Unstructured Data Analytics **Homework 1**
Fall 2017 **October 5, 2017**

Deadlines Homework 1 is due on Oct. 23th 11:55pm. Therefore, you have in total 18 days to finish this homework. The late policy for all our homework has been carefully discussed in the first lecture.

How to submit: Please submit a zipped file to the following email address: kaizhaofrank AT gmail.com. The title of the email should be 'Unstructured Data Analytics-HW1' and the file name should be 'Yourname-Pantherid.zip'. In the zipped folder it should contain the homework data set and two separate ipython notebook files '1-zipf.ipynb' and '2-tf-idf.ipynb', for the first and second problem respectively.

Data Set: The instructor has prepared a small collection of Amazon reviews (626 review comments from 19 products) in the file Homework_1.csv. The data set is in the iCollege under the folder Homeworks. The columns are named and organized in the following manner: ProductID, ReviewID, ReviewTitle, ReviewTime, Verified, ReviewContent, ReviewRating.

---

1. (7.5 points) Understand Zipf's Law

   First, let's examine the Zipf's law with the provided Amazon review data set. This can be achieved by the following steps: First, breaks the comments into meaningful units (Tokenization). You do not need to do Stopword removal or Stemming for this task. Second, for each token, go over all the review comments containing it (in both train and test folder), and accumulate its frequency, i.e., total term frequency (TTF). Third, order the tokens by their TTF in a descending order. Fourth, create a dot plot by treating each word's rank as x-axis and its TTF as y-axis. Please use log-log scale for this plot.

   What to submit:

   An ipython notebook file with the plot of the word frequency based on all comments in the end.

2. (7.5 points) TF-IDF (compute comment similarity between different ratings)

   The Amazon rating system is 1 to 5 stars, with 5 stars being the best. The comments with each rating can be viewed as a separate document. In total we will have 5 separate documents for 1 star, 2 star, 3 star, 4 star, or 5 star comments. Using this document representation to categorize all the comments with different ratings into 5 documents.

   Each document (1 star, 2 star, 3 star, 4 star, or 5 star comments) can be represented as a N-dimension vector. You need to preform Stopwords removal and Stemming before generating each document. Each dimension in this vector space is defined by the unigrams of the all comments from all documents; while the weight for each unigram in this rating can defined by TF-IDF. Specifically, we need to use "Sub-linear TF scaling" to compute the normalized TF of each unigram in a document (e.g., sklearn.feature_extraction.text.TfidfVectorizer(sublinear_tf=True)).

Construct the vector space representations for these 1 to 5 star reviews and find out the most similar reviews to 1 star, 3 star and 5 star reviews, where the similarity metric is defined as cosine similarity.

What to submit:

For each star rating (1 star, 3 star, 5 star), list the most similar review document and the corresponding cosine similarity. E.g., for the 5 star comments, the most similar review documents is 4 star comments, with a cosine similarity 0.52.