**CS 4375 Fall 2019**

**Introduction to Machine Learning**

**Assignment #3**

**Due Date: Nov 5, 2019**

**Section A**

**Problem Statement:**

1. Given the following web pages and how they are linked in between, give an order of collection by running a web-crawler. This question should be solved by hand.

A

C

E

D

B

F

***Note*:**

1. An arrow from X → Y means X has a link to Y
2. Consider A as the seed page
3. To break any tie, use lexicographic order.
4. Given the following corpus of 3 documents, compute TF-IDF for each of the words. Apply stemming to improve the representation. You can use the following link provided to solve this question.

Machine learning is a key area of focus by computer science researchers.

Deep learning is and advanced method of learning that involves neural network based computation.

Any involvance in scientific research will help to progress in life.

***Note:***

**Use PorterStemmer algorithm for stemming, remove stopwords where applicable (To remove the stopwords, you can use the online source** [**https://demos.datasciencedojo.com/demo/stopwords/**](https://demos.datasciencedojo.com/demo/stopwords/)**). (To do the stemming go here,** [**https://text-processing.com/demo/stem/**](https://text-processing.com/demo/stem/)**)**

***Answers Expected:***

1. Vocabulary size before TF-IDF computation.
2. 2 words with highest TF-IDF score, 2 words with lowest TF-IDF score per document

**Section B**

Answer Question 4.1, 4.2, 4.7 from Tom Mitchell’s Machine Learning book.

**Section C**

Please solve the following problem by coding Python programs:

***Data:***

For this analysis we’ll be using a dataset of 50,000 movie reviews taken from IMDb. IMDb lets users rate movies on a scale from 1 to 10. To label these reviews the curator of the data labeled anything with ≤ 4 stars as negative and anything with ≥ 7 stars as positive. Reviews with 5 or 6 stars were left out.

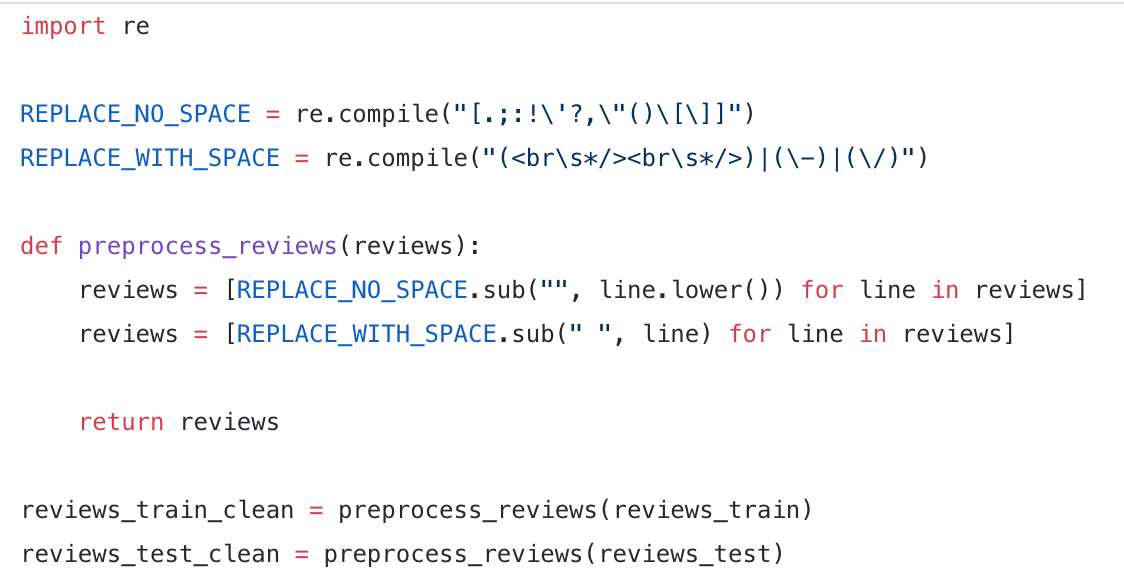
The data is split evenly with 25k reviews intended for training and 25k for testing your classifier, where the first 12.5k are positive and the last 12.5k are negative in each set.

***Problem Statement:***

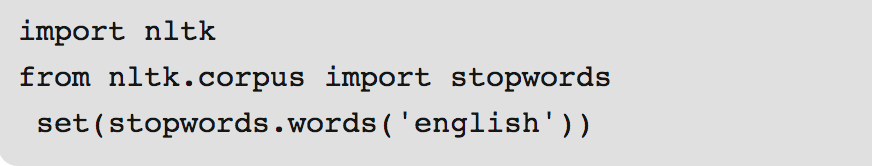
1. Clean and **preprocess**: remove any special characters with space, remove the stop-words.
2. Convert each review to a numeric representation, which is called **vectorization.**
3. Now that we’ve transformed our dataset into a format suitable for modeling, we can start building a **classifier**.

***Task 1:***

You can use regular expression operations to clean the data like this:



And you can use package NLTK to find out the common stop words:



***Task 2:***

The simplest form of this is to transform each review into one row containing 0s and 1s, where 1 means that the word in the corpus (where the corpus is all 50k reviews in our case) corresponding to that column appears in that review. (you can also use other vectorization methods such as bag of words and tf-idf).



***Task 3:***

Apply **K-Nearest Neighbo**r and **SVM** on the training dataset and predict on the test dataset, report the testing accuracy and plot the ROC curve for each method.

***Deliverables:***

1. Python source codes.