Artificial Intelligence (CS 534)
Assignment 4 - Bonus (130 points)
Due: November 23th, 11:59 PM

In this assignment you are required to build a Naïve Bayes email spam filter.

Data Description

The data can be downloaded from here.
This dataset was created from 64 emails collected from the DBWorld mailing list.
Please note, the actual emails are not given to you, and the emails have already been processed using NLP.
There are two datasets, **dbworld_bodies_stemmed** and **dbworld_subjects_stemmed** corresponding to the email body and email subject respectively
The data is currently represented as a *binary stemmed* bag-of-words and requires no additional NLP.

- Each dataset is in a table form with 64 rows and *n* columns.
- The *1st* column is "id" and has values from 1 to 64, corresponding to each of the 64 emails (this column can be removed).
- The *2* to *n-1* columns are unique words found in all the emails, they have binary values i.e. 0 means that the word did not appear in the email and 1 means that the word appeared.
- The *nth* column is CLASS, 0 means discard email and 1 means keep email.

Naïve Bayes Classifier

a. You should implement from scratch a Naïve Bayes classifier (using the spam filter example discussed in class).
Also implement Laplacian smoothing to handle words not in the dictionary. (6**0 points**)

b. Using the implemented algorithm, train and test the model for each dataset.
Use 80% of each class data to train your classifier and the remaining 20% to test it.
Which dataset provides better classification i.e. email body or email subject? (3**0 points**)

$$f\text{ - }measure = \frac{2 \times Pre \times Rec}{Pre + Rec}$$

$$\text{where } Pre = \frac{TP}{TP + FP}; \quad Rec = \frac{TP}{TP + FN};$$

and *TP* is the number of true positives (class 1 members predicted as class 1),
TN is the number of true negatives (class 2 members predicted as class 2),
FP is the number of false positives (class 2 members predicted as class 1),
and *FN* is the number of false negatives (class 1 members predicted as class 2).

c. Compare your classifier with the scikit-learn implementation (sklearn.naive_bayes.MultinomialNB).
Repeat the analysis from (b). (3**0 points**)