**Preliminary Results**

Cameron Gallien

DSE6212OM\_DSE6311OM\_SU2023S5\_Text and Image Mining

Merrimack University

Prof. Peter Salemi

August 4, 2023

**Introduction**

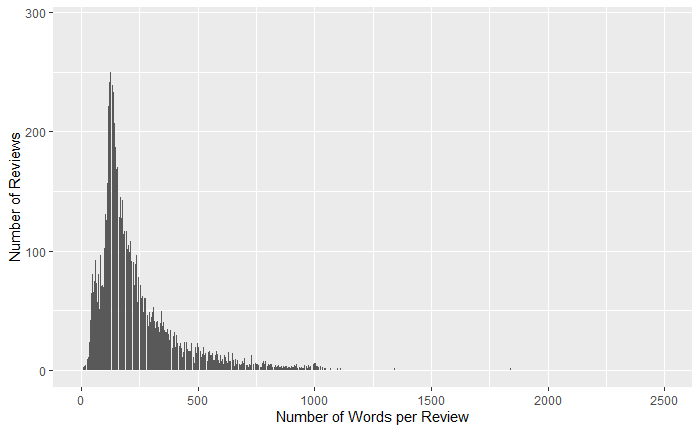
The ABC Company is looking to add a page to their website where users can discuss and review movies. In addition, the company is looking to develop an algorithm that can generate a sentiment score for each review on a scale of 0 to 1. This report will encompass the steps that were outlined in the previously submitted analytic plan, and provide a preliminary model that can be utilized to generate the required movie review sentiment scores.

**Data Processing**

I started my analysis by performing the basic data processing steps that were outlined in the analytic plan. This consisted of first cleaning all the data to eliminate punctuation, unnecessary characters, and numbers that would not provide much value to our movie review model.

Next, the data was split into training and test data sets. By splitting the data we are able to train the model on one subset of data, before testing it out on another, smaller subset. The subsets that were chosen were 37,186 (75%) reviews for training, and 12,396 (25%) for testing. Without this step there would be no way to truly determine how accurate our model would be, since it would be tested on the same information that was used to train it. This would cause overly optimistic performance metrics. The test data allows us to test the model out on a simulation of what future reviews may look like once the algorithm is implemented on the site.

From there, further data processing steps were taken. The tokenization step separates each word into an individual token that the model can then analyze. This tokenization was done at the word level to capture the meaning of individual words, and to make the model more computationally efficient. Tokens were originally limited to 500 per review, as there were not many reviews with that many words, however after doing some quick tests of the models utilized, this was later increased to 1000.

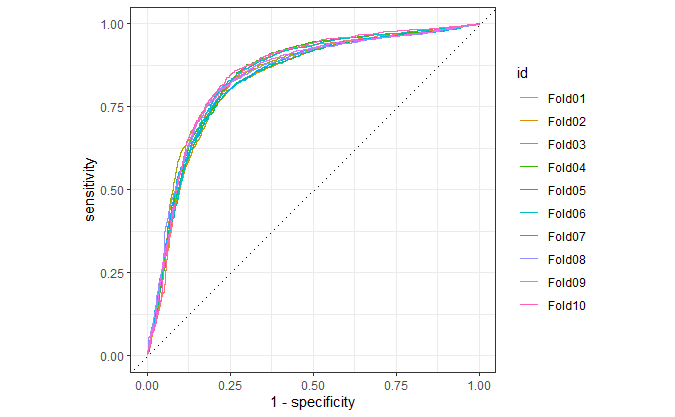


Stemming techniques were then implemented in order to reduce words to their base or “stem” which would allow different variations of the same word to be considered as the same. Finally, all of these words were placed in a TF-IDF matrix, which would store the text data numerically, and take into account the frequency of each word. This allows the model to highlight terms that are important within the reviews, and downplay the common terms that are used frequently.

**Training the Models**

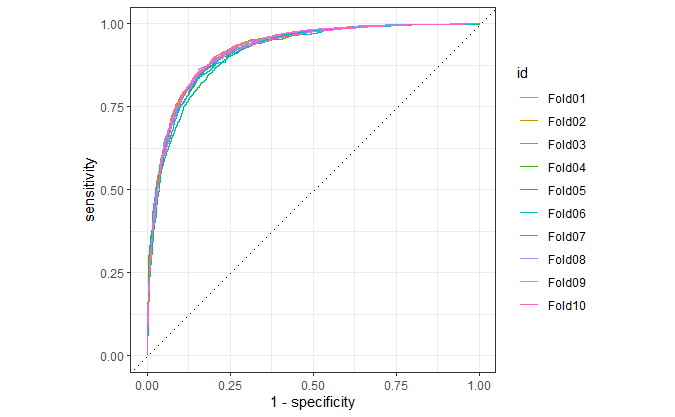
Two different models were utilized in my preliminary analysis: naïve bayes, and lasso classification. The naïve bayes model was chosen for its ability to properly analyze a large number of features, and the lasso classification model was chosen because of its ability to penalize features, a value that can be customized (Hvitfeld & Silge, 2002).

The first model chosen was the naïve bayes model. As stated above this model can handle a large number of features, which made it perfect for dealing with word counts as high as those in this project. Unfortunately, this model produced less than ideal results. The mean accuracy for this model was only 52.7%, which is definitely not reliable. The ROC curve does show however that the model does work better than random guessing:



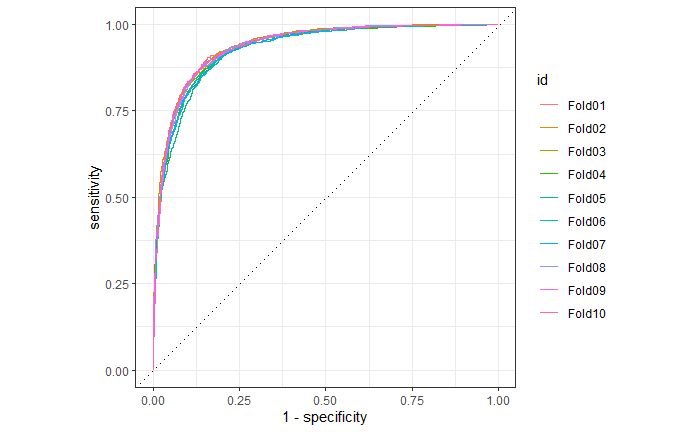
The area under the ROC curve was .833, which does prove that it is better than random guessing, but as mentioned above the mean accuracy of 52.7% is very worrisome, especially if this is going to become a tool that people go to on a regular basis for movie review scores.

Since this model produced less than stellar results, I decided to also attempt to fit a lasso classification model for the data. This ultimately ended up being a more appropriate choice, as it produced much better results than that of the naïve bayes model. Originally, I used an arbitrary value of 0.01 for the feature penalty simply to see how the model performed in a generic setting. After fitting the model, it showed an 84.3% accuracy on movie review sentiment predictions. The ROC curve also showed considerable improvement from the naïve bayes model:



The area under the ROC curve was .92, which shows much more desirable results than the naïve bayes model, and also reinforces that this model is much better than random guessing. That being said, 84.3% is still not good enough to compete with other movie review sites, so more improvement was needed.

Since the penalty of 0.01 was a somewhat randomly chosen value, I decided to try to better tune our lasso hyperparameters to see if we could achieve more successful results. After creating a model specification to properly tune the lasso, it was determined that the best penalty for the model was 0.0007880463. After adjusting the original lasso model and implementing this new penalty amount, our results showed an encouraging level of improvement. The new average accuracy of the model was 86.2%, an increase of just under 2%. While this improvement is not exactly ground-breaking, it was reassuring to see that the model was getting closer to 100% accuracy. This model also produced the following ROC curve, another slight improvement on the original lasso model:



The new AUC that was produced was equal to .935. Just like the average accuracy, this is not a massive increase from the original model, however it does show that we are on the right track, and that greater overall accuracy is possible.

**Further Data Processing and Engineering**

It is obvious after utilizing both of the above models that further work is required before our algorithm is a finished product. As far as data processing is concerned, there are a few ways that we can potentially improve our model accuracy. The first is by using varying lengths of n-grams in our further analysis. Bigrams and trigrams could possibly increase the accuracy of the model, especially if there are certain wordings in the reviews that would negate the original word meaning. For example, if the model sees a word like “exciting” and associates it with a positive movie review, this would prove problematic if the review actually said “not exciting” or “less than exciting.” These nuanced relationships between words would be better captured by utilizing bigrams or trigrams which will hopefully increase model accuracy even if it is at the expense of computational resources.

Another potential improvement could be made by employing a different stopword list. The models above utilized the stopword list from the “SnowballC” package. By doing additional analysis of smaller or bigger stopword lists, we may be able to improve the model as well. In addition, since there is a large amount of movie reviews available to us, different sized subsets of training/testing data will be investigated. It may prove fruitful to increase the size of the training set, providing the model with more examples of movie reviews to learn from.

As far as engineering goes, we may need to implement more advanced models in order to truly capture the sentiment of each review. Experimentation with deep learning methods such as dense neural networks, long short-term memory networks (LSTM), and convolutional neural networks may show substantial improvement over the more basis models that we utilized so far. An LSTM seems especially promising, as Hvitfeldt & Silge (2022) describe them as being “capable of learning long-range dependencies and broader context.” Further research will need to be done on these models to learn how to build them and implement them on the review data that has been provided.

**Conclusion**

While we were able to create models that predict movie sentiments with better accuracy than random guessing, there is still much improvement required before we have a reliable algorithm to roll out onto the company website. The highest accuracy that we were able to produce was 86.2%, which is not high enough for it to be considered a dependable resource for consumers. Hopefully after further experimentation and analysis we can produce a model that accurately predicts the sentiment score greater than 95% of the time. This is the benchmark we will shoot for once the processing and engineering steps listed above are investigated. With so many places for movie lovers to find aggregate reviews and opinions, the sentiment algorithm will need to be extremely accurate in order to keep viewers coming back to the site. If we can design a more accurate prediction model we will be able to position the ABC Company’s website as the go-to place for cinephiles and movie viewers of all types.

**References**

Hvitfeldt, E., & Silge, J. (2022). Supervised machine learning for text analysis in R. CRC Press, Taylor & Francis Group. Retrieved Aug 4, 2023, from https://smltar.com/.