**Data Description:** The four attached json files (savedtweets\_americalatina.json, savedtweets\_machinelearning.json, savedtweets\_superleague.json, savedtweets\_weibo.json), represent four separate classes of 100 tweets collected using a search query with the appropriate suffix. For example, **saved\_tweetsamericalatina.json** has 100 tweets with the query “América Latina.”

Each tweet has up to seven characteristics (stored as key-value pairs): screen\_name, text, location, lang, retweet\_count, latitude\*, and longitude\*.

\* Many tweets are missing these characteristics: see instructions below.

**Instructions (Four parts in total):**

**Part 1.** Load each json file into Python (obtaining a list of dictionaries for each) and perform the following:   
 a. discard any tweets that lack latitude (those without latitude will also lack longitude, and vice-versa)  
 b. Use the tweet-preprocessor to clean the text for each tweet using all available (default) options.

For each collection, save the modified list of tweets *back* into a **new** json file with the name **prep\_tweets\_class#.json**, where # matches the order of json files cited above (0=…americalatina, 1= …machinelearning, 2=..superleague, 3=weibo). You should have files **savedtweets\_class0.json**, **savedtweets\_class1.json**, **savedtweets\_class2.json**, and **savedtweets\_class3.json** at the end of the process.

**Part 2.** For **each** *modified* collection of tweets (i.e. after the transformation from part 1) calculate the # tweets with positive, negative, and neutral sentiment and depict these on a simple bar plot. You should have 3 bars per plot (one bar for positive, one bar for negative, one bar for neutral), and 4 plots total (one per tweet query class).

**Part 3.** Pool together all *modified* tweets into a single list, but maintain a combined *secondary list* of equal size that dictates the class (0, 1, 2, or 3) to which each tweet belongs. Ex: If there are 44 América Latina tweets at the beginning of the pooled list of tweets, the first 44 elements of the secondary list should be 0.

**Part 4.** Assume your combined lists each have a length of **n**. Your next goal is to construct a **n x 5 numpy feature array** suited for machine learning, where each row matches the corresponding index in your lists, and the 5 columns represent the features for the tweet at that position as follows:  
Feature 1: The length of the tweet’s text.  
Feature 2: The tweet’s retweet count.  
Feature 3: The tweet’s latitude.  
Feature 4: The tweet’s longitude.  
Feature 5: one of two values as follows: 0 if the tweet is in English, or 100 otherwise.

For example, the first row in your feature array may look like the below:  
**[80. , 1. , 46.2380576 , 6.15323095, 100. ]**

**Part 5.** Convert your secondary list of classes into an array, and then perform 10-fold cross-validation using three distinct classification estimators (either the ones we used in class, or those of your own choosing) to determine the accuracy available in using our features from part 4 in predicting the class of tweets.

**Part 6.** Using the t-SNE estimator to compress our features into 2 dimensions, visualize the tweets on a scatter-plot with 4 different colors for 4 different classes. Briefly comment (inline code comments are fine) on where you see distinct clusters of classes on the plot, and where you do *not* see any distinction.