CS114 (Spring 2017) Homework 3
Naive Bayes Classifier and Evaluation

Due March 15, 2017

You're given *movie_reviews.zip*, the NLTK movie review corpus. Reviews are sorted by sentiment (positive/negative). The files are already tokenized. Each review is in its own file. Before you begin the assignment, you should separate the data into a training set (80% of the data), a dev-test (10% of the data)—used for tuning your classifier, and a test set (10% of the data).

We suggest you using Ipython Notebook to do this programming assignment (You should write a monolithic file with extension *.ipynb*, similar to the one we use for lab session, for example, this one Lab note 3). You should write your code and comments using Code box, write any report and calculation using Markdown box.

Naive Bayes Classifier

We will train the sentiment classifier using Naive Bayes Classifier. For the following questions, use the notations provided below. We use three features to classify the label. More formally,

- X1 = 1 if the review containing the word *'great'*. 0 otherwise.
- X2 = 1 if the review containing the word *'poor'*. 0 otherwise.
- X3 = 1 if the review containing the word *'long'*. 0 otherwise.
- Y = 1 if the review is labeled positive. 0 if labeled negative.

1. Fit the Naive Bayes classifier. The likelihood $P(X_i|Y)$ should be a Bernoulli distribution. Show the calculation steps and list all of the probability distributions involved in the model.
2. If a review contains the word *'great'* and the word *'poor'*, but not the word *'long'*, what is the probability that the review is positive according to your Naive Bayes Classifier? Show your calculation steps (both formula and code).
3. Compute the mutual information $I(X_i, Y)$ for all three features. Which feature is the best feature according to this metric?

Evaluation Metrics

Given a performance report of a classifier in the form of a confusion matrix:

	predicted A	predicted B	predicted C
true A	5	1	1
true B	1	4	2
true C	0	2	4

Table 1: Evaluation

1. Compute precision, recall, and F1 for each class.
2. Compute macroaveraged F1
3. Compute microaveraged F1
4. Compute accuracy rate

NOTE: "Macroaveraging" gives equal weight to all classes, better measuring the effectiveness of the classifier on smaller classes. "Microaveraging" gives equal weight to every per-document classification, better measuring the effectiveness of the classifier on larger classes. The two class size in this dataset should be about equal. For more information on macroaveraging vs. microaveraging, including formulas, visit:

http://www.clips.uantwerpen.be/vincent/pdf/microaverage.pdf

Submission Instructions

Submit your notebook file only (if you write other helper python files, compress your file to .zip or .gz before submission). Don't include *movie_reviews.zip*