CSE 6240 Spr 2017 - Homework #3
Due: Apr 19th, 2017
Analyzing a Movie Review Dataset - Part 2

Submit a single ipython notebook that contains clear, concise code and comments and all output requested.

Read through this tutorial on kaggle, https://www.kaggle.com/c/word2vec-nlp-tutorial/details/part-1-for-beginners-bag-of-words , as well as parts 2-4. Write your own annotated ipython notebook(s) to complete the exercises below. You can start with the sample code provided in the tutorial, but should clean it up, document and refactor as necessary.

1. **(30 pts): Word2Vec**
   Using the blogs as a reference:
   a. Create vector representations for each movie post in your training set by training word2vec with context=5, embedding dimension=100, min_words=40. We'll call the collection of these representations Z1.
   b. Create vector representations for each movie post in your training set by loading the pretrained Google word2vec model. We'll call the collection of these representations Z2.
   c. With k=10, do k-means clustering on each set Z1, Z2. Print a table of the words in each cluster for Z1 and for Z2.
   d. Featurize the training and test reviews in Z1, Z2 to produce design matrices X1, X2 as described in part 3 of the blog series. Basically, each review is converted into a bag of centroids feature vector with each vector component representing the count of the number of words in that review that belong in that component's cluster.
   e. Save X1, X2 for
2. **(30 pts): Topic Modeling**
   Using this LDA example blog and others as a reference:
   a. Perform LDA topic modeling on your training set with ntopics=10. Featurize each email by its topic composition. We'll call this representation X3.
   b. Repeat part (a) with ntopics=20 (this will be X4).
   c. Print tables of words for each of the topics from (a) and (b)
3. **(40 pts): Classification Experiment**
   Using the Kaggle blog series as a guide:
   a. Properly train and tune a collection of random forest classifiers using cross-validation for each of the design matrices X1...X4. You should end up with four classifiers, M1...M4.
   b. Plot the ROCs for each classifier computed on the test set. Plot all ROCs on the same plot, but use different linestyles.

c. Which featurization technique works best for sentiment classification? Is this better or worse than the simple bag-of-words approach? What are at least three things you could do to improve the efficacy of the classifier?