Assignment 2

Nathan Cho
nc5123@bard.edu

Implementation

This sentiment analysis model used Naive Bayes algorithm with Laplace smoothing and $\log_2$. `Pandas` and `sklearn` packages were used for training and testing computations.

Results

The following is the metrics from the provided training and testing dataset, which includes 25,000 reviews each. Training process took 1 hour 12 minutes, and the testing process took 17 minutes[1].

Accuracy:	0.817
Precision:	0.870
Recall:	0.746

Discussions

What assumptions does the Naive Bayes model make?

The model makes two major assumptions. Firstly, it assumes that the position of words within a document does not affect their classification. Therefore, the model only considers whether a word exists or not, rather than where it appears in the text. Secondly, the model assumes that the probability of a word given a particular class (i.e., $P(word_i \mid class)$) is conditionally independent from the probabilities of other words. This assumption, known as the naive Bayes assumption, allows the model to predict the probability of a document given a class (i.e., $P(document \mid class)$) by simply multiplying the individual probabilities of each word given that class.

Based on your results, how do you believe these assumptions affected your model's performance?

The first assumption of disregarding word order has a significant impact on the performance of the model, especially for sentiment analysis tasks. In particular, the sentiment of a sentence can be reversed by the presence of negation words (e.g. "bad" conveys a negative sentiment, but "not bad at all" implies a positive sentiment). By ignoring word order, the model fails to capture these negation patterns, resulting in limited performance, as demonstrated by the results presented above.

What is one change you could make to the model that you believe would improve performance?

Adding support to detecting negation will greatly improve the performance. This can be implementing by recognizing negating words (e.g., “not”, “no” or “never”) and its predicates (e.g. “disappoints”) into single tokens such as *never_disappoints*. This approach allows the model to capture the effect of negation more accurately and thus improve its performance in this sentiment analysis task.

[1] Executed on MacBook Pro (2021, M1 Pro).