Stats 315B Final Project Report

Arthur Chen
Department of Statistics
Stanford University
Stanford, CA 94305
aulchen@stanford.edu

Abstract

Of the eighty-five anonymously published Federalist Papers, twelve of them have their authorship status disputed between James Madison and Alexander Hamilton. This presentation attempts to use machine learning techniques to determine the author of these papers. I find that it is likely that Madison wrote most of the disputed papers, but the exact details depend strongly on modeling choices.

1 Introduction

In 1787 and 1788, as New York considered whether to ratify the proposed United States constitution, 77 opinion pieces arguing for the new constitution appeared in New York newspapers, under the pseudonym 'Publius.' Later published in book form with an additional 8 essays, bringing the total up to 85, the *Federalist Papers* have remained an inspiration for American political thought ever since.

Eventually, it was revealed that the Federalist Papers were a collaboration between Alexander Hamilton, James Madison, and John Jay. Hamilton and Madison eventually released lists indicating which papers were written by which author. The two lists agreed on 73 of the papers, but disagreed on the authorship of 12, which Hamilton and Madison both claimed to have written - Nos. 49-58, and 62-63 (Table 1).

This project attempts to determine, for each of the twelve disputed papers, whether Madison or Hamilton wrote it. As there are only two authors, this is a binary classification problem. Unlike most other text classification problems, which try to extract content words and remove function words, past treatments of this problem have focused on function words over content words [Mosteller and Wallace, 1963].

Table 1: Authorship of the *Federalist Papers*. Reproduced from [Holmes and Forsyth, 1995]

Paper Number	Author
1	Hamilton
2–5	Jay
6–9	Hamilton
10	Madison
11–13	Hamilton
14	Madison
15–17	Hamilton
18–20	Joint: Hamilton and Madison
21–36	Hamilton
37–48	Madison
49–58	**Disputed**
59–61	Hamilton
62–63	**Disputed**
64	Jay
65–85	Hamilton

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

2 Data

The data consists of the 80 Federalist Papers that were either written by Hamilton, Madison, the two jointly, or disputed between the two, as indicated in Table 1. The versions used were published by Project Gutenberg [Hamilton et al., 1788]. I make the simplifying assumption that jointly written papers, No. 18-20, were solely written by Madison. Of the 190,156 words in the corpus, 111,779 were written by Hamilton, 43,775 were written by Madison, and 24,313 are in dispute.

The total vocabulary size, i.e. the number of unique words in the corpus, is 8,535 words. The size of Hamilton's vocabulary is 6,965 words, while the size of Madison's vocabulary is 4,710 words. The size of Hamilton and Madison's common vocabulary, i.e. words that both authors use, is 3,583 words.

The papers have a mean word length of 2,237 words with a standard deviation of 758 words. The shortest paper is No. 13 at 968 words, and the longest paper being No. 83 at 5,838 words.

Each paper as listed on Project Gutenberg has four components: a title, a header, the text of the paper, and references. The title is of the form "`FEDERALIST No.  `" with the paper number following the dot. The header contains a descriptive title, the original newspaper and date of publication, and the editors' attribution of the paper to Madison, Hamilton, or Jay. The text of every paper begins with the salutation `To the People of the State of New York:` and ends with the a pseudonymous sign-off, "`PUBLIUS`." The references consist of bulleted footnotes from the main paper.

3 Pre-Processing and Feature Extraction

I downloaded the text using the `urllib` library. To process text, I used the Natural Language Toolkit package in Python [Bird et al., 2009] to convert all of the text to lowercase and strip all of the punctuation and diacritic marks. The main feature I used is the word frequencies for each paper. Word frequencies uses a bag-of-words model to measure the frequency of every word in a given text [Kowsari et al., 2019]. The number of features equals the total vocabulary size of the corpus - in this case, 8,535 words. For example, the following two texts:

`I like Eggs.  I eat Ham.`

`I like ham.  Eggs sleep.`

would be converted into the collections of words

`{'I', 'like', 'eggs', 'I', 'eat', 'ham'}`

`{'I', 'like', 'ham', 'eggs', 'sleep'}`

which then are represented as the frequencies in Table 2.

Table 2: Sample feature data frame

Paper	'I'	'like'	'eggs'	'eat'	'ham'	'sleep'
Sentence1	1/3	1/6	1/6	1/6	1/6	0
Sentence2	1/5	1/5	1/5	0	1/5	1/5

4 Word Frequencies

Over the entire corpus, the most common words are listed in Table 3. As a dimensionality reduction technique, we rank all words in common between the two authors by the difference in the respective authors' usage frequency. Words that Madison uses more frequently than Hamilton are pro-Madison words, while the reverse are pro-Hamilton words.

We look at two measures of frequency differences: absolute and relative. The absolute usage difference looks at the words with the largest absolute usage difference between Madison and Hamilton, while the relative usage difference looks at the words with the largest percentage difference in usages. The top ten absolute pro-Madison and pro-Hamilton words, in order of decreasing usage difference, are listed in Table 4, while the top ten relative pro-Madison and Hamilton words are as follows: `'whilst', 'composing', 'achaeans', 'sphere', 'diet', 'residence',`

Table 3: Top ten most frequent words in the corpus

Word	Frequency / 1,000 Words
the	94.11
of	63.42
to	39.03
and	25.98
in	23.59
a	21.42
be	29.61
that	14.65
it	13.32
is	11.04

Table 4: Top ten pro-Madison and pro-Hamilton words, absolute frequency

Pro-Madison Words			Pro-Hamilton Words		
Word	Madison Freq. / 1,000	Hamilton Freq. / 1,000	Word	Hamilton Freq. / 1,000	Madison Freq. / 1,000
the	100.88	92.38	to	40.62	31.82
and	30.99	24.05	of	64.79	60.13
by	11.82	7.56	would	8.18	3.82
on	7.53	3.39	in	24.97	20.85
powers	3.21	0.88	upon	3.31	0.20
states	5.73	3.70	a	22.13	19.33
government	5.35	3.70	be	20.33	17.55
was	2.81	1.37	there	3.36	0.83
federal	2.63	1.21	that	15.18	13.15
their	6.47	5.12	this	8.15	6.43

`'oath'`, `'sovereigns'`, `'recommend'`, `'respectively'` are pro-Madison, while, `'kind'`, `'upon'`, `'enough'`, `'afford'`, `'exclusion'`, `'think'`, `'jury'`, `'massachusetts'`, `'matter'`, `'courts'` are pro-Hamilton.

Note that the top eight most frequent words in Table 3 also appear in Table 4. In addition, of the top ten pro-Hamilton and Madison words by relative usage, only 'upon' appears in either Table 3 or 4.

5 Methods

Train-Test Split Of the 68 papers available for training and evaluating the model, 51 were written by Hamilton and 17 were written by Madison, for an approximately three-to-one class imbalance. I held out 16 papers for a roughly 75-25 training-test split. Of the 16 test papers, 12 (Nos. 13, 16, 21, 32, 35, 59, 61, 67, 76, 80, 84) were randomly selected from Hamilton's papers, while the other 4 (Nos. 20, 39, 44, 46) were randomly selected from Madison's papers. Thus, the class balance among the test data matches the training data.

Metrics All models were implemented with the scikit-learn package in Python [Pedregosa et al., 2011]. All models had their quality assessed using two metrics: 10-fold cross-validation on the training set, and prediction accuracy on the held-out test set. All models that required hyperparameter tuning were tuned with by using grid search and choosing the model with the highest 10-fold cross-validation accuracy on the training data. All models were initialized with the random state of 500.

The methods can be broken down into three categories: logistic regression, dimensionality reduction, and trees.

Logistic Regression In logistic regression, the conditional probability of a point with features X_i belonging to the positive class (in this case, Madison writing the paper) is modeled as

$$\hat{p}(X_i) = \frac{1}{1 + \exp(-\omega^T X_i)}$$

where X_i are the features of observation i, and the following loss function is optimized via gradient descent:

$$\min_{\omega} C \sum_{i=1}^{n} \left(-y_i \log(\hat{p}(X_i)) - (1 - y_i) \log(1 - \hat{p}(X_i))\right) + r(\omega)$$

In the unregularized case, $r(\omega) = 0$. In the L1 regularized case, $r(\omega) = ||\omega||_1$. For the unregularized case, there are no hyperparameters to tune. For the L1 regularized case, I tuned over C, the strength of the regularization. Note that as opposed to most textbook treatments of regularization, smaller values of C imply stronger regularization.

All models were trained with all 8,535 features. For 'balanced' regression, two changes were made. First, observations corresponding to Madison's papers were given a weight of 3 to make the classes in the training set balanced, as in the training set, Hamilton wrote three times more papers (39) than Madison (13). Second, the intercept term in the regression was fixed to zero, representing fixing a 50-50 prior on whether the model believes Hamilton or Madison wrote a give paper before seeing any features.

Dimensionality Reduction In these methods, I used the vanilla logistic regression, but instead of using regularization to automatically reduce the number of features, I hand-picked features that looked like they might be discriminatory. I also put a regression I trained on the first two principal components in this category, for lack of better category to put it in.

All models used unregularized logistic regression, with the only differences being the features used to train the models. "Length" uses the word count of each paper as the only feature. The single-word models "the" and "upon" only use the frequencies of the word mentioned, and the "upon, there" model only uses the frequencies of the two words mentioned. "10 Most Freq" uses the ten words listed in Table 3, while "20 Abs. Freq. Diff" uses the twenty words listed in Table 4. "PCA" uses the first two principal components of the data covariance matrix.

Decision Trees For all tree-based methods, the loss function is the Gini index. In the binary classification case, the Gini index simplifies as follows. For a given observation i, let $\hat{p}_i$ be the empirical probability that i is in the positive class, which in this case is the proportion of observations in observation i's leaf that are in the positive class. Then the loss of observation i is $G_i = 2\hat{p}_i(1-\hat{p}_i)$, and the cost of a tree is the sum of the Gini indexes of all of the observations [Hastie et al., 2009], [Murphy, 2022]. At each split, we use the split that leads to the lowest cost tree. For decision trees, the hyperparameters I tuned over were the maximum depth of the tree, and the minimum number of observations per leaf. I did not use pruning.

Random Forests Random forests is an ensemble estimator that aggregates multiple decision trees. For all variations, I used the default 100 trees. The default random forests used all features and all data with no bootstrapping. The only randomness in the default method comes from sklearn randomly permuting the order of the features before training the classifier. Bootstrap variants generate bootstrap samples of the test data before training the classifiers. Square root max features (Sqrt.) randomly selects $\sqrt{8535} \approx 92$ of the inputted features to train the classifiers.

AdaBoost AdaBoost is a forward stagewise additive model that trains an ensemble of weak classifiers, whose prediction is a weighted sum of the predictions from the individual classifiers [Freund and Schapire, 1997], [Hastie et al., 2009]. In this case, the weak learners are decision stumps with depth one. After each iteration, misclassified points are reweighted to have higher weight and correctly classified points are reweighted to have lower weight, so that later learners focus on misclassified points.

The two hyperparameters I tuned are the number of estimators in the ensemble, and the learning rate. The learning rate is a parameter v such that if classifier $h_m(x)$ normally has weight α_m in the ensemble, it instead has weight αv in the ensemble.

Table 5: Results of dimension reduction methods

Model	Length	"The"	"Upon"	"Upon", "There"	10 Most Freq	20 Abs. Freq. Diff	20 Rel. Freq. Diff	PCA
10-Fold Acc.	71.33	79.33	96.00	98.00	90.33	94.33	98.00	90.33
Test Acc.	75.00	75.00	100.00	100.00	94.00	100.00	100.00	100.00
49	17.01	39.31	99.93	99.92	100.00	95.74	100.00	46.29
50	12.78	20.65	75.19	87.88	100.00	100.00	100.00	97.95
51	19.48	35.97	99.93	99.82	100.00	100.00	100.00	46.27
52	18.83	30.21	99.93	99.97	0.00	100.00	100.00	29.80
53	22.00	19.23	99.93	99.94	100.00	100.00	100.00	62.87
54	20.32	31.74	61.24	70.60	0.00	100.00	100.00	48.94
55	20.69	18.93	99.93	99.76	0.00	0.00	100.00	6.92
56	17.00	17.43	99.93	99.76	0.00	95.71	100.00	53.54
57	22.50	26.96	99.93	99.86	0.00	100.00	100.00	63.33
58	21.23	31.70	99.93	99.94	100.00	100.00	100.00	39.92
62	24.41	12.69	99.93	99.97	0.00	98.82	100.00	11.52
63	32.59	24.73	99.93	99.71	100.00	100.00	100.00	70.66

Table 6: Results of tree-based methods

Model	Decision Tree	RF, Sqrt	RF, Sqrt, Bootstrap	RF	RF, Bootstrap	AdaBoost
10-Fold Acc.	94.00	77.33	75.33	94	94	94.33
Test Acc.	100	100	100.00	100.00	100.00	100.00
49	92.86	22	27	100	30	100
50	92.86	23	27	100	30	100
51	92.86	25	27	47	27	100
52	92.86	25	28	100	39	100
53	92.86	21	21	100	31	100
54	92.86	20	17	100	27	99.55
55	92.86	24	18	47	23	100
56	92.86	26	27	47	21	0.58
57	92.86	21	21	47	28	100
58	92.86	20	28	100	37	100
62	92.86	28	32	100	39	100
63	92.86	29	33	47	30	100

6 Results

The results are listed in Tables 5, 6, and 7. '10-Fold Acc.' and 'Test Acc.' are the accuracy using 10-fold cross validation on the train set, and accuracy on the test set, respectively. The rows below indicate the predicted probabilities of Madison writing the given paper for each model, with green representing Madison and blue representing Hamilton.

7 Discussion

Cross-Validation and Test Accuracies Statistically, classifying the papers was easy, as I got 10-fold classification accuracies in the mid 90's and perfect test accuracies for nearly every model. However, the classification of the disputed papers depended on the exact modeling and feature selection choices.

This result can be explained by looking at the projection of the data down to the first two principal components, displayed in Figure 1. Figure 1a uses the same colors as Tables 5-7: green is Madison, blue is Hamilton, and red is the disputed papers. We can see that even with just two principal components, the known Madison and Hamilton papers separate. However, the disputed papers are intermingled between the known Madison and Hamilton papers, and so their classification depends on the exact nature of the decision boundary.

Similarly, the 100% test accuracy can be explained with reference to Figure 1b. The orange papers are the test papers, the blue the train papers, and the green the disputed papers. The squares represent

Table 7: Results of logistic regression-based methods

Model	Vanilla	Vanilla, Balanced	L1	L1, Balanced
10-Fold Acc.	94.33	94.33	98	100
Test Acc.	100	100	100	100
49	49.39	98.9	52.96	91.71
50	100	99.99	0.76	64.99
51	100	100	99.98	99.99
52	99.23	99.07	88.58	99.75
53	100	99.58	99.81	99.94
54	100	99.97	99.82	99.99
55	0	0.59	0.57	0.93
56	86.6	57.97	95.97	97.57
57	94.79	95.93	97.56	99.88
58	99.99	99.81	99.08	99.71
62	72.29	67.38	99.94	100
63	100	99.95	96.25	99.66

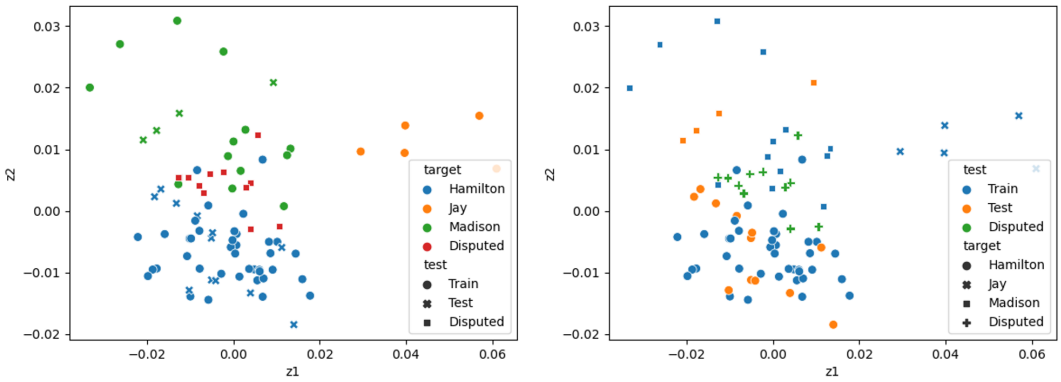

Figure 1: Projection of data onto two principal components. Left is colored by author, right is colored by train/test.

Madison's papers, while the circles represent Hamilton's papers. We see that by random chance, the test papers chosen split very well, even in two principal components. Thus the test set was small enough that it can be split by nearly any reasonable method.

Classifying the Disputed Papers Most reasonable methods that don't use the bootstrap (e.g. get more than 92% 10-fold cross-validation accuracy) agree that Madison very likely wrote most of the disputed papers. However, the details differ by method used. The dimension reduction methods with the highest 10-fold accuracy agree that Madison wrote all of the papers, while logistic-regression based methods and the '20 Abs. Freq. Diff' dimension reduction method agree that Hamilton probably wrote No. 55. Meanwhile, random forests gives Hamilton basically a 50-50 chance of writing five papers, while AdaBoost thinks that Hamilton wrote No. 56.

Interestingly, bootstrap random forest methods think that Hamilton wrote all of the papers, a result that I can not satisfactorily explain. Maybe this is due to some bootstrap samples containing so few Madison papers that those classifiers just predict everything as Hamilton, which influences the ensemble to vote 'Hamilton' for the disputed papers. However, this is little more than speculation.

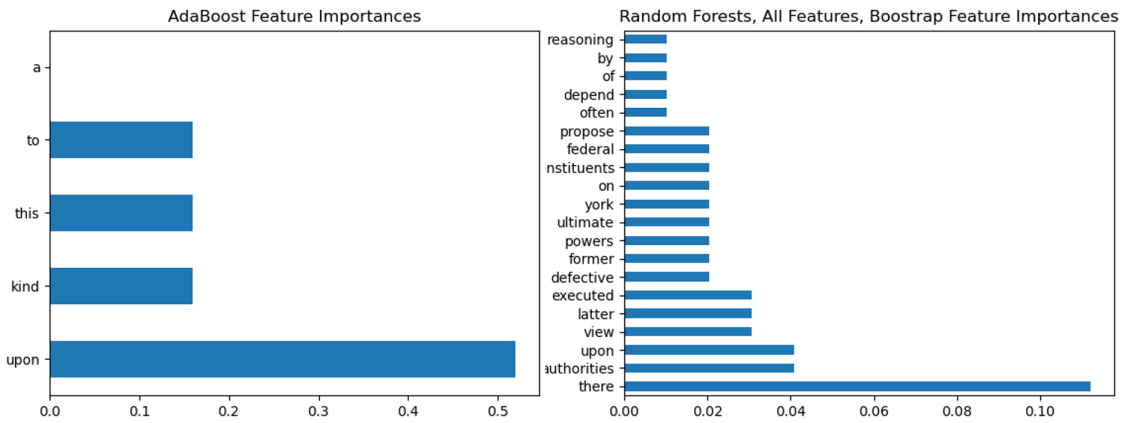

Figure 2: Top feature importances in AdaBoost (left) and random forests with bootstrap samples (right). Feature importances are the sum of the reduction in Gini indices over each classifier.

Decision Trees, Non-Bootstrap Random Forests, and AdaBoost After tuning the hyperparameters, the optimal decision tree turned out to be a decision stump splitting on the frequency of 'upon.' Papers with a usage of 'upon' greater than 1 word per 1,000 were predicted to be Hamilton, and going deeper in the tree didn't result in any improvements in the 10-fold cross-validation metric. Similarly, looking at feature importances, random forests without bootstrapping only splits on two features with roughly equal weight: 'there' and 'upon.'

These results are understandable in the context of Mosteller and Wallace [1963], who found that 'upon' was the most discriminatory single word among Hamilton and Madison's writings. This also makes sense in reference to Table 4. The ratio between Hamilton's usage of 'upon' and Madison's usage is roughly sixteen to one, and Hamilton uses 'upon' somewhat frequently. For random forests feature importances, Hamilton uses 'there' roughly four times more frequently than Madison does. Thus this combination of high relative usage and reasonable absolute usage makes 'upon' and 'there' good discriminatory words.

AdaBoost is a similar story as seen in Figure 2a, splitting on primarily on 'upon.' 'This' and 'to' are pro-Hamilton words found in Table 4. 'Kind' is the word with the single most pro-Hamilton word that both Madison and Hamilton use, when measured by usage ratios. Hamilton's usage of 'kind' is .690 words per thousand, while Madison's is .022 per thousand - Hamilton uses 'kind' roughly thirty-times more frequently than Madison. Thus AdaBoost's features are explainable as various pro-Hamilton words.

Bootstrapping The feature importances from using bootstrapped samples in random forests (Figure 2b) are not quite as explainable. 'There' and 'upon' are pro-Hamilton words in Table 4, but the rest of the features can not be found in Tables 3 and 4. I suspect that these feature importances come from the bootstrap artificially inflating the frequencies of rare features. If a paper that contains rare features is randomly selected to enter a bootstrap sample multiple times, then the model might consider that feature discriminative. Due to the high number of features, it's likely that any given paper contains rare features that can have their importance artificially inflated.

Dimensionality Reduction Surprisingly few features are required to get good results in classifying the papers, provided those features are well selected. This is most obviously borne out by the "upon" and "upon, there" models, which manage to get 96% and 98% 10-fold cross-validation accuracy and perfect test accuracy off of one and two features, respectively. 'Upon' and 'there' are both words in the "20 Abs. Freq. Diff" feature set; increasing the number of features from two to twenty slightly decreases the model performance.

Most of the features likely contain little discriminatory information. As seen in the 'Sqrt' random forest models, accuracy suffers when random forests can't split on all of the features, indicating that it's splitting on low information features.

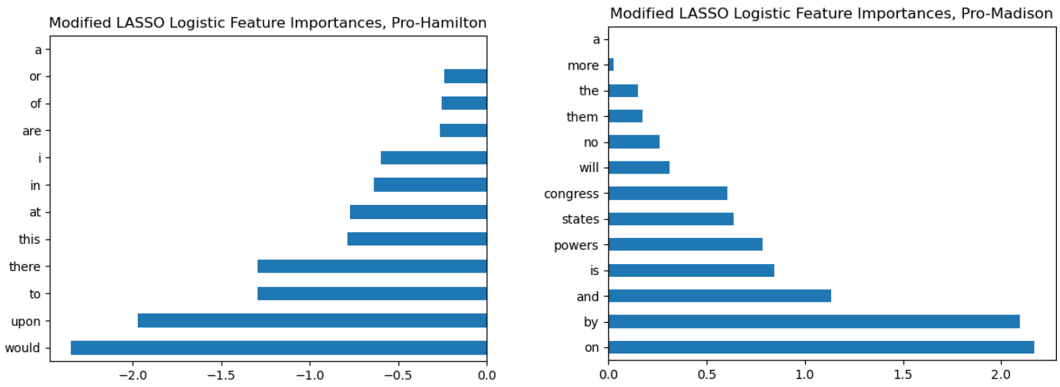

Figure 3: Top twenty most important features in the balanced L1 logistic model. Importances are the coefficient for each features times its standard deviation. Left features are pro-Hamilton, right are pro-Madison.

Logistic Methods We see significant improvements in the model by using L1 regularization combined with balancing the classes and setting an uninformative prior. L1 regularized logistic regression selects 46 features, while balanced L1 regularized logistic regression selects 23 features. Regularization likely helps because most features seem to be extraneous to model performance.

I suspect that the performance increase from the 'balanced' method is due to balancing the classes. In the vanilla L1 regularized logistic model, the prior probability of Madison writing a paper is 56.4%, as implied by the intercept term. This is close enough to the 50-50 forced by the 'balanced' model that I don't think the priors are significantly different.

Looking at the feature importances in Figure 3, we see that the first few features in both the pro-Madison and pro-Hamilton lists are in the corresponding lists in Table 4, which is reassuring.

8 Conclusion and Next Steps

I found that according to the models that don't use the bootstrap, Madison likely wrote most of the disputed papers. Bootstrap random forest methods predict that Hamilton wrote all of the papers, but at the moment I consider this result an anomaly. However, the exact details of which papers Hamilton might have written depend significantly on the modeling choices, with several models predicting that Hamilton wrote No. 55.

Most of the best models used some sort of feature selection, and ended up choosing functional words over content words. Due to the relative ease of the problem, it is difficult to distinguish between different reasonable models and pick a 'best' model, as the high cross-validation accuracies and small sample sizes mean that the difference between model metrics might just due to chance.

The immediate next step on technical grounds would be to see whether the result that bootstrap random forests predict that Hamilton wrote all of the papers extends to other models. If we draw bootstrap samples and create an ensemble of L1 regularized logistic learners, will we predict Hamilton wrote all of the papers?

A major extension of this project would be trying to expand the training set. The problem as formulated here has $p >> N$, where the dimension of the features (8,535) is much higher than the number of datapoints (68). Expanding the corpus with additional writings by Hamilton and Madison might be productive, provided that the additional writings can be shown to be similar to the Federalist Papers themselves.

A speculative extension would be to classify not at the level of paper, but at the level of sentences. This would allow the various part of speech tagging and sentence diagramming features in the Natural Language Toolkit to be used, and could try to identify which parts of the jointly written papers, No. 18-20, were written by which author.

References

Frederick Mosteller and David L. Wallace. Inference in an authorship problem: A comparative study of the discrimination methods applied to the authorship of the disputed federalist papers. *Journal of the American Statistical Association*, 58(302):275–309, 1963. URL https://www.jstor.org/stable/2283270.

David I. Holmes and R.S. Forsyth. The federalist revisited: New directions in authorship attribution. *Literary and Linguistic Computing*, 10(2), January 1995. URL https://academic.oup.com/dsh/article-abstract/10/2/111/956290.

Alexander Hamilton, James Madison, and John Jay. The federalist papers, 1788. URL https://www.gutenberg.org/files/1404/1404-h/1404-h.htm.

Steven Bird, Edward Loper, and Ewan Klein. *Natural Language Processing with Python*. O'Reilly Media Inc, 2009. URL https://www.nltk.org/.

Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura E. Barnes, and Donald E. Brown. Text classification algorithms: A survey. *Information*, 10(4), 2019. URL https://www.mdpi.com/2078-2489/10/4/150.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2009. ISBN 9780387848570.

Kevin P. Murphy. *Probabilistic Machine Learning: An Introduction*. Adaptive computation and machine learning series. MIT Press, 2022. ISBN 9780262046824.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, August 1997.