Jacob Dineen

NETID: Jdineen

IST565

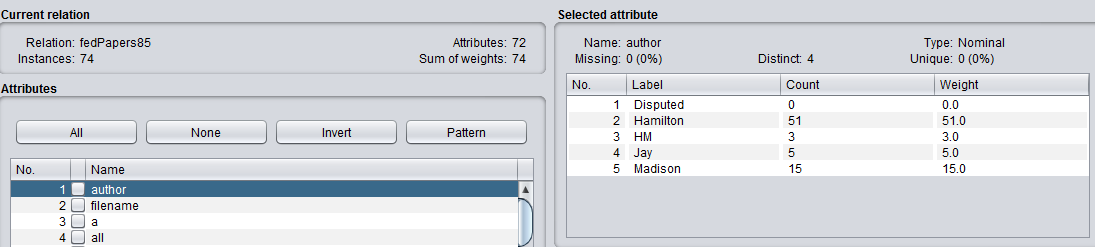
Homework 5: Decision Trees

The first order of business will be deciding how to split the data for training and testing purposes. Normally a cutpoint of 2/3 is utilized, with the larger portion going towards the training data. Because we are simply trying to decipher the authorship of the 11 disputed paper, I will include those in my testing set.

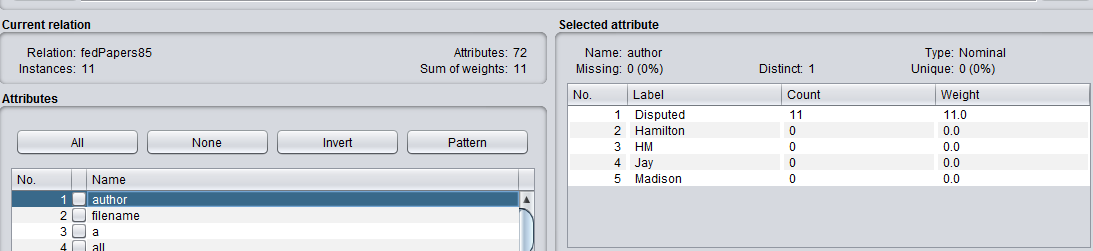
I will load the all-encompassing CSV file into excel and manually split the allocation of instances per the above specifications, leaving me with two CSV files.

I notice that filename is distinctly unique, and will thus register a higher information gain than other attributes. Recalling that the J48 algorithm has a way of adjusting for this type of scenario using gain ratio, I will initially leave it in.

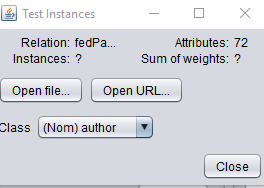
**Training set:**



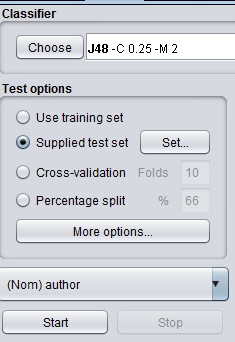
**Testing set:**



Next, I will upload the test data set into the ‘Supplied Data Set’ portal, and change the class to (Nom) Author:



I will also change the desired class output to author, as that is the classification problem we are dealing with here:



**Section 2: Build and tune decision tree models**

**First build a DT model using default setting, and then tune the parameters to see if better model can be generated. Compare these models using appropriate evaluation measures. Describe and compare the patterns learned in these models.**

First, I will build a decision tree model using the default settings, so I will not adjust anything after completing the above preprocessing and loading steps. The default settings of note:

Batch size: 100

Binary Splits: False

minNumObj: 2

Seed: 1

Generating a model exclusively using the training data set results in the following:

=== Summary ===

Correctly Classified Instances 73 98.6486 %

Incorrectly Classified Instances 1 1.3514 %

a b c d e <-- classified as

0 0 0 0 0 | a = Disputed

0 50 1 0 0 | b = Hamilton

0 0 3 0 0 | c = HM

0 0 0 5 0 | d = Jay

1. 0 0 0 15 | e = Madison

So it was able to classify 73 out of 74 of the instances correctly, with the lone exception being a jointly written paper by Hamilton and Madison.

Now we will use the supplied test set with this model, which generates the following:

=== Summary ===

Correctly Classified Instances 0 0 %

Incorrectly Classified Instances 11 100 %

=== Confusion Matrix ===

a b c d e <-- classified as

0 0 0 0 11 | a = Disputed

0 0 0 0 0 | b = Hamilton

0 0 0 0 0 | c = HM

0 0 0 0 0 | d = Jay

0 0 0 0 0 | e = Madison

So it is noting that they are incorrectly classified instances, but this is mainly a preprocessing issue. The confusion matrix shows that all of the disputed essays were bucketed into the Madison column, which coincides with the cluster analysis that was previous performed.

**I was unable to partition the documents to the point where I would not receive the combability issue. When a placed a “?” instead of ‘disputed’, my class would automatically change to string, rather than nominal. After converting to nominal, it would automatically change back when saving the file as an AARF.**

Because this model predicted outcome as all being Madison, I will now load the joint papers, and the Jay papers into the test set as well.

The model generated using just the training set:

=== Summary ===

Correctly Classified Instances 65 98.4848 %

Incorrectly Classified Instances 1 1.5152 %

=== Confusion Matrix ===

a b c d e <-- classified as

0 0 0 0 0 | a = Disputed

0 50 0 0 1 | b = Hamilton

0 0 0 0 0 | c = HM

0 0 0 0 0 | d = Jay

1. 0 0 0 15 | e = Madison

So now there is one Madison essay that the algorithm predicted was more linguistically styled like Hamilton.

When running the 19 instances now involved in our testing set against this model, we see the following output:

=== Confusion Matrix ===

a b c d e <-- classified as

0 0 0 0 11 | a = Disputed

0 0 0 0 0 | b = Hamilton

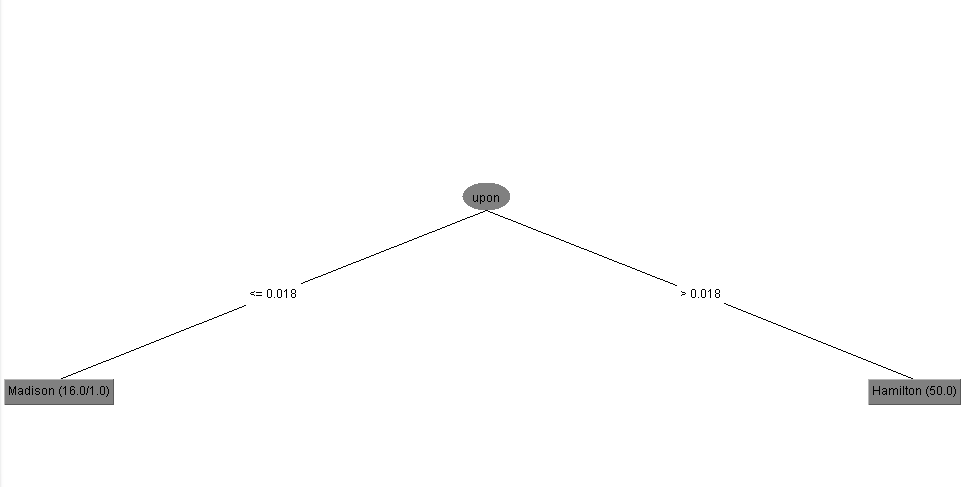
0 0 0 0 3 | c = HM

0 0 0 0 5 | d = Jay

1. 0 0 0 0 | e = Madison

All essays were more similar to Madison than to Hamilton.

This displays a tree that looks like this:



With ‘upon’ being the only root node here, it makes me curious about the other attributes in comparison, and why they aren’t present.

Using Information gain to rank the attributes, we see the following:

Ranked attributes:

0.773 2 filename

0.691 62 upon

0.428 57 there

0.36 42 on

0.279 60 to

0.237 15 by

0.175 11 at

0.156 6 an

0.152 7 and

0.145 8 any

From here, I was not able to change the parameters to produce meaningful results. I will move to R to see what I can do.

Below is my R preprocessing. In this cutpoint, I reverted back to using all of the known authorship instances as my training data, and all of the disputed authorship instances as my Testing data.

all <- read.csv(file.choose())

Train <- all[12:85,]

Test <- all[1:11,]

str(all)

MS <- make\_Weka\_filter("weka/filters/unsupervised/attribute/ReplaceMissingValues") #build a function using RWeka filter interface

#Apply the filter function to both training and test datasets.

trainset <-MS(data=Train, na.action = NULL)

testset <-MS(data=Test, na.action = NULL)

m=J48(author~., data = trainset)

m=J48(author~., data = trainset, control=Weka\_control(U=FALSE, M=2, C=0.5))

m

e <- evaluate\_Weka\_classifier(m,numFolds = 10, seed = 1, class = TRUE)

e

The model generated on my training set was not as accurate as the one generated in Weka:

J48 pruned tree

------------------

upon <= 0.018

| of <= 0.724: Jay (5.0)

| of > 0.724

| | not <= 0.061: HM (4.0/1.0)

| | not > 0.061: Madison (15.0)

upon > 0.018: Hamilton (50.0)

=== 10 Fold Cross Validation ===

=== Summary ===

Correctly Classified Instances 67 90.5405 %

Incorrectly Classified Instances 7 9.4595 %

Kappa statistic 0.7984

Mean absolute error 0.0473

Root mean squared error 0.2153

Relative absolute error 19.1244 %

Root relative squared error 62.0151 %

Total Number of Instances 74

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.980 0.043 0.980 0.980 0.980 0.937 0.968 0.975 Hamilton

0.000 0.000 0.000 0.000 0.000 0.000 0.500 0.041 HM

0.600 0.014 0.750 0.600 0.667 0.650 0.797 0.627 Jay

0.933 0.085 0.737 0.933 0.824 0.781 0.924 0.701 Madison

Weighted Avg. 0.905 0.048 0.876 0.905 0.888 0.848 0.929 0.858

=== Confusion Matrix ===

a b c d <-- classified as

50 0 0 1 | a = Hamilton

0 0 1 2 | b = HM

0 0 3 2 | c = Jay

1 0 0 14 | d = Madison

Again, many of the jointly written articles, and Jay, had writing styles more similar to Madison. What is interesting about this R v Weka interpretation is the difference in the tree. ‘Of’ and ‘Not’ appear in the R pruned tree, but not in Weka. This could be due to different parameterization, or implementation of the algorithm.

Running this on the testing data yields similar results- Madison was likely to have authored the disputed essays.

> #Apply the model with test dataset

> pred <- predict(m, newdata = testset, type = c("class" ))

> pred

[1] Madison Madison Madison Madison Madison Madison Madison Madison Madison Madison Madison

Levels: Disputed Hamilton HM Jay Madison

Again, changing the minNumberObjects or the confidence factor doesn’t seem to change the display of the tree.

My thinking is that the information gain of the attribute “Upon” is essential enough to the algorithm that it is not necessary to have a deep tree in this instance. According to these results, it would be very accurate to look at an essay and analyze the feature value of a particular word, defined as the total number of appearances of a word divided by the total word count of the essay, and be able to identify the authorship of an essay, in this dataset.