Decision Tree

Sudhanshu Kulkarni

# Background

We have been provided with a data set of the Federal Papers which consisted of 85 essays which were written by Alexander Hamilton, James Madison, and John Jay.Out of these 85 essays, 74 essays were identified with their authors, i.e. 51 essays written by Hamilton, 15 by Madison, 3 by Hamilton and Madison, 5 by Jay. The remaining 11 essays did not have an atuhor attributed to them, hence termed as disputed papers. In the data set, we are provided of function words. The occurance of these function words vary with different authors. So our task is, by using decision trees find the authors of these disputed papers.

# Solution

## Importing Libraries

First, we begin with installing and importing libraries. We have installed and imported the “dplyr” library to help us with data preprocessing measures. Also, we have imported the “tree” to build and visualize trees.

library(dplyr)

## Warning: package 'dplyr' was built under R version 3.5.2

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tree)

## Warning: package 'tree' was built under R version 3.5.2

## Loading Data

Then, we load the csv data into a data set and check its structure to know if any data preprocessing is required.

setwd("C:\\Sudhanshu\\SU\\Semester 2\\707\\Assignment 5")  
fedPapersDS <- read.csv("fedPapers85.csv")  
View(fedPapersDS)  
str(fedPapersDS)

## 'data.frame': 85 obs. of 72 variables:  
## $ author : Factor w/ 5 levels "dispt","Hamilton",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ filename: Factor w/ 85 levels "dispt\_fed\_49.txt",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ a : num 0.28 0.177 0.339 0.27 0.303 0.245 0.349 0.414 0.248 0.442 ...  
## $ all : num 0.052 0.063 0.09 0.024 0.054 0.059 0.036 0.083 0.04 0.062 ...  
## $ also : num 0.009 0.013 0.008 0.016 0.027 0.007 0.007 0.009 0.007 0.006 ...  
## $ an : num 0.096 0.038 0.03 0.024 0.034 0.067 0.029 0.018 0.04 0.075 ...  
## $ and : num 0.358 0.393 0.301 0.262 0.404 0.282 0.335 0.478 0.356 0.423 ...  
## $ any : num 0.026 0.063 0.008 0.056 0.04 0.052 0.058 0.046 0.034 0.037 ...  
## $ are : num 0.131 0.051 0.068 0.064 0.128 0.111 0.087 0.11 0.154 0.093 ...  
## $ as : num 0.122 0.139 0.203 0.111 0.148 0.252 0.073 0.074 0.161 0.1 ...  
## $ at : num 0.017 0.114 0.023 0.056 0.013 0.015 0.116 0.037 0.047 0.031 ...  
## $ be : num 0.411 0.393 0.474 0.365 0.344 0.297 0.378 0.331 0.289 0.379 ...  
## $ been : num 0.026 0.165 0.015 0.127 0.047 0.03 0.044 0.046 0.027 0.025 ...  
## $ but : num 0.009 0 0.038 0.032 0.061 0.037 0.007 0.055 0.027 0.037 ...  
## $ by : num 0.14 0.139 0.173 0.167 0.209 0.186 0.102 0.092 0.168 0.174 ...  
## $ can : num 0.035 0 0.023 0.056 0.088 0 0.058 0.037 0.047 0.056 ...  
## $ do : num 0.026 0.013 0 0 0 0 0.015 0.028 0 0 ...  
## $ down : num 0 0 0.008 0 0 0.007 0 0 0 0 ...  
## $ even : num 0.009 0.025 0.015 0.024 0.02 0.007 0.007 0.018 0 0.006 ...  
## $ every : num 0.044 0 0.023 0.04 0.027 0.007 0.087 0.064 0.081 0.05 ...  
## $ for. : num 0.096 0.076 0.098 0.103 0.141 0.067 0.116 0.055 0.127 0.1 ...  
## $ from : num 0.044 0.101 0.053 0.079 0.074 0.096 0.08 0.083 0.074 0.124 ...  
## $ had : num 0.035 0.101 0.008 0.016 0 0.022 0.015 0.009 0.007 0 ...  
## $ has : num 0.017 0.013 0.015 0.024 0.054 0.015 0.036 0.037 0.02 0.019 ...  
## $ have : num 0.044 0.152 0.023 0.143 0.047 0.119 0.044 0.074 0.074 0.044 ...  
## $ her : num 0 0 0 0 0 0 0.007 0 0.034 0.025 ...  
## $ his : num 0.017 0 0 0.024 0.02 0.067 0 0.018 0.02 0.05 ...  
## $ if. : num 0 0.025 0.023 0.04 0.034 0.03 0.029 0 0 0.025 ...  
## $ in. : num 0.262 0.291 0.308 0.238 0.263 0.401 0.189 0.267 0.248 0.274 ...  
## $ into : num 0.009 0.025 0.038 0.008 0.013 0.037 0 0.037 0.013 0.037 ...  
## $ is : num 0.157 0.038 0.15 0.151 0.189 0.26 0.167 0.083 0.208 0.23 ...  
## $ it : num 0.175 0.127 0.173 0.222 0.108 0.156 0.102 0.165 0.134 0.131 ...  
## $ its : num 0.07 0.038 0.03 0.048 0.013 0.015 0 0.046 0.02 0.019 ...  
## $ may : num 0.035 0.038 0.12 0.056 0.047 0.074 0.08 0.092 0.027 0.106 ...  
## $ more : num 0.026 0 0.038 0.056 0.067 0.045 0.08 0.064 0.06 0.081 ...  
## $ must : num 0.026 0.013 0.083 0.071 0.013 0.015 0.044 0.018 0.027 0.068 ...  
## $ my : num 0 0 0 0 0 0 0.007 0 0 0 ...  
## $ no : num 0.035 0 0.03 0.032 0.047 0.059 0.022 0.018 0.02 0.044 ...  
## $ not : num 0.114 0.127 0.068 0.087 0.128 0.134 0.102 0.101 0.094 0.106 ...  
## $ now : num 0 0 0 0 0 0 0.007 0 0.007 0.012 ...  
## $ of : num 0.9 0.747 0.858 0.802 0.869 ...  
## $ on : num 0.14 0.139 0.15 0.143 0.054 0.141 0.051 0.083 0.127 0.118 ...  
## $ one : num 0.026 0.025 0.03 0.032 0.047 0.052 0.073 0.046 0.06 0.031 ...  
## $ only : num 0.035 0 0.023 0.048 0.027 0.022 0.007 0.046 0.02 0.012 ...  
## $ or : num 0.096 0.114 0.06 0.064 0.081 0.074 0.153 0.037 0.154 0.081 ...  
## $ our : num 0.017 0 0 0.016 0.027 0.03 0.051 0 0.007 0.025 ...  
## $ shall : num 0.017 0 0.008 0.016 0 0.015 0.007 0 0.02 0 ...  
## $ should : num 0.017 0.013 0.068 0.032 0 0.03 0.007 0 0 0.012 ...  
## $ so : num 0.035 0.013 0.038 0.04 0.027 0.007 0.051 0.018 0.04 0.05 ...  
## $ some : num 0.009 0.063 0.03 0.024 0.067 0.045 0.007 0.028 0.027 0.025 ...  
## $ such : num 0.026 0 0.045 0.008 0.027 0.015 0.015 0 0.013 0.031 ...  
## $ than : num 0.009 0 0.023 0 0.047 0.03 0.109 0.055 0.067 0.044 ...  
## $ that : num 0.184 0.152 0.188 0.238 0.162 0.208 0.233 0.165 0.208 0.218 ...  
## $ the : num 1.42 1.25 1.49 1.33 1.19 ...  
## $ their : num 0.114 0.165 0.053 0.071 0.027 0.089 0.109 0.083 0.154 0.081 ...  
## $ then : num 0 0 0.015 0.008 0.007 0.007 0.015 0.009 0.007 0.012 ...  
## $ there : num 0.009 0 0.015 0 0.007 0.007 0.036 0.028 0.02 0 ...  
## $ things : num 0.009 0 0 0 0 0 0 0 0 0.012 ...  
## $ this : num 0.044 0.051 0.075 0.103 0.094 0.126 0.08 0.11 0.067 0.093 ...  
## $ to : num 0.507 0.355 0.361 0.532 0.485 0.445 0.56 0.34 0.49 0.498 ...  
## $ up : num 0 0 0 0 0 0 0.007 0 0 0 ...  
## $ upon : num 0 0.013 0 0 0 0 0 0 0 0 ...  
## $ was : num 0.009 0.051 0.008 0.087 0.027 0.007 0.015 0.018 0.027 0 ...  
## $ were : num 0.017 0 0.015 0.079 0.02 0.03 0.029 0.009 0.007 0 ...  
## $ what : num 0 0 0.008 0.008 0.02 0.015 0.015 0.009 0.02 0.025 ...  
## $ when : num 0.009 0 0 0.024 0.007 0.037 0.007 0 0.02 0.012 ...  
## $ which : num 0.175 0.114 0.105 0.167 0.155 0.186 0.211 0.175 0.201 0.199 ...  
## $ who : num 0.044 0.038 0.008 0 0.027 0.045 0.022 0.018 0.04 0.031 ...  
## $ will : num 0.009 0.089 0.173 0.079 0.168 0.111 0.145 0.267 0.154 0.106 ...  
## $ with : num 0.087 0.063 0.045 0.079 0.074 0.089 0.073 0.129 0.027 0.081 ...  
## $ would : num 0.192 0.139 0.068 0.064 0.04 0.037 0.073 0.037 0.04 0.031 ...  
## $ your : num 0 0 0 0 0 0 0 0 0 0 ...

## Data Preprocessing

Decision trees can only be built on nominal values. So, we assign AuthorCode to every author and remove the filename column.

fedPapersDS$AuthorCode <- dplyr::recode(fedPapersDS$author, dispt = 1, Hamilton = 2, HM = 3, Jay = 4, Madison = 5)  
fedPapersDS <- fedPapersDS[,-2]  
str(fedPapersDS)

## 'data.frame': 85 obs. of 72 variables:  
## $ author : Factor w/ 5 levels "dispt","Hamilton",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ a : num 0.28 0.177 0.339 0.27 0.303 0.245 0.349 0.414 0.248 0.442 ...  
## $ all : num 0.052 0.063 0.09 0.024 0.054 0.059 0.036 0.083 0.04 0.062 ...  
## $ also : num 0.009 0.013 0.008 0.016 0.027 0.007 0.007 0.009 0.007 0.006 ...  
## $ an : num 0.096 0.038 0.03 0.024 0.034 0.067 0.029 0.018 0.04 0.075 ...  
## $ and : num 0.358 0.393 0.301 0.262 0.404 0.282 0.335 0.478 0.356 0.423 ...  
## $ any : num 0.026 0.063 0.008 0.056 0.04 0.052 0.058 0.046 0.034 0.037 ...  
## $ are : num 0.131 0.051 0.068 0.064 0.128 0.111 0.087 0.11 0.154 0.093 ...  
## $ as : num 0.122 0.139 0.203 0.111 0.148 0.252 0.073 0.074 0.161 0.1 ...  
## $ at : num 0.017 0.114 0.023 0.056 0.013 0.015 0.116 0.037 0.047 0.031 ...  
## $ be : num 0.411 0.393 0.474 0.365 0.344 0.297 0.378 0.331 0.289 0.379 ...  
## $ been : num 0.026 0.165 0.015 0.127 0.047 0.03 0.044 0.046 0.027 0.025 ...  
## $ but : num 0.009 0 0.038 0.032 0.061 0.037 0.007 0.055 0.027 0.037 ...  
## $ by : num 0.14 0.139 0.173 0.167 0.209 0.186 0.102 0.092 0.168 0.174 ...  
## $ can : num 0.035 0 0.023 0.056 0.088 0 0.058 0.037 0.047 0.056 ...  
## $ do : num 0.026 0.013 0 0 0 0 0.015 0.028 0 0 ...  
## $ down : num 0 0 0.008 0 0 0.007 0 0 0 0 ...  
## $ even : num 0.009 0.025 0.015 0.024 0.02 0.007 0.007 0.018 0 0.006 ...  
## $ every : num 0.044 0 0.023 0.04 0.027 0.007 0.087 0.064 0.081 0.05 ...  
## $ for. : num 0.096 0.076 0.098 0.103 0.141 0.067 0.116 0.055 0.127 0.1 ...  
## $ from : num 0.044 0.101 0.053 0.079 0.074 0.096 0.08 0.083 0.074 0.124 ...  
## $ had : num 0.035 0.101 0.008 0.016 0 0.022 0.015 0.009 0.007 0 ...  
## $ has : num 0.017 0.013 0.015 0.024 0.054 0.015 0.036 0.037 0.02 0.019 ...  
## $ have : num 0.044 0.152 0.023 0.143 0.047 0.119 0.044 0.074 0.074 0.044 ...  
## $ her : num 0 0 0 0 0 0 0.007 0 0.034 0.025 ...  
## $ his : num 0.017 0 0 0.024 0.02 0.067 0 0.018 0.02 0.05 ...  
## $ if. : num 0 0.025 0.023 0.04 0.034 0.03 0.029 0 0 0.025 ...  
## $ in. : num 0.262 0.291 0.308 0.238 0.263 0.401 0.189 0.267 0.248 0.274 ...  
## $ into : num 0.009 0.025 0.038 0.008 0.013 0.037 0 0.037 0.013 0.037 ...  
## $ is : num 0.157 0.038 0.15 0.151 0.189 0.26 0.167 0.083 0.208 0.23 ...  
## $ it : num 0.175 0.127 0.173 0.222 0.108 0.156 0.102 0.165 0.134 0.131 ...  
## $ its : num 0.07 0.038 0.03 0.048 0.013 0.015 0 0.046 0.02 0.019 ...  
## $ may : num 0.035 0.038 0.12 0.056 0.047 0.074 0.08 0.092 0.027 0.106 ...  
## $ more : num 0.026 0 0.038 0.056 0.067 0.045 0.08 0.064 0.06 0.081 ...  
## $ must : num 0.026 0.013 0.083 0.071 0.013 0.015 0.044 0.018 0.027 0.068 ...  
## $ my : num 0 0 0 0 0 0 0.007 0 0 0 ...  
## $ no : num 0.035 0 0.03 0.032 0.047 0.059 0.022 0.018 0.02 0.044 ...  
## $ not : num 0.114 0.127 0.068 0.087 0.128 0.134 0.102 0.101 0.094 0.106 ...  
## $ now : num 0 0 0 0 0 0 0.007 0 0.007 0.012 ...  
## $ of : num 0.9 0.747 0.858 0.802 0.869 ...  
## $ on : num 0.14 0.139 0.15 0.143 0.054 0.141 0.051 0.083 0.127 0.118 ...  
## $ one : num 0.026 0.025 0.03 0.032 0.047 0.052 0.073 0.046 0.06 0.031 ...  
## $ only : num 0.035 0 0.023 0.048 0.027 0.022 0.007 0.046 0.02 0.012 ...  
## $ or : num 0.096 0.114 0.06 0.064 0.081 0.074 0.153 0.037 0.154 0.081 ...  
## $ our : num 0.017 0 0 0.016 0.027 0.03 0.051 0 0.007 0.025 ...  
## $ shall : num 0.017 0 0.008 0.016 0 0.015 0.007 0 0.02 0 ...  
## $ should : num 0.017 0.013 0.068 0.032 0 0.03 0.007 0 0 0.012 ...  
## $ so : num 0.035 0.013 0.038 0.04 0.027 0.007 0.051 0.018 0.04 0.05 ...  
## $ some : num 0.009 0.063 0.03 0.024 0.067 0.045 0.007 0.028 0.027 0.025 ...  
## $ such : num 0.026 0 0.045 0.008 0.027 0.015 0.015 0 0.013 0.031 ...  
## $ than : num 0.009 0 0.023 0 0.047 0.03 0.109 0.055 0.067 0.044 ...  
## $ that : num 0.184 0.152 0.188 0.238 0.162 0.208 0.233 0.165 0.208 0.218 ...  
## $ the : num 1.42 1.25 1.49 1.33 1.19 ...  
## $ their : num 0.114 0.165 0.053 0.071 0.027 0.089 0.109 0.083 0.154 0.081 ...  
## $ then : num 0 0 0.015 0.008 0.007 0.007 0.015 0.009 0.007 0.012 ...  
## $ there : num 0.009 0 0.015 0 0.007 0.007 0.036 0.028 0.02 0 ...  
## $ things : num 0.009 0 0 0 0 0 0 0 0 0.012 ...  
## $ this : num 0.044 0.051 0.075 0.103 0.094 0.126 0.08 0.11 0.067 0.093 ...  
## $ to : num 0.507 0.355 0.361 0.532 0.485 0.445 0.56 0.34 0.49 0.498 ...  
## $ up : num 0 0 0 0 0 0 0.007 0 0 0 ...  
## $ upon : num 0 0.013 0 0 0 0 0 0 0 0 ...  
## $ was : num 0.009 0.051 0.008 0.087 0.027 0.007 0.015 0.018 0.027 0 ...  
## $ were : num 0.017 0 0.015 0.079 0.02 0.03 0.029 0.009 0.007 0 ...  
## $ what : num 0 0 0.008 0.008 0.02 0.015 0.015 0.009 0.02 0.025 ...  
## $ when : num 0.009 0 0 0.024 0.007 0.037 0.007 0 0.02 0.012 ...  
## $ which : num 0.175 0.114 0.105 0.167 0.155 0.186 0.211 0.175 0.201 0.199 ...  
## $ who : num 0.044 0.038 0.008 0 0.027 0.045 0.022 0.018 0.04 0.031 ...  
## $ will : num 0.009 0.089 0.173 0.079 0.168 0.111 0.145 0.267 0.154 0.106 ...  
## $ with : num 0.087 0.063 0.045 0.079 0.074 0.089 0.073 0.129 0.027 0.081 ...  
## $ would : num 0.192 0.139 0.068 0.064 0.04 0.037 0.073 0.037 0.04 0.031 ...  
## $ your : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ AuthorCode: num 1 1 1 1 1 1 1 1 1 1 ...

## Decision Tree Model

For decision trees we must have training dataset and testing dataset. By using the training dataset we will train the model i.e. understand the use of function words and then run the classification algorithm on the test dataset to understand the authors of disputed papers.

fedTrainingData <- subset(fedPapersDS, AuthorCode == 2 | AuthorCode == 3 | AuthorCode == 4 | AuthorCode == 5)  
str(fedTrainingData)

## 'data.frame': 74 obs. of 72 variables:  
## $ author : Factor w/ 5 levels "dispt","Hamilton",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ a : num 0.213 0.369 0.305 0.391 0.327 0.26 0.261 0.449 0.392 0.194 ...  
## $ all : num 0.083 0.07 0.047 0.045 0.096 0.065 0.108 0.022 0.05 0.081 ...  
## $ also : num 0 0.006 0.007 0.015 0 0 0 0 0.004 0 ...  
## $ an : num 0.083 0.076 0.068 0.03 0.086 0.087 0.072 0.074 0.075 0.089 ...  
## $ and : num 0.343 0.411 0.386 0.27 0.356 0.274 0.467 0.353 0.329 0.413 ...  
## $ any : num 0.056 0.023 0.047 0.045 0.014 0.079 0.018 0.044 0.029 0.065 ...  
## $ are : num 0.111 0.053 0.102 0.06 0.086 0.022 0.045 0.059 0.075 0.138 ...  
## $ as : num 0.093 0.117 0.108 0.09 0.072 0.13 0.027 0.133 0.104 0.186 ...  
## $ at : num 0.065 0.065 0.088 0.015 0.115 0.079 0.063 0.029 0.05 0.024 ...  
## $ be : num 0.315 0.258 0.271 0.376 0.211 0.397 0.216 0.295 0.221 0.356 ...  
## $ been : num 0.028 0.018 0.054 0.03 0.067 0.051 0.027 0.081 0.067 0.049 ...  
## $ but : num 0 0.023 0.041 0.03 0.034 0.036 0.009 0.015 0.05 0.024 ...  
## $ by : num 0.13 0.106 0.095 0.075 0.154 0.094 0.081 0.162 0.125 0.089 ...  
## $ can : num 0.028 0.029 0.014 0.06 0.067 0.007 0.018 0.066 0.025 0.089 ...  
## $ do : num 0.009 0.012 0 0.015 0.019 0.014 0 0.015 0.008 0 ...  
## $ down : num 0 0 0 0 0.005 0.007 0 0 0 0 ...  
## $ even : num 0.019 0.018 0.014 0 0.014 0.029 0 0.015 0.021 0 ...  
## $ every : num 0.028 0.012 0.027 0.045 0.038 0.007 0.009 0.022 0.029 0.024 ...  
## $ for. : num 0.093 0.106 0.054 0.03 0.086 0.065 0.099 0.037 0.088 0.13 ...  
## $ from : num 0.102 0.111 0.129 0.075 0.101 0.079 0.045 0.118 0.096 0.049 ...  
## $ had : num 0.009 0.006 0.02 0 0.01 0.036 0.018 0.007 0.013 0.008 ...  
## $ has : num 0.056 0.047 0.088 0.03 0.062 0.051 0.036 0.066 0.083 0.041 ...  
## $ have : num 0.093 0.088 0.088 0.06 0.144 0.087 0.036 0.066 0.058 0.081 ...  
## $ her : num 0 0.076 0.007 0.15 0 0 0 0.007 0.004 0 ...  
## $ his : num 0 0.006 0.014 0 0.005 0 0.063 0.015 0.029 0 ...  
## $ if. : num 0.028 0.029 0.027 0.045 0.014 0.058 0.018 0.044 0.025 0.016 ...  
## $ in. : num 0.232 0.387 0.311 0.24 0.317 0.274 0.252 0.331 0.321 0.211 ...  
## $ into : num 0.019 0.023 0.047 0.105 0.019 0.022 0.027 0 0.017 0.016 ...  
## $ is : num 0.12 0.053 0.149 0.135 0.207 0.101 0.099 0.199 0.192 0.154 ...  
## $ it : num 0.139 0.088 0.142 0.09 0.096 0.188 0.09 0.103 0.146 0.154 ...  
## $ its : num 0.074 0.035 0.047 0.03 0.038 0.116 0.054 0.081 0.067 0.089 ...  
## $ may : num 0.102 0.053 0.02 0.12 0.034 0.036 0.036 0.081 0.083 0.049 ...  
## $ more : num 0.065 0.053 0.034 0.09 0.029 0.043 0.081 0.022 0.025 0.016 ...  
## $ must : num 0.028 0.012 0.108 0.045 0.043 0.065 0.009 0.029 0.038 0.073 ...  
## $ my : num 0.056 0 0.007 0 0.014 0 0.009 0 0.004 0 ...  
## $ no : num 0.028 0.029 0.007 0.045 0.029 0.029 0 0.081 0.021 0.016 ...  
## $ not : num 0.13 0.117 0.041 0.075 0.077 0.137 0.036 0.059 0.104 0.081 ...  
## $ now : num 0 0.006 0.007 0 0.01 0.007 0 0.022 0.004 0.008 ...  
## $ of : num 0.908 1.061 0.961 0.887 0.947 ...  
## $ on : num 0.074 0.029 0.081 0.045 0.048 0.029 0.018 0.044 0.038 0.016 ...  
## $ one : num 0.028 0.029 0.027 0.135 0.005 0.014 0.018 0.007 0.038 0.024 ...  
## $ only : num 0 0.029 0 0 0.038 0.022 0 0.015 0.004 0 ...  
## $ or : num 0.056 0.07 0.047 0.045 0.183 0.123 0.117 0.147 0.088 0.049 ...  
## $ our : num 0.028 0.199 0.034 0 0.091 0.007 0.018 0.007 0.05 0.057 ...  
## $ shall : num 0.065 0.018 0.007 0.045 0.005 0.007 0.009 0.037 0.008 0.008 ...  
## $ should : num 0.009 0.018 0.027 0 0.029 0.043 0.018 0.037 0.033 0.016 ...  
## $ so : num 0.019 0.029 0.041 0.045 0.01 0.022 0.036 0.052 0.038 0.008 ...  
## $ some : num 0 0.006 0.014 0 0.01 0.022 0.009 0.022 0.017 0.008 ...  
## $ such : num 0.009 0.029 0.02 0.015 0.005 0.022 0 0.022 0.038 0.057 ...  
## $ than : num 0.102 0.035 0.027 0.15 0.048 0.029 0.045 0.029 0.046 0 ...  
## $ that : num 0.241 0.141 0.176 0.27 0.178 0.116 0.189 0.177 0.221 0.227 ...  
## $ the : num 1.09 1.04 1.1 1.05 1.16 ...  
## $ their : num 0.13 0.059 0.095 0.015 0.034 0.087 0.09 0.059 0.092 0.065 ...  
## $ then : num 0 0.012 0 0 0 0 0 0 0.004 0 ...  
## $ there : num 0.019 0.023 0.041 0.105 0.072 0.022 0.018 0.029 0.054 0.032 ...  
## $ things : num 0 0.006 0.014 0 0.01 0 0.009 0.015 0.008 0 ...  
## $ this : num 0.083 0.106 0.095 0.03 0.086 0.101 0.045 0.096 0.108 0.073 ...  
## $ to : num 0.575 0.481 0.555 0.646 0.557 0.628 0.521 0.39 0.579 0.753 ...  
## $ up : num 0 0 0 0.015 0.005 0 0 0 0.004 0 ...  
## $ upon : num 0.056 0.035 0.041 0.03 0.048 0.043 0.045 0.044 0.05 0.057 ...  
## $ was : num 0 0 0 0 0.014 0 0.054 0.007 0.029 0.032 ...  
## $ were : num 0 0.012 0 0 0.038 0.051 0.036 0.015 0.038 0.008 ...  
## $ what : num 0 0.012 0.007 0 0.014 0.014 0.009 0.015 0.017 0 ...  
## $ when : num 0.009 0.012 0 0.045 0.019 0.014 0 0.007 0.008 0 ...  
## $ which : num 0.158 0.147 0.156 0.165 0.264 0.159 0.216 0.147 0.15 0.243 ...  
## $ who : num 0.074 0.029 0.007 0.045 0.029 0.036 0.018 0.022 0.013 0.016 ...  
## $ will : num 0.222 0.094 0.074 0.135 0.091 0.043 0.117 0.044 0.046 0.122 ...  
## $ with : num 0.046 0.129 0.122 0.15 0.086 0.094 0.09 0.11 0.083 0.13 ...  
## $ would : num 0.019 0.27 0.149 0.21 0.062 0.26 0.108 0.088 0.083 0.032 ...  
## $ your : num 0.074 0 0 0 0.01 0 0 0 0 0 ...  
## $ AuthorCode: num 2 2 2 2 2 2 2 2 2 2 ...

fedTestData <- subset(fedPapersDS, AuthorCode == 1)  
str(fedTestData)

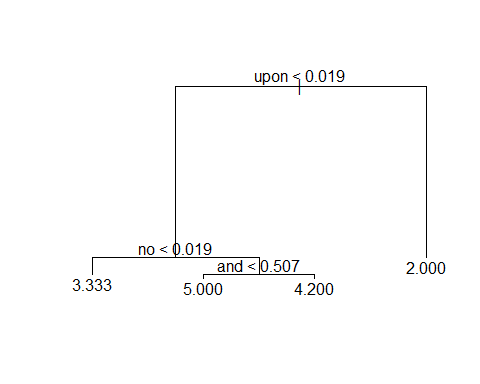
## 'data.frame': 11 obs. of 72 variables:  
## $ author : Factor w/ 5 levels "dispt","Hamilton",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ a : num 0.28 0.177 0.339 0.27 0.303 0.245 0.349 0.414 0.248 0.442 ...  
## $ all : num 0.052 0.063 0.09 0.024 0.054 0.059 0.036 0.083 0.04 0.062 ...  
## $ also : num 0.009 0.013 0.008 0.016 0.027 0.007 0.007 0.009 0.007 0.006 ...  
## $ an : num 0.096 0.038 0.03 0.024 0.034 0.067 0.029 0.018 0.04 0.075 ...  
## $ and : num 0.358 0.393 0.301 0.262 0.404 0.282 0.335 0.478 0.356 0.423 ...  
## $ any : num 0.026 0.063 0.008 0.056 0.04 0.052 0.058 0.046 0.034 0.037 ...  
## $ are : num 0.131 0.051 0.068 0.064 0.128 0.111 0.087 0.11 0.154 0.093 ...  
## $ as : num 0.122 0.139 0.203 0.111 0.148 0.252 0.073 0.074 0.161 0.1 ...  
## $ at : num 0.017 0.114 0.023 0.056 0.013 0.015 0.116 0.037 0.047 0.031 ...  
## $ be : num 0.411 0.393 0.474 0.365 0.344 0.297 0.378 0.331 0.289 0.379 ...  
## $ been : num 0.026 0.165 0.015 0.127 0.047 0.03 0.044 0.046 0.027 0.025 ...  
## $ but : num 0.009 0 0.038 0.032 0.061 0.037 0.007 0.055 0.027 0.037 ...  
## $ by : num 0.14 0.139 0.173 0.167 0.209 0.186 0.102 0.092 0.168 0.174 ...  
## $ can : num 0.035 0 0.023 0.056 0.088 0 0.058 0.037 0.047 0.056 ...  
## $ do : num 0.026 0.013 0 0 0 0 0.015 0.028 0 0 ...  
## $ down : num 0 0 0.008 0 0 0.007 0 0 0 0 ...  
## $ even : num 0.009 0.025 0.015 0.024 0.02 0.007 0.007 0.018 0 0.006 ...  
## $ every : num 0.044 0 0.023 0.04 0.027 0.007 0.087 0.064 0.081 0.05 ...  
## $ for. : num 0.096 0.076 0.098 0.103 0.141 0.067 0.116 0.055 0.127 0.1 ...  
## $ from : num 0.044 0.101 0.053 0.079 0.074 0.096 0.08 0.083 0.074 0.124 ...  
## $ had : num 0.035 0.101 0.008 0.016 0 0.022 0.015 0.009 0.007 0 ...  
## $ has : num 0.017 0.013 0.015 0.024 0.054 0.015 0.036 0.037 0.02 0.019 ...  
## $ have : num 0.044 0.152 0.023 0.143 0.047 0.119 0.044 0.074 0.074 0.044 ...  
## $ her : num 0 0 0 0 0 0 0.007 0 0.034 0.025 ...  
## $ his : num 0.017 0 0 0.024 0.02 0.067 0 0.018 0.02 0.05 ...  
## $ if. : num 0 0.025 0.023 0.04 0.034 0.03 0.029 0 0 0.025 ...  
## $ in. : num 0.262 0.291 0.308 0.238 0.263 0.401 0.189 0.267 0.248 0.274 ...  
## $ into : num 0.009 0.025 0.038 0.008 0.013 0.037 0 0.037 0.013 0.037 ...  
## $ is : num 0.157 0.038 0.15 0.151 0.189 0.26 0.167 0.083 0.208 0.23 ...  
## $ it : num 0.175 0.127 0.173 0.222 0.108 0.156 0.102 0.165 0.134 0.131 ...  
## $ its : num 0.07 0.038 0.03 0.048 0.013 0.015 0 0.046 0.02 0.019 ...  
## $ may : num 0.035 0.038 0.12 0.056 0.047 0.074 0.08 0.092 0.027 0.106 ...  
## $ more : num 0.026 0 0.038 0.056 0.067 0.045 0.08 0.064 0.06 0.081 ...  
## $ must : num 0.026 0.013 0.083 0.071 0.013 0.015 0.044 0.018 0.027 0.068 ...  
## $ my : num 0 0 0 0 0 0 0.007 0 0 0 ...  
## $ no : num 0.035 0 0.03 0.032 0.047 0.059 0.022 0.018 0.02 0.044 ...  
## $ not : num 0.114 0.127 0.068 0.087 0.128 0.134 0.102 0.101 0.094 0.106 ...  
## $ now : num 0 0 0 0 0 0 0.007 0 0.007 0.012 ...  
## $ of : num 0.9 0.747 0.858 0.802 0.869 ...  
## $ on : num 0.14 0.139 0.15 0.143 0.054 0.141 0.051 0.083 0.127 0.118 ...  
## $ one : num 0.026 0.025 0.03 0.032 0.047 0.052 0.073 0.046 0.06 0.031 ...  
## $ only : num 0.035 0 0.023 0.048 0.027 0.022 0.007 0.046 0.02 0.012 ...  
## $ or : num 0.096 0.114 0.06 0.064 0.081 0.074 0.153 0.037 0.154 0.081 ...  
## $ our : num 0.017 0 0 0.016 0.027 0.03 0.051 0 0.007 0.025 ...  
## $ shall : num 0.017 0 0.008 0.016 0 0.015 0.007 0 0.02 0 ...  
## $ should : num 0.017 0.013 0.068 0.032 0 0.03 0.007 0 0 0.012 ...  
## $ so : num 0.035 0.013 0.038 0.04 0.027 0.007 0.051 0.018 0.04 0.05 ...  
## $ some : num 0.009 0.063 0.03 0.024 0.067 0.045 0.007 0.028 0.027 0.025 ...  
## $ such : num 0.026 0 0.045 0.008 0.027 0.015 0.015 0 0.013 0.031 ...  
## $ than : num 0.009 0 0.023 0 0.047 0.03 0.109 0.055 0.067 0.044 ...  
## $ that : num 0.184 0.152 0.188 0.238 0.162 0.208 0.233 0.165 0.208 0.218 ...  
## $ the : num 1.42 1.25 1.49 1.33 1.19 ...  
## $ their : num 0.114 0.165 0.053 0.071 0.027 0.089 0.109 0.083 0.154 0.081 ...  
## $ then : num 0 0 0.015 0.008 0.007 0.007 0.015 0.009 0.007 0.012 ...  
## $ there : num 0.009 0 0.015 0 0.007 0.007 0.036 0.028 0.02 0 ...  
## $ things : num 0.009 0 0 0 0 0 0 0 0 0.012 ...  
## $ this : num 0.044 0.051 0.075 0.103 0.094 0.126 0.08 0.11 0.067 0.093 ...  
## $ to : num 0.507 0.355 0.361 0.532 0.485 0.445 0.56 0.34 0.49 0.498 ...  
## $ up : num 0 0 0 0 0 0 0.007 0 0 0 ...  
## $ upon : num 0 0.013 0 0 0 0 0 0 0 0 ...  
## $ was : num 0.009 0.051 0.008 0.087 0.027 0.007 0.015 0.018 0.027 0 ...  
## $ were : num 0.017 0 0.015 0.079 0.02 0.03 0.029 0.009 0.007 0 ...  
## $ what : num 0 0 0.008 0.008 0.02 0.015 0.015 0.009 0.02 0.025 ...  
## $ when : num 0.009 0 0 0.024 0.007 0.037 0.007 0 0.02 0.012 ...  
## $ which : num 0.175 0.114 0.105 0.167 0.155 0.186 0.211 0.175 0.201 0.199 ...  
## $ who : num 0.044 0.038 0.008 0 0.027 0.045 0.022 0.018 0.04 0.031 ...  
## $ will : num 0.009 0.089 0.173 0.079 0.168 0.111 0.145 0.267 0.154 0.106 ...  
## $ with : num 0.087 0.063 0.045 0.079 0.074 0.089 0.073 0.129 0.027 0.081 ...  
## $ would : num 0.192 0.139 0.068 0.064 0.04 0.037 0.073 0.037 0.04 0.031 ...  
## $ your : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ AuthorCode: num 1 1 1 1 1 1 1 1 1 1 ...

Then we build the decision tree. We exclude author field this time as we already have AuthorCode. To understand the tree in detail we check its summary and plot the tree.

tree.fedPapers = tree(AuthorCode~.-author, data=fedTrainingData)  
summary(tree.fedPapers)

##   
## Regression tree:  
## tree(formula = AuthorCode ~ . - author, data = fedTrainingData)  
## Variables actually used in tree construction:  
## [1] "upon" "no" "and"   
## Number of terminal nodes: 4   
## Residual mean deviance: 0.08762 = 6.133 / 70   
## Distribution of residuals:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -1.333 0.000 0.000 0.000 0.000 0.800

plot(tree.fedPapers)  
text(tree.fedPapers, splits = TRUE, all = FALSE ,pretty = 0)



From the above graph we can see that the algorithm has build a tree with four leaf nodes for each author.

tree.fedPapers

## node), split, n, deviance, yval  
## \* denotes terminal node  
##   
## 1) root 74 112.500 2.784   
## 2) upon < 0.019 24 17.830 4.417   
## 4) no < 0.019 6 3.333 3.333 \*  
## 5) no > 0.019 18 5.111 4.778   
## 10) and < 0.507 13 0.000 5.000 \*  
## 11) and > 0.507 5 2.800 4.200 \*  
## 3) upon > 0.019 50 0.000 2.000 \*

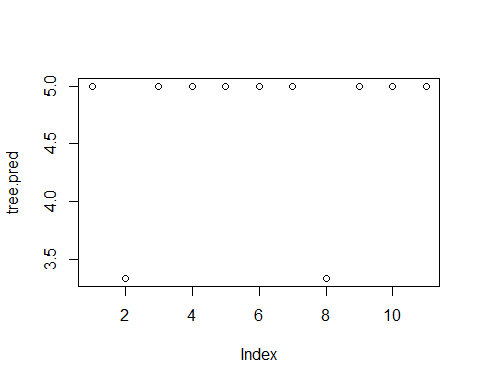
## Prediction

Now we use this tree to predict the result of test dataset using the predict method.

tree.pred = predict(tree.fedPapers, fedTestData)  
summary(tree.pred)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 3.333 5.000 5.000 4.697 5.000 5.000

plot(tree.pred)



with(fedTestData, table(tree.pred, AuthorCode))

## AuthorCode  
## tree.pred 1  
## 3.33333333333333 2  
## 5 9

From the above matrix, we can see that the model has predicted two of the disputed essays were written by Hamilton and Madison, and remainng nine were written by Madison.

## Conclusion

This result is similar to the clustering analysis. Last week we had concluded that two disputed essays were written by Hamilton and remaining nine essays were written by Madison. After the decision tree analysis, we can conclude that two of the disputed essays were written by Hamilton and Madison, and remainng nine were written by Madison.