Homework-5

Subbu Kandhaswamy

11/6/2019

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

# Homework Instructions - Last week we used Clustering and in this week, we are using the decision tree algorithm to solve the disputed essay 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

Now loading the data. Since this is the same dataset as last week’s homework, we will skip summary, structure, and head.

dat <- read.csv("C:/Users/Administrator/Documents/MSDatascience/IST707/Week5/fedPapers85.csv")

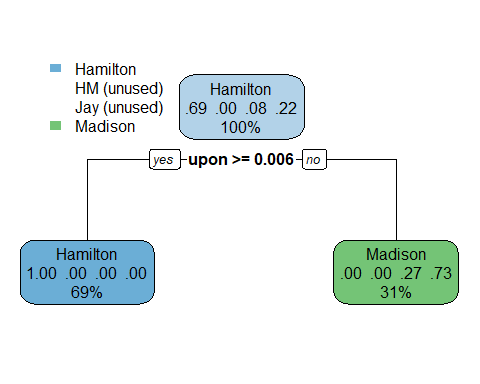
Then separate the dataset into 3 groups: Training, Testing & Verification set

Now 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 removing verification set; and the test set will be the remaining 1/3.

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")

## 2. Build and train the model.

# We'll create a first tree using rpart. and we exclude filename variable  
# considering the type (factor) and doesn't add much to the decision tree.  
feds.tree <- rpart(author ~ . - filename, data = train, method = "class", control = rpart.control(cp = 0))  
  
# Plotting to see how the model works vefore pruning  
rpart.plot(feds.tree)

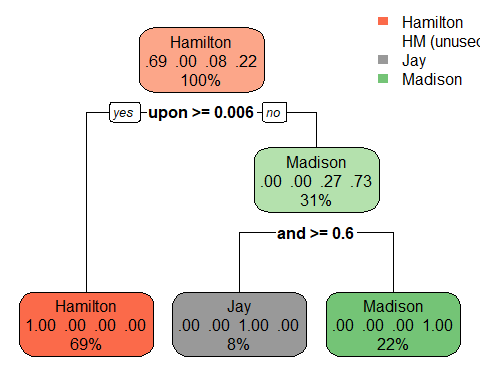


# And also look under the hood.  
summary(feds.tree)

## Call:  
## rpart(formula = author ~ . - filename, data = train, method = "class",   
## control = rpart.control(cp = 0))  
## n= 49   
##   
## CP nsplit rel error xerror xstd  
## 1 0.7333333 0 1.0000000 1.0 0.2150779  
## 2 0.0000000 1 0.2666667 0.4 0.1529750  
##   
## Variable importance  
## upon on and there to of   
## 27 18 14 14 14 13   
##   
## Node number 1: 49 observations, complexity param=0.7333333  
## predicted class=Hamilton expected loss=0.3061224 P(node) =1  
## class counts: 34 0 4 11  
## probabilities: 0.694 0.000 0.082 0.224   
## left son=2 (34 obs) right son=3 (15 obs)  
## Primary splits:  
## upon < 0.006 to the right, improve=16.745580, (0 missing)  
## on < 0.0795 to the left, improve=10.427560, (0 missing)  
## there < 0.016 to the right, improve= 8.536487, (0 missing)  
## to < 0.503 to the right, improve= 8.536487, (0 missing)  
## and < 0.4175 to the left, improve= 6.630263, (0 missing)  
## Surrogate splits:  
## on < 0.0795 to the left, agree=0.898, adj=0.667, (0 split)  
## and < 0.4175 to the left, agree=0.857, adj=0.533, (0 split)  
## there < 0.0115 to the right, agree=0.857, adj=0.533, (0 split)  
## to < 0.503 to the right, agree=0.857, adj=0.533, (0 split)  
## of < 0.7775 to the right, agree=0.837, adj=0.467, (0 split)  
##   
## Node number 2: 34 observations  
## predicted class=Hamilton expected loss=0 P(node) =0.6938776  
## class counts: 34 0 0 0  
## probabilities: 1.000 0.000 0.000 0.000   
##   
## Node number 3: 15 observations  
## predicted class=Madison expected loss=0.2666667 P(node) =0.3061224  
## class counts: 0 0 4 11  
## probabilities: 0.000 0.000 0.267 0.733

Notice 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 the 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 accredit the text to Madison. Also notice 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)

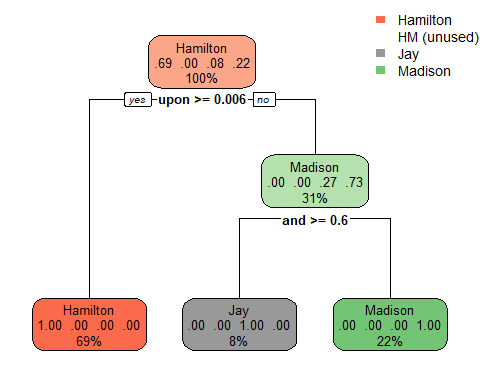


summary(feds.tree2)

## 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 0.7333333 0 1.0000000 1.0000000 0.2150779  
## 2 0.2666667 1 0.2666667 0.4000000 0.1529750  
## 3 0.0000000 2 0.0000000 0.2666667 0.1277753  
##   
## Variable importance  
## upon and of on there to an been the which   
## 17 15 14 11 9 9 6 6 6 6   
##   
## Node number 1: 49 observations, complexity param=0.7333333  
## predicted class=Hamilton expected loss=0.3061224 P(node) =1  
## class counts: 34 0 4 11  
## probabilities: 0.694 0.000 0.082 0.224   
## left son=2 (34 obs) right son=3 (15 obs)  
## Primary splits:  
## upon < 0.006 to the right, improve=16.745580, (0 missing)  
## on < 0.0795 to the left, improve=10.427560, (0 missing)  
## there < 0.016 to the right, improve= 8.536487, (0 missing)  
## to < 0.503 to the right, improve= 8.536487, (0 missing)  
## and < 0.4175 to the left, improve= 6.630263, (0 missing)  
## Surrogate splits:  
## on < 0.0795 to the left, agree=0.898, adj=0.667, (0 split)  
## and < 0.4175 to the left, agree=0.857, adj=0.533, (0 split)  
## there < 0.0115 to the right, agree=0.857, adj=0.533, (0 split)  
## to < 0.503 to the right, agree=0.857, adj=0.533, (0 split)  
## of < 0.7775 to the right, agree=0.837, adj=0.467, (0 split)  
##   
## Node number 2: 34 observations  
## predicted class=Hamilton expected loss=0 P(node) =0.6938776  
## class counts: 34 0 0 0  
## probabilities: 1.000 0.000 0.000 0.000   
##   
## Node number 3: 15 observations, complexity param=0.2666667  
## predicted class=Madison expected loss=0.2666667 P(node) =0.3061224  
## class counts: 0 0 4 11  
## probabilities: 0.000 0.000 0.267 0.733   
## left son=6 (4 obs) right son=7 (11 obs)  
## Primary splits:  
## and < 0.5955 to the right, improve=5.866667, (0 missing)  
## an < 0.043 to the left, improve=5.866667, (0 missing)  
## been < 0.027 to the left, improve=5.866667, (0 missing)  
## of < 0.697 to the left, improve=5.866667, (0 missing)  
## the < 1.098 to the left, improve=5.866667, (0 missing)  
## Surrogate splits:  
## an < 0.043 to the left, agree=1, adj=1, (0 split)  
## been < 0.027 to the left, agree=1, adj=1, (0 split)  
## of < 0.697 to the left, agree=1, adj=1, (0 split)  
## the < 1.098 to the left, agree=1, adj=1, (0 split)  
## which < 0.112 to the left, agree=1, adj=1, (0 split)  
##   
## Node number 6: 4 observations  
## predicted class=Jay expected loss=0 P(node) =0.08163265  
## class counts: 0 0 4 0  
## probabilities: 0.000 0.000 1.000 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

We can see that 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)



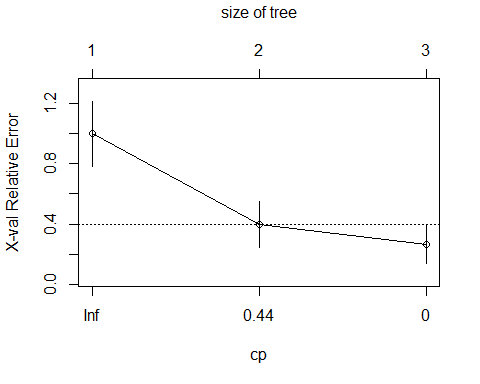
summary(feds.tree3)

## 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 0.7333333 0 1.0000000 1.0000000 0.2150779  
## 2 0.2666667 1 0.2666667 0.4000000 0.1529750  
## 3 0.0000000 2 0.0000000 0.2666667 0.1277753  
##   
## Variable importance  
## upon and of on there to an been the which   
## 17 15 14 11 9 9 6 6 6 6   
##   
## Node number 1: 49 observations, complexity param=0.7333333  
## predicted class=Hamilton expected loss=0.3061224 P(node) =1  
## class counts: 34 0 4 11  
## probabilities: 0.694 0.000 0.082 0.224   
## left son=2 (34 obs) right son=3 (15 obs)  
## Primary splits:  
## upon < 0.006 to the right, improve=16.745580, (0 missing)  
## on < 0.0795 to the left, improve=10.427560, (0 missing)  
## there < 0.016 to the right, improve= 8.536487, (0 missing)  
## to < 0.503 to the right, improve= 8.536487, (0 missing)  
## and < 0.4175 to the left, improve= 6.630263, (0 missing)  
## Surrogate splits:  
## on < 0.0795 to the left, agree=0.898, adj=0.667, (0 split)  
## and < 0.4175 to the left, agree=0.857, adj=0.533, (0 split)  
## there < 0.0115 to the right, agree=0.857, adj=0.533, (0 split)  
## to < 0.503 to the right, agree=0.857, adj=0.533, (0 split)  
## of < 0.7775 to the right, agree=0.837, adj=0.467, (0 split)  
##   
## Node number 2: 34 observations  
## predicted class=Hamilton expected loss=0 P(node) =0.6938776  
## class counts: 34 0 0 0  
## probabilities: 1.000 0.000 0.000 0.000   
##   
## Node number 3: 15 observations, complexity param=0.2666667  
## predicted class=Madison expected loss=0.2666667 P(node) =0.3061224  
## class counts: 0 0 4 11  
## probabilities: 0.000 0.000 0.267 0.733   
## left son=6 (4 obs) right son=7 (11 obs)  
## Primary splits:  
## and < 0.5955 to the right, improve=5.866667, (0 missing)  
## an < 0.043 to the left, improve=5.866667, (0 missing)  
## been < 0.027 to the left, improve=5.866667, (0 missing)  
## of < 0.697 to the left, improve=5.866667, (0 missing)  
## the < 1.098 to the left, improve=5.866667, (0 missing)  
## Surrogate splits:  
## an < 0.043 to the left, agree=1, adj=1, (0 split)  
## been < 0.027 to the left, agree=1, adj=1, (0 split)  
## of < 0.697 to the left, agree=1, adj=1, (0 split)  
## the < 1.098 to the left, agree=1, adj=1, (0 split)  
## which < 0.112 to the left, agree=1, adj=1, (0 split)  
##   
## Node number 6: 4 observations  
## predicted class=Jay expected loss=0 P(node) =0.08163265  
## class counts: 0 0 4 0  
## probabilities: 0.000 0.000 1.000 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 amazing. 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 predict the test set using this model.  
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"))))  
# let’s introduce some bias here to explicitly statethat 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 17 2 1 2  
## HM 0 0 0 0  
## Jay 0 0 0 0  
## Madison 0 1 0 2  
##   
## Overall Statistics  
##   
## Accuracy : 0.76   
## 95% CI : (0.5487, 0.9064)  
## No Information Rate : 0.68   
## P-Value [Acc > NIR] : 0.2657   
##   
## Kappa : 0.3724   
##   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: Hamilton Class: HM Class: Jay Class: Madison  
## Sensitivity 1.0000 0.00 0.00 0.5000  
## Specificity 0.3750 1.00 1.00 0.9524  
## Pos Pred Value 0.7727 NaN NaN 0.6667  
## Neg Pred Value 1.0000 0.88 0.96 0.9091  
## Prevalence 0.6800 0.12 0.04 0.1600  
## Detection Rate 0.6800 0.00 0.00 0.0800  
## Detection Prevalence 0.8800 0.00 0.00 0.1200  
## Balanced Accuracy 0.6875 0.50 0.50 0.7262

Reviewing the confusion matrix, we could see that that the model correctly assigns the Hamilt, Jay , and Madison texts to each author. Also confirms the issue lies with the HM papers (HM or disputed papers).

## 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)  
  
# Skipping the confusion matrix in this case and just check the results.  
ver\_pred

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

To conclude, out of the 11 disputed papers, the model classifies 3 as authored by Madison, one by Jay, and the remaining as a 50/50 chance being authored either by Hamilton or Hamilton & Madison.

In comparison, last week using clustering technique was less effective, where there was less evidence to determine the authors where the average method showed that some of the papers were more likely to be authored by Hamilton, some by Madison, and a couple that couldn’t be determined since the language used by the authors were closely similar.