Expertise through Online Reviews" by Julian McAuley and Jure Leskovec. This study utilized the BeerAdvocate and RateBeer datasets, as well as other non-beer datasets, to not only recommend beer (and other products) that a given user would like at that moment, but to also model how as a user gains experiencing in consuming a particular good, their tastes change. This model was tied to largely temporal aspects such as the age of the user and product. While this study is very interesting, we felt that it was utilizing the datasets with a goal that was fairly different from ours. While both did work with beer and their ratings, we had a final goal of rating prediction accuracy based on what the user says while this study was analyzing other factors about the user for it's model.

Next, we looked at "Learning Attitudes and Attributes from Multi-Aspect Reviews" by Julian McAuley, Jure Leskovec, and Dan Jurafsky. This study also utilized both the BeerAdvocate and RateBeer datasets, as well as other datasets, to learn about which words or phrases are discussing an aspect of the beer or product rather than just the connection to overall rating. For example, while the word "superb" may correlate with a very positive overall rating, the model hopes to learn that the phrase "enchanting aroma" is not only is discussing the "smell" aspect but is also correlated with a positive rating for this aspect. We found that this is much more similar to our study, but also much more advanced. This study seeks to not just understand the correlation between words and phrases and the overall rating but also with the various other aspects users can rate. However, we are purely focusing on not just how the text of the review correlates to the overall rating, but also how user and brewer review history affects this.

As the bulk of our model is based on sentiment analysis, we looked for state of the art methods currently being used for sentiment analysis. The most popular method for sentiment analysis that we found was BERT. BERT stands for Bidirectional Encoder Representations from Transformers and is used to help recognize the meaning behind the language being used in a body of text and using the text that appears around it for context. It is a deep learning model that is highly successful because it is able to not just read left-to-right and right-to-left but can actually do both at the same time. BERT was developed by Google and open-sourced in 2018 and subsequently achieved many groundbreaking results. It was able to outperform other language models because it was able to most effectively deal with ambiguity. BERT is currently being used in many ways that involve sentiment analysis such as text classification and search query optimization. While, BERT may not be specifically tied to beer and rating prediction, it seems to currently be at the forefront of sentiment analysis.

The conclusions from existing work, such as the two studies we discussed earlier, seem to be in line with our findings. They conclude, as do we, that the text in a review and specific word choice can be extrapolated to meaningfully and accurately predict ratings.

5. Results and Conclusions

In the end, our TF-IDF enhanced with brewery and user history had the best mean absolute error of .3965. This was better than bag of words, bag of words enhanced with user and brewery history, and regular tf-idf. Our mean absolute error of .3965 means that on average our model made predictions that were off by 0.3965 in terms of star ratings. So if the actual rating of the beer was a 4, on average our model would be off by 0.3965 and could guess 4.3965 or 3.6035. Tf-idf with user and brewery history worked the best because Tf-idf works well on capturing the sentiment of a review and the user and brewery history boost the sentiment analysis portion of the review towards the user and breweries typical tendencies. When we regressed on the tf-idf model we found that not using a lambda actually worked the best on the test set, which is interesting because using lambda lowered the mean absolute error on the bag of words model. While we did see improvement by adding user and brewery review history, we were disappointed at how slight the improvement actually ended up being. We believe that the improvement was slight because in our implementation, the user and brewery review history were used to modify the prediction that the TF-IDF model already made. We did try to create a feature vector the incorporated both the user and brewery review history along with tf-idf but were unsuccessful in doing do. We needed more understanding of the inner workings of the TF-IDF model's prediction system to successfully create. Despite this, we were satisfied that we continued to build upon our model and its complexity and showed improved results with each model iteration.