**Text Analytics Group Assignment #2 (Revised)**

**Date Due: By 11:59 p.m. on 9/21 (now Sunday)**

**The data for this assignment (Yelp Restaurant Review Data) is posted on Canvas.**

This Yelp dataset has information on restaurants (e.g., type of food, price range, etc.) as well as reviews written by patrons. The output variable is the star rating (1-5). It will be best to convert this rating to high (say, ratings of 4 & 5) and low (1, 2, 3).

**For all the tasks below, you can use any tool (previously I had asked you to choose between WEKA and LightSIDE).**

**Task A.** Ignore the text (reviews) and run a classification model with the numeric data (you can use standard methods like logistic regression or k-nearest neighbors). What is the accuracy of your model? Show the confusion matrix. Would this analytic model be a good basis for a recommendation engine?

**Task B.** Perform a supervised classification on a subset of the corpus using the reviews only. What accuracy do you get from this text mining exercise? Explain what you did: e.g., TF-IDF scores, Naïve Bayes, etc. What is the best overall accuracy you got? What was the best accuracy for identifying highly rated restaurants?

**Task C.** Combine the numeric data and the text classification model (in task B) to create a “hybrid” model. It is your task to figure out how to do this. Now run this hybrid classification model and compare the results with those in A and B. Explain and justify how you created a mixed model with both numbers (e.g., price ranges, type of food) and text (the reviews). Did you get better results? Discuss the implications of these “hybrid” models based on your “best” attempt.

**Task D.** Use unsupervised sentiment analysis (with SentiStrength or any other tool) and use the sentiment score to predict high/low rating. Compare and contrast the results of tasks B and D.

**Task E.** Use unsupervised clustering on the text with Euclidean distances. Does this clustering achieve satisfactory separation between high and low rated restaurants relative to the results obtained above?

**I have removed the extra credit question since for WEKA users it involves writing quite a bit of Java code while those using python (or R) will be able to simply call a library with cosine similarity built in.**

Note: I have posted primers for XLMiner (in case you want to use it for Task A – always good to know more packages) and SentiStrength on Canvas. I have not written a primer on WEKA, but many are available online (including videos on Youtube).

For Mac users, download the Java version of SentiStrength from here:

<http://sentistrength.wlv.ac.uk/SentiStrengthCom.jar> (please copy and paste rather than clicking the link)

and use it with the text files extracted from this zip file

<http://sentistrength.wlv.ac.uk/SentStrength_Data_Sept2011.zip>

There is no JavaDoc but there is a brief manual online.

<http://sentistrength.wlv.ac.uk/documentation/SentiStrengthJavaManual.doc>

There is a bit of extra information for the Java version on the web site too. This is for academic use only.