CSE 190 Assignment 2

Lamar Cimafranca
A10629461
lcimafra@ucsd.edu

EXPLORATORY ANALYSIS

The data that will be used comes from the "*reviews_Beauty.json.gz*" file which contains information about beauty products that were bought and reviewed on Amazon.com. Each data point contains the following information: 'reviewerID', 'asin' (productID), 'reviewerName', 'helpful' (number of helpful votes / total number of votes), 'unixReviewTime', 'reviewText', 'overall' (rating on a scale from 1 to 5), 'reviewTime'(MM DD YYYY) , and 'summary'. In addition data about each product comes from "*meta_Beauty.json.gz*", which includes 'asin', 'description', 'title', 'salesRank', and category for each item. An exploratory analysis is performed on the data in order to gain a better understanding of the data, and may help in choosing an appropriate model for the predictive task. Additionally, this analysis may help in determining how to split the data into training and test sets.

- There are a total of 2023082 reviews on beauty products
- There are 1210281 unique reviewers and 249274 unique products purchased that have been reviewed at least once.
- The average rating across all product reviews is 4.14903745869.
- The user with the most reviews written has the reviewerID: 'A3KEZLJ59C1JVH' and reviewerName: 'Melissa Niksic', and has written 389 reviews on beauty products.
- The item that was reviewed the most has the asin (productID): 'B001MA0QY2' and the title (productName): HSI Professional 1 Ceramic Tourmaline Ionic Flat Iron Hair Straightener', and has been bought and reviewed a total of 7533 times.
- Out of all the reviewers, 887409 of them only made 1 review. This accounts for approximately 73.32% of reviewers.
- Out of all the items, 103483 of them were only purchased a single time. This accounts for approximately 41.51% of items.
- 858935 of the reviews received at least one 'helpfulness' vote, indicating whether another user found the review helpful. This makes up approximately 42.46% of reviews.
- 176621 of the reviews received were voted on 'helpfulness' at least 5 times. This is about 8.68% of reviews.
- The average helpfulness ratio the reviews is .804296171837

Some interesting points that can be discovered from the data:

- In addition, we can see that the majority of users have only reviewed a single item.
- Some instances in the item metadata dataset include items that were never reviewed.

SELECTING RELEVANT DATA

The data that will be used for analysis will be determined in the following ways:

- Reviews with 0 votes on helpfulness were removed from the dataset because a helpfulness ratio does not exist for them. They do not provide any useful information for the predictive task.
- Reviews with less than 5 votes on helpfulness were also removed because the helpfulness ratio on reviews with a very low amount of votes do not provide as much meaningful information as reviews with a higher amount of votes.
- Reviews with more than 1000 votes on helpfulness were removed to prevent bias in the dataset. Almost all the reviews with a large amount of votes have an extremely high helpfulness ratio (over .9).
- The final size of the data to be analyzed is 176,621 reviews. From these, 100000 review data will be randomly chosen.

PREDICTIVE TASK

In this case, the predictive task is: Given the review data, predict whether a reviewer's review of a beauty product will be 'helpful' to other users. This is a classification task in which a review is classified as one of the following classes: 'helpful' or 'unhelpful'. A review that is helpful, in this case, will be defined as a review has a helpfulness ratio greater than .60.

This threshold was chosen because .60 is the average helpfulness ratio for the majority of the votes to be helpful when examining data with the minimum number of votes. For example, 5 is the minimum number of votes, so at least 3 (or 60%) of these votes must be positive in order for the review to have mostly positive reviews. The model that handles the predictive task will be evaluated on the percent of reviews that it classifies incorrectly. The evaluator will be:

$$\frac{1}{|N|}\sum_{i=1}^{|N|}\frac{\sum_{j=1}^{|L|} xor(y_{i,j}, z_{i,j})}{|L|}$$

The best possible value for the evaluator is 0, in which case all of the reviews were classified correctly. When evaluating a model, half of the data will be chosen at random and will be designated to the training set, and the other half will be designated to the test set.

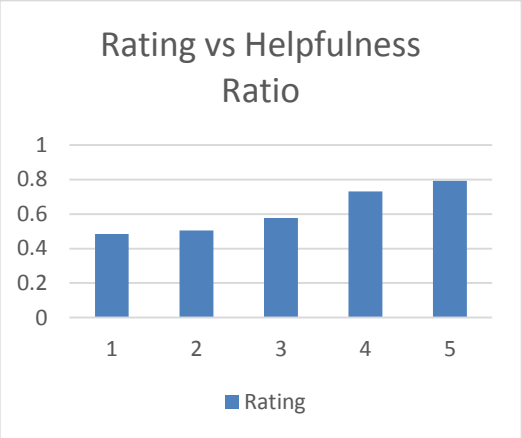
Some of the simple baselines for this task are to

- Predict either 'helpful' all the time or predict 'unhelpful' all the time. In this case, we expect that the classification error will be about .5, incorrectly classifying half of the data.
- Use the reviewer's average helpfulness ratio and multiply it by the total number of votes a review received in order to obtain the helpfulness ratio. This baseline model is expected to perform slightly better than the previous predictor, but will still classify too many reviews incorrectly.
- Predict helpful all the time, in which case it will predict correctly about 65.5% of the time, because that is the percentage of reviews where the positive vote ratio is greater than .60.

FEATURE SELECTION

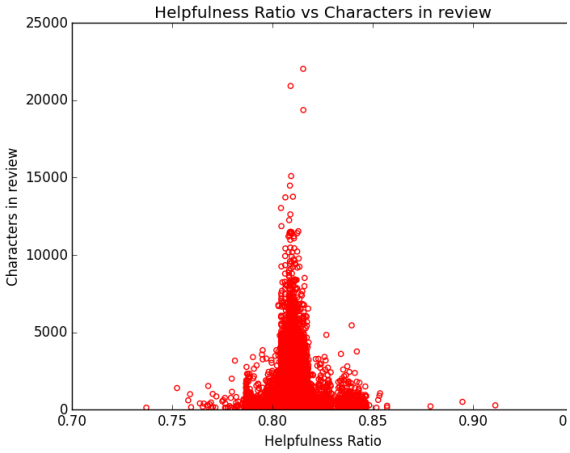
The features that were chosen for this predictive task were:

- Rating the user gave to the product – This feature was chosen because users who rate items poorly tend to also write poor reviews. This may be caused by dissatisfied customers who wrote biased reviews that were not helpful at all to other users. Also, the average helpfulness ratio for items that were given high ratings is higher than the average helpfulness ratio for items that were given low ratings (see graph below). The average helpfulness ratio for 1-star rated items was a mere .484, but the average helpfulness ratio for 5-star rated items is a lot higher at .794. For these reasons, rating should be a reasonable feature to use.



- Deviation from the reviewer's rating and the average rating – Ratings that deviate too much from the mean may be biased and unhelpful. In the exploratory analysis, we already saw that beauty products in this dataset have a high average rating. This may be part of the reason why reviews with low ratings have significantly lower helpfulness ratios.

- Length of the review – Longer reviews may be more descriptive and in-depth, and are therefore more useful to other readers.



- Number of exclamation marks in text – A review with too many exclamation marks may be too enthusiastic about the review, which may be indicative to extremeness. This type of review would not be as helpful.

- Number of question marks in text – A review with too many question marks may more focused on asking questions than actually giving out useful information about the product. This type of review should not be as helpful as well.

- Number of words in all-caps in text – Like with exclamation marks, a review with a lot of capitalized words indicates that the review may be too extreme. For example, it is not uncommon for upset purchasers to post biased, negative reviews in all-caps.

- Number of votes – The number of helpfulness votes an item receives may have an effect on the number of additional positive votes it receives. For example, the Amazon user interface tends to display the most helpful reviews at the top, where it will get more exposure and as a result, receive even more positive helpfulness votes (Amazon rarely displays unhelpful reviews). The more votes a review has, the more likely it will be to have a high positive vote ratio.

MODEL AND RESULTS

The approach this task will take is to use a Support Vector Machine (SVM) in order to classify these reviews as either helpful or unhelpful. We use sklearn.svm.svc to train the classifier. The first half of the pruned data is taken to be training set and the other half split into validation and test sets. The penalty parameter of the error term is set to be 1. After running the SVM, we get an accuracy of approximately .7859, which is several percent better than our baseline. The parameter that was tuned to optimize the classification accuracy on the test set was the penalty parameter of the error term. After trying multiple values for this term, we used the default value (1) because it gave the greatest performance on the validation set. The most important features of this model were the rating, deviation from the average rating, and the number of votes.

Another approach that was previously considered was to use linear regression using the features described in the previous section to first predict the amount of positive helpfulness votes a review received, then divide it by the total number of votes it received. The resulting ratio was then compared to the threshold. If this ratio was greater than .6 then we would predict that the review was helpful, otherwise it was predicted the review was unhelpful. The performance of this model was only slightly better than the baseline, classifying only about 67% of the reviews correctly. The weakness of this unsuccessful attempt was that it was optimized to predict the number of positive helpfulness votes (data ['helpful'][0]). Since this is a classification task, it is no surprise why linear regression did not work so well.

In addition, utilizing the Naïve Bayes model was considered. This model gave a classification accuracy of about 73.38%. Although the Naïve Bayes model was did not perform as well as the SVM classifier, it did not perform as terribly as the linear regression model, or the SVM classifier with the kernel parameter set to kernel='sigmoid' (default='rbf'), which gave a classification accuracy of approximately .507, which was worse than the baseline model.

Another attempt that was used was text mining, in which the words with the most positive weights associated with them were extracted. Then sklearn.svm.svc was again used for training, but the parameters above were replaced by parameters which indicated whether a unigram with a high positive

weight was used in the text of the review. Unfortunately, with unigram features of the text, this model could not beat the support vector machine model (with the default kernel value). In order for this to be effective, I think that this can be used with the other parameters that are based of the review data along with some type of dimensionality reduction or decomposition. This is due to the fact that each unigram will comprise one dimension. Since there are many words that can be strongly associated with either positive or negative reviews, we would want to include many unigram features. However, this type of model would be very expensive to train, so decomposition may be necessary to filter out some of the weaker unigrams.

RELATED LITERATURE

The Amazon beauty product reviews were retrieved from the SNAP web data site. Similar datasets (from Amazon) have been used before to make this predictive task. All of my features were used in these other studies, but some include interesting features. Some features that were considered by others were:

- Time of the review – The longer a review has been posted the more likely it is to receive more votes. In the exploratory analysis we know votes is correlated with helpfulness, so it may be beneficial to include this parameter.
- Average sentence length – A review that is with sentences that are too long are may not be easily readable by others.
- Term inverse document frequency – The high dimensional result representing unigrams was decomposed using single value decomposition into only a few important dimensions.
- Normalized tf-idf – Used in order to prevent bias towards longer reviews and takes a value between 0.5 and 1.
- Automated Readability Index – estimates how many years of education are required in order to understand the text

Many of these other studies have also employed the use of support vector machines to classify helpful reviews. A popular method was using the linear, rbf (radical basis function), sigmoid, and polynomial kernels on the SVM function to train a classifier. In these instances, the results of my model is very similar to the results of these other models. The of the SVM classifiers is the best performing model, except for the SVM with the sigmoid kernel which performs the worst with a classification of a mere 50%. The SVM model gives a result of between 70% and 80% accuracy, which is consistent with the results of my model. Additionally, when other people have attempted the Naïve Bayes Model, they also similar accuracy values around 70% classified correctly. In another study, the SVM with the linear kernel set achieved a significantly higher classification accuracy on their dataset. However, I did not use the linear SVM model because it is expensive to train on a data set this large.

CONCLUSION

(See ***results*** in the third section). We can conclude that a classifier can made using features only from the review data. Some of the most important features were the rating and the number of votes. It was suspected that lowly rated products tend to receive worse reviews that highly rated products, and we can see this in our exploratory analysis. Reviews with many votes also tended to have high helpfulness ratings. I suspect this may be because Amazon displays the most helpful reviews on the first page where more people can see it and give it a positive rating, or it may be because nobody will read a review if they see it does not have positive helpfulness ratings. When training the classifier, this becomes apparent that rating (and deviation from average rating), and number of votes is important because they make a large difference in classification accuracy.

The rbf SVM achieved the highest classification accuracy, significantly higher than some of the other models tested. The linear regression model did not work because it is not optimized to do classification tasks. The sigmoid model performed no better than predicting at random. Furthermore, trying to incorporate unigram features from the text did not improve the classification accuracy and is expensive to train. For this reason, I decided not to use unigram features for this task. The Naïve Bayes model had a decent classification accuracy, but not as high as the rbf classification. With some tuning of the parameters and addition of some more useful features, it is likely that this SVM classifier is able to achieve a higher accuracy.