IST736 Text Mining

Oliver Posewitz

Fall 2021

Multinomial Naïve Bayes Sentiment and Fake Review Detection

Overview:

In this analysis, Multinomial naïve Bayes algorithm is used to detect whether a document is positive or negative sentiment as well as detect whether each document is real or fake. The data comes from students who have written each document describing their experience at a particular restaurant with labels indicating the ground truth.

Preprocessing:

In order to reduce the amount of noise introduced into the model, several preprocessing steps were taken before building the model. Each document was tokenized using spaCy package, then all stop words, punctuation, and numbers were removed from each document.

Analysis:

Since MN Bayes is a supervised learning technique, the next step in the process is splitting the data into test and train partitions. In this case, there are only ~90 documents in all, so in order to maximize the training data, I made the test partition 30% of the overall data. The training data is what helps the model learn the probabilities that indicate which class each document would most likely be in, so the more training data, the greater potential for a more accurate model. The test data is split with respect to the proportion of the predictive feature of the entire data set to decrease likelihood of skewed predictions. For predicting both sentiment and authenticity, both features contain two classes that are of equal proportion. So the baseline accuracy for either prediction model is 50%.

Sentiment Detection:

For sentiment analysis, several different vectorization methods were tested, including count vectorization (unigram, unigram/bigram) with maximum document term frequency at 90% and 80% and term frequency inverse document frequency. Each method was fit to the multinomial naïve Bayes model, and the count vectorization unigram method showed to be the most accurate with an average 86% with 5-fold cross validation. Some of the top features with the strongest indicative probability for positive sentiment include; fresh, friendly, amazing, best. And for negative; service, terrible, wasn’t. The confusion matrix (see confusion\_matrix\_sentiment.png) describes the performance of the model. The model did exceptionally well detecting negative reviews, only missing one out of 24 in the training dataset.

In the test dataset, the model was very precise predicting negative reviews, as 95% of the predicted negative reviews were truly negative. However, of all the truly negative reviews, only 65% were labeled correctly.

Fake review Detection:

Several of the same vectorization techniques that were used in sentiment detection were also used in the fake review prediction model. The results were substantially less accurate, so further steps were taken to improve the model including; leaving in certain stop words, broader range of unigram and bigram methods, and adjusting the test split size. None of these methods had a significant impact on the resulting 55% accuracy the TFIDF vectorization method produced. Looking at the confusion matrix (see confusion\_matrix\_lie.png), the model mostly labels the documents as authentic (lie=False), and only labeled 4 of the fake reviews as fake in the training data.

Testing the model led to a surprisingly high recall (92%) for authentic reviews. This is less surprising after realizing the model is skewed to predict the reviews are mostly authentic. Perhaps the naïve-ness is due to the Naïve Bayes model? (sorry for the pun) You can observe the skewedness in the test data as only 8% of the predicted fake reviews were actually fake.

Conclusion:

Detecting sentiment is a much better suited task for a multinomial naïve Bayes model than detecting fake reviews. This intuitively makes sense once you look at how the algorithm works. It uses the probabilities of finding tokens in each class and uses those probabilities to predict the classification. With sentiment detection, this performs well as there are certain words that are inherently positive and negative, so building probabilities on these words with strong tendencies results in greater prediction accuracy. With deception, it is harder for this model to detect as there are less apparent patterns that can be captured with this type of model. Even with a larger data set, this task would not likely be well suited for detecting fake reviews.