**HLT – Fall 2020**

**Homework 9: Authorship Attribution of the Federalist Papers**

Worth 100 points

**Objective**: Gain experience in text classification using machine learning with Python and sklearn. This is a multi-class classification problem. All sklearn classifiers do multiclass classification with no special parameters required. The do multi-class classification by decomposing the problem into binary classification problems. Read more in the docs.

**Turn in:** Your Jupyter notebook or Google Colab notebook, printed to pdf. In either platform, go to File -> Print Preview and print the finished notebook to pdf. Upload the pdf, not the notebook.

**Background**: *The Federalist Papers* is a collection of documents written by Alexander Hamilton, James Madison, and John Jay collectively under the pseudonym Publius. These documents were written to persuade voters to ratify the US Constitution. These documents continue to be influential to this day, as they are frequently cited in Federal court rulings, as well as law blogs, and political opinions.

**Overview**: The data set used in this homework (in Piazza) is a collection of Federalist Papers from Project Gutenberg. There are 83 documents in this data set which has two columns: one for the author(s), and one for the text of the document.

The NLP task of *authorship attribution* is the attempt to identify the author of a document, given samples of authors' work. In this data set, the breakdown by author is as follows:

- Alexander Hamilton 49
- James Madison 15
- John Jay 5

There are several documents for which authorship is in dispute by historians:

- Hamilton or Madison 11
- Hamilton and Madison 3

**Instructions**: Create a Jupyter notebook (or use Google Colab) for this homework solution.

1. Read in the csv file using pandas. Convert the author column to categorical data. Display the first few rows. Display the counts by author.
2. Divide into train and test, with 80% in train. Use random state 1234. Display the shape of train and test.
3. Process the text by removing stop words and performing tf-idf vectorization, fit to the training data only, and applied to train and test. Output the training set shape and the test set shape.
4. Try a Bernoulli Naïve Bayes model. What is your accuracy on the test set?
5. The results from step 4 will be disappointing. The classifier just guessed the predominant class, Hamilton, every time. Looking at the train data shape above, there are 7876 unique words in the vocabulary. This may be too much, and many of those words may not be helpful. Redo the vectorization with max_features option set to use only the 1000 most frequent words. In addition to the words, add bigrams as a feature. Try Naïve Bayes again on the new train/test vectors and compare your results.
6. Try logistic regression. Adjust at least one parameter in the LogisticRegression() model to see if you can improve results over having no parameters. What are your results?
7. Try a neural network. Try different topologies until you get good results. What is your final accuracy?

Grading Rubric:

| Element | Points |
|---|---|
| Step 1 | 10 |
| Step 2 | 10 |
| Step 3 | 20 |
| Step 4 | 10 |
| Step 5 | 20 |
| Step 6 | 10 |
| Step 7 | 20 |
| Total | 100 |