Predicting Amazon Review Helpfulness

Many online retailers such as Amazon allow users to review products and rate them. Those reviews in turn can be voted up or down based on how helpful they are to other users. The ratio of positive votes (nhelpful) over total votes (Outof) shows the overall quality of the review (which we will call helpful ratio). In this assignment, we consider models for predicting helpfulness of Amazon reviews. We use features such as the total number of votes, number of sentences, text readability, and a few others.

We began by examining review length and readability. Initially we simply took the total length of the review and use that as a feature but because there are simply too many factors that could skew review length, we decided to use number of sentences instead. In addition to this feature, we also took into account how readable each amazon review is. We calculated this feature according to the Automated Readability Index formula:

$$4.71\left(\frac{characters}{words}\right) + 0.5\left(\frac{words}{sentences}\right) - 21.43$$

where the resulting score gives an approximation representation of the grade school level needed to comprehend the text (1 for kindergarten level and 14 for college level). In addition, we also postulated that many peer reviewers do not want to read an entire product review and instead tend to read the summaries instead. To capture this, we created one more feature for conciseness of the summary by simply taking the length of the summary divided by the length of the review. It would have been preferable to use number of sentences, but we discovered that oftentimes, summaries are not written in complete sentences.

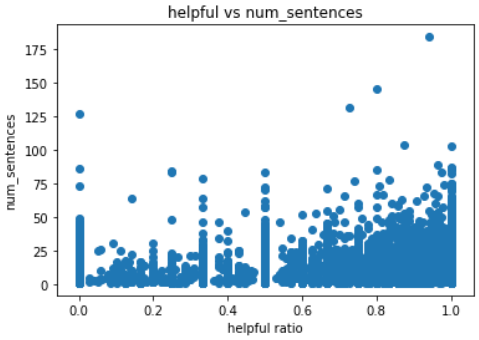

Figure 1: (helpful vs num_sentences

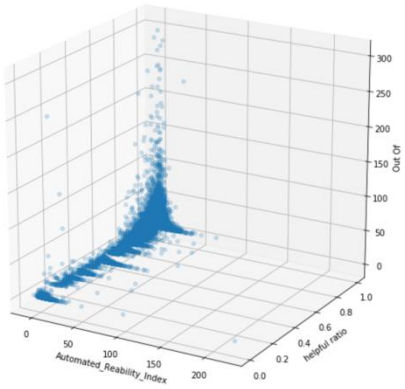

Figure 2: Readability Index vs helpful vs Outof

Additional features we used were the number of reviews written by each user and the number of reviews written about each product. The logic behind this is that users who write more reviews would be better at writing them and would thus tend to write more helpful reviews. In the same line of thought, products which have more reviews written about them would see more helpful reviews.

We used the product rating of the review as well as the product category as two more features. We saw that the ratings which reviews gave products affected their helpful ratio. Reviews which gave the product a 5-star rating were more likely to see a higher helpful ratio. Also, we noticed some differences between products in different categories, perhaps due to the demographics of people buying those products. There were

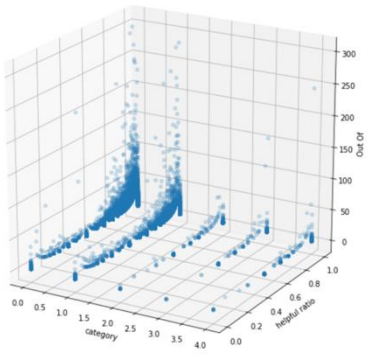

Figure 3: Category vs helpfulratio vs Outof

far less product reviews in category 3-5, making it harder to interpret. Given this info, we decided to "bucket" the categories and make each category its own feature.

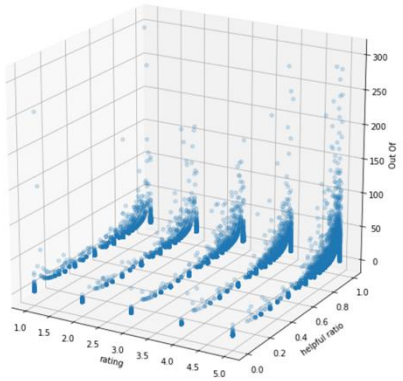

Figure 4: Rating vs helpfulratio vs Outof

Another trend of the dataset we noticed is that the distribution of Outof is skewed towards lower numbers (e.g. most reviews have very few total votes. See Figure .). We also examined the correlation between Outof and our helpful ratio and noticed that the data behaved very differently at low Outofs and higher Outofs (see figures 2, 3, and 4). We can clearly see that at high values of Outof, the the vast majority of reviews have a very high helpful ratio and at lower values, the data assumes a more concave curve. Using this information, we decided to split our training dataset into 4 different datasets on ranges of Outof: from 1-10, 10-57, and 57-500.

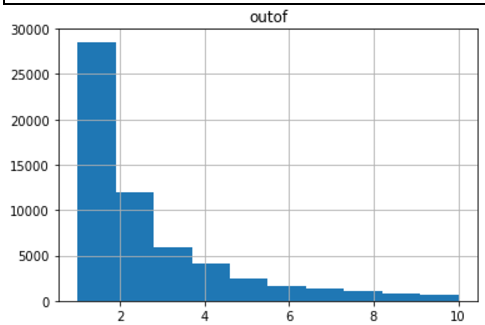

Figure 5: Distribution of reviews Outof

For our regression model, we used Gradient Boosting Regression which we determined to have the best accuracy out of various regression models. For our Outof between 57-500, we used learning_rate =0.1, loss=quantile, max_depth=7, and n_estimators=140. For our Outof between 10-57, we used learning_rate=0.1, loss=ls, max_depth=3, and n_estimators=120. For our Outof between 1-10, we noticed a rather high mean absolute error (MAE) despite excessive tuning. As a result, we decided to futher split this into two datasets with Outof =1, and Outof between 2-10. For the Outof between 2-10, we used learning_rate=0.1, loss='ls', max_depth=3, and n_estimators=120. For our Outof=1, we discovered that a Gradient Boosting Classifier yielded better results with parameters learning_rate= 0.1, loss='deviance', max_depth= 2, n_estimators= 160. All models predicted on the helpful ratio, which we rounded to 2 decimal points before multiplying by the Outof for the testdata. These results were then rounded to the nearest whole number and submitted.

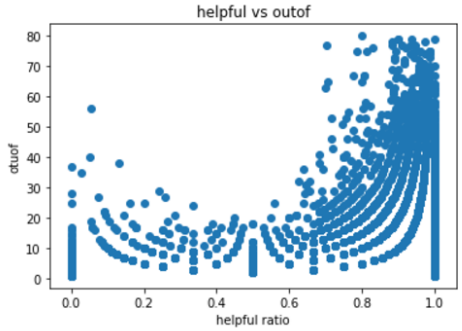

Figure 6: helpful ratio vs Outof for Outof<80

Overall, with our 4 models and 12 features (including the 5 buckets for categories), we were able to achieve an MAE of 0.17214 on the private leaderboard. Some improvements for the future would be better data cleaning. We did not remove many outliers and that may have caused extra noise in our data. We also could have attempted to do a sentiment analysis of each review instead of simply the readability. There is also room for improvement in the choice of cutoffs for segmenting our training data.