Classifying movie reviews

1. Download movies-students.py from Nexus.
2. Explore the code.
    1. What does the feature vector that is created to represent documents look like? Sketch an example.
    2. How many items are in the test set and the training set?
3. Evaluate
    1. What is the accuracy?
    2. Examine the 20 most informative features. Can you explain why these particular features are informative? Do you find any of them surprising?
4. Explore improvements
    1. Come up with additional features and test them. (Note that the nltk classifiers cannot deal with numeric features. So, if you want to use features with a numeric value, you have to break it into categories such as high vs. average vs. low.) Some suggestions:
        1. Add bigram features, that is, in addition to individual words look at pairs of words that appear next to each other in a document. The NLTK function `bigrams` takes a list of words and returns a list of bigrams.
        2. Word features can be very useful for performing document classification, since the words that appear in a document give a strong indication about what its semantic content is. However, many words occur very infrequently, and some of the most informative words in a document may never have occurred in our training data. One solution is to make use of a lexicon, which describes how different words relate to one another. Using WordNet lexicon, augment the movie review document classifier presented in this chapter to use features that generalize the words that appear in a document, making it more likely that they will match words found in the training data. (See Chapter 2 of the NLTK book to learn about WordNet.)
    2. Try out different machine learning algorithms. See Chapter 6 of the NLTK book and http://www.nltk.org/howto/classify.html.