Text classification report:

1. I use two libraries: math for the avoiding underflow by using log and increase speed; string for delete all punctuations from the sentence.
2. First I define a function to help me tokenize the sentence, I lower all the letters and remove all the punctuations and use .split() to remove the id from the sentence.
3. Then I want to build a Naïve Bayes Classifier.

From the book we know the formula:

So, for the positive class and negative class:

We first calculate total word in each training file:

count\_total\_pos and count\_total\_neg

Then we create three dictionaries:

dictforvac: key is all vocabulary, value is 1.

dict\_pos: key is positive word, value is times it occurs in training set

dict\_neg: key is negtive word, value is times it occurs in training set

We then use formula:

To calculate:

Then we use formula to do the add-1 smoothing:

The bottom are:

pos\_bottom = log(sum(dict\_pos.values) + len(dicforvac))

neg\_bottom = log(sum(dict\_neg.values) + len(dicforvac))

The top are:

dict\_pos[word] + 1

dict\_neg[word] + 1

And :

Which we have P(c), and use a for loop to iterate though all test sentences, and for each sentence we keep adding new log values using a for loop of all words in sentence to calculate a positive probability and a negative probability, and compare them.

Then if positive probability is bigger then the sentence is positive

Or it is negative.