1. [Movie review classification using Naïve Bayes - 10 points]

   Assume that you have trained a Naïve Bayes classifier for the task of sentiment classification (please refer to Chapter 4 in the J&M book). The classifier uses only bag-of-word features. Assume the following parameters for each word being part of a positive or negative movie review, and the prior probabilities are 0.4 for the positive class and 0.6 for the negative class.

	pos	**neg**
I	0.09	0.16
always	0.07	0.06
like	0.29	0.06
foreign	0.04	0.15
films	0.08	0.11

   Question: What class will Naïve Bayes assign to the sentence "I always like foreign films"? **Show your work.**

$$p(pos) * p(s|pos) = .4 * .09 * .07 * .29 * .04 * .08 = 2.34 * 10^{-6}$$
$$p(neg) * p(s|neg) = .6 * .16 * .06 * .06 * .15 * .11 = 5.7 * 10^{-6}$$

According to the product above, the Naïve Bayes model would assign this sentence to the negative class.

2a: Implemented in NB.py
2b: Parameters hard coded into testnb.py. The parameters are stored as dictionaries.
2c: Action
I predict that the test set is a part of the action class. The P(action) is 0.00028577960676726106 and the P(comedy) is 0.00018310546875.

2d: Files preprocessed and pickled into vectors and dictionaries. You can easily generate this set by running preprocess.py and having the following directory structure in your current working directory:

```
#positive class of training data
path_to_pos_train = r'movie-review-HW2/aclImdb/train/pos'
#negative class of training data
path_to_neg_train = r'movie-review-HW2/aclImdb/train/neg'
#positive class of testing data
path_to_pos_test = r'movie-review-HW2/aclImdb/test/pos'
#negative class of testing data
path_to_neg_test = r'movie-review-HW2/aclImdb/test/neg'
```

Result analysis:

The model successfully predicted about 5500 of the 25000 test cases accurately. The reason for such a high error is because of precision errors. There were approximately 12500 cases where python couldn't compare likelihoods of classes. I tried this in both regular non-log math and with log math and either way, the precision eventually didn't work with whatever values I choose for the log base or however big/small the original probabilities were.

Instead of discluding those results, I randomly chose neg or pos for those classes and the result was that 12416 of the results were predicted correctly or about 50%.

I noticed that when the reviews are smaller, it is more likely that the prediction is correct and as the reviews get larger, the accuracy decreases. I predict that this is because of the naïve approach of multiplying probabilities.

Happy holidays!