Project 2: Author Recognition

Naïve Bayes has been successfully applied to a variety of document classification problems: author recognition, spam filtering, sentiment analysis, genre identification, bias detection, etc. For this project I opted for author recognition (also known as stylometry), but you are free to choose a different application if you prefer. Stylometry has been used to identify authors of anonymous or disputed historical documents, modern authors penning works under pseudonyms, and manifesto-publishing terrorists like the Unabomber.

Dataset:

The Federalist Papers are a collection of 85 essays published in several New York newspapers during the 1780's arguing on behalf of ratifying the newly written Constitution. The papers were written by three of America's founding fathers – John Jay, James Madison and Alexander Hamilton – under the pseudonym *Publius*. Many years after the publishing of the Federalist Papers and the eventual ratification of the Constitution, notes were separately found among Hamilton and Madison's personal effects identifying who wrote the different essays. The two separate lists agreed upon the authors of 73 of the essays, but differed as to who authored the remaining 12.

Scholars from various disciplines have studied the 12 disputed Federalist Papers and the general consensus is that they were likely written by James Madison. However, some still argue on behalf of Hamilton as the author. (Jay is considered unlikely because he fell seriously ill during the time-period when these specific essays were published.)

The Federalist Papers are available from Project Gutenberg and I have separated them into four separate subfolders: Hamilton, Jay, Madison and Disputed. You are going to write a Naïve Bayes classifier to detect who wrote the 12 disputed essays. You can use the undisputed essays as training data to help you identify the author(s) of the disputed ones – it is up to you whether you wish to include John Jay's essays in your analysis or whether you wish to exclusively consider James Madison and Alexander Hamilton. You should use some of the undisputed papers as a validation set to confirm the accuracy of your model before you apply it to the disputed papers.

NLP Tools:

There are several natural language processing (NLP) techniques that might be helpful to you in your project and you are free to use them if you wish:

- *Stop Words:* These are lists of common words like 'a', 'the', 'me' that appear frequently in most texts and don't have much meaning, in contrast with words like 'constitution', 'tennis' or 'eating'. In applications like spam filtering or genre identification, it is typical to ignore or remove stop words from the training and testing texts. The idea being that stop words are highly frequent, but not very informative for those problems, so it is better to reduce the added computational cost of considering them. On a similar basis, punctuation is frequently ignored or removed in such applications. In applications like stylometry, stopwords and punctuation can sometimes be more helpful than content words because they are used more consistently across different topics. For example, in identifying JK Rowling as the author of the crime novels written under the pseudonym Robert Galbraith, stop words were much more useful than content words like 'muggle' or 'wand' from Rowling's earlier works.

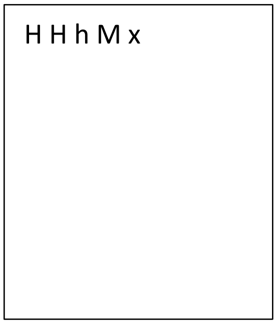
- *Stemming:* It can be useful in some applications to treat words with the same root, e.g. ‘educate’ ‘education’ ‘educator’ ‘educators’, as identical. Stemming is commonly used for document retrieval (i.e. search engines) to return documents that relate to the keywords provided by the user even if they don’t contain the exact words sought. It can be similarly useful in applications like genre identification and sentiment analysis.
- *N-Grams*: While we naturally tend to think about analyzing documents on a word-by-word basis, because that is how we read them, sometimes combinations of words or combinations of letters can be more informative. Models based on analyzing the frequencies of individual words or individual letters are considered 1-gram models. Models that analyze pairs of words or pairs of letters are 2-gram models, etc.

You are free to use libraries like Python’s NLTK library to help you with NLP tasks. In your readme document, you should provide a brief description of what each tool does and roughly how it works.

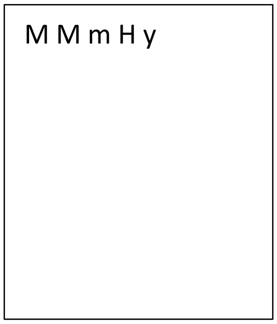
Model: You are going to build a Bayesian learning system to perform a document classification task on a real-world dataset. The version of the assignment I described here is a Naïve Bayes model to perform author recognition on the Federalist Paper dataset. If you wish to use a different dataset, or perform a different document classification task or implement a different Bayesian algorithm, you are free to do so. Even within the task of identifying the author(s) of the Federalist Papers, you have a variety of choices of how to apply your Bayesian analysis and you are encouraged to play around with different options to find one that performs well.

Given the huge vocabulary of potential words available in the English language and the relatively modest size of your training set, you may want to consider applying smoothing and/or UNK estimation when building your model. You will very likely also need to use log-probabilities.

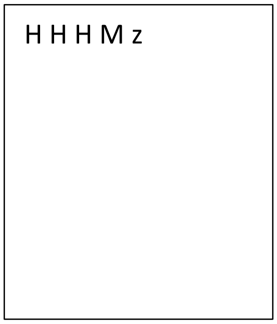
As before, I **strongly** recommend that you create a set of debugging data to work with when initially developing your model. I started out with the following debugging dataset:


H H h M x

Hamilton.txt


M M m H y

Madison.txt


H H H M z

Disputed.txt