## *Assignment 2 Part 2: Algorithm Description and Results*

Implementing Naive Bayes Classifier using Spark MapReduce

**Naïve Bayes Algorithm description:**

A math problem with text

Description automatically generated with medium confidence

**Map Reduce Algorithm Description:**

1. Log prior calculation:

For each label, the calculate the prior probability by dividing the count of the label by the total count of all labels.

The prior probability is added as a new column (prior) and the log prior is added as a new column (log\_prior).

1. Log Likelihood calculation:

* Count Smoothing: The count of each word is incremented using the incr function, which likely adds a smoothing value (like Laplace smoothing) to avoid zero probabilities. The result is stored in a new column count\_smoothed.
* Total Word Count Smoothing: Similarly, the total word count for each sentiment is incremented using the incr\_v function, and the result is stored in a new column total\_word\_count\_smoothed.
* The likelihood of each word given its sentiment is calculated by dividing the smoothed word count (count\_smoothed) by the smoothed total word count (total\_word\_count\_smoothed). This is stored in a new column likelihood.
* The logarithm of the likelihood is computed to facilitate calculations in logarithmic space, improving numerical stability and efficiency. This is stored in a new column log\_likelihood.

1. Model and result creation:

* The test\_df DataFrame is exploded so that each word in the words\_stemmed column gets its own row. This helps in calculating the probabilities for each word separately.
* The exploded DataFrame is joined with the priors DataFrame on the sentiment column to add the log prior probabilities.
* It is then joined with the likelihood DataFrame on both the sentiment and word columns to add the log likelihood probabilities for each word.
* The data is grouped by words\_stemmed, sentiment, and log\_prior.
* The sum of the log\_likelihood values for all words in each review is calculated and stored in a new column sum\_log\_likelihood.
* The log-probability for each review is computed by adding the log\_prior to the sum\_log\_likelihood. This value is stored in a new column log\_probability.
* intermediate columns log\_prior and sum\_log\_likelihood is dropped for clarity.
* The resulting DataFrame, which contains words\_stemmed and their corresponding log\_probability, is joined back with the original test\_df (with sentiment renamed to label) to retain the original structure of the DataFrame.

**Priors for each class:**

**A screenshot of a phone number

Description automatically generated**

**Evaluation:** The accuracy is low as the implementation of Naïve Bayes classifier is very simple without any optimizations. For comparison, SKlearn’s vanilla NB gives accuracy of 73% with TFIDF vectorization.

|  |  |
| --- | --- |
| *Accuracy* | *0.66* |
| *Precision* | *0.59* |
| *Recall* | *0.68* |
| *F1 measure*: | *0.64* |