**Text Analytics Group Assignment 1 (Fall 2018)**

**Date due: Sep 13 by 11:59 p.m. on Canvas.**

This assignment has two parts, A and B. In A, you will perform some basic text mining tasks just to familiarize yourself with the nuances of putting text mining theories to practice with Python scripts. If you are more comfortable with R, please feel free to use it (however, the demos in class will only involve Python). Part B involves building and testing classification models to predict salaries from the text contained in the job descriptions. The data for this assignment can be found at http://www.kaggle.com/c/job-salary-prediction

**Randomly select 2500 data points (the training dataset train\_rev1 contains nearly 250k) for ease of analysis. Now perform the tasks in Part A and B.**

**Part A (basic text mining, just to make sure you are familiar with the fundamentals)**

A1. What are the top 5 parts of speech in the job descriptions? How frequently do they appear? How do these numbers change if you exclude stopwords?

*Hint: nltk.org is a great resource for exploring text mining with Python. There are many examples that are similar to the questions in this assignment.*

A2. Does this data support Zipf’s law? Plot the most common 100 words in the data against the theoretical prediction of the law. For this question, do not remove stopwords. Also do not perform stemming or lemmatization.

*Hint: Check* [*http://www.garysieling.com/blog/exploring-zipfs-law-with-python-nltk-scipy-and-matplotlib*](http://www.garysieling.com/blog/exploring-zipfs-law-with-python-nltk-scipy-and-matplotlib)

A3. If we remove stopwords and lemmatize the data, what are the 10 most common words? What are their frequencies?

**Part B (predict salary from job description; the idea here is to test the predictive power of text and compare it with that of numeric data)**

In this section, you will create classification models to predict *high* (75th percentile and above) or *low* (below 75th percentile) salary from the text contained in the job descriptions.

B1. Ignore the job descriptions, and train a model to predict high/low salary from all the numeric columns, e.g., part time/full time, contract vs. others, type of job (a lot of dummy variables), location (instead of using a huge number of dummy variables, you can use a list of cities in England with highest cost of living, and create a 0/1 variable which is 1 if the job is in one of those cities, else 0). Use the Naïve Bayes classifier. What is the accuracy of your model?

Now build a classification model with text (full job description) as the only predictor. There are two versions of Naïve Bayes – binomial and multinomial; for the moment, let’s not bother about the theoretical underpinnings! You can use either one for this assignment. For all models, show the *confusion matrix*.

*Hint: For part B, check out* [*http://www.nltk.org/book/ch06.html*](http://www.nltk.org/book/ch06.html) *(esp 1.3) for illustrations.*

*Also look at* [*http://nbviewer.ipython.org/gist/rjweiss/7158866*](http://nbviewer.ipython.org/gist/rjweiss/7158866) *(you may have to download additional Python libraries than the ones I have mentioned before)*

B2. Create a classification model with *all* words and the bag-of-words approach. How accurate is the model (show the confusion matrix)?

**Also show the top 10 words (excluding stopwords) that are most indicative of (i) high salary, and (ii) low salary.**

B3. Train a “hybrid” model to predict high/low salary using both numeric and text data. Show the accuracy of the model.

**Which model – numeric only, text only and hybrid – provided the highest accuracy in predicting high/low salary? Did the result surprise you? Why or why not?**

**Deliverables**

The deliverable for this assignment is a file with the python scripts, outputs (including plots & tables where applicable) and your answers to the various questions. Do not forget to write the names of all group members.