# Sentiment Analysis Using Naive Bayes(Multinomial and Bernoulli)

## Workflow:

1. Dataset used is the Multi-Domain Sentiment Dataset.
2. Preparing data:

1a. Parsing for Training:

Positive (found in “positive.review” file of directory of a product-category) and negative (found in “negative.review” file of directory of a product-category) reviews were separated and written to files as:

- “1.review” for reviews of rating 1.0
- “2.review” for reviews of rating 2.0
- “4.review” for reviews of rating 4.0
- “5.review” for reviews of rating 5.0

Each file described above has only the reviews, each review being separated by a line.

1b. Parsing for Testing:

Reviews from files “all.review” and “unlabeled.review” were taken, that weren’t present in “positive.review” or “negative.review” of corresponding categories, and stored accordingly by rating in their respective product-category directories as:

- “1_Test.review” for reviews of rating 1.0
- “2_Test.review” for reviews of rating 2.0
- “4_Test.review” for reviews of rating 4.0
- “5_Test.review” for reviews of rating 5.0

Each file described above has only the reviews, each review being separated by a line. Also, generally, in the dataset, the test set is put in the “unlabeled.review” file in the directory of a product-category, but some of them had empty “unlabeled.review” files. All product-categories have an “all.review” file, that contains the reviews that are present in “positive.review”, “negative.review” and in “unlabeled.review”. So, unlabeled reviews were extracted from the “all.review” and “unlabeled.review”(if not empty) files.

2. Removing Stop Words:

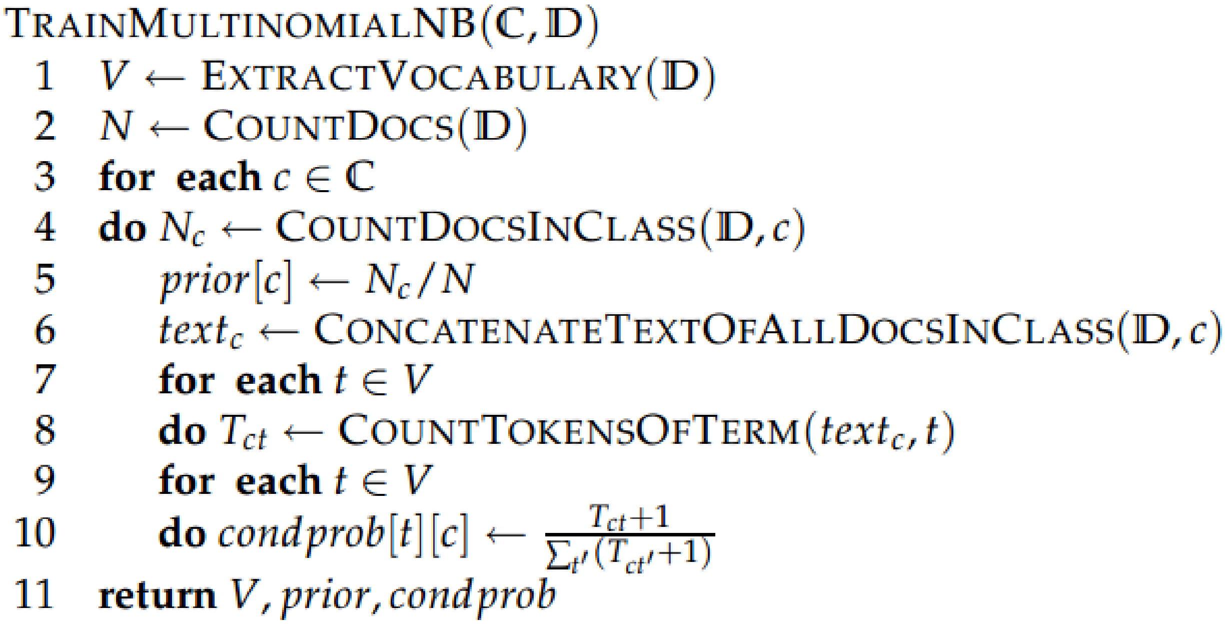
The reviews obtained after the “Parsing” step were read and all the stop words found in the file “stopwords” were removed from them. Plus, all the special characters have also been omitted.

3. Stemming of Review Tokens:

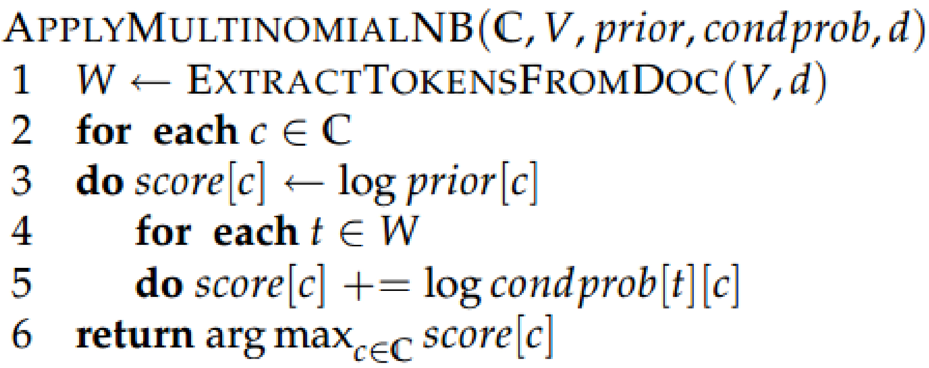
Tokens are converted into their base stems so that words of same stem are treated equally, but the output may or may not be an actual English word e.g. “because” is stemmed to “becaus”.

4. Naive Bayes Classification:
    1. Multinomial:

       The following algorithm, with some changes some of the steps or their arrangement, is used to classify a review as positive or negative based on the frequency of occurrence of the tokens of the review.



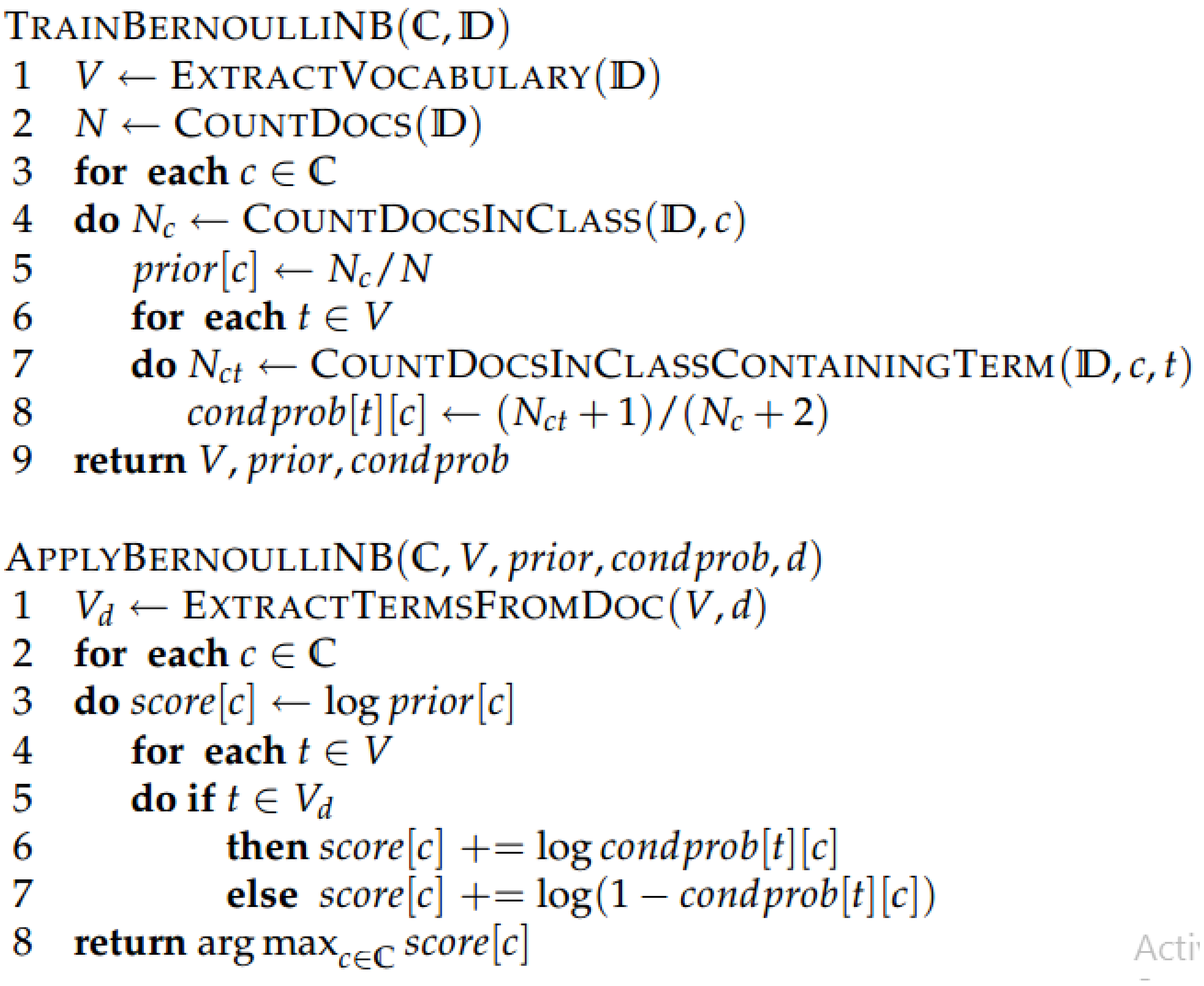
```
TRAINMULTINOMIALNB($\mathbb{C}, \mathbb{D}$)
$V \leftarrow$ EXTRACTVOCABULARY($\mathbb{D}$)
$N \leftarrow$ COUNTDOCS($\mathbb{D}$)
for each $c \in \mathbb{C}$
do $N_c \leftarrow$ COUNTDOCSINCLASS($\mathbb{D}, c$)
    $prior[c] \leftarrow N_c / N$
    $text_c \leftarrow$ CONCATENATETEXTOFALLDOCSINCLASS($\mathbb{D}, c$)
    for each $t \in V$
    do $T_{ct} \leftarrow$ COUNTTOKENSOFTERM($text_c, t$)
    for each $t \in V$
    do $condprob[t][c] \leftarrow \frac{T_{ct}+1}{\sum_{t'}(T_{ct'}+1)}$
return $V, prior, condprob$
```



```
APPLYMULTINOMIALNB($\mathbb{C}, V, prior, condprob, d$)
$W \leftarrow$ EXTRACTTOKENSFROMDOC($V, d$)
for each $c \in \mathbb{C}$
do $score[c] \leftarrow \log prior[c]$
    for each $t \in W$
    do $score[c] \mathrel{+}= \log condprob[t][c]$
return $\arg\max_{c \in \mathbb{C}} score[c]$
```

    2. Bernoulli:

The following algorithm, with some changes some of the steps or their arrangement, is used to classify a review as positive or negative based on the presence/absence of the tokens of the review.



```
TrainBernoulliNB($\mathbb{C}, \mathbb{D}$)
  $V \leftarrow$ ExtractVocabulary($\mathbb{D}$)
  $N \leftarrow$ CountDocs($\mathbb{D}$)
  for each $c \in \mathbb{C}$
  do $N_c \leftarrow$ CountDocsInClass($\mathbb{D}, c$)
      $prior[c] \leftarrow N_c / N$
      for each $t \in V$
      do $N_{ct} \leftarrow$ CountDocsInClassContainingTerm($\mathbb{D}, c, t$)
          $condprob[t][c] \leftarrow (N_{ct} + 1)/(N_c + 2)$
  return $V, prior, condprob$

ApplyBernoulliNB($\mathbb{C}, V, prior, condprob, d$)
  $V_d \leftarrow$ ExtractTermsFromDoc($V, d$)
  for each $c \in \mathbb{C}$
  do $score[c] \leftarrow \log prior[c]$
      for each $t \in V$
      do if $t \in V_d$
            then $score[c] \mathrel{+}= \log condprob[t][c]$
            else $score[c] \mathrel{+}= \log(1 - condprob[t][c])$
  return $\arg\max_{c \in \mathbb{C}} score[c]$
```

## Interpretation of Results:

Result describing files of each product-category are located in directory of that product-category in the folder "Sentiment Analysis Using Naive Bayes\Sentiment Analysis Using Naive Bayes\bin\sorted_data", "Sentiment Analysis Using Naive Bayes" being folder of the project.

- "Labels_Bernoulli.review" has actual labels found in dataset, and labels calculated by our Bernoulli Multi-Variate Naive Bayes classifier, separated by a space, one line for each review that was classified.

- “Stats_Bernoulli.review” has the following format:
  <# of reviews having rating 1.0 in test dataset>
  <# of reviews having rating 2.0 in test dataset>
  <# of reviews having rating 4.0 in test dataset>
  <# of reviews having rating 5.0 in test dataset>
  <Accuracy of the classifier>
  <Precision of the classifier>
  <Recall of the classifier>

- “Count.review” has the following format:
  <# of reviews having rating 1.0 in training dataset>
  <# of reviews having rating 2.0 in training dataset>
  <# of reviews having rating 4.0 in training dataset>
  <# of reviews having rating 5.0 in training dataset>
  <# of total reviews in training dataset>

- “Priors.review” has the following format:
  <prior probability of reviews from training set of rating 1.0>
  <prior probability of reviews from training set of rating 2.0>
  <prior probability of reviews from training set of rating 4.0>
  <prior probability of reviews from training set of rating 5.0>

- “Probabilities_Bernoulli.review” has Vc # of lines, Vc being the size of the vocabulary of training set of product-category c; each line corresponding to a term of the vocabulary, each line being in the format:
  <termID><space><term><space><Probability of term being present in reviews of rating 1.0><space><Probability of term being present in reviews of rating 2.0><space><Probability of term being present in reviews of rating 4.0><space><Probability of term being present in reviews of rating 5.0>

- “SortedOnTerms.review” has Vc number of lines, Vc being the size of the vocabulary of training set of product-category c, alphabetically sorted by terms; each line corresponding to a term of the vocabulary, each line being in format:
  <term><space><termID>

- “Labels_Multinomial.review” has actual labels found in dataset, and labels calculated by our Multinomial Naive Bayes classifier, separated by a space, one line for each review that was classified.

- "Stats_Multinomial.review" has the following format:
  <# of reviews having rating 1.0 in test dataset>
  <# of reviews having rating 2.0 in test dataset>
  <# of reviews having rating 4.0 in test dataset>
  <# of reviews having rating 5.0 in test dataset>
  <Accuracy of the classifier>
  <Precision of the classifier>
  <Recall of the classifier>

- "Probabilities_Multinomial.review" has Vc # of lines, Vc being the size of the vocabulary of training set of product-category c; each line corresponding to a term of the vocabulary, each line being in the format:
  <termID><space><term><space><Probability of term being present in reviews of rating 1.0><space><Probability of term being present in reviews of rating 2.0><space><Probability of term being present in reviews of rating 4.0><space><Probability of term being present in reviews of rating 5.0>

## Using the Code:

1. Multinomial:
   - Train_Multinomial( ) takes the "positive.review" and "negative.review" files in directory of each product-category and outputs the file "Probabilities_Multinomial.review" in the same directory.

   - Test_Multinomial( ) takes the "TestSet.review" file in directory of each product-category and outputs the files:
     1. "Labels_Multinomial.review"
     2. "Stats_Multinomial.review"

2. Bernoulli:
   - Train_Bernoulli( ) takes the "positive.review" and "negative.review" files in directory of each product-category and outputs the file "Probabilities_Bernoulli.review" in the same directory.

- Test_Bernoulli( ) takes the “TestSet.review” file in directory of each product-category and outputs the files:
    1. “Labels_Bernoulli.review”
    2. “Stats_Bernoulli.review”

## **Notes:**

1. The dataset is located in the directory of the project as “Sentiment Analysis Using Naive Bayes\Sentiment Analysis Using Naive Bayes\bin\sorted_data\”.
2. The images of the algorithms are taken from the book “Introduction to Information Retrieval by christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze”, Multinomial from page 260 and Bernoulli from page 263.