Beer Rating Prediction

Abstract

For this project, we are trying to predict the overall rating through feature extraction from the review text (bag-of-words) and through the subjective numeric ratings provided by different users. We used these features to perform linear regression on the dataset in order to achieve a highly accurate result.

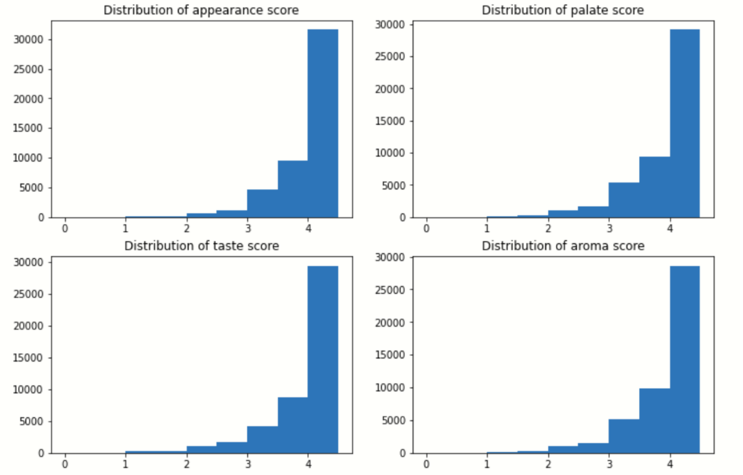
Keywords

bag-of-words, linear regression, feature extraction, sentiment analysis, prediction.

I. Introduction

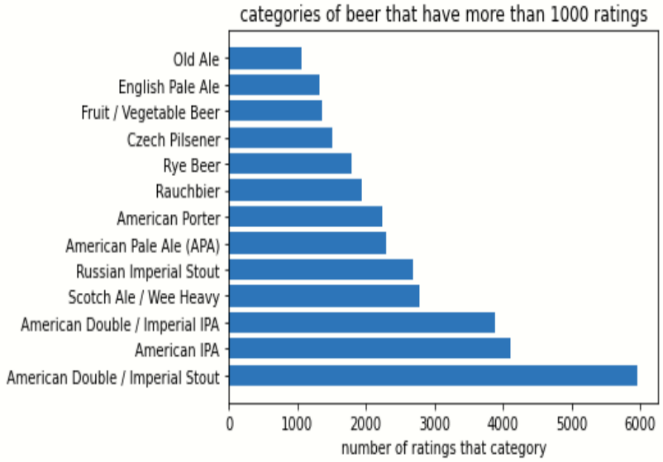
Many people have drunk beers at least once in their lives. Beers are one of the most popular alcoholic drinks and we often drink beers with friends, families, classmates, etc. Since there is a high demand for beers, there are many types of beers in this world, and we are often wondering which beer we should buy. Beers made by different factories or brands may have different appearance, palate, taste, and aroma. Sometimes it is hard to decide which one of these attributes is more important. However, if we have a list of ratings and comments made by the customers, we could gain some understanding of which beer to buy. Thus, in this paper, we are going to explore how people rate beers and try to predict the overall rating of beers given by customers.

We are using the Beer rating dataset from the CSE158 https://cseweb.ucsd.edu/classes/fa22/cse258-a/. The data is in json format and has 50000 dictionaries. There are 14 keys in total: appearance, style, palate, taste, name, review time, ABV, beer ID, brewer ID, time structure, overall rating, profile name, review text, and aroma. Style, name, and review text are composed of strings, and others contain integers.



For the data cleaning purpose, we first remove any dictionary that contains empty value, since empty value will be useless for our analysis. The length of the data goes down from 50000 to 49971, which means that we remove 29 dictionaries that contain at least 1 empty value. We also plot all the variables and clean out any outliers. All the beer ratings should be between 0 and 5.

As we could see from the plots on the left, the distributions are similar among the appearance score, palate score, taste score, and aroma score. One interesting finding is that there are almost zero people who give the rate of 0 and most people give rates between 4 to 5.



We also see the top popular categories among all the beer. We plot all the categories that have more than 1000 ratings. The American Double and American IPA are the most popular categories and may indicate a higher overall rating.

Finally, we plot the distribution of the overall rating, which is the number that we want to predict. From the distribution graph of overall score, we could see that the pattern that occurs in the distributions of appearance score, palate score, taste score, aroma score also exists in the distribution of overall score. It indicates that we may be able to use the other scores to predict the overall score.

II. Predictive task

The predictive task we identified is to predict the overall score of the review through the features we extracted from the text and numerical features provided in the review. For the model evaluation, we picked mean squared error (MSE) since it is one of the most universal and recognized evaluators out there.

Our baseline model is rather a simple one; first remove all the punctuations and capitalizations in all 50,000 reviews, then construct a bag-of-words model. With the bag-of-words model, we sorted out the most frequent word from all unique review words in the dataset. The last step is to fit the most frequent word onto the dataset, forming a X for the function:

$$y = mx + b$$

Since this is an extremely rudimentary model with an MSE of 0.507 (rounded to the thousandth), we would later use this as our starting ground and compare our improved model against this baseline MSE. For improved models, we decided to go with bag-of-words with the combination of linear regression. Such a combination was decided based on the rich features contained both within the review text and the numeric features from each user. We will try two different methods for extracting features from the review text, one will continue to utilize the bag-of-words model from baseline, and the other will improve the model through removal of non-essential words.

III.Model

Version I. Bag of Words

For the version I model, we choose to implement a bag of words model on the review text for the overall rating prediction. We choose this model mainly because it is a key concept in the lecture and in assignment 1. Different from what is being implemented in assignment 1 (a multi-class classification model), here we have a regression model.

We optimized the model by experimenting with various techniques for choosing the vocabulary that builds the feature vector for the regression model, and the alpha penalty term (regularization) for the ridge regression model specifically.

On various techniques of choosing the vocabulary, we implemented a model that contained vocabulary without punctuation and another model that contained vocabulary without both punctuation and stopwords.

Another choice that we made is a different vocabulary size (i.e. choosing the top n words to be included in the feature vector, where n will be denoted as the vocabulary size) (we tested 1000 to 5000 with increments of 1000), as the larger vocabulary size can possibly contribute to more fittable features which could lead to increased model accuracy. However, the drawback to this is that the algorithm will take much more time and space to run and the increase in model accuracy does not follow a linear relationship but rather resembles a logarithmic scale.

On the two techniques of building the vocabulary above, we further divide our model into testing the effect of regularization. More specifically, we implemented models without regularization and models with regularization on a logarithmic scale from 1 to 10,000.

Combining the vocabulary method and the regularization method above, after finding the optimal alpha for the ridge regression (through the hyper-parameter search described above), we use that specific alpha to train the model of different vocabulary sizes (i.e. 1000 to 5000 with increments of 1000). Further discussion will be directed to the result section.

Version II. LInear Regression

A. Model Description

This model uses linear regression along with its variants (with or without regularization term) using features extracted from each user. We choose this model to be the second one because linear regression with relevant features is one of the most popular methods, and usually the go-to method to try when given a prediction task.

B. Feature Engineering

Each user provides 14 attributes, including smell, taste, review text, beer name, etc. Considering the nature of overall review would be closely related to features such as taste rating or text length, we assigned 6 features: appearance score, palate, taste, aroma, number of exclamation review in text, and length of review. We assigned 80 percent to be training data, and the rest to be testing data, without random selection.

C. Without Regularization

We will first try the model without regularization to see how it performs. If the model behaves well, we would like to see which covariate is the most relevant to the overall score we are predicting.

D. With Ridge Regression

Next, we would like to see if it is possible to improve the model by including a regularization term. Given the theoretical solver of Ridge regression,

$$\hat{\beta}_R = (\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y}$$

Since the value in the design matrix is relatively small (only spans from 0 to 5), the validation value for regularization does not need to be large, hence we are trying values from 0.01 to 0.09, total 9 values.

In addition, Lasso regression would also be tested on the test dataset. However, since our design matrix would not be 0 almost everywhere, which is the situation Lasso regression is designed for, it is unlikely that including L1 regularization would improve the model noticeably.

IV. Literature

The dataset we used comes from the website BeerAdvocate, which is a site that aggregates the beer reviews from consumers since 1996. The reviews in the file have been anonymized in order to protect privacy. The reviews covered both the objective aspect, such as the ABV and the style of the beer, and the subjective aspects of the beer, including taste, aroma, appearance, aroma, palate, overall rating, and most importantly, the review text itself. BeerAdvocate's dataset is excellent in terms of the sheer amount of reviews (1,386,259 reviews), but due to the limited computing power, we are only going to use a subset of that data that is provided in professor Julian's website (50,000 reviews). The number of parameters available for predictions is also an important factor when we decide on this dataset since the aspects of the beer mentioned above are all numeric values that are relatively easy to process than categorical values. All the factors combined, BeerAdvocate dataset became one of the most popular datasets for prediction and recommendation, and different reports and papers using various methods set numerous precedents for us to learn from. Similar datasets are also available, such as BeerReview, however, since we already had previous experience with BeerAdvocate, we decided to continue with it. Benjamin et al. [3] also used the original BeerAdvocate in their model and a similar bag-of-words model. However, it is different in the way words are handled. They tested separating the words into different categories and found out adjectives are the most useful when it comes to prediction, meanwhile we just used a regular bag-of-words model.

V. Result

Version I. Bag of Words

1. Removed Punctuation

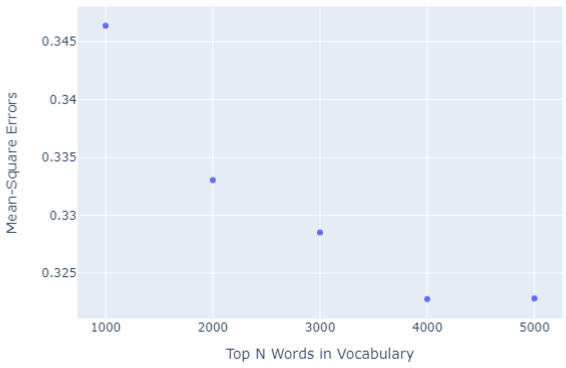


Fig. A

No Regularization

vocab. size	MSE
1000	0.34637
2000	0.33304
3000	0.32851
4000	0.32276
5000	0.32281

For methods with no regularization, we can see an increase in vocab. size will generally increase the accuracy of the model. However, there is a bottleneck/limitation of vocab. size as presented by the fact that the MSE started to increase as we increased our vocab. size from 4000 to 5000.

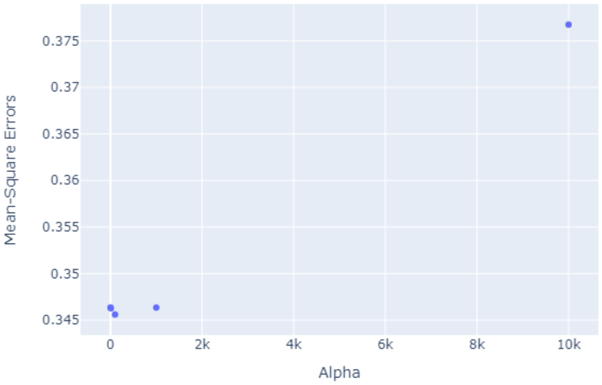


Fig. B

Regularization (vocab. size = 1000)

alpha	MSE
1	0.34636
10	0.34627
100	0.34560
1000	0.34635
10000	0.37673

By fixing the vocab. size to 1000, we tested the performance of the ridge regression model on different alpha. As presented by both the graph and the table, the optimal alpha landed to be 100 within all the alphas we tested. When we compared the model with the optimal alpha and the vocab. size of 1000 to the model without regularization, we observed that the former model outperformed the latter by a small amount (0.00777 less in MSE), which is not a significant improvement (significant improvement: MSE difference >= 0.01)

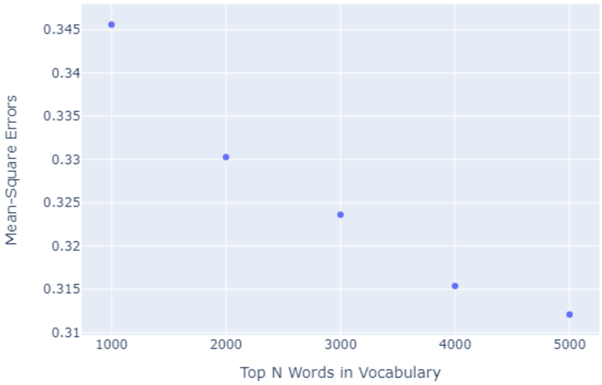


Fig. C

Best Alpha (100) + Different Vocab. size

vocab. size	MSE
1000	0.34560
2000	0.33029
3000	0.32363
4000	0.31538
5000	0.31208

After finding the “optimal” alpha term, namely, 100, we explored the ridge regression with different vocab. size shown in the table. As expected, the model with vocab. The size of 5000 has the lowest MSE. When we compared the model without regularization to this model with the same vocab. size (5000), we observed the latter have a significant improvement in performance (i.e. 0.01073 less in MSE).

2. Removed Punctuation AND stopwords

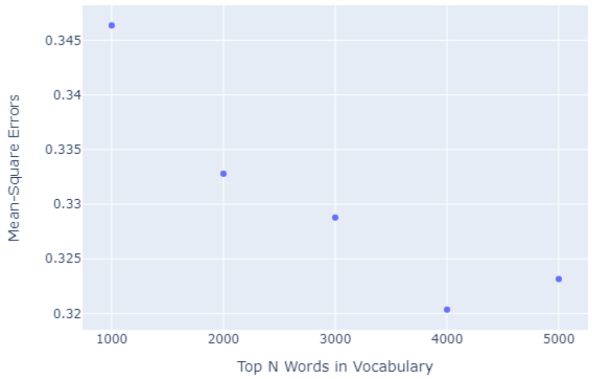


Fig. D
No Regularization

vocab. size	MSE
1000	0.34636
2000	0.33279
3000	0.32878
4000	0.32035
5000	0.32315

For the second variation of the regression model, we employed the technique of filtering both punctuation and stopwords in building the vocabulary. For the model performance with no regularization, we observed a similar trend as to the similar model (no regularization) of the first variation. And if we further observe the model here with the optimal vocab. size (4000) and compared it with the first variation with the same vocab. size, it is clear that there is no significant improvement (according to our previous definition of significant improvement).

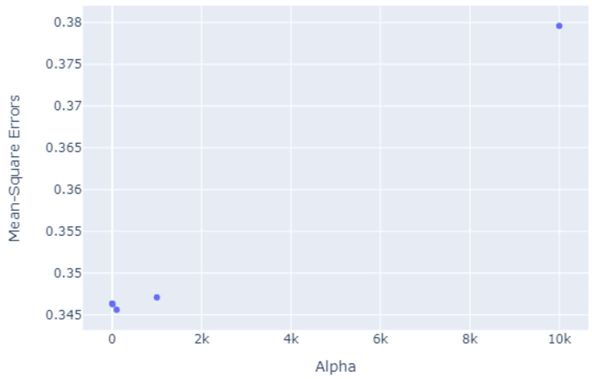


Fig. E
Regularization (vocab. size =1000)

alpha	MSE
1	0.34635
10	0.34626
100	0.34561
1000	0.34709
100000	0.37961

Similar to the procedure to produce Fig. B, this model has a slight variation in the way of choosing the vocabulary. Furthermore, this slight variety did not affect the performance in either a positive or negative manner when we compared the MSE of the optimal alpha in this model (100) to the MSE of the same alpha in Fig.B. So, we can infer that the two varieties of choosing vocabulary will produce a model of similar performance, and sometimes the first variety (filtering only punctuation) will outperform the other. Additionally, when we compare the MSE of the optimal alpha in this model to the MSE of the model with the same vocab. size (1000) in Fig. D, there is no significant difference. The same statement also applies when we compare the MSE of the same parameter (optimal alpha, fixed vocab. size) between Fig A. and Fig. B. Hence, we have arrived at the conclusion that regularization generally does not contribute to the model performance of our task.

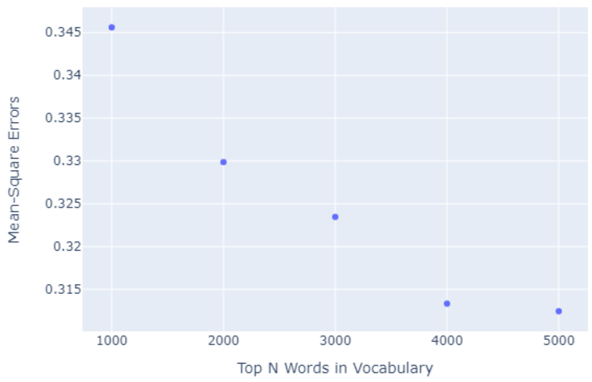


Fig. F
Best Alpha (100) + Different Vocab. size

vocab. size	MSE
1000	0.34561
2000	0.32987
3000	0.32346
4000	0.31333
5000	0.31244

As we observed from the MSE in the table, the model in Fig. F performs similarly to the model in Fig.C. The models used in the two figures are the same except for the only difference in the technique employed in choosing the vocabulary. So, adding to our discussion in Fig. E, this result further reinforced the claim that the two varieties of choosing vocabulary do not affect the model performance significantly.

Overall, for the Bag of Words model for our beer rating prediction task, we claim that the vocab. size will be one of the most important hyper-parameters in improving the model performance, whereas the two ways of choosing the vocabulary (filtering punctuation and/or stopwords) and regularization do not.

Version II. Linear Regression

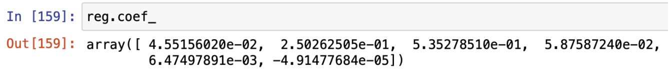
1. Linear Regression

Using MSE as the metric, the regression on testing data gives

0.17141494601851723

which is a relatively good performance.

Looking at the coefficients of our model,

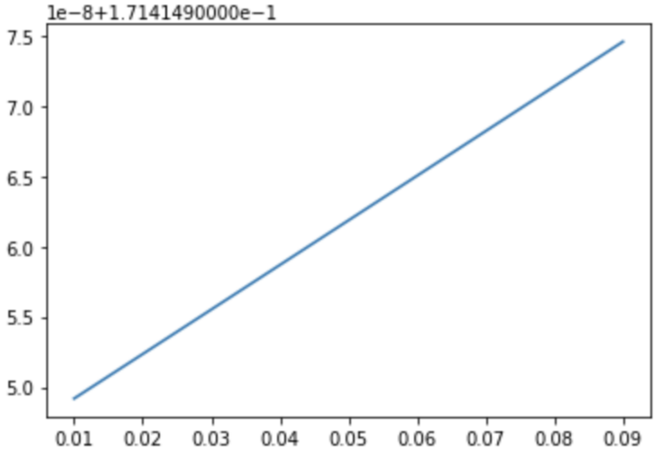

```
In [159]: reg.coef_
Out[159]: array([ 4.55156020e-02,  2.50262505e-01,  5.35278510e-01,  5.87587240e-02,
                  6.47497891e-03, -4.91477684e-05])
```

The largest coefficients are that of taste and palate, which is around 0.5 and 0.25, respectively. From this, we know that overall score is largely correlated with the rating of beer's taste and palate, which refers to the mouthfeel. To confirm this observation, we take the difference between overall score versus taste score for all the data, and found that the variance is around 0.21, which indicates a small difference. This small variance explains the good performance of a simply linear model.

2. Ridge Regression

The result MSE is as follows,

Fig. A



(The y value corresponds to 7.5 means

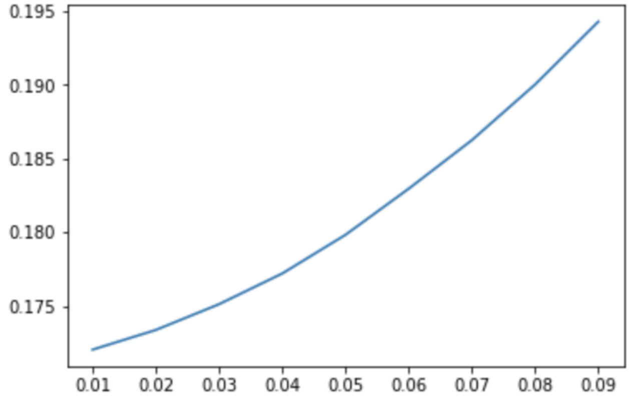
$$(7.5 * 1e^{-8}) + 1.7141490000e^{-1} \approx 0.171414975$$
)

Notice that the MSE increases almost linearly with the increase of regularization term. Also, looking at the rank of the design matrix, it gives 6, which corresponds to the number of columns in the design matrix. This suggests that the design matrix is of full rank (not the usual sense of full rank, but having independent column vectors). Since the purpose of Ridge Regression is to nudge each column vector a little so that they are not linearly dependent on each other, Ridge regression should not improve our model at all for the reason that column vectors are independent.

3. Lasso Regression

Finally, we want to try the Lasso Regression, but theoretically it should not improve the model because our matrix is relatively dense. We will use the same regularization values.

Fig. B



Looking at the graph, MSE increases almost quadratically with the regularization values. Hence, Lasso regression should not be included as expected.

VI. Conclusion

After trying the bag of words models and different regression models, we successfully created a linear regression model with Mean Squared Error 0.1714. The input features of the model are: appearance score, palate, taste, aroma, number of exclamation review in text, and length of review. Using our model, we could predict the overall rating of a beer with a high accuracy. Our model also indicates that there is a strong correlation between the appearance, palate, taste, aroma, and the overall rating of a beer. Besides, by using feature engineering on the people’s comments, our model is very accurate for predicting what rate customers will give to a specific beer.

With a strong model we built here, it is possible to predict other components of the review; predicting the style of the beer (multi-class classification) or the ABV (investigate the relationship between ratings and style/ABV of the beer) could be areas of interest.

References

[1] **Learning attitudes and attributes from multi-aspect reviews**
Julian McAuley, Jure Leskovec, Dan Jurafsky
International Conference on Data Mining (ICDM), 2012

[2] **From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews**
Julian McAuley, Jure Leskovec
WWW, 2013

[3] B. Braun, R. Timpe, “**Text based rating predictions from beer and wine reviews**,” 2015.