Martin\_Alonso\_Hw5

Martin Alonso

November 5, 2018

# Hw5 Instructions

In this homework assignment, you are going to use the decision tree algorithm to solve the disputed essay problem. Last week you used clustering techniques to tackle this problem.

## 1. Data Preparation

We’ll start by loading the necesary packages to work with data partitioning, model design, and graphics outputs.

require(caret)

## Loading required package: caret

## Loading required package: lattice

## Loading required package: ggplot2

require(dplyr)

## Loading required package: dplyr

##   
## 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

require(ggplot2)  
require(rpart)

## Loading required package: rpart

require(rpart.plot)

## Loading required package: rpart.plot

require(e1071)

## Loading required package: e1071

With the packages now loaded, it’s time to load the data. Since this is the same dataset as last week’s homework, we’ll eschew from checking the summary, structure, and head of the dataset.

dat <- read.csv('fedPapers85.csv')

Now, we’ll separate the dataset into three group: 1. Training set 2. Testing set 3. Verification set - the disputed papers

We’ll extract the disputed papers from the set as we want to know if the model can predict to who these belong to.

The training set will consist of 2/3 of the dataset after the verification set has been removed; and the test set will be the remaining 1/3.

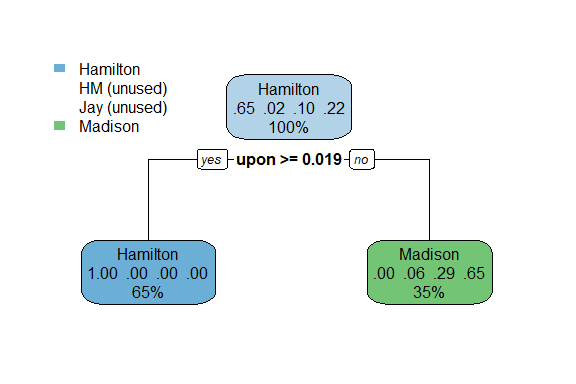
During the Prediction part, we’ll test the model with the

fed <- dat %>% filter(author != 'dispt')  
fed$author <- as.factor(as.character(fed$author))  
  
# Randomly select 2/3 of the dataset - these will be the training set.   
split <- sample(nrow(fed), nrow(fed) \* 2/3)  
train <- fed[split, ]  
test <- fed[-split, ]  
  
# The verification set,  
ver <- dat %>% filter(author == 'dispt')

Now that we have our datasets split, time to build and train the model.

## 2. Build and tune decision tree models

# We'll create a first tree using rpart. We exclude the filename variable as it is a factor and doesn't add much to the decision tree.   
feds.tree <- rpart(author ~ . - filename, data = train, method = 'class', control = rpart.control(cp = 0))  
  
# Without actually pruning the tree, let's check how the model works  
rpart.plot(feds.tree)

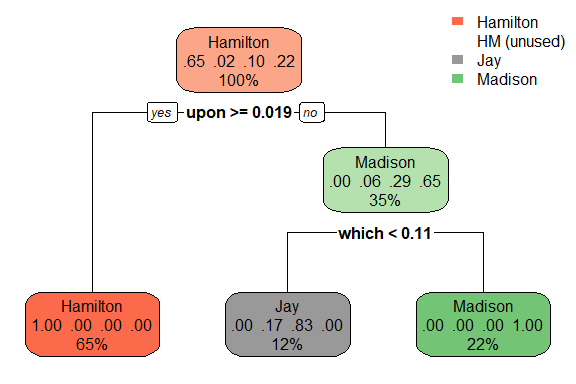


# And also look under the hood.  
summary(feds.tree)

## Call:  
## rpart(formula = author ~ . - filename, data = train, method = "class",   
## control = rpart.control(cp = 0))  
## n= 49   
##   
## CP nsplit rel error xerror xstd  
## 1 0.6470588 0 1.0000000 1.0000000 0.1959984  
## 2 0.0000000 1 0.3529412 0.4117647 0.1440876  
##   
## Variable importance  
## upon and there of to a   
## 25 16 16 15 15 13   
##   
## Node number 1: 49 observations, complexity param=0.6470588  
## predicted class=Hamilton expected loss=0.3469388 P(node) =1  
## class counts: 32 1 5 11  
## probabilities: 0.653 0.020 0.102 0.224   
## left son=2 (32 obs) right son=3 (17 obs)  
## Primary splits:  
## upon < 0.019 to the right, improve=16.749100, (0 missing)  
## there < 0.0145 to the right, improve= 9.539541, (0 missing)  
## and < 0.3945 to the left, improve= 8.488070, (0 missing)  
## of < 0.8655 to the right, improve= 7.873469, (0 missing)  
## to < 0.4885 to the right, improve= 7.759184, (0 missing)  
## Surrogate splits:  
## and < 0.3945 to the left, agree=0.878, adj=0.647, (0 split)  
## there < 0.0145 to the right, agree=0.878, adj=0.647, (0 split)  
## of < 0.8655 to the right, agree=0.857, adj=0.588, (0 split)  
## to < 0.4885 to the right, agree=0.857, adj=0.588, (0 split)  
## a < 0.255 to the right, agree=0.837, adj=0.529, (0 split)  
##   
## Node number 2: 32 observations  
## predicted class=Hamilton expected loss=0 P(node) =0.6530612  
## class counts: 32 0 0 0  
## probabilities: 1.000 0.000 0.000 0.000   
##   
## Node number 3: 17 observations  
## predicted class=Madison expected loss=0.3529412 P(node) =0.3469388  
## class counts: 0 1 5 11  
## probabilities: 0.000 0.059 0.294 0.647

We can see that the model has a starting node as Hamilton - 71 percent of the texts in the training set belong to Hamilton. From there, the model asks whether tho word ‘upon’ is used above 1.9 percent. If the ‘upon’ appears more frequently, then the model asigns the paper to Hamilton. But, if ‘upon’ is used less than 1.9 percent, then the model immediately ascribes the text to Madison. It is interesting that - at this point in the model - no papers are assigned to Jay or to the dual Hamilton-Madison papers. The model will have to be tuned to fix this. Furthermore, when checking the model summary, we find that at two splits, the relative error of the model is 0.429. This is not desirable as we want a more accurate model.

# We'll set a minimum split of 10 papers in a bucket, and a max depth of 4 leaf nodes, and check how the model works from there.   
feds.tree2 <- rpart(author ~ . - filename, data = train, method = 'class', control = rpart.control(cp = 0, minsplit = 10, maxdepth = 4))  
  
rpart.plot(feds.tree2)

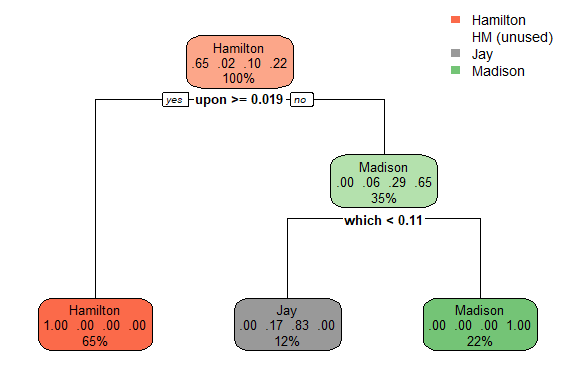


summary(feds.tree2)

## Call:  
## rpart(formula = author ~ . - filename, data = train, method = "class",   
## control = rpart.control(cp = 0, minsplit = 10, maxdepth = 4))  
## n= 49   
##   
## CP nsplit rel error xerror xstd  
## 1 0.6470588 0 1.00000000 1.0000000 0.1959984  
## 2 0.2941176 1 0.35294118 0.4117647 0.1440876  
## 3 0.0000000 2 0.05882353 0.4117647 0.1440876  
##   
## Variable importance  
## upon and there of to a which an been is no   
## 16 16 11 10 10 9 7 5 5 5 5   
##   
## Node number 1: 49 observations, complexity param=0.6470588  
## predicted class=Hamilton expected loss=0.3469388 P(node) =1  
## class counts: 32 1 5 11  
## probabilities: 0.653 0.020 0.102 0.224   
## left son=2 (32 obs) right son=3 (17 obs)  
## Primary splits:  
## upon < 0.019 to the right, improve=16.749100, (0 missing)  
## there < 0.0145 to the right, improve= 9.539541, (0 missing)  
## and < 0.3945 to the left, improve= 8.488070, (0 missing)  
## of < 0.8655 to the right, improve= 7.873469, (0 missing)  
## to < 0.4885 to the right, improve= 7.759184, (0 missing)  
## Surrogate splits:  
## and < 0.3945 to the left, agree=0.878, adj=0.647, (0 split)  
## there < 0.0145 to the right, agree=0.878, adj=0.647, (0 split)  
## of < 0.8655 to the right, agree=0.857, adj=0.588, (0 split)  
## to < 0.4885 to the right, agree=0.857, adj=0.588, (0 split)  
## a < 0.255 to the right, agree=0.837, adj=0.529, (0 split)  
##   
## Node number 2: 32 observations  
## predicted class=Hamilton expected loss=0 P(node) =0.6530612  
## class counts: 32 0 0 0  
## probabilities: 1.000 0.000 0.000 0.000   
##   
## Node number 3: 17 observations, complexity param=0.2941176  
## predicted class=Madison expected loss=0.3529412 P(node) =0.3469388  
## class counts: 0 1 5 11  
## probabilities: 0.000 0.059 0.294 0.647   
## left son=6 (6 obs) right son=7 (11 obs)  
## Primary splits:  
## which < 0.112 to the left, improve=6.686275, (0 missing)  
## of < 0.734 to the left, improve=6.519608, (0 missing)  
## the < 1.1205 to the left, improve=6.519608, (0 missing)  
## and < 0.5955 to the right, improve=6.519608, (0 missing)  
## an < 0.043 to the left, improve=5.210084, (0 missing)  
## Surrogate splits:  
## an < 0.043 to the left, agree=0.941, adj=0.833, (0 split)  
## and < 0.5115 to the right, agree=0.941, adj=0.833, (0 split)  
## been < 0.034 to the left, agree=0.941, adj=0.833, (0 split)  
## is < 0.111 to the left, agree=0.941, adj=0.833, (0 split)  
## no < 0.0305 to the left, agree=0.941, adj=0.833, (0 split)  
##   
## Node number 6: 6 observations  
## predicted class=Jay expected loss=0.1666667 P(node) =0.122449  
## class counts: 0 1 5 0  
## probabilities: 0.000 0.167 0.833 0.000   
##   
## Node number 7: 11 observations  
## predicted class=Madison expected loss=0 P(node) =0.2244898  
## class counts: 0 0 0 11  
## probabilities: 0.000 0.000 0.000 1.000

This tree is much more accurate, producing a relative error of 0.143 at two splits. We also see that the tree further elaborates beyond the Madison leaf node, going so far as to identify the Jay papers. Returning to the Madison leaf node, if the author uses ‘been’ less than 5.3 percent of the time, then it is a Jay paper. Anything more is still attributed to Madison. Unfortunately, the model is still failing to determine the Hamilton-Madison papers, and we see that it’s struggling to correctly identify the Jay papers, as only 67 percent of the papers attributed to Jay are in fact his.

# Let's reduce the minimum split to five, while also increasing the max depth to five.   
feds.tree3 <- rpart(author ~ . - filename, data = train, method = 'class', control = rpart.control(cp = 0, minsplit = 5, maxdepth = 5))  
  
rpart.plot(feds.tree3, cex = 0.8)



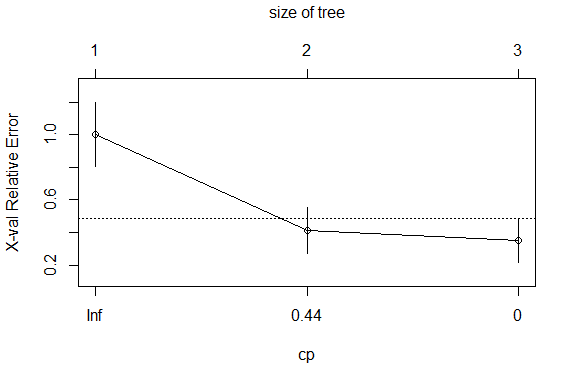
summary(feds.tree3)

## Call:  
## rpart(formula = author ~ . - filename, data = train, method = "class",   
## control = rpart.control(cp = 0, minsplit = 5, maxdepth = 5))  
## n= 49   
##   
## CP nsplit rel error xerror xstd  
## 1 0.6470588 0 1.00000000 1.0000000 0.1959984  
## 2 0.2941176 1 0.35294118 0.4117647 0.1440876  
## 3 0.0000000 2 0.05882353 0.3529412 0.1349780  
##   
## Variable importance  
## upon and there of to a which an been is no   
## 16 16 11 10 10 9 7 5 5 5 5   
##   
## Node number 1: 49 observations, complexity param=0.6470588  
## predicted class=Hamilton expected loss=0.3469388 P(node) =1  
## class counts: 32 1 5 11  
## probabilities: 0.653 0.020 0.102 0.224   
## left son=2 (32 obs) right son=3 (17 obs)  
## Primary splits:  
## upon < 0.019 to the right, improve=16.749100, (0 missing)  
## there < 0.0145 to the right, improve= 9.539541, (0 missing)  
## and < 0.3945 to the left, improve= 8.488070, (0 missing)  
## of < 0.8655 to the right, improve= 7.873469, (0 missing)  
## to < 0.4885 to the right, improve= 7.759184, (0 missing)  
## Surrogate splits:  
## and < 0.3945 to the left, agree=0.878, adj=0.647, (0 split)  
## there < 0.0145 to the right, agree=0.878, adj=0.647, (0 split)  
## of < 0.8655 to the right, agree=0.857, adj=0.588, (0 split)  
## to < 0.4885 to the right, agree=0.857, adj=0.588, (0 split)  
## a < 0.255 to the right, agree=0.837, adj=0.529, (0 split)  
##   
## Node number 2: 32 observations  
## predicted class=Hamilton expected loss=0 P(node) =0.6530612  
## class counts: 32 0 0 0  
## probabilities: 1.000 0.000 0.000 0.000   
##   
## Node number 3: 17 observations, complexity param=0.2941176  
## predicted class=Madison expected loss=0.3529412 P(node) =0.3469388  
## class counts: 0 1 5 11  
## probabilities: 0.000 0.059 0.294 0.647   
## left son=6 (6 obs) right son=7 (11 obs)  
## Primary splits:  
## which < 0.112 to the left, improve=6.686275, (0 missing)  
## of < 0.734 to the left, improve=6.519608, (0 missing)  
## the < 1.1205 to the left, improve=6.519608, (0 missing)  
## and < 0.5955 to the right, improve=6.519608, (0 missing)  
## an < 0.043 to the left, improve=5.210084, (0 missing)  
## Surrogate splits:  
## an < 0.043 to the left, agree=0.941, adj=0.833, (0 split)  
## and < 0.5115 to the right, agree=0.941, adj=0.833, (0 split)  
## been < 0.034 to the left, agree=0.941, adj=0.833, (0 split)  
## is < 0.111 to the left, agree=0.941, adj=0.833, (0 split)  
## no < 0.0305 to the left, agree=0.941, adj=0.833, (0 split)  
##   
## Node number 6: 6 observations  
## predicted class=Jay expected loss=0.1666667 P(node) =0.122449  
## class counts: 0 1 5 0  
## probabilities: 0.000 0.167 0.833 0.000   
##   
## Node number 7: 11 observations  
## predicted class=Madison expected loss=0 P(node) =0.2244898  
## class counts: 0 0 0 11  
## probabilities: 0.000 0.000 0.000 1.000

This model gives us a relative error of 0.071, which is amamzing. Most of the papers are being assigned correctly to Hamilton, Jay, and Madison. However, at the first Jay node, if the word ‘also’ appears less than one percent of the time, the model assigns the text to Hamilton with a 50 percent probability of getting it right. We can assume that both Hamilton’s and the Hamilton Madison papers are being classified here.

Let’s check the cp plot and see how the model works on the test set.

plotcp(feds.tree3)



# Now we'll use this model to predict the test set.   
test\_pred <- data.frame(predict(feds.tree3, newdata = test))  
  
# Some data transformation needs to be done to check the confusion matrix.  
results <- test\_pred %>% mutate(results = ifelse(Madison == 1, 'Madison', ifelse(Hamilton == 1, 'Hamilton', ifelse(Jay == 1, 'Jay', 'HM'))))  
# I'm introducing some bias here as I'm explicitly stating that if none of the rules apply to the results, then the result should be HM This will cause some texts which are 50-50 to be classified as HM.   
  
row.names(test) <- NULL  
testResult <- test %>% bind\_cols(results)  
testResult$results <- as.factor(testResult$results)  
  
confusionMatrix(testResult$result, testResult$author)

## Warning in confusionMatrix.default(testResult$result, testResult$author):  
## Levels are not in the same order for reference and data. Refactoring data  
## to match.

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Hamilton HM Jay Madison  
## Hamilton 18 0 0 0  
## HM 0 0 0 0  
## Jay 0 0 0 0  
## Madison 1 2 0 4  
##   
## Overall Statistics  
##   
## Accuracy : 0.88   
## 95% CI : (0.6878, 0.9745)  
## No Information Rate : 0.76   
## P-Value [Acc > NIR] : 0.1166   
##   
## Kappa : 0.7059   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: Hamilton Class: HM Class: Jay Class: Madison  
## Sensitivity 0.9474 0.00 NA 1.0000  
## Specificity 1.0000 1.00 1 0.8571  
## Pos Pred Value 1.0000 NaN NA 0.5714  
## Neg Pred Value 0.8571 0.92 NA 1.0000  
## Prevalence 0.7600 0.08 0 0.1600  
## Detection Rate 0.7200 0.00 0 0.1600  
## Detection Prevalence 0.7200 0.00 0 0.2800  
## Balanced Accuracy 0.9737 0.50 NA 0.9286

The results are really good! By checking the confusion matrix, we find that the model correctly assigns the Hamilt, Jay , and Madison texts to each author. The issue lies with the HM papers which are either HM or disputed papers. Not bad for a very simple classifier.

## 3. Prediction

But, we have one more test to actually see how the model fares: can the model predict who the disputed texts belong to?

ver\_pred <- predict(feds.tree3, newdata = ver)  
  
# We'll skip the confusion matrix in this case and just check the results.  
ver\_pred

## Hamilton HM Jay Madison  
## 1 0 0.0000000 0.0000000 1  
## 2 0 0.0000000 0.0000000 1  
## 3 0 0.1666667 0.8333333 0  
## 4 0 0.0000000 0.0000000 1  
## 5 0 0.0000000 0.0000000 1  
## 6 0 0.0000000 0.0000000 1  
## 7 0 0.0000000 0.0000000 1  
## 8 0 0.0000000 0.0000000 1  
## 9 0 0.0000000 0.0000000 1  
## 10 0 0.0000000 0.0000000 1  
## 11 0 0.0000000 0.0000000 1

We can see that, out of the 11 disputed papers, the model classifies three as authored by Madison, one by Jay, and the remaining seven as a 50/50 chance being authored either by Hamilton or Hamilton *and* Madison! Overall, not a bad piece of work by the decision tree classifier.