Spam Classifier using Naive Bayes

Dataset reference:

I have used the below mentioned dataset to train my spam classifier model:

http://openclassroom.stanford.edu/MainFolder/DocumentPage.php?course=MachineLearning&doc=exercises/ex6/ex6.html

Algorithm:

I am using naive bayes algorithm for training my spam classifier.The steps involved in this algorithm are as follows:

i) Preprocessing:

In this step, I am pre-processing each mail present in files ”spam” and ”non-spam”.
I am then taking the raw emails provided to us and cleaning or pre-processing them such that all the special characters or any punctuation are removed and all the alphabets are changed into small letters. In other words, after the pre-processing stage, I'll only have words separated by spaces.
This pre-processing is also done to the test dataset.

ii) Training the model:

While training my spam classifier model, I am creating CSV files, in order to store the all the vocabulary present in those emails, and I am also storing the frequency of their occurrences.
There are three CSV files that are being created during training the model:
a)”vocabulary.csv” : This file contains all the vocabularies and their corresponding frequencies present in both spam emails as well as non-spam emails.
b)”spam.csv” : This file contains as much as 5000 vocabularies and their corresponding frequencies that are present in the spam emails.
c)”non-spam.csv” : This file contains as much as 5000 vocabularies and their corresponding frequencies that are present in the non-spam emails.

iii) Prediction of emails:

I am assuming the prior probabilities of P(spam)=0.5, and P(non-spam)=0.5.
Suppose an email contain n words($w_1, \ldots, w_n$).

Then, given those words, I am calculating the probabilities in the following ways:

$$P(spam/w_1 \cap w_2 \cap .... \cap w_n) = \Pi_{i=1}^{n}\left(\frac{P(w_i/spam).P(spam)}{P(w_i)}\right)$$

$$P(non - spam/w_1 \cap w_2 \cap .... \cap w_n) = \Pi_{i=1}^{n}\left(\frac{P(w_i/non - spam).P(non - spam)}{P(w_i)}\right)$$

I am calculating the probability of a given word in the following way:

$$P(word) = P(word/spam) \times P(spam) + P(word/non - spam) \times P(non - spam)$$

Suppose a new word is coming for the first time, then it will make P(word)=0, so I am using additive smoothing to avoid this problem.

At last, after comparing $P(spam/w_1 \cap w_2 \cap ...... \cap w_n)$ and $P(non - spam/w_1 \cap w_2 \cap .... \cap w_n)$, I am predicting whether the given mail is spam or non-spam, i.e.,
if the $P(spam/w_1 \cap w_2 \cap ...... \cap w_n) > P(non - spam/w_1 \cap w_2 \cap ...... \cap w_n)$, then the email will be classified as spam, else non-spam.