Term Assignment (1st notice)

Work on one of the following problems by coding and executing adequate Python programs.
Record the entire workflow (data collection, preprocessing, and analysis) on Jupyter Notebook. Also attach brief introductory and concluding notes to the same notebook.
Finally submit this notebook to the upload site (which will open soon) at ITC-LMS by January 26th, 2018 at the latest (this is a serious deadline!).
If the submission via ITC-LMS doesn't work, send the notebook to my email address: sakamoto@hsp.c.u-tokyo.ac.jp.

(1) Make a bot (web crawler) that is able to collect data from at least 10,000 websites. Using this bot and starting from some webpage, crawl over the Web, gather 10,000 URLs along the way, and derive the distribution of the domain names of these URLs at the level of your choice (for example, you can choose country-code top-level domains such as '.jp' and '.ch'). It is even better to repeat the scraping process from several different starting points and compare the resultant domain-name distributions.

(2) Collect at least 5,000 Wikipedia articles by using either web scraping techniques or API. Clean each of these articles to obtain only the main body of the text (probably in the form of a bag-of-words), removing all the other elements. Apply some unsupervised machine learning algorithm such as clustering and topic models to automatically classify these documents. Examine the results and adequately interpret the classification you have obtained. Repeat the process while changing main training parameters (e.g., the number of clusters, topics…) if necessary.

(3) Collect at least 300 news feeds for each of the news categories you choose (e.g., 'domestic' vs 'international', 'culture' vs 'sports' vs 'others' …). Clean each of these feeds to obtain only the main body of the text (probably in the form of a bag-of-words), removing all the other elements. Using this data, train some supervised machine learning algorithms such as Naive Bayes, support vector machine (SVM) and decision tree to obtain a classifier that can accurately classify news feeds into these categories. Evaluate the performance of each classifier in the framework of cross-validation (division of data into a training dataset and a test dataset). It is even better to apply the best classifier you get to larger documents such as news articles to see its performance beyond the original sample of feeds.