ST707 Applied Machine Learning
School of Information Studies
Syracuse University

Hendi Kushta

Section 1: Data preparation

Load Data

The first step in data preparation involves loading the dataset and examining it. It is essential to comprehend the significance of the attributes and the feasible values they can hold before diving into the exploration process. The table below displays this information.

Attribute	Description
Author	Who wrote the essay. (disputed, Hamilton, Madison, Jay, HM)
Filename	.txt files (The Federalist Papers which were a series of eighty-five essays)
Other	70 columns with most frequent used words

Data exploratory and preprocessing

In order to identify the author of the essays we can analyze the characteristics and the writing styles of the authors. This analysis starts by verifying the given dataset, some exploratory analysis is also performed, and finally we analyze through unsupervised learning (with k-means algorithm and hierarchical) the possibility of authorship for the essays.

R code written in RStudio is used to develop the project. When the dataset is imported from the csv file into the project, 85 rows (observations) and 72 columns are seen (attributes). The columns are written essays, while the qualities are made up of "function words." The distribution of these essays among their writers can be seen using the table function:

```
dispt Hamilton        HM       Jay   Madison
   11        51         3         5        15
```

51 writings by Hamilton, 5 by Jay, 15 by Madison, and 3 by Hamilton and Madison are included in this dataset. The 11 essays that are left are the ones whose authorship is under question. The following is a list of all function words:

```
 [1] "author"   "filename" "a"        "all"      "also"     "an"       "and"
 [8] "any"      "are"      "as"       "at"       "be"       "been"     "but"
[15] "by"       "can"      "do"       "down"     "even"     "every"    "for."
[22] "from"     "had"      "has"      "have"     "her"      "his"      "if."
[29] "in."      "into"     "is"       "it"       "its"      "may"      "more"
[36] "must"     "my"       "no"       "not"      "now"      "of"       "on"
[43] "one"      "only"     "or"       "our"      "shall"    "should"   "so"
[50] "some"     "such"     "than"     "that"     "the"      "their"    "then"
[57] "there"    "things"   "this"     "to"       "up"       "upon"     "was"
[64] "were"     "what"     "when"     "which"    "who"      "will"     "with"
[71] "would"    "your"
```

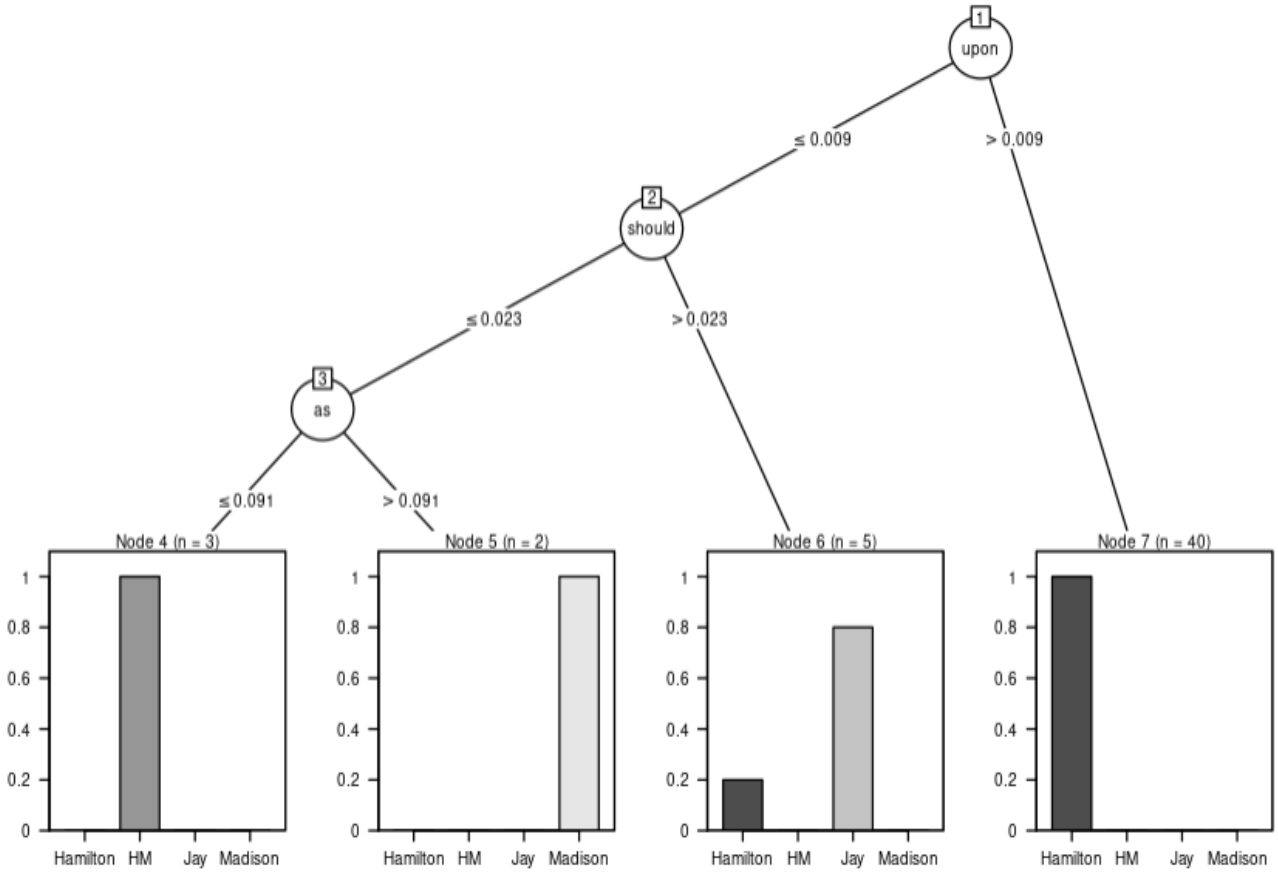
Regarding classification, it is not mandatory to normalize the data. Additionally, it is important to note that the disputed essays will be segregated from the remaining dataset. This means that the disputed

essays will not be included in the model development and training, but will instead be analyzed separately once the model is established.

My next step was to check if there is any missing value or different from numerical and there is not any.

Section 2: Build and tune decision tree models

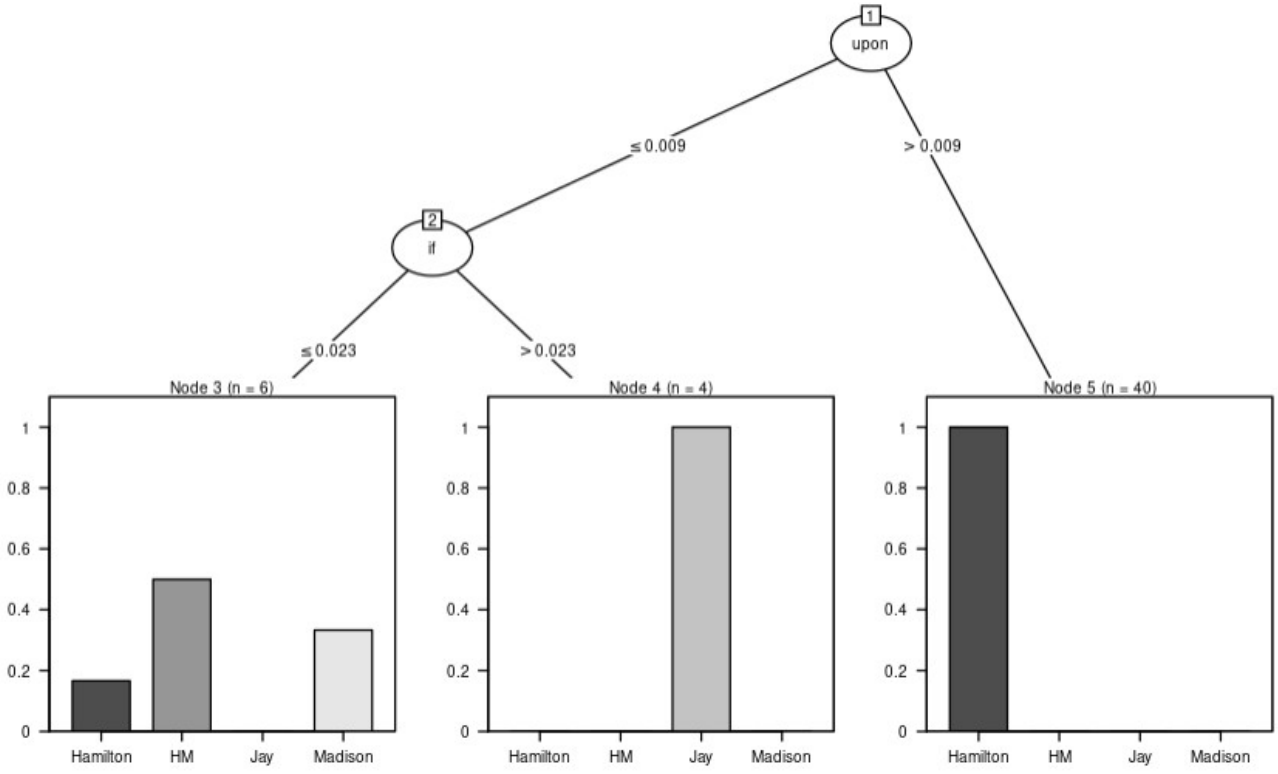
To classify the disputed essays, the J48 Classifier was chosen for its ability to generate a decision tree using the C4.5 algorithm. Since the dataset contains 72 attributes, only the author attribute will be used as the dependent variable, and function words will be fed into the model for training. The filename attribute was not taken into consideration as it does not provide valuable information for classification. Below is the decision tree plot before predicting.



If the word upon is repeated more than 0.9%, the essay is written from Hamilton, else if the word upon is repeated less, we check the second word which is should. If the word should is repeated more than 2.3%, than the essays might be written from Hamilton and Jay, and if the words is repeated less than 2.3%, we check the word as. If as is repeated more than 9.1%, than the essay is written from Madisson, else they are written from HM.

We start by using the J48 function provided by the Weka library to build a decision tree model. The J48 function takes several arguments that control the behavior of the decision tree, such as the confidence threshold for pruning (-C), whether to use reduced error pruning (-R), the minimum number of instances per leaf (-M), and whether to use an unpruned tree (-U).

To evaluate the performance of the model, we use the evaluate_Weka_classifier() function from the Weka library. This function takes a Weka_classifier object that was built using the J48 function, as well as a testing dataset, the number of folds to use in cross-validation, and other optional arguments such as whether to include entropy-based and class statistics. We run several iterations of the J48 function with different values for the minimum number of instances and confidence levels for pruning, and evaluate each model using the evaluate_Weka_classifier() function.
After testing multiple models, we select the best model with an accuracy of 82%, a minimum number of instances equal to 2, and a confidence level of 0.01.
To improve the model, we test other iterations of the J48 function using k-fold cross-validation. We find that the best model has an accuracy of 86%, a minimum number of instances equal to 4, and a confidence level of 0.01. This model has a kappa statistics value of 0.39, which is an improvement over the previous model. We also calculate the root mean squared error for this model, which is a measure of the difference between the predicted and actual values. In this case, the root mean squared error is 0.2, which means that the model is able to predict the class of an instance within 0.2 of the true value on average. Overall, these results suggest that the decision tree model is effective at classifying instances, but there is still room for improvement. Below is the decision tree for the improved model.



Section 3: Prediction

The decision tree model and the clustering algorithms are two different types of machine learning algorithms that can be used for different purposes. The decision tree model is a supervised learning algorithm that predicts the class label of a data point based on its features. In contrast, clustering algorithms are unsupervised learning algorithms that group data points based on their similarity.

In this case, the decision tree model has been trained on a set of essays that are known to be written by Hamilton or Madison. The model has then been applied to a set of disputed essays to predict their authorship. According to the model, the disputed essays are not written by only one author. However, it is not clear why the model has reached this conclusion.

One possible reason for the discrepancy between the decision tree model and the clustering algorithms could be the choice of hyperparameters. The decision tree model has been trained with a specific set of hyperparameters, namely the confidence factor and the minimum number of instances per leaf node. Changing these hyperparameters can lead to different results. Therefore, it is possible that the choice of hyperparameters in the decision tree model has led to different results than the clustering algorithms.

Another possible reason could be the structure of the data. The decision tree model relies on a set of features to predict the authorship of the essays. If the features are not informative or are not representative of the underlying patterns in the data, the model may not perform well. In contrast, clustering algorithms group data points based on their similarity, regardless of their features. Therefore, clustering algorithms may be more robust to variations in the structure of the data.

In conclusion, it is important to carefully evaluate the results obtained from different machine learning algorithms and to consider the strengths and limitations of each algorithm. The choice of algorithm should depend on the specific problem and the characteristics of the data. Furthermore, it is important to explore different hyperparameters and feature sets to ensure that the chosen algorithm performs well for the problem at hand.

Decision Tree Prediction

Filename	Author
dispt_fed_49.txt	HM
dispt_fed_50.txt	Hamilton
dispt_fed_51.txt	HM
dispt_fed_52.txt	Jay
dispt_fed_53.txt	Jay
dispt_fed_54.txt	Jay
dispt_fed_55.txt	Jay
dispt_fed_56.txt	HM
dispt_fed_57.txt	HM
dispt_fed_62.txt	Jay
dispt_fed_63.txt	Jay

K-Means Prediction

	clusters			
author	1	2	3	4
dispt	10	0	0	1
Hamilton	0	1	46	4
HM	0	0	0	3
Jay	1	3	0	1
Madison	13	0	0	2