Lab 4: Feature Wars

October 5th 2017

In this lab you'll use nltk's MaxEnt (logistical regression) classifier to classify book reviews as positive or negative. You may work with a partner on this assignment if you'd like, just make sure to specify who you worked with in a comment at the top of your code. Please turn in your solution by midnight on 10/11/17.

Relevant Resources:

- http://www.nltk.org/book/ch06.html
- http://www.nltk.org/howto/classify.html

Logistics: 20 points

You have been provided an xml dataset of 1900 book reviews. A test set of 100 reviews has been held out for grading. Each review contains a unique id, the review text, which you'll use to generate a set of features, and a rating. We'll consider ratings of 4 or 5 to be positive, and ratings or 1 or 2 to be negative. Your program must take two xml files as parameters, the first being the reviews you'll train on, and the second being the reviews you'll test on. For example, when I grade your assignment, I'll be expecting to use the following command:

```
python3 maxent.py TrainingSet.xml TestSet.xml
```

Your program should output the list of unique id's from the test set, as well as the class ("Positive" or "Negative") that you've assigned to it, separated by a tab. It should look like this:

```
0679749004    Positive
0140116168    Positive
0471477540    Positive
...
...
0312273193    Negative
```

For grading, your classifier will be trained on the entire TrainingSet.xml that you've been provided and then tested on the held out TestSet.xml (from which the <rating> tags have been removed). This will be done automatically so you mustn't get

cute with your formatting.

Since it would defeat the purpose of the assignment to give you my TestSet.xml, you'll have to create your own by lopping off a piece of the training data. You can decide how much to lop off, but I would recommend testing your solution using a couple different training/test splits. Just make sure you remove any use of the <rating> tag in the test set before you submit, since that tag won't be there in the test set I'll be using.

Part 1: 50 points

Get your classifier up and running. My recommendation is to make up 2 or 3 dummy features (review contains the word "great") to use while you're initially setting up your classifier. You'll probably also want to create a function that outputs your accuracy (aka the percentage of reviews in your test set that you classified correctly), though you'll have to remove this before submiting since the grading test set won't have <rating> tags.

Part 2: 30 points

You now have 70 points on this assignment. For the remaining 30 points, you'll have to outwit your classmates. Design an improved set of features that will boost the accuracy of your classifier. The person/team who can produce the highest accuracy will get a 100 on the assignment. Everyone else will get a scaled score based on their accuracy.

Disclaimer: The above scaling does **not** mean that one team will get 0 points here. It just means that you will not get more points than someone who produced a higher accuracy. It is totally possible for everyone to get an A on this assignment.

Step 7

Submit your code using the command:

```
provide comp150nlp maxent maxent.py
```