3 PRIDICTIVE TASKS

Naturally, the rating customers give to merchandise would be a reasonable reflection of their evaluation upon that. And since we already have a clear overview about the properties of the Amazon fine food dataset, we can now conduct a predictive task to predict the rating based on other information for every review record. These predictions can be used to help recommend potential attracting products to the customer's taste.

3.1 Data Pre-processing

As introduced in section 2.1, the Amazon fine food database has 568462 reviews as total, and after we eliminate those without review text we end up with 568454 reviews left. And we find that many of these review texts contain HTML formatting tags such as $< \mathrm{br}/>$ and they will dominate in our word frequency statistics but they contain no information, so we have to remove these HTML formatting tags by BeautifulSoup Python library firstly.

Since the uppercase and lowercase of the same word usually don't contain much different information, we convert the review text to all-lowercase form for each piece of review. And also we remove the punctuations due to that they contain little information. Finally, we shuffle the dataset, split the dataset and use the 1-100000 reviews to be our training set, 100001-200000 reviews to be validation set and 200001-300000 reviews to be the test set.

3.2 Model Evaluation

In order to evaluate how well our models perform in the predictive task, we will use Mean Squared Error (MSE). Compared to Mean Absolute Error(MAE), MSE has the property of penalizing large magnitude errors much more greatly, thus we can end up with a model whose errors are distributed more evenly across the whole training set. The MSE is defined as below:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

where y_i is the true (or label) value and $\hat{y}_i$ is the value predicted by our model.

3.3 Baseline

In order to better evaluate our models, we set up a baseline which makes prediction intuitively by using only the number of words in each review text and bias term as the feature and use linear regression with no regularization to get the parameters. This is one of the most straightforward machine learning method, but has a reasonable overall performance to act as a baseline. Then we will use mathematical equation to represent the baseline, and it is defined as below.

$$\text{score} \simeq \theta_0 + \theta_1 \times (the\ number\ of\ words\ in\ review)$$

Now we can apply our baseline model on the data we obtain as section 3.1. On the test dataset, the MSE achieves to be 1.7103 using the baseline model. And then we can assess the validity of the prediction of our models by calculating how much the MSE of our model exceeds that of the baseline on test set.

3.4 Feature Selection

In this section, we analyze the dataset to choose features, use these features to train models and apply the models on test set and pick features by their performance concerning MSE.

Firstly, we count the top 500 frequent adjectives used in 1-5 score reviews respectively, thus we get 2500 adjectives in total. We use the word_tokenize in nltk Python library to distinguish the part of speech of word. But there are many duplicates in them, after eliminating the duplicates we end up with about 760 adjectives. Then we use the number of each adjective as features and use these 760 features plus a bias term to train a linear regression model. We use this model to predict the rating on test set. Then we do the same process for top 500 frequent nouns and verbs, and their number of features and corresponding MSE results on test set are listed as **Table 2**.

Table 2. Comparison of MSE Using Different Kinds of Words

Category	The number of features	MSE
Adjective	761	1.2248
Verb	748	1.2547
Noun	738	1.3174

Since the number of features are all around 750, we can see from Table 2 that adjectives are most informative, thus we will use adjectives as part of our features in the following explorations.

Then we try to test if it is useful to add bigrams and trigrams as our features. Similarly, we count the top 100 frequent bigrams in review text used in 1-5 score reviews respectively, and remove the duplicates and use them as extra features plus adjectives to train models and run tests on test set. And then we do the same thing for trigrams. And also we test $\text{adjectives} + \text{bigrams} + \text{trigrams}$ model. The results are shown below as Table 3.

Table 3. Comparison of MSE Using Bigrams and Trigrams

Category	Features	MSE
Adj.	761	1.2248
Adj.&Bi.	988	1.2239
Adj.&Tri	1097	1.2174
Adj.&Bi.&Tri	1324	1.2084

From Table 3, we can see that bigram and trigram features do have some extra information, so we will use them as well as the high frequent adjectives as the features.

And choosing these features is also accord with common sense. The adjectives are usually used to describe something instead of narrating a plain fact, which has more things to do with subjective factors. Thus they are likely to be more informative when predicting one's evaluation upon a product. Introducing bigrams and trigrams will decrease the loss of information by splitting

phrases (e.g. “not bad” into “not” and “bad”, “would not recommend” into “would”, “not” and “recommend”).