George_Smith_HW5_IST772

George Smith

8/8/2021

Introduction

The purpose of this assignment is to use the decision tree algorithm to solve the disputed essay problem. Previously clustering techniques were used to tackle this problem, however this time around I will use decision tree algorithm techniques to see if I get the same results.

Installing packages

#install.packages("rattle") #install.packages("rpart.plot")

```
library(tm)
```

```
## Loading required package: NLP
```

```
library(stringr)
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
library(stringi)
library(Matrix)
library(tidytext)
library(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
```

```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:NLP':
##
##     annotate
```

```
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(rpart)
library(rattle)
```

```
## Loading required package: tibble

## Loading required package: bitops

##
## Attaching package: 'bitops'

## The following object is masked from 'package:Matrix':
##
##     %&%

## Rattle: A free graphical interface for data science with R.
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(rpart.plot)
library(RColorBrewer)
library(tidyr)
```

```
##
## Attaching package: 'tidyr'

## The following objects are masked from 'package:Matrix':
##
##     expand, pack, unpack
```

```
library(caret)
```

```
## Loading required package: lattice
```

#Load the data # Below, loading data (Federalist Papers) in Corpus format. #### Load Fed Papers Corpus

```
setwd("C:/Users/GeorgeSmith/Documents")
FedPapersCorpus <- Corpus(DirSource("FedPapersCorpus"))
#checks to see if it was loaded correctly - commented out after 1st run
(numberFedPapers<-length(FedPapersCorpus))
```

```
## [1] 85
```

```
(summary(FedPapersCorpus))
```

```
##                      Length Class             Mode
## dispt_fed_49.txt     2      PlainTextDocument list
## dispt_fed_50.txt     2      PlainTextDocument list
## dispt_fed_51.txt     2      PlainTextDocument list
## dispt_fed_52.txt     2      PlainTextDocument list
## dispt_fed_53.txt     2      PlainTextDocument list
## dispt_fed_54.txt     2      PlainTextDocument list
## dispt_fed_55.txt     2      PlainTextDocument list
## dispt_fed_56.txt     2      PlainTextDocument list
## dispt_fed_57.txt     2      PlainTextDocument list
## dispt_fed_62.txt     2      PlainTextDocument list
## dispt_fed_63.txt     2      PlainTextDocument list
## Hamilton_fed_1.txt   2      PlainTextDocument list
## Hamilton_fed_11.txt  2      PlainTextDocument list
## Hamilton_fed_12.txt  2      PlainTextDocument list
## Hamilton_fed_13.txt  2      PlainTextDocument list
## Hamilton_fed_15.txt  2      PlainTextDocument list
## Hamilton_fed_16.txt  2      PlainTextDocument list
## Hamilton_fed_17.txt  2      PlainTextDocument list
## Hamilton_fed_21.txt  2      PlainTextDocument list
## Hamilton_fed_22.txt  2      PlainTextDocument list
## Hamilton_fed_23.txt  2      PlainTextDocument list
## Hamilton_fed_24.txt  2      PlainTextDocument list
## Hamilton_fed_25.txt  2      PlainTextDocument list
## Hamilton_fed_26.txt  2      PlainTextDocument list
## Hamilton_fed_27.txt  2      PlainTextDocument list
## Hamilton_fed_28.txt  2      PlainTextDocument list
## Hamilton_fed_29.txt  2      PlainTextDocument list
## Hamilton_fed_30.txt  2      PlainTextDocument list
## Hamilton_fed_31.txt  2      PlainTextDocument list
## Hamilton_fed_32.txt  2      PlainTextDocument list
## Hamilton_fed_33.txt  2      PlainTextDocument list
## Hamilton_fed_34.txt  2      PlainTextDocument list
## Hamilton_fed_35.txt  2      PlainTextDocument list
## Hamilton_fed_36.txt  2      PlainTextDocument list
## Hamilton_fed_59.txt  2      PlainTextDocument list
## Hamilton_fed_6.txt   2      PlainTextDocument list
## Hamilton_fed_60.txt  2      PlainTextDocument list
## Hamilton_fed_61.txt  2      PlainTextDocument list
## Hamilton_fed_65.txt  2      PlainTextDocument list
## Hamilton_fed_66.txt  2      PlainTextDocument list
## Hamilton_fed_67.txt  2      PlainTextDocument list
## Hamilton_fed_68.txt  2      PlainTextDocument list
```

```
## Hamilton_fed_69.txt 2      PlainTextDocument list
## Hamilton_fed_7.txt  2      PlainTextDocument list
## Hamilton_fed_70.txt 2      PlainTextDocument list
## Hamilton_fed_71.txt 2      PlainTextDocument list
## Hamilton_fed_72.txt 2      PlainTextDocument list
## Hamilton_fed_73.txt 2      PlainTextDocument list
## Hamilton_fed_74.txt 2      PlainTextDocument list
## Hamilton_fed_75.txt 2      PlainTextDocument list
## Hamilton_fed_76.txt 2      PlainTextDocument list
## Hamilton_fed_77.txt 2      PlainTextDocument list
## Hamilton_fed_78.txt 2      PlainTextDocument list
## Hamilton_fed_79.txt 2      PlainTextDocument list
## Hamilton_fed_8.txt  2      PlainTextDocument list
## Hamilton_fed_80.txt 2      PlainTextDocument list
## Hamilton_fed_81.txt 2      PlainTextDocument list
## Hamilton_fed_82.txt 2      PlainTextDocument list
## Hamilton_fed_83.txt 2      PlainTextDocument list
## Hamilton_fed_84.txt 2      PlainTextDocument list
## Hamilton_fed_85.txt 2      PlainTextDocument list
## Hamilton_fed_9.txt  2      PlainTextDocument list
## HM_fed_18.txt       2      PlainTextDocument list
## HM_fed_19.txt       2      PlainTextDocument list
## HM_fed_20.txt       2      PlainTextDocument list
## Jay_fed_2.txt       2      PlainTextDocument list
## Jay_fed_3.txt       2      PlainTextDocument list
## Jay_fed_4.txt       2      PlainTextDocument list
## Jay_fed_5.txt       2      PlainTextDocument list
## Jay_fed_64.txt      2      PlainTextDocument list
## Madison_fed_10.txt  2      PlainTextDocument list
## Madison_fed_14.txt  2      PlainTextDocument list
## Madison_fed_37.txt  2      PlainTextDocument list
## Madison_fed_38.txt  2      PlainTextDocument list
## Madison_fed_39.txt  2      PlainTextDocument list
## Madison_fed_40.txt  2      PlainTextDocument list
## Madison_fed_41.txt  2      PlainTextDocument list
## Madison_fed_42.txt  2      PlainTextDocument list
## Madison_fed_43.txt  2      PlainTextDocument list
## Madison_fed_44.txt  2      PlainTextDocument list
## Madison_fed_45.txt  2      PlainTextDocument list
## Madison_fed_46.txt  2      PlainTextDocument list
## Madison_fed_47.txt  2      PlainTextDocument list
## Madison_fed_48.txt  2      PlainTextDocument list
## Madison_fed_58.txt  2      PlainTextDocument list
```

```
(meta(FedPapersCorpus[[1]]))
```

```
##   author       : character(0)
##   datetimestamp: 2021-08-08 21:47:38
##   description  : character(0)
##   heading      : character(0)
##   id           : dispt_fed_49.txt
##   language     : en
##   origin       : character(0)
```

```
(meta(FedPapersCorpus[[1]],5))
```

```
## [1] "dispt_fed_49.txt"
```

Cleaning and Preparing

#Choosing some good stop words can really go a long way to improve modeling results. There are also many #other parameters one can tweak and tune using the DocumentTermMatrix function. See many below. #Data Preparation and Transformation on Fed Papers

##Remove punctuation,numbers, and space

```
(getTransformations())
```

```
## [1] "removeNumbers"     "removePunctuation" "removeWords"
## [4] "stemDocument"      "stripWhitespace"
```

```
(nFedPapersCorpus<-length(FedPapersCorpus))
```

```
## [1] 85
```

```
(minTermFreq <-30)
```

```
## [1] 30
```

```
(maxTermFreq <-1000)
```

```
## [1] 1000
```

Create a personalized list of stop words

```
MyStopwords <- c("will","one","two", "may","less","publius","Madison","Alexand", "Alexander", "James",
                 "without", "small", "single" ,"several", "but", "very", "can", "must", "also", "any",
                 "almost", "for", "add", "Author")

STOPS <-stopwords('english')

Papers_DTM <- DocumentTermMatrix(FedPapersCorpus,
                                 control = list(stopwords = TRUE, wordLengths=c(3, 15),
                                 removePunctuation = T, removeNumbers = T,
                                 tolower=T, stemming = T,
                                 remove_separators = T,
                                 stopwords = MyStopwords,
                                 removeWords=STOPS,
                                 removeWords=MyStopwords,
                                 bounds = list(global = c(minTermFreq, maxTermFreq))
                                                 ))
```

##inspect FedPapers Document Term Matrix (DTM)

```
DTM <- as.matrix(Papers_DTM)
```

Confirming 1st 11 are disputed

```
(DTM[1:11,1:10])
```

```
##                   Terms
## Docs               abl absolut accord act addit administr admit adopt advantag
##   dispt_fed_49.txt   2       0      0   0     0         1     1     0        4
##   dispt_fed_50.txt   0       2      0   0     0         2     0     0        1
##   dispt_fed_51.txt   1       2      0   0     1         1     3     0        0
##   dispt_fed_52.txt   1       1      0   1     1         0     0     1        2
##   dispt_fed_53.txt   0       0      1   2     0         0     1     0        2
##   dispt_fed_54.txt   0       0      2   1     0         0     5     1        4
##   dispt_fed_55.txt   0       0      2   0     0         0     2     0        0
##   dispt_fed_56.txt   0       0      1   1     0         0     0     0        1
##   dispt_fed_57.txt   0       0      1   0     1         1     1     0        0
##   dispt_fed_62.txt   1       0      0   1     1         0     0     1        7
##   dispt_fed_63.txt   4       0      1   3     1         1     1     0        5
##                   Terms
## Docs               affair
##   dispt_fed_49.txt      0
##   dispt_fed_50.txt      0
##   dispt_fed_51.txt      1
##   dispt_fed_52.txt      0
##   dispt_fed_53.txt      9
##   dispt_fed_54.txt      0
##   dispt_fed_55.txt      1
##   dispt_fed_56.txt      5
##   dispt_fed_57.txt      0
##   dispt_fed_62.txt      4
##   dispt_fed_63.txt      1
```

Vectorization

Vectorizing words is often done by encoding frequency information. Below we take a peak at the frequency

of the words. Next some normalization techniques are tried. Which works best . . . ?? Try many and assess

the results!!!

##Look at word frequencies

```r
WordFreq <- colSums(as.matrix(Papers_DTM))
(head(WordFreq, 20))
```

```
##       abl   absolut    accord       act     addit administr     admit     adopt
##        74        63        71       139        61        90       107        57
##  advantag    affair    affect    afford   alexand    almost      alon   alreadi
##       142        65        56        64        67        45        70        56
##      also     alway   america     among
##        96        84       114       131
```

```r
(length(WordFreq))
```

```
## [1] 427
```

```r
ord <- order(WordFreq)
(WordFreq[head(ord, 20)])
```

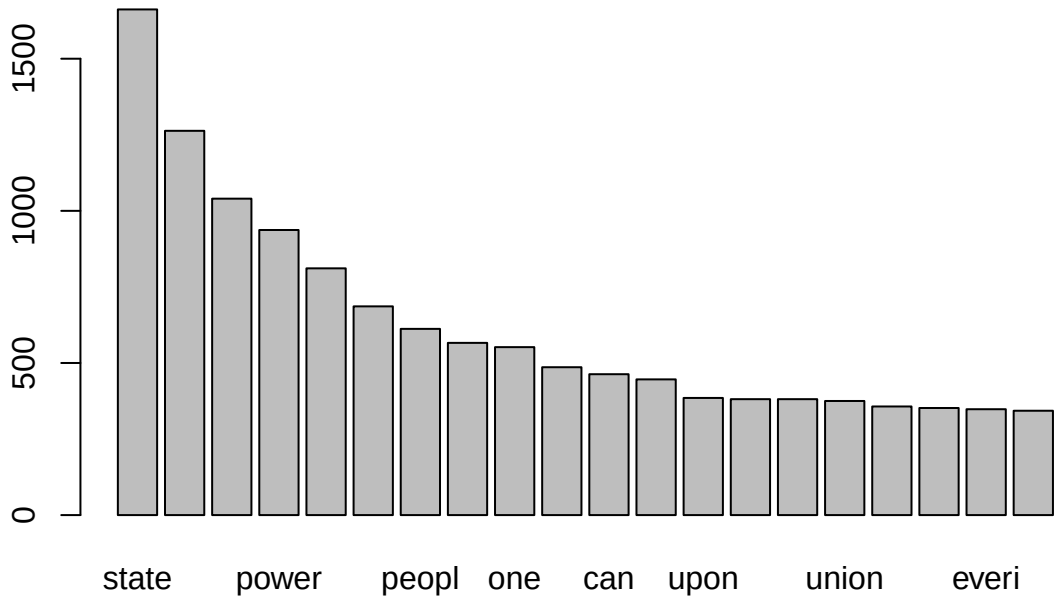
```
##      jame     expos   furnish      word    unless     bound  descript     drawn
##        30        34        36        36        37        38        38        38
##      leav    design     fulli   tendenc    applic apprehens     avoid   portion
##        38        39        39        39        40        40        40        40
##    preced   foundat    extrem      fall
##        40        41        42        42
```

```r
(WordFreq[tail(ord)])
```

```
## constitut       may     power    govern      will     state
##       686       811       937      1040      1263      1662
```

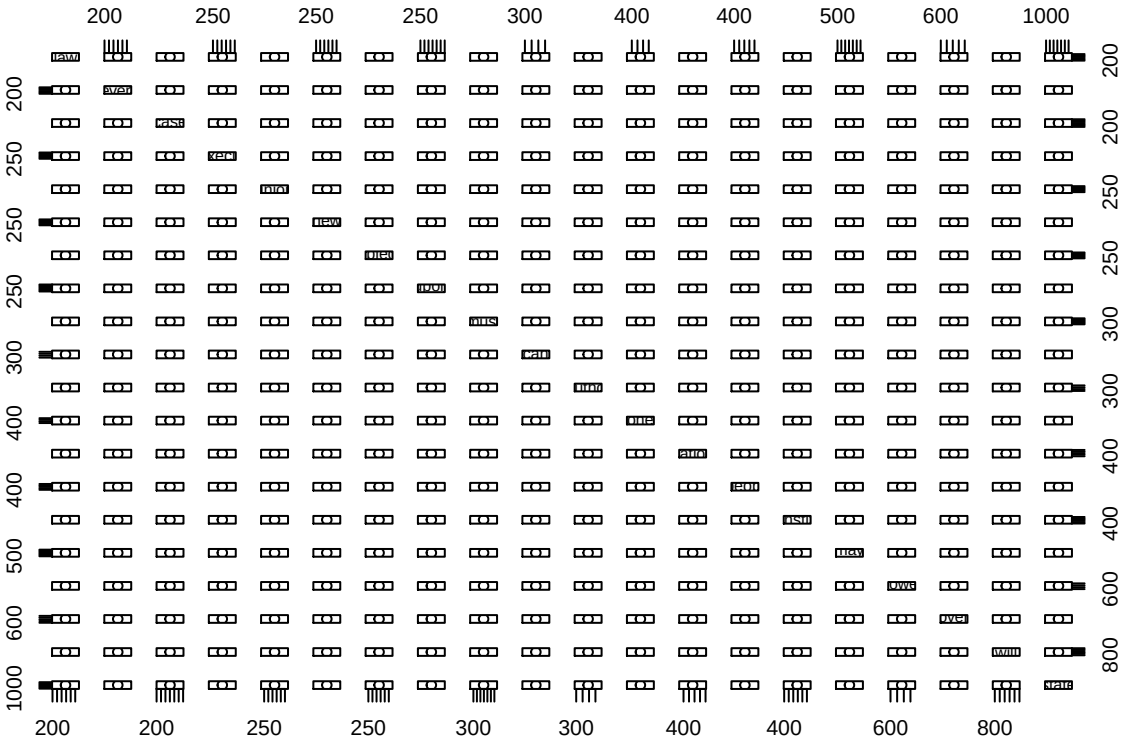
Creating a barplot for the top 20 words

```r
barplot(head(sort(WordFreq, decreasing = T),20))
```

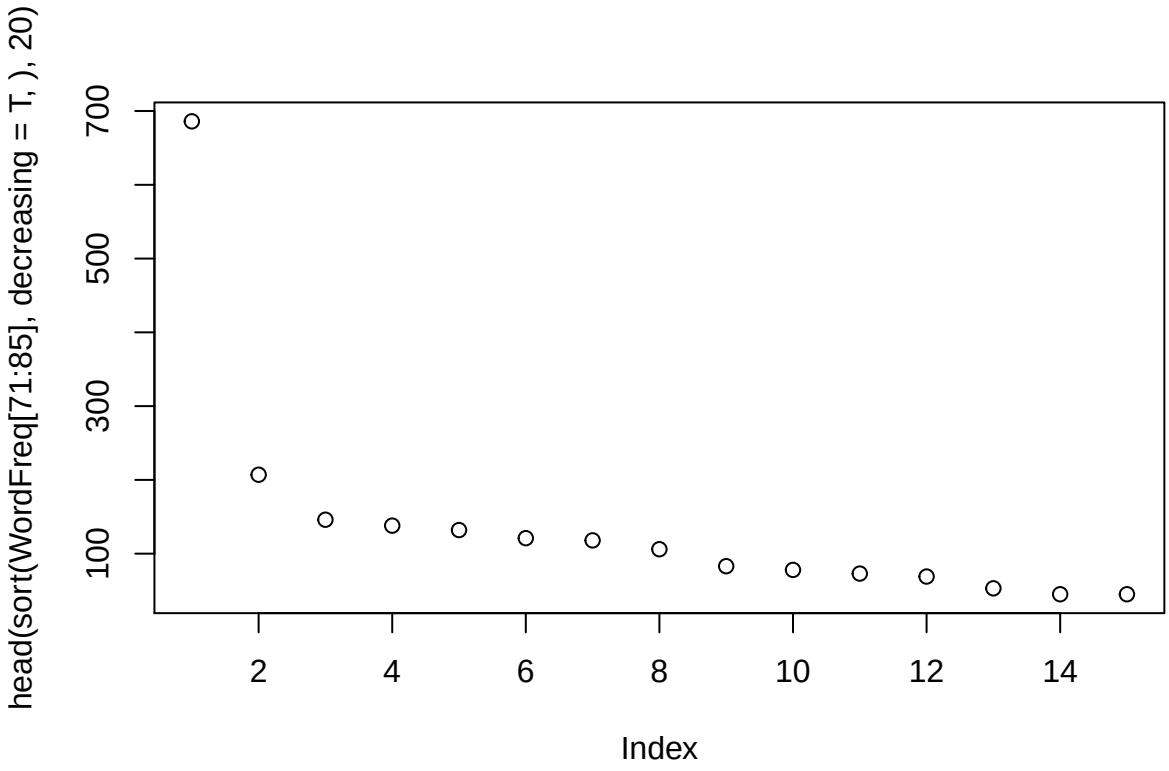


Creating aplot for the top 20 words

```
WF_2 <- t(WordFreq[tail(ord, 20, decreasing = F)])
plot(as.data.frame(WF_2))
```



```
plot(head(sort(WordFreq[71:85], decreasing = T, ),20),xlab = colnames(WordFreq[71:85]))
```



Row Sums per Fed Papers

```
Row_Sum_Per_doc <- rowSums((as.matrix(Papers_DTM)))
```

Create a normalized version of Papers_DTM

```
Papers_M <- as.matrix(Papers_DTM)
Papers_M_N1 <- apply(Papers_M, 1, function(i) round(i/sum(i),3))
Papers_Matrix_Norm <- t(Papers_M_N1)
                ## Convert to matrix and view
Papers_dtm_matrix = as.matrix(Papers_DTM)
str(Papers_dtm_matrix)
```

```
##  num [1:85, 1:427] 2 0 1 1 0 0 0 0 0 1 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ Docs : chr [1:85] "dispt_fed_49.txt" "dispt_fed_50.txt" "dispt_fed_51.txt" "dispt_fed_52.txt"
##   ..$ Terms: chr [1:427] "abl" "absolut" "accord" "act" ...
```

```
(Papers_dtm_matrix[c(1:11),c(2:10)])
```

```
##                   Terms
## Docs               absolut accord act addit administr admit adopt advantag
##   dispt_fed_49.txt       0      0   0     0         1     1     0        4
```

```
##   dispt_fed_50.txt       2      0   0     0         2     0     0        1
##   dispt_fed_51.txt       2      0   0     1         1     3     0        0
##   dispt_fed_52.txt       1      0   1     1         0     0     1        2
##   dispt_fed_53.txt       0      1   2     0         0     1     0        2
##   dispt_fed_54.txt       0      2   1     0         0     5     1        4
##   dispt_fed_55.txt       0      2   0     0         0     2     0        0
##   dispt_fed_56.txt       0      1   1     0         0     0     0        1
##   dispt_fed_57.txt       0      1   0     1         1     1     0        0
##   dispt_fed_62.txt       0      0   1     1         0     0     1        7
##   dispt_fed_63.txt       0      1   3     1         1     1     0        5
##                   Terms
## Docs               affair
##   dispt_fed_49.txt      0
##   dispt_fed_50.txt      0
##   dispt_fed_51.txt      1
##   dispt_fed_52.txt      0
##   dispt_fed_53.txt      9
##   dispt_fed_54.txt      0
##   dispt_fed_55.txt      1
##   dispt_fed_56.txt      5
##   dispt_fed_57.txt      0
##   dispt_fed_62.txt      4
##   dispt_fed_63.txt      1
```

```
Papers_DF <- as.data.frame(as.matrix(Papers_Matrix_Norm))
Papers_DF1<- Papers_DF%>%add_rownames()
```

```
## Warning: `add_rownames()` was deprecated in dplyr 1.0.0.
## Please use `tibble::rownames_to_column()` instead.
```

Labeling the data only for Hamilton and Madison.

```
names(Papers_DF1)[1]="Author"
Papers_DF1[1:11,1]="dispt"
Papers_DF1[12:62,1]="hamil"
Papers_DF1[63:85,1]="madis"
head(Papers_DF1)
```

```
## # A tibble: 6 x 428
##   Author   abl absolut accord   act addit administr admit adopt advantag affair
##   <chr>  <dbl>   <dbl>  <dbl> <dbl> <dbl>     <dbl> <dbl> <dbl>    <dbl>  <dbl>
## 1 dispt  0.004   0      0     0     0         0.002 0.002 0        0.008  0
## 2 dispt  0       0.006  0     0     0         0.006 0     0        0.003  0
## 3 dispt  0.002   0.003  0     0     0.002     0.002 0.005 0        0      0.002
## 4 dispt  0.002   0.002  0     0.002 0.002     0     0     0.002    0.004  0
## 5 dispt  0       0      0.001 0.003 0         0     0.001 0        0.003  0.013
## 6 dispt  0       0      0.003 0.002 0         0     0.009 0.002    0.007  0
## # ... with 417 more variables: affect <dbl>, afford <dbl>, alexand <dbl>,
## #   almost <dbl>, alon <dbl>, alreadi <dbl>, also <dbl>, alway <dbl>,
## #   america <dbl>, among <dbl>, amount <dbl>, anoth <dbl>, answer <dbl>,
```

```
## #   appear <dbl>, appli <dbl>, applic <dbl>, appoint <dbl>, apprehens <dbl>,
## #   argument <dbl>, aris <dbl>, articl <dbl>, assembl <dbl>, attempt <dbl>,
## #   attend <dbl>, attent <dbl>, author <dbl>, avoid <dbl>, becom <dbl>,
## #   best <dbl>, better <dbl>, bodi <dbl>, bound <dbl>, branch <dbl>,
## #   britain <dbl>, calcul <dbl>, call <dbl>, can <dbl>, capac <dbl>,
## #   care <dbl>, carri <dbl>, case <dbl>, caus <dbl>, certain <dbl>,
## #   chang <dbl>, charact <dbl>, circumst <dbl>, citizen <dbl>, civil <dbl>,
## #   class <dbl>, clear <dbl>, collect <dbl>, combin <dbl>, commit <dbl>,
## #   common <dbl>, communiti <dbl>, complet <dbl>, compos <dbl>, concern <dbl>,
## #   conclus <dbl>, conduct <dbl>, confeder <dbl>, confederaci <dbl>,
## #   confid <dbl>, confin <dbl>, congress <dbl>, connect <dbl>, consequ <dbl>,
## #   consid <dbl>, consider <dbl>, consist <dbl>, constitu <dbl>,
## #   constitut <dbl>, contend <dbl>, continu <dbl>, contrari <dbl>,
## #   control <dbl>, convent <dbl>, council <dbl>, countri <dbl>, cours <dbl>,
## #   danger <dbl>, decid <dbl>, decis <dbl>, declar <dbl>, defect <dbl>,
## #   defens <dbl>, degre <dbl>, deliber <dbl>, depart <dbl>, depend <dbl>,
## #   deriv <dbl>, descript <dbl>, design <dbl>, desir <dbl>, determin <dbl>,
## #   differ <dbl>, difficulti <dbl>, direct <dbl>, dispos <dbl>, disposit <dbl>,
## #   ...
```

Experimental Design

Randomly selecting training (train) and testing (test) data sets using function: sample.int() .

```
(head(sort(as.matrix(Papers_dtm_matrix)[11,], decreasing = TRUE), n=50))
```

```
##      peopl      senat       will        may     repres     govern       bodi
##         42         24         19         18         18         16         15
##        can      elect       must     measur      state     nation        one
##         14         14         12         11         11          9          9
##  constitut     former      power     reason       year    assembl     exampl
##          8          8          8          8          8          7          7
##        two     danger      everi       evid      feder     import     latter
##          7          6          6          6          6          6          6
##     object particular     public   advantag     answer     appear     author
##          6          6          6          5          5          5          5
##    charact       fact      first       hous   institut       less       mani
##          5          5          5          5          5          5          5
##     member      might       oper      order       part    popular    probabl
##          5          5          5          5          5          5          5
##      small
##          5
```

##Make Train and Test sets

```
numDisputed = 11
numTotalPapers = nrow(Papers_DF1)

trainRatio <- .60
set.seed(11) # Set Seed so that same sample can be reproduced in future also

sample <- sample.int(n = numTotalPapers-numDisputed, size = floor(trainRatio*numTotalPapers), replace =
newSample = sample + numDisputed

train <- Papers_DF1[newSample, ]
test <- Papers_DF1[-newSample, ]
```

train / test ratio

```
length(newSample)/nrow(Papers_DF1)
```

```
## [1] 0.6
```

Classification

Training and testing using classifiers

And using different decision tree models, parameters and pruning

Using fancyRpartPlot to visualize the learned tree models.

##Decision Tree Models #Train Tree Model 1

```
train_tree1 <- rpart(Author ~ ., data = train, method="class", control=rpart.control(cp=0))
summary(train_tree1)
```

```
## Call:
## rpart(formula = Author ~ ., data = train, method = "class", control = rpart.control(cp = 0))
##   n= 51
##
##       CP nsplit rel error xerror      xstd
## 1 0.9375      0    1.0000 1.0000 0.2071042
## 2 0.0000      1    0.0625 0.4375 0.1535926
##
## Variable importance
##     upon  alexand hamilton     jame     form     also
##       24       19       19       19       10        8
##
## Node number 1: 51 observations,    complexity param=0.9375
##   predicted class=hamil  expected loss=0.3137255  P(node) =1
```

```
##     class counts:    35    16
##    probabilities: 0.686 0.314
##   left son=2 (36 obs) right son=3 (15 obs)
##   Primary splits:
##       upon     < 0.0035 to the right, improve=20.016340, (0 missing)
##       alexand  < 5e-04  to the right, improve=18.177000, (0 missing)
##       hamilton < 5e-04  to the right, improve=18.177000, (0 missing)
##       jame     < 5e-04  to the left,  improve=18.177000, (0 missing)
##       york     < 0.0025 to the right, improve= 7.843137, (0 missing)
##   Surrogate splits:
##       alexand  < 5e-04  to the right, agree=0.941, adj=0.800, (0 split)
##       hamilton < 5e-04  to the right, agree=0.941, adj=0.800, (0 split)
##       jame     < 5e-04  to the left,  agree=0.941, adj=0.800, (0 split)
##       form     < 0.0065 to the left,  agree=0.824, adj=0.400, (0 split)
##       also     < 0.0035 to the left,  agree=0.804, adj=0.333, (0 split)
##
## Node number 2: 36 observations
##   predicted class=hamil  expected loss=0.02777778  P(node) =0.7058824
##     class counts:    35     1
##    probabilities: 0.972 0.028
##
## Node number 3: 15 observations
##   predicted class=madis  expected loss=0  P(node) =0.2941176
##     class counts:     0    15
##    probabilities: 0.000 1.000
```

#predict the test dataset using the model for train tree No. 1

```
predicted1= predict(train_tree1, test, type="class")
(Results1 <- data.frame(Actual=test$Author, TrainTreeModel1 = predicted1))
```

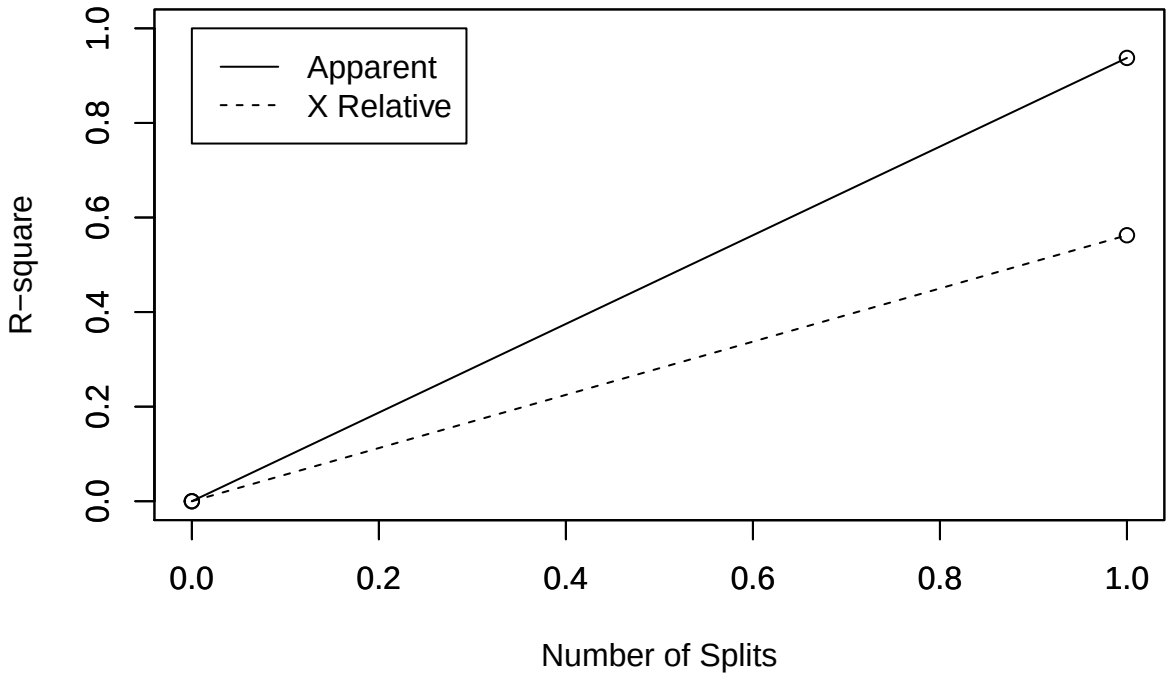
```
##    Actual TrainTreeModel1
## 1   dispt           madis
## 2   dispt           madis
## 3   dispt           madis
## 4   dispt           madis
## 5   dispt           madis
## 6   dispt           madis
## 7   dispt           madis
## 8   dispt           madis
## 9   dispt           madis
## 10  dispt           madis
## 11  dispt           madis
## 12  hamil           hamil
## 13  hamil           hamil
## 14  hamil           hamil
## 15  hamil           hamil
## 16  hamil           hamil
## 17  hamil           hamil
## 18  hamil           hamil
## 19  hamil           hamil
## 20  hamil           hamil
## 21  hamil           hamil
```

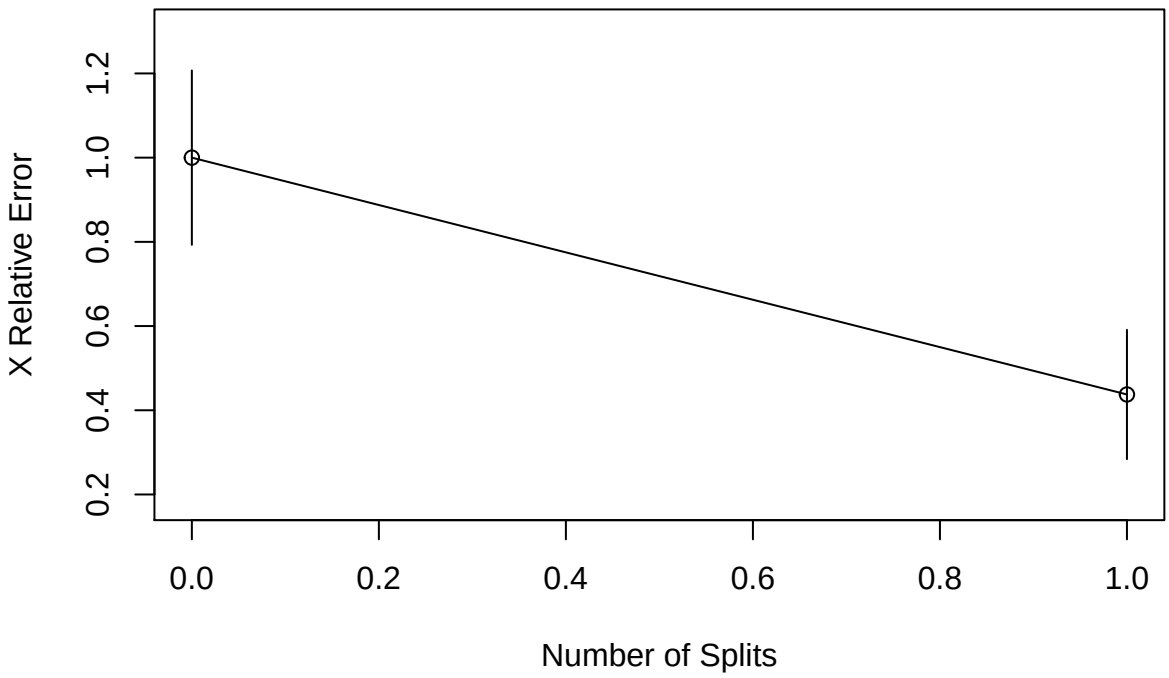
```
## 22  hamil           hamil
## 23  hamil           hamil
## 24  hamil           hamil
## 25  hamil           hamil
## 26  hamil           hamil
## 27  hamil           hamil
## 28  madis           madis
## 29  madis           madis
## 30  madis           madis
## 31  madis           madis
## 32  madis           madis
## 33  madis           madis
## 34  madis           madis
```

#plot number of splits

```
rsq.rpart(train_tree1)
```

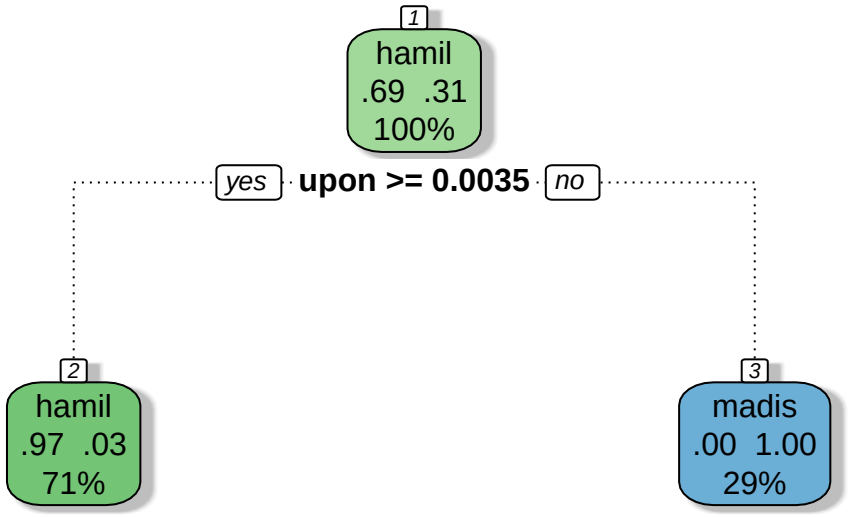
```
##
## Classification tree:
## rpart(formula = Author ~ ., data = train, method = "class", control = rpart.control(cp = 0))
##
## Variables actually used in tree construction:
## [1] upon
##
## Root node error: 16/51 = 0.31373
##
## n= 51
##
##       CP nsplit rel error xerror    xstd
## 1 0.9375      0    1.0000 1.0000 0.20710
## 2 0.0000      1    0.0625 0.4375 0.15359

## Warning in rsq.rpart(train_tree1): may not be applicable for this method
```


Apparent
X Relative
R−square
0.0
0.2
0.4
0.6
0.8
1.0
Number of Splits



#plot the decision tree

```
fancyRpartPlot(train_tree1)
```



#confusion matrix to find correct and incorrect predictions

```
table(Authorship=predicted1, true=test$Author)
```

```
##           true
## Authorship dispt hamil madis
##      hamil     0    16     0
##      madis    11     0     7
```

from this point below we try different parameters

```
train_tree2 <- rpart(Author ~ ., data = train, method="class", control=rpart.control(cp=0), minsplit = 1
(summary(train_tree2))
```

```
## Call:
## rpart(formula = Author ~ ., data = train, method = "class", control = rpart.control(cp = 0),
##     minsplit = 2, maxdepth = 5)
##   n= 51
##
##       CP nsplit rel error xerror      xstd
## 1 0.9375      0    1.0000  1.000 0.2071042
## 2 0.0000      1    0.0625  0.375 0.1438059
##
```

```
## Variable importance
##     upon  alexand hamilton     jame     form     also
##       24       19       19       19       10        8
##
## Node number 1: 51 observations,    complexity param=0.9375
##   predicted class=hamil  expected loss=0.3137255  P(node) =1
##     class counts:    35    16
##    probabilities: 0.686 0.314
##   left son=2 (36 obs) right son=3 (15 obs)
##   Primary splits:
##       upon     < 0.0035 to the right, improve=20.016340, (0 missing)
##       alexand  < 5e-04  to the right, improve=18.177000, (0 missing)
##       hamilton < 5e-04  to the right, improve=18.177000, (0 missing)
##       jame     < 5e-04  to the left,  improve=18.177000, (0 missing)
##       york     < 0.0025 to the right, improve= 7.843137, (0 missing)
##   Surrogate splits:
##       alexand  < 5e-04  to the right, agree=0.941, adj=0.800, (0 split)
##       hamilton < 5e-04  to the right, agree=0.941, adj=0.800, (0 split)
##       jame     < 5e-04  to the left,  agree=0.941, adj=0.800, (0 split)
##       form     < 0.0065 to the left,  agree=0.824, adj=0.400, (0 split)
##       also     < 0.0035 to the left,  agree=0.804, adj=0.333, (0 split)
##
## Node number 2: 36 observations
##   predicted class=hamil  expected loss=0.02777778  P(node) =0.7058824
##     class counts:    35     1
##    probabilities: 0.972 0.028
##
## Node number 3: 15 observations
##   predicted class=madis  expected loss=0  P(node) =0.2941176
##     class counts:     0    15
##    probabilities: 0.000 1.000

## n= 51
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 51 16 hamil (0.68627451 0.31372549)
##   2) upon>=0.0035 36  1 hamil (0.97222222 0.02777778) *
##   3) upon< 0.0035 15  0 madis (0.00000000 1.00000000) *
```

#predict the test dataset using the model for train tree No. 1

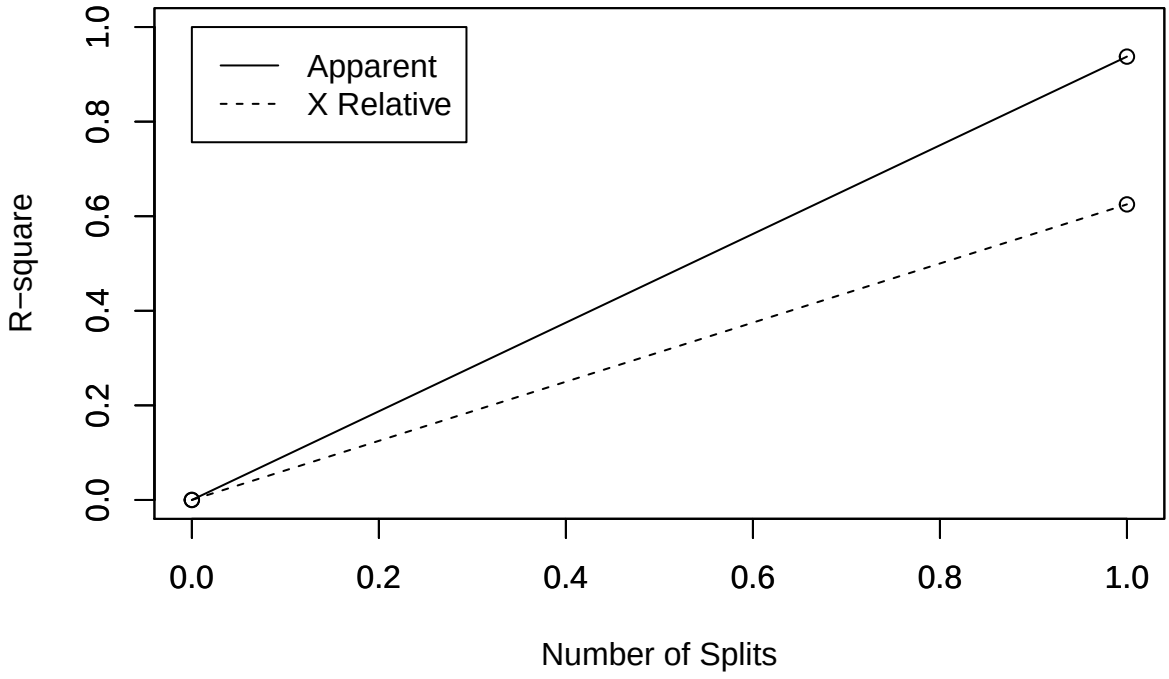
```
predicted2= predict(train_tree2, test, type="class")
```

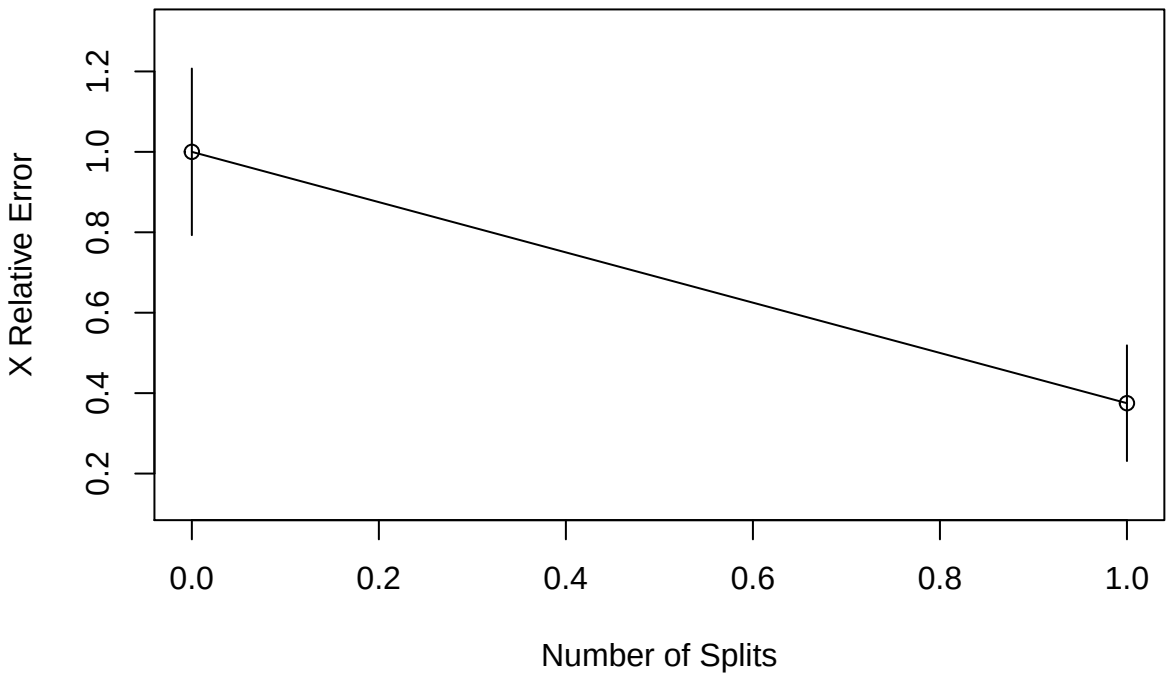
#plot number of splits

```
rsq.rpart(train_tree2)
```

```
##
## Classification tree:
## rpart(formula = Author ~ ., data = train, method = "class", control = rpart.control(cp = 0),
```

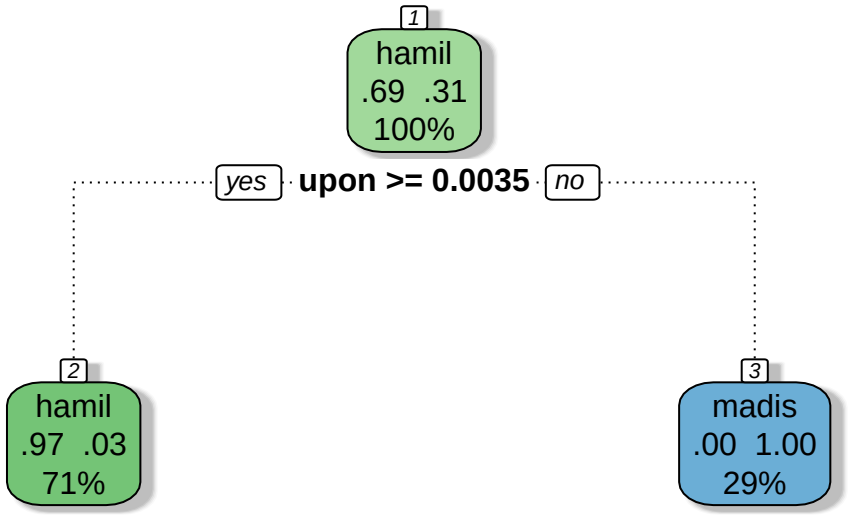
```
##     minsplit = 2, maxdepth = 5)
##
## Variables actually used in tree construction:
## [1] upon
##
## Root node error: 16/51 = 0.31373
##
## n= 51
##
##       CP nsplit rel error xerror    xstd
## 1 0.9375      0    1.0000  1.000 0.20710
## 2 0.0000      1    0.0625  0.375 0.14381

## Warning in rsq.rpart(train_tree2): may not be applicable for this method
```





#plot the decision tree

```
fancyRpartPlot(train_tree2)
```



Rattle 2021−Aug−08 17:47:39 GeorgeSmith

#Table

```
(Results1 <- data.frame(Actual=test$Author, TrainTreeModel1 = predicted1, TrainTreeModel2 = predicted2))
```

```
##    Actual TrainTreeModel1 TrainTreeModel2
## 1   dispt           madis           madis
## 2   dispt           madis           madis
## 3   dispt           madis           madis
## 4   dispt           madis           madis
## 5   dispt           madis           madis
## 6   dispt           madis           madis
## 7   dispt           madis           madis
## 8   dispt           madis           madis
## 9   dispt           madis           madis
## 10  dispt           madis           madis
## 11  dispt           madis           madis
## 12  hamil           hamil           hamil
## 13  hamil           hamil           hamil
## 14  hamil           hamil           hamil
## 15  hamil           hamil           hamil
## 16  hamil           hamil           hamil
## 17  hamil           hamil           hamil
## 18  hamil           hamil           hamil
## 19  hamil           hamil           hamil
## 20  hamil           hamil           hamil
## 21  hamil           hamil           hamil
```

```
## 22  hamil           hamil           hamil
## 23  hamil           hamil           hamil
## 24  hamil           hamil           hamil
## 25  hamil           hamil           hamil
## 26  hamil           hamil           hamil
## 27  hamil           hamil           hamil
## 28  madis           madis           madis
## 29  madis           madis           madis
## 30  madis           madis           madis
## 31  madis           madis           madis
## 32  madis           madis           madis
## 33  madis           madis           madis
## 34  madis           madis           madis
```

#confusion matrix to find correct and incorrect predictions

```
table(Authorship=predicted2, true=test$Author)
```

```
##           true
## Authorship dispt hamil madis
##      hamil     0    16     0
##      madis    11     0     7
```

DT with words taken out

```
FedPapersCorpus2 <- Corpus(DirSource("FedPapersCorpus"))
(numberFedPapers<-length(FedPapersCorpus2))
```

```
## [1] 85
```

```
getTransformations()
```

```
## [1] "removeNumbers"     "removePunctuation" "removeWords"
## [4] "stemDocument"      "stripWhitespace"
```

```
(nFedPapersCorpus2<-length(FedPapersCorpus2))
```

```
## [1] 85
```

```
(minTermFreq <-30)
```

```
## [1] 30
```

```
(maxTermFreq <-1000)
```

```
## [1] 1000
```

```
#  Stopwords
(MyStopwords2 <- c("will","one","two", "may","less","publius","Madison","Alexand", "alexand", "james",
                   "madison", "jay", "hamilton", "jame", "author", "Alexander", "James", "Hamilton","Jay
                   "well","might","without","small", "single", "several", "but", "very", "can", "must",
                   "also", "any", "and", "are", "however", "into", "almost", "can","for", "add", "Autho
                   "alexander", "people", "peoples" , "author", "authors", "member", "latter", "members
                   "alexand", "james" ))
```

```
##  [1] "will"      "one"       "two"       "may"       "less"      "publius"
##  [7] "Madison"   "Alexand"   "alexand"   "james"     "madison"   "jay"
## [13] "hamilton"  "jame"      "author"    "Alexander" "James"     "Hamilton"
## [19] "Jay"       "well"      "might"     "without"   "small"     "single"
## [25] "several"   "but"       "very"      "can"       "must"      "also"
## [31] "any"       "and"       "are"       "however"   "into"      "almost"
## [37] "can"       "for"       "add"       "Author"    "alexander" "people"
## [43] "peoples"   "author"    "authors"   "member"    "latter"    "members"
## [49] "alexand"   "james"
```

```
#(STOPS <-stopwords('english'))
FedPapersCorpus2<- tm_map(FedPapersCorpus2, tolower)
FedPapersCorpus2<- tm_map(FedPapersCorpus2, removeWords, MyStopwords)
FedPapersCorpus2<- tm_map(FedPapersCorpus2, removeWords,
                          c("author", "latter", "members", "constitution", "communiti", "communities",
                            "long", "act", "alexander", "alexand", "james", "jame", "madison", "hamil",
                            "hamilton"))

Papers_DTM2 <- DocumentTermMatrix(FedPapersCorpus2,
                                  control = list(
                                    stopwords = TRUE,
                                    wordLengths=c(3, 15),
                                    removePunctuation = T,
                                    removeNumbers = T,
                                    tolower=T,
                                    stemming = T,
                                    remove_separators = T,
                                    stopwords = MyStopwords2,
                                    removeWords=STOPS,
                                    bounds = list(global = c(minTermFreq, maxTermFreq))
                                  ))

DTM2 <- as.matrix(Papers_DTM2)
(DTM2[12:65,1])
```

```
##  Hamilton_fed_1.txt Hamilton_fed_11.txt Hamilton_fed_12.txt Hamilton_fed_13.txt
##                   1                   4                   2                   1
## Hamilton_fed_15.txt Hamilton_fed_16.txt Hamilton_fed_17.txt Hamilton_fed_21.txt
##                   0                   2                   2                   0
## Hamilton_fed_22.txt Hamilton_fed_23.txt Hamilton_fed_24.txt Hamilton_fed_25.txt
##                   3                   0                   1                   1
## Hamilton_fed_26.txt Hamilton_fed_27.txt Hamilton_fed_28.txt Hamilton_fed_29.txt
##                   1                   2                   2                   0
## Hamilton_fed_30.txt Hamilton_fed_31.txt Hamilton_fed_32.txt Hamilton_fed_33.txt
##                   2                   1                   0                   0
```

```
## Hamilton_fed_34.txt Hamilton_fed_35.txt Hamilton_fed_36.txt Hamilton_fed_59.txt
##                   1                   1                   1                   0
##  Hamilton_fed_6.txt Hamilton_fed_60.txt Hamilton_fed_61.txt Hamilton_fed_65.txt
##                   0                   0                   0                   0
## Hamilton_fed_66.txt Hamilton_fed_67.txt Hamilton_fed_68.txt Hamilton_fed_69.txt
##                   0                   1                   1                   0
##  Hamilton_fed_7.txt Hamilton_fed_70.txt Hamilton_fed_71.txt Hamilton_fed_72.txt
##                   2                   1                   2                   0
## Hamilton_fed_73.txt Hamilton_fed_74.txt Hamilton_fed_75.txt Hamilton_fed_76.txt
##                   0                   0                   1                   0
## Hamilton_fed_77.txt Hamilton_fed_78.txt Hamilton_fed_79.txt  Hamilton_fed_8.txt
##                   0                   1                   0                   2
## Hamilton_fed_80.txt Hamilton_fed_81.txt Hamilton_fed_82.txt Hamilton_fed_83.txt
##                   0                   0                   0                   0
## Hamilton_fed_84.txt Hamilton_fed_85.txt  Hamilton_fed_9.txt       HM_fed_18.txt
##                   0                   1                   3                   0
##       HM_fed_19.txt       HM_fed_20.txt
##                   0                   0
```

```
#Vectorizing
WordFreq2 <- colSums(as.matrix(Papers_DTM2))
(head(WordFreq2))
```

```
##       abl   absolut    accord       act     addit administr
##        74        63        71        58        61        90
```

```
(length(WordFreq2))
```

```
## [1] 406
```

```
ord2 <- order(WordFreq2)
(WordFreq2[head(ord2)])
```

```
##    expos  furnish     word   unless    bound descript
##       34       36       36       37       38       38
```

```
(WordFreq2[tail(ord2)])
```

```
## author nation  peopl  power govern  state
##    390    566    612    937   1040   1662
```

```
(Row_Sum_Per_doc <- rowSums((as.matrix(Papers_DTM2))))
```

```
##    dispt_fed_49.txt    dispt_fed_50.txt    dispt_fed_51.txt    dispt_fed_52.txt
##                 458                 286                 554                 500
##    dispt_fed_53.txt    dispt_fed_54.txt    dispt_fed_55.txt    dispt_fed_56.txt
##                 598                 508                 554                 482
##    dispt_fed_57.txt    dispt_fed_62.txt    dispt_fed_63.txt  Hamilton_fed_1.txt
##                 529                 595                 821                 413
## Hamilton_fed_11.txt Hamilton_fed_12.txt Hamilton_fed_13.txt Hamilton_fed_15.txt
```

```
##                 498                 475                 272                 729
## Hamilton_fed_16.txt Hamilton_fed_17.txt Hamilton_fed_21.txt Hamilton_fed_22.txt
##                 506                 441                 482                 878
## Hamilton_fed_23.txt Hamilton_fed_24.txt Hamilton_fed_25.txt Hamilton_fed_26.txt
##                 501                 455                 510                 608
## Hamilton_fed_27.txt Hamilton_fed_28.txt Hamilton_fed_29.txt Hamilton_fed_30.txt
##                 388                 445                 496                 510
## Hamilton_fed_31.txt Hamilton_fed_32.txt Hamilton_fed_33.txt Hamilton_fed_34.txt
##                 457                 408                 468                 544
## Hamilton_fed_35.txt Hamilton_fed_36.txt Hamilton_fed_59.txt  Hamilton_fed_6.txt
##                 597                 715                 521                 420
## Hamilton_fed_60.txt Hamilton_fed_61.txt Hamilton_fed_65.txt Hamilton_fed_66.txt
##                 566                 375                 486                 559
## Hamilton_fed_67.txt Hamilton_fed_68.txt Hamilton_fed_69.txt  Hamilton_fed_7.txt
##                 401                 390                 712                 542
## Hamilton_fed_70.txt Hamilton_fed_71.txt Hamilton_fed_72.txt Hamilton_fed_73.txt
##                 753                 413                 485                 610
## Hamilton_fed_74.txt Hamilton_fed_75.txt Hamilton_fed_76.txt Hamilton_fed_77.txt
##                 247                 536                 523                 525
## Hamilton_fed_78.txt Hamilton_fed_79.txt  Hamilton_fed_8.txt Hamilton_fed_80.txt
##                 762                 259                 474                 694
## Hamilton_fed_81.txt Hamilton_fed_82.txt Hamilton_fed_83.txt Hamilton_fed_84.txt
##                1059                 448                1450                1086
## Hamilton_fed_85.txt  Hamilton_fed_9.txt       HM_fed_18.txt       HM_fed_19.txt
##                 662                 454                 395                 419
##       HM_fed_20.txt       Jay_fed_2.txt       Jay_fed_3.txt       Jay_fed_4.txt
##                 348                 439                 449                 398
##       Jay_fed_5.txt      Jay_fed_64.txt  Madison_fed_10.txt  Madison_fed_14.txt
##                 361                 604                 767                 472
##  Madison_fed_37.txt  Madison_fed_38.txt  Madison_fed_39.txt  Madison_fed_40.txt
##                 619                 764                 767                 773
##  Madison_fed_41.txt  Madison_fed_42.txt  Madison_fed_43.txt  Madison_fed_44.txt
##                 886                 716                 851                 826
##  Madison_fed_45.txt  Madison_fed_46.txt  Madison_fed_47.txt  Madison_fed_48.txt
##                 631                 718                 804                 496
##  Madison_fed_58.txt
##                 549
```

```
Papers_M2 <- as.matrix(Papers_DTM2)
Papers_M_N12 <- apply(Papers_M2, 1, function(i) round(i/sum(i),3))
Papers_Matrix_Norm2 <- t(Papers_M_N12)
Papers_dtm_matrix2 = as.matrix(Papers_DTM2)
#  Relabeling the data assigning essays to each author
#                  Below we label the data, prepare for modeling, and create some wordclouds
                 ## Also convert to DF
Papers_DF2 <- as.data.frame(as.matrix(Papers_Matrix_Norm2))
Papers_df1_1<- Papers_DF2%>%add_rownames()

names(Papers_df1_1)[1] = "Author"
Papers_df1_1[1:11,1] = "dispt"
Papers_df1_1[12:62,1] = "hamil"
Papers_df1_1[63:65,1] = "ham-mad"
Papers_df1_1[66:70,1] = "jay"
Papers_df1_1[71:85,1 ]="madis"
```

```
head(Papers_df1_1, 15)
```

```
## # A tibble: 15 x 407
##    Author   abl absolut accord   act addit administr admit adopt advantag affair
##    <chr>  <dbl>   <dbl>  <dbl> <dbl> <dbl>     <dbl> <dbl> <dbl>    <dbl>  <dbl>
##  1 dispt  0.004   0      0     0     0         0.002 0.002 0        0.009  0
##  2 dispt  0       0.007  0     0     0         0.007 0     0        0.003  0
##  3 dispt  0.002   0.004  0     0     0.002     0.002 0.005 0        0      0.002
##  4 dispt  0.002   0.002  0     0     0.002     0     0     0.002    0.004  0
##  5 dispt  0       0      0.002 0.002 0         0     0.002 0        0.003  0.015
##  6 dispt  0       0      0.004 0.002 0         0     0.01  0.002    0.008  0
##  7 dispt  0       0      0.004 0     0         0     0.004 0        0      0.002
##  8 dispt  0       0      0.002 0     0         0     0     0        0.002  0.01
##  9 dispt  0       0      0.002 0     0.002     0.002 0.002 0        0      0
## 10 dispt  0.002   0      0     0.002 0.002     0     0     0.002    0.012  0.007
## 11 dispt  0.005   0      0.001 0.001 0.001     0.001 0.001 0        0.006  0.001
## 12 hamil  0.002   0      0.002 0     0.002     0     0.002 0.007    0.002  0
## 13 hamil  0.008   0      0     0     0.002     0     0     0        0.01   0.002
## 14 hamil  0.004   0      0.004 0     0.002     0.002 0.004 0        0.002  0
## 15 hamil  0.004   0      0.004 0     0.004     0.004 0.004 0        0.004  0.004
## # ... with 396 more variables: affect <dbl>, afford <dbl>, alon <dbl>,
## #   alreadi <dbl>, alway <dbl>, america <dbl>, among <dbl>, amount <dbl>,
## #   anoth <dbl>, answer <dbl>, appear <dbl>, appli <dbl>, applic <dbl>,
## #   appoint <dbl>, apprehens <dbl>, argument <dbl>, aris <dbl>, articl <dbl>,
## #   assembl <dbl>, attempt <dbl>, attend <dbl>, attent <dbl>, author <dbl>,
## #   avoid <dbl>, becom <dbl>, best <dbl>, better <dbl>, bodi <dbl>,
## #   bound <dbl>, branch <dbl>, britain <dbl>, calcul <dbl>, call <dbl>,
## #   capac <dbl>, care <dbl>, carri <dbl>, case <dbl>, caus <dbl>,
## #   certain <dbl>, chang <dbl>, charact <dbl>, circumst <dbl>, citizen <dbl>,
## #   civil <dbl>, class <dbl>, clear <dbl>, collect <dbl>, combin <dbl>,
## #   commit <dbl>, common <dbl>, communiti <dbl>, complet <dbl>, compos <dbl>,
## #   concern <dbl>, conclus <dbl>, conduct <dbl>, confeder <dbl>,
## #   confederaci <dbl>, confid <dbl>, confin <dbl>, congress <dbl>,
## #   connect <dbl>, consequ <dbl>, consid <dbl>, consider <dbl>, consist <dbl>,
## #   constitu <dbl>, constitut <dbl>, construct <dbl>, contend <dbl>,
## #   continu <dbl>, contrari <dbl>, control <dbl>, convent <dbl>, council <dbl>,
## #   countri <dbl>, cours <dbl>, danger <dbl>, decid <dbl>, decis <dbl>,
## #   declar <dbl>, defect <dbl>, defens <dbl>, degre <dbl>, deliber <dbl>,
## #   depart <dbl>, depend <dbl>, deriv <dbl>, descript <dbl>, design <dbl>,
## #   desir <dbl>, determin <dbl>, differ <dbl>, difficulti <dbl>, direct <dbl>,
## #   dispos <dbl>, disposit <dbl>, distinct <dbl>, doubt <dbl>, drawn <dbl>, ...
```

```
tail(Papers_df1_1, 20)
```

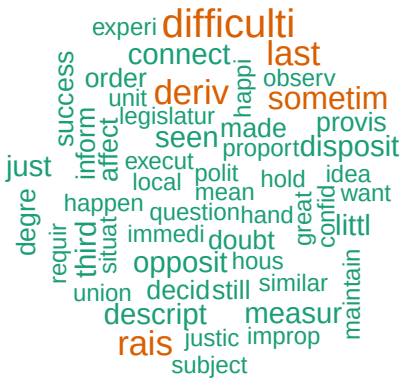
```
## # A tibble: 20 x 407
##    Author   abl absolut accord   act addit administr admit adopt advantag affair
##    <chr>  <dbl>   <dbl>  <dbl> <dbl> <dbl>     <dbl> <dbl> <dbl>    <dbl>  <dbl>
##  1 jay    0       0      0     0     0         0     0.002 0.005    0      0
##  2 jay    0.004   0      0.002 0     0.002     0.002 0     0.002    0.002  0
##  3 jay    0.003   0.005  0.003 0     0         0     0.003 0        0.005  0
##  4 jay    0       0      0     0.003 0         0     0.003 0        0      0.003
##  5 jay    0.008   0.003  0     0.005 0         0     0.002 0        0.007  0.01
```

```
##  6 madis  0.003   0      0.004 0.001 0         0.003 0.001 0.001    0.005  0
##  7 madis  0       0.002  0     0     0.002     0.004 0     0.002    0.002  0.002
##  8 madis  0.002   0      0.002 0     0.002     0.003 0.002 0.002    0.002  0.002
##  9 madis  0.001   0.005  0.004 0     0         0.003 0.003 0.001    0      0
## 10 madis  0       0.004  0.01  0     0         0     0     0.001    0      0
## 11 madis  0       0.004  0.003 0.004 0         0     0.003 0        0      0
## 12 madis  0.003   0      0     0.001 0.002     0     0.001 0        0.005  0.001
## 13 madis  0       0.001  0.001 0.003 0.001     0.003 0.003 0.001    0      0.003
## 14 madis  0.001   0.002  0     0.001 0         0     0.006 0.002    0.002  0
## 15 madis  0       0.002  0.004 0.004 0.002     0     0.001 0.001    0.001  0
## 16 madis  0       0.003  0.002 0     0.003     0.002 0     0        0.006  0.002
## 17 madis  0.007   0.001  0.003 0     0.001     0.004 0.003 0        0.01   0.001
## 18 madis  0       0.001  0.002 0.004 0         0.001 0.004 0.001    0      0
## 19 madis  0       0      0.002 0.006 0         0.004 0     0        0.002  0
## 20 madis  0.002   0.002  0     0     0.005     0     0.005 0        0.009  0.004
## # ... with 396 more variables: affect <dbl>, afford <dbl>, alon <dbl>,
## #   alreadi <dbl>, alway <dbl>, america <dbl>, among <dbl>, amount <dbl>,
## #   anoth <dbl>, answer <dbl>, appear <dbl>, appli <dbl>, applic <dbl>,
## #   appoint <dbl>, apprehens <dbl>, argument <dbl>, aris <dbl>, articl <dbl>,
## #   assembl <dbl>, attempt <dbl>, attend <dbl>, attent <dbl>, author <dbl>,
## #   avoid <dbl>, becom <dbl>, best <dbl>, better <dbl>, bodi <dbl>,
## #   bound <dbl>, branch <dbl>, britain <dbl>, calcul <dbl>, call <dbl>,
## #   capac <dbl>, care <dbl>, carri <dbl>, case <dbl>, caus <dbl>,
## #   certain <dbl>, chang <dbl>, charact <dbl>, circumst <dbl>, citizen <dbl>,
## #   civil <dbl>, class <dbl>, clear <dbl>, collect <dbl>, combin <dbl>,
## #   commit <dbl>, common <dbl>, communiti <dbl>, complet <dbl>, compos <dbl>,
## #   concern <dbl>, conclus <dbl>, conduct <dbl>, confeder <dbl>,
## #   confederaci <dbl>, confid <dbl>, confin <dbl>, congress <dbl>,
## #   connect <dbl>, consequ <dbl>, consid <dbl>, consider <dbl>, consist <dbl>,
## #   constitu <dbl>, constitut <dbl>, construct <dbl>, contend <dbl>,
## #   continu <dbl>, contrari <dbl>, control <dbl>, convent <dbl>, council <dbl>,
## #   countri <dbl>, cours <dbl>, danger <dbl>, decid <dbl>, decis <dbl>,
## #   declar <dbl>, defect <dbl>, defens <dbl>, degre <dbl>, deliber <dbl>,
## #   depart <dbl>, depend <dbl>, deriv <dbl>, descript <dbl>, design <dbl>,
## #   desir <dbl>, determin <dbl>, differ <dbl>, difficulti <dbl>, direct <dbl>,
## #   dispos <dbl>, disposit <dbl>, distinct <dbl>, doubt <dbl>, drawn <dbl>, ...
```

```
Papers_df1_1[62:71,1]  #  Checking row names
```

```
## # A tibble: 10 x 1
##    Author
##    <chr>
##  1 hamil
##  2 ham-mad
##  3 ham-mad
##  4 ham-mad
##  5 jay
##  6 jay
##  7 jay
##  8 jay
##  9 jay
## 10 madis
```

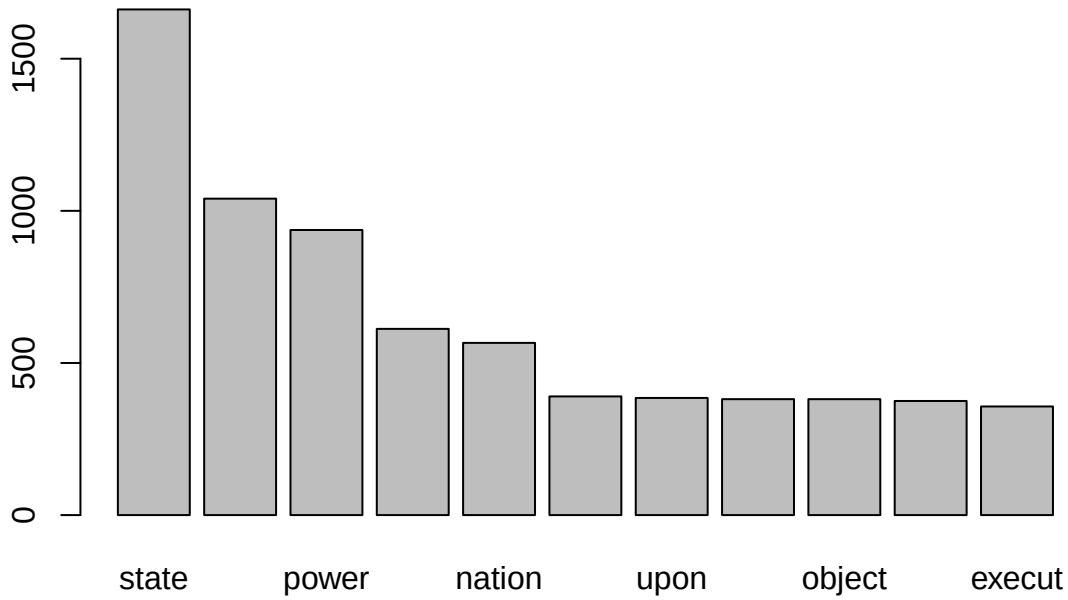
```
#Storing disputed essays in its own dataframe
disputedEssays <- Papers_df1_1[1:11,]

dE_Matrix <- as.matrix(disputedEssays)

#Wordcloud Visualization Hamilton, Madison and Disputed Papers
DisputedPapersWC<- wordcloud(colnames(Papers_dtm_matrix2), Papers_dtm_matrix2[1:11,],
                             rot.per = .35, colors = brewer.pal(n = 8, name = "Dark2"))
```



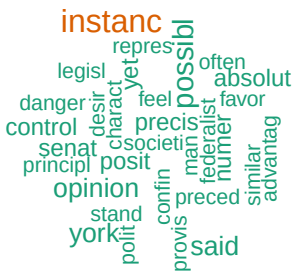

```
(head(sort(as.matrix(WordFreq2[1:11]), decreasing = TRUE), n=50))
```

```
##  [1] 142 107  90  74  71  65  63  61  58  57  56
```

```
barplot(head(sort(WordFreq2, decreasing = T),11))
```




```
HamiltonPapersWC <-wordcloud(colnames(Papers_dtm_matrix2),Papers_dtm_matrix2[12:62,],
                             rot.per = .35, colors = brewer.pal(n = 8, name = "Dark2"))
```



#barplot for Hamilton

#barplot(head(sort(WordFreq2[12:62], decreasing = T),20))

```
(head(sort(as.matrix(Papers_DTM[12:62,])[11,], decreasing = TRUE), n=50))
```

```
##      state  constitut legislatur       peac      power       will     appear
##         10          9          9          9          9          9          8
##  establish        one      right       time        two     author     nation
##          8          8          8          8          8          7          7
##      stand       upon    without       even       must     necess     object
##          7          7          7          6          6          6          6
##    increas      natur        new      peopl       plan    respect    subject
##          5          5          5          5          5          5          5
##        can     either      exist     govern      great        man     matter
##          4          4          4          4          4          4          4
##        may      secur       view     articl    britain     consid     danger
##          4          4          4          3          3          3          3
##       find       forc       form    general      howev     legisl    liberti
##          3          3          3          3          3          3          3
##        now
##          3
```

```
MadisonPapersWC <-wordcloud(colnames(Papers_dtm_matrix), Papers_dtm_matrix[71:85,],
                            rot.per = .35, colors = brewer.pal(n = 8, "Dark2"))
```

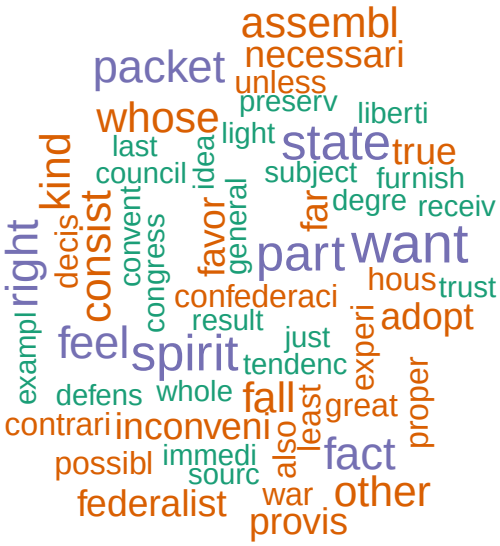

```
(head(sort(as.matrix(Papers_DTM[71:85,])[11,], decreasing = TRUE), n=50))
```

```
##       state      govern        will       power       peopl       feder
##          41          35          32          23          18          17
##       union      latter         may       offic      author      former
##          11           9           9           9           8           8
##    influenc     essenti  particular      danger        less      member
##           8           7           7           6           6           6
##        much        must      object        peac     probabl   constitut
##           6           6           6           6           6           5
##       everi         far       great       happi         new      propos
##           5           5           5           5           5           5
##        side    advantag     appoint        case confederaci     consequ
##           5           4           4           4           4           4
##    consider       degre      depart        form     general      import
##           4           4           4           4           4           4
##    individu     instanc        like       local      number       numer
##           4           4           4           4           4           4
##         one        part
##           4           4
```

#barplot for Madison

#barplot(head(sort(WordFreq2[71:85], decreasing = T),20))

```
JayPapersWC <- wordcloud(colnames(Papers_dtm_matrix), Papers_dtm_matrix[63:70,],
                                        rot.per = .35, colors = brewer.pal(n = 8, "Dark2"))
```




```
(head(sort(as.matrix(Papers_DTM[63:70,])[70-63,], decreasing = TRUE), n=50))
```

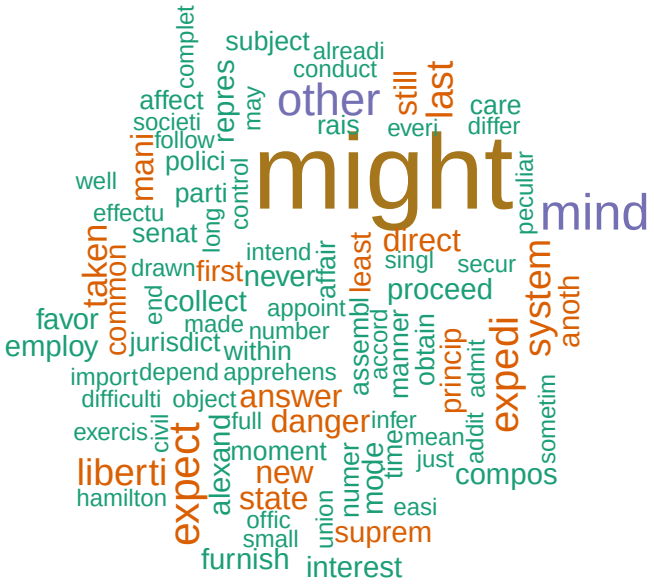
```
##      nation confederaci      differ     foreign         one    interest
##          13          11          11          11          10           8
##       union        will    jealousi       might       secur     america
##           7           7           6           6           6           5
##        form       anoth      danger       equal      happen    independ
##           5           4           4           4           4           4
##      polici     probabl       three      affect   apprehens       degre
##           4           4           4           3           3           3
##     general        good      govern       great      import        long
##           3           3           3           3           3           3
##        mani        much        must      observ       other        part
##           3           3           3           3           3           3
##       peopl       state         use         war         yet        also
##           3           3           3           3           3           2
##      appear     britain     charact    circumst      combin     concern
##           2           2           2           2           2           2
```

```
##      confid      consid
##           2           2
```

#barplot for Jay

#barplot(head(sort(WordFreq2[63:70], decreasing = T),20))

```
Ham_MadWC <-wordcloud(colnames(Papers_dtm_matrix), Papers_dtm_matrix[60:62,],
                      rot.per = .35, colors = brewer.pal(n = 8, "Dark2"))
```




```
(head(sort(as.matrix(Papers_DTM[60:62,])[3,], decreasing = TRUE), n=50))
```

```
##      govern       state         one    confeder    principl      author
##          19          19          12           9           8           7
##        mean         new       union        will confederaci       great
##           7           7           7           7           6           6
##      member   constitut         far         may      object        part
##           6           5           5           5           5           5
##    distinct        forc        form        kind     liberti       power
##           4           4           4           4           4           4
##        seem       shall       singl     societi        upon         abl
##           4           4           4           4           4           3
##    advantag       anoth      compos     council      either       equal
```

```
##           3           3           3           3           3           3
##     general       howev       idea       offic       peopl     popular
##           3           3           3           3           3           3
##     possess     respect        time        view     without   administr
##           3           3           3           3           3           2
##     america     appoint
##           2           2
```

#barplot for Coauthors

#barplot(head(sort(WordFreq2[60:62], decreasing = T),20))

Experimental Design

Now that the data is labeled, its time to design an experiment. Below we randomly select a train and test

set for validation using function: sample.int() .

```
(head(sort(as.matrix(Papers_dtm_matrix)[11,], decreasing = TRUE), n=50))
```

```
##      peopl      senat       will        may     repres     govern       bodi
##         42         24         19         18         18         16         15
##        can      elect       must     measur      state     nation        one
##         14         14         12         11         11          9          9
##  constitut     former      power     reason       year    assembl     exampl
##          8          8          8          8          8          7          7
##        two     danger      everi       evid      feder     import     latter
##          7          6          6          6          6          6          6
##     object particular     public   advantag     answer     appear     author
##          6          6          6          5          5          5          5
##    charact       fact      first       hous   institut       less       mani
##          5          5          5          5          5          5          5
##     member      might       oper      order       part    popular    probabl
##          5          5          5          5          5          5          5
##      small
##          5
```

##Make Train and Test sets

```
numDisputed = 11
numTotalPapers = nrow(Papers_df1_1)

trainRatio <- .60
set.seed(11) # Set Seed so that same sample can be reproduced in future also

sample <- sample.int(n = numTotalPapers-numDisputed, size = floor(trainRatio*numTotalPapers), replace =
```

```
newSample = sample + numDisputed

train <- Papers_df1_1[newSample, ]
test <- Papers_df1_1[-newSample, ]
```

train / test ratio

```
length(newSample)/nrow(Papers_df1_1)
```

```
## [1] 0.6
```

Classification

Repeating above using different parameters

##Decision Tree Models #Train Tree Model 1

```
train_tree1 <- rpart(Author ~ ., data = train, method="class", control=rpart.control(cp=0))
summary(train_tree1)
```

```
## Call:
## rpart(formula = Author ~ ., data = train, method = "class", control = rpart.control(cp = 0))
##   n= 51
##
##       CP nsplit rel error xerror      xstd
## 1 0.6875      0    1.0000 1.0000 0.2071042
## 2 0.0000      1    0.3125 0.4375 0.1535926
##
## Variable importance
##  upon  form   men natur other paper
##    41    16    11    11    11    11
##
## Node number 1: 51 observations,    complexity param=0.6875
##   predicted class=hamil  expected loss=0.3137255  P(node) =1
##     class counts:     2    35     2    12
##    probabilities: 0.039 0.686 0.039 0.235
##   left son=2 (36 obs) right son=3 (15 obs)
##   Primary splits:
##       upon      < 0.0035 to the right, improve=15.655560, (0 missing)
##       kind      < 0.0015 to the right, improve= 7.242236, (0 missing)
##       form      < 0.0075 to the left,  improve= 5.692683, (0 missing)
##       thing     < 0.0015 to the right, improve= 5.638889, (0 missing)
##       communiti < 0.0015 to the right, improve= 5.161290, (0 missing)
##   Surrogate splits:
##       form  < 0.0075 to the left,  agree=0.824, adj=0.400, (0 split)
##       men   < 5e-04  to the right, agree=0.784, adj=0.267, (0 split)
##       natur < 0.0015 to the right, agree=0.784, adj=0.267, (0 split)
```

```
##       other < 0.0065 to the left,  agree=0.784, adj=0.267, (0 split)
##       paper < 0.0045 to the left,  agree=0.784, adj=0.267, (0 split)
##
## Node number 2: 36 observations
##   predicted class=hamil  expected loss=0.02777778  P(node) =0.7058824
##     class counts:     0    35     0     1
##    probabilities: 0.000 0.972 0.000 0.028
##
## Node number 3: 15 observations
##   predicted class=madis  expected loss=0.2666667  P(node) =0.2941176
##     class counts:     2     0     2    11
##    probabilities: 0.133 0.000 0.133 0.733
```

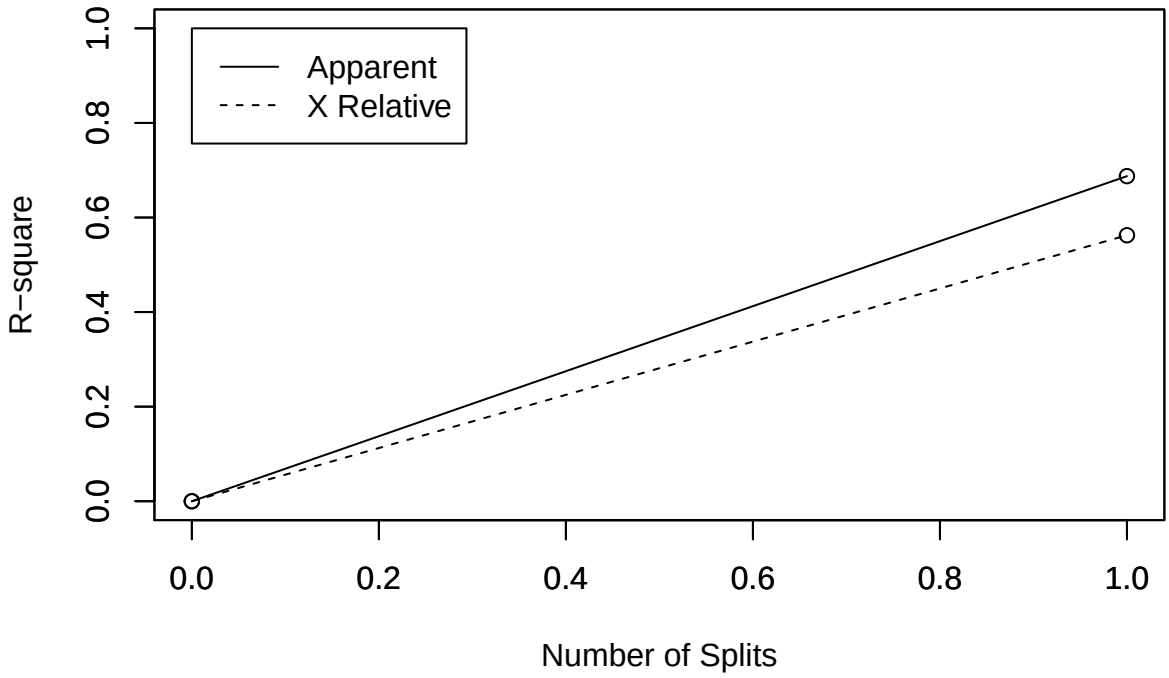
#predict the test dataset using the model for train tree No. 1

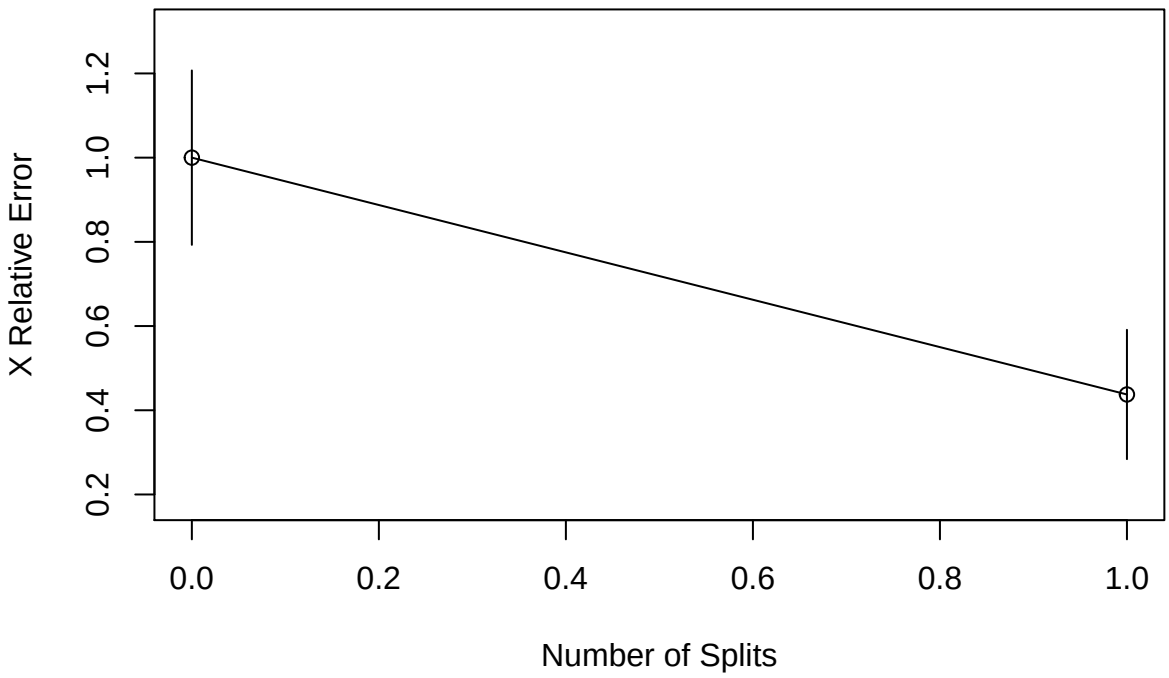
```
predicted1= predict(train_tree1, test, type="class")
```

#plot number of splits

```
rsq.rpart(train_tree1)
```

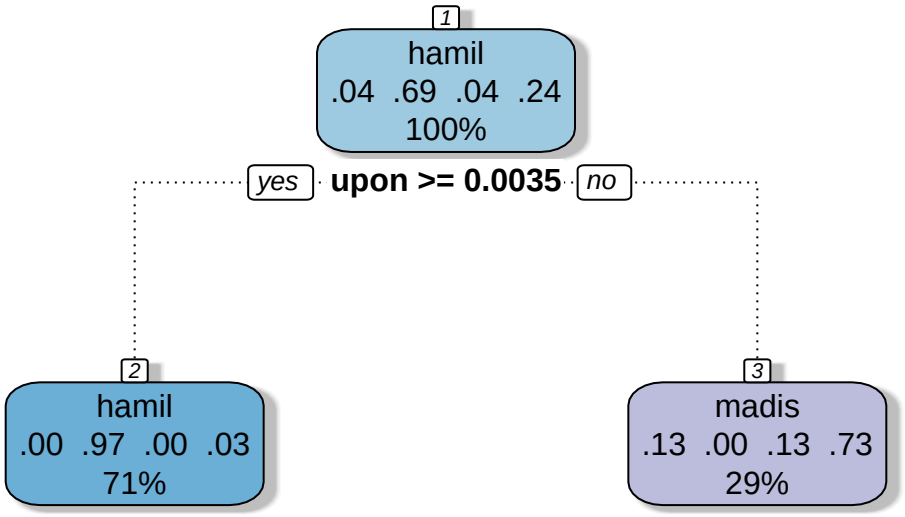
```
##
## Classification tree:
## rpart(formula = Author ~ ., data = train, method = "class", control = rpart.control(cp = 0))
##
## Variables actually used in tree construction:
## [1] upon
##
## Root node error: 16/51 = 0.31373
##
## n= 51
##
##       CP nsplit rel error xerror    xstd
## 1 0.6875      0    1.0000 1.0000 0.20710
## 2 0.0000      1    0.3125 0.4375 0.15359

## Warning in rsq.rpart(train_tree1): may not be applicable for this method
```


Apparent
X Relative
R−square
0.0
0.2
0.4
0.6
0.8
1.0
Number of Splits



#plot the decision tree

```
fancyRpartPlot(train_tree1)
```



#confusion matrix to find correct and incorrect predictions

```
table(Authorship=predicted1, true=test$Author)
```

```
##           true
## Authorship dispt ham-mad hamil jay madis
##    ham-mad     0       0     0   0     0
##    hamil       1       0    16   0     0
##    jay         0       0     0   0     0
##    madis      10       1     0   3     3
```

```
train_tree2 <- rpart(Author ~ ., data = train, method="class", control=rpart.control(cp=0), minsplit = 1
(summary(train_tree2))
```

```
## Call:
## rpart(formula = Author ~ ., data = train, method = "class", control = rpart.control(cp = 0),
##     minsplit = 2, maxdepth = 5)
##   n= 51
##
##       CP nsplit rel error xerror      xstd
## 1 0.6875      0    1.0000 1.0000 0.2071042
## 2 0.0000      1    0.3125 0.4375 0.1535926
##
## Variable importance
##  upon  form   men natur other paper
```

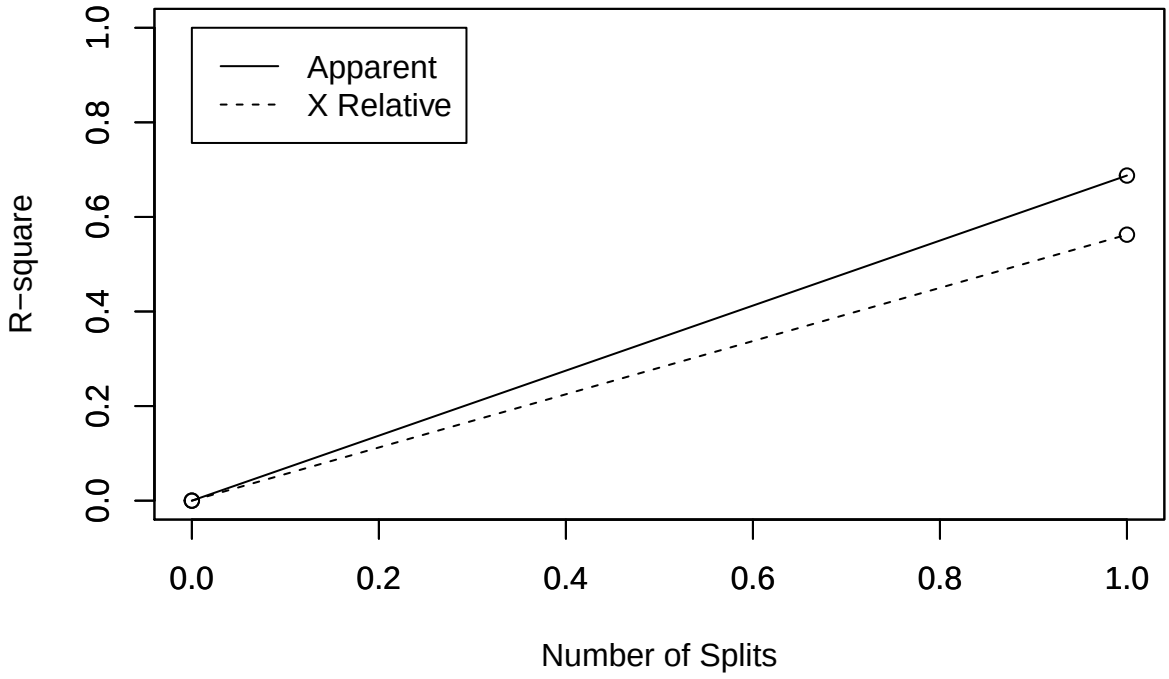
```
##    41    16    11    11    11    11
##
## Node number 1: 51 observations,    complexity param=0.6875
##   predicted class=hamil  expected loss=0.3137255  P(node) =1
##     class counts:     2    35     2    12
##    probabilities: 0.039 0.686 0.039 0.235
##   left son=2 (36 obs) right son=3 (15 obs)
##   Primary splits:
##       upon      < 0.0035 to the right, improve=15.655560, (0 missing)
##       kind      < 0.0015 to the right, improve= 7.242236, (0 missing)
##       form      < 0.0075 to the left,  improve= 5.692683, (0 missing)
##       thing     < 0.0015 to the right, improve= 5.638889, (0 missing)
##       communiti < 0.0015 to the right, improve= 5.161290, (0 missing)
##   Surrogate splits:
##       form  < 0.0075 to the left,  agree=0.824, adj=0.400, (0 split)
##       men   < 5e-04  to the right, agree=0.784, adj=0.267, (0 split)
##       natur < 0.0015 to the right, agree=0.784, adj=0.267, (0 split)
##       other < 0.0065 to the left,  agree=0.784, adj=0.267, (0 split)
##       paper < 0.0045 to the left,  agree=0.784, adj=0.267, (0 split)
##
## Node number 2: 36 observations
##   predicted class=hamil  expected loss=0.02777778  P(node) =0.7058824
##     class counts:     0    35     0     1
##    probabilities: 0.000 0.972 0.000 0.028
##
## Node number 3: 15 observations
##   predicted class=madis  expected loss=0.2666667  P(node) =0.2941176
##     class counts:     2     0     2    11
##    probabilities: 0.133 0.000 0.133 0.733

## n= 51
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 51 16 hamil (0.03921569 0.68627451 0.03921569 0.23529412)
##   2) upon>=0.0035 36  1 hamil (0.00000000 0.97222222 0.00000000 0.02777778) *
##   3) upon< 0.0035 15  4 madis (0.13333333 0.00000000 0.13333333 0.73333333) *
```

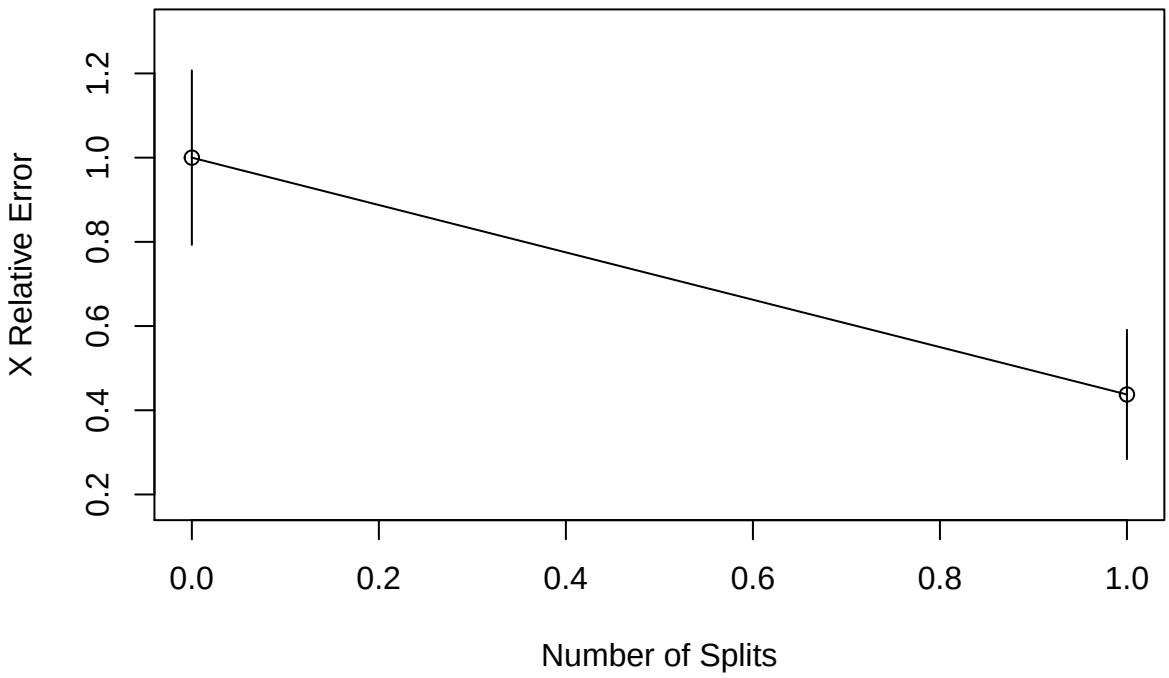
#predict the test dataset using the model for train tree No. 1 predicted2= predict(train_tree2, test, type="class") #plot number of splits

```
rsq.rpart(train_tree2)
```

```
##
## Classification tree:
## rpart(formula = Author ~ ., data = train, method = "class", control = rpart.control(cp = 0),
##     minsplit = 2, maxdepth = 5)
##
## Variables actually used in tree construction:
## [1] upon
##
## Root node error: 16/51 = 0.31373
```

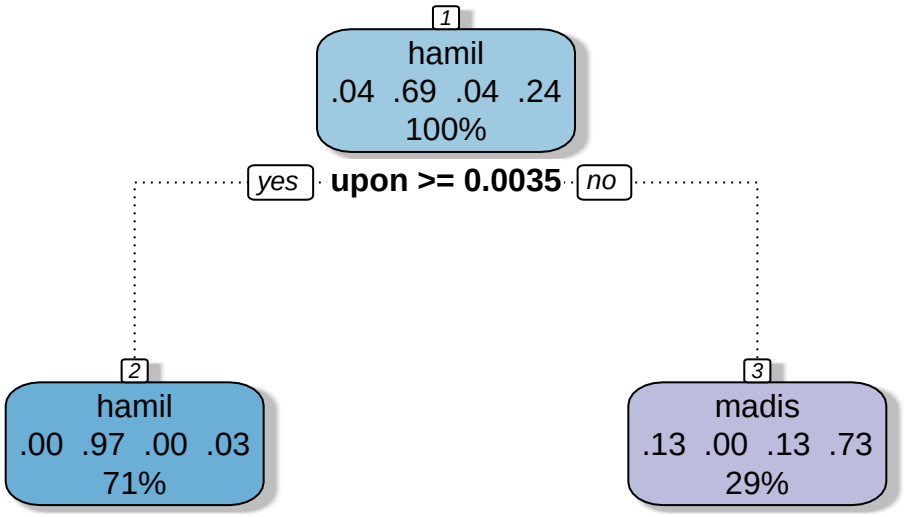
```
##
## n= 51
##
##       CP nsplit rel error xerror    xstd
## 1 0.6875      0    1.0000 1.0000 0.20710
## 2 0.0000      1    0.3125 0.4375 0.15359

## Warning in rsq.rpart(train_tree2): may not be applicable for this method
```





#plot the decision tree

```
fancyRpartPlot(train_tree2)
```



#confusion matrix to find correct and incorrect predictions

```
table(Authorship=predicted2, true=test$Author)
```

```
##           true
## Authorship dispt ham-mad hamil jay madis
##      hamil     0       0    16   0     0
##      madis    11       1     0   3     3
```

DT with words taken out

```
FedPapersCorpus2 <- Corpus(DirSource("FedPapersCorpus"))
(numberFedPapers<-length(FedPapersCorpus2))
```

```
## [1] 85
```

```
(getTransformations())
```

```
## [1] "removeNumbers"     "removePunctuation" "removeWords"
## [4] "stemDocument"      "stripWhitespace"
```

```
(nFedPapersCorpus2<-length(FedPapersCorpus2))
```

```
## [1] 85
```

```
(minTermFreq <-30)
```

```
## [1] 30
```

```
(maxTermFreq <-1000)
```

```
## [1] 1000
```

Stopwords remain the same

#(MyStopwords2 <- c("will","one","two", "may","less","publius","Madison","Alexand", "alexand", "james", # "madison", "jay", "hamilton", "jame", "author", "Alexander", "James", "Hamilton","Jay", # "well","might","without","small", "single", "several", "but", "very", "can", "must", # "also", "any", "and", "are", "however", "into", "almost", "can","for", "add", "Author", # "alexander", "people", "peoples" , "author", "authors", "member", "latter", "members", # "alexand", "james")) #(STOPS <-stopwords('english'))

```
FedPapersCorpus2<- tm_map(FedPapersCorpus2, tolower)
FedPapersCorpus2<- tm_map(FedPapersCorpus2, removeWords, MyStopwords)
FedPapersCorpus2<- tm_map(FedPapersCorpus2, removeWords, c("author", "latter", "members", "constitution

Papers_DTM2 <- DocumentTermMatrix(FedPapersCorpus2,
                                  control = list(
                                    stopwords = TRUE,
                                    wordLengths=c(3, 15),
                                    removePunctuation = T,
                                    removeNumbers = T,
                                    tolower=T,
                                    stemming = T,
                                    remove_separators = T,
                                    stopwords = MyStopwords2,
                                    removeWords=STOPS,
                                    bounds = list(global = c(minTermFreq, maxTermFreq))
                                  ))

DTM2 <- as.matrix(Papers_DTM2)
(DTM[12:65,1])
```

```
##  Hamilton_fed_1.txt Hamilton_fed_11.txt Hamilton_fed_12.txt Hamilton_fed_13.txt
##                   1                   4                   2                   1
## Hamilton_fed_15.txt Hamilton_fed_16.txt Hamilton_fed_17.txt Hamilton_fed_21.txt
##                   0                   2                   2                   0
## Hamilton_fed_22.txt Hamilton_fed_23.txt Hamilton_fed_24.txt Hamilton_fed_25.txt
##                   3                   0                   1                   1
## Hamilton_fed_26.txt Hamilton_fed_27.txt Hamilton_fed_28.txt Hamilton_fed_29.txt
```

```
##                   1                   2                   2                   0
## Hamilton_fed_30.txt Hamilton_fed_31.txt Hamilton_fed_32.txt Hamilton_fed_33.txt
##                   2                   1                   0                   0
## Hamilton_fed_34.txt Hamilton_fed_35.txt Hamilton_fed_36.txt Hamilton_fed_59.txt
##                   1                   1                   1                   0
##  Hamilton_fed_6.txt Hamilton_fed_60.txt Hamilton_fed_61.txt Hamilton_fed_65.txt
##                   0                   0                   0                   0
## Hamilton_fed_66.txt Hamilton_fed_67.txt Hamilton_fed_68.txt Hamilton_fed_69.txt
##                   0                   1                   1                   0
##  Hamilton_fed_7.txt Hamilton_fed_70.txt Hamilton_fed_71.txt Hamilton_fed_72.txt
##                   2                   1                   2                   0
## Hamilton_fed_73.txt Hamilton_fed_74.txt Hamilton_fed_75.txt Hamilton_fed_76.txt
##                   0                   0                   1                   0
## Hamilton_fed_77.txt Hamilton_fed_78.txt Hamilton_fed_79.txt  Hamilton_fed_8.txt
##                   0                   1                   0                   2
## Hamilton_fed_80.txt Hamilton_fed_81.txt Hamilton_fed_82.txt Hamilton_fed_83.txt
##                   0                   0                   0                   0
## Hamilton_fed_84.txt Hamilton_fed_85.txt  Hamilton_fed_9.txt       HM_fed_18.txt
##                   0                   1                   3                   0
##       HM_fed_19.txt       HM_fed_20.txt
##                   0                   0
```

```
#Vectorizing
WordFreq2 <- colSums(as.matrix(Papers_DTM2))
(head(WordFreq2))
```

```
##       abl   absolut    accord       act     addit administr
##        74        63        71        58        61        90
```

```
(length(WordFreq2))
```

```
## [1] 406
```

```
ord2 <- order(WordFreq2)
(WordFreq2[head(ord2)])
```

```
##    expos  furnish     word   unless    bound descript
##       34       36       36       37       38       38
```

```
(WordFreq2[tail(ord2)])
```

```
## author nation  peopl  power govern  state
##    390    566    612    937   1040   1662
```

```
(Row_Sum_Per_doc <- rowSums((as.matrix(Papers_DTM2))))
```

```
##    dispt_fed_49.txt    dispt_fed_50.txt    dispt_fed_51.txt    dispt_fed_52.txt
##                 458                 286                 554                 500
##    dispt_fed_53.txt    dispt_fed_54.txt    dispt_fed_55.txt    dispt_fed_56.txt
##                 598                 508                 554                 482
```

```
##    dispt_fed_57.txt    dispt_fed_62.txt    dispt_fed_63.txt  Hamilton_fed_1.txt
##                 529                 595                 821                 413
## Hamilton_fed_11.txt Hamilton_fed_12.txt Hamilton_fed_13.txt Hamilton_fed_15.txt
##                 498                 475                 272                 729
## Hamilton_fed_16.txt Hamilton_fed_17.txt Hamilton_fed_21.txt Hamilton_fed_22.txt
##                 506                 441                 482                 878
## Hamilton_fed_23.txt Hamilton_fed_24.txt Hamilton_fed_25.txt Hamilton_fed_26.txt
##                 501                 455                 510                 608
## Hamilton_fed_27.txt Hamilton_fed_28.txt Hamilton_fed_29.txt Hamilton_fed_30.txt
##                 388                 445                 496                 510
## Hamilton_fed_31.txt Hamilton_fed_32.txt Hamilton_fed_33.txt Hamilton_fed_34.txt
##                 457                 408                 468                 544
## Hamilton_fed_35.txt Hamilton_fed_36.txt Hamilton_fed_59.txt  Hamilton_fed_6.txt
##                 597                 715                 521                 420
## Hamilton_fed_60.txt Hamilton_fed_61.txt Hamilton_fed_65.txt Hamilton_fed_66.txt
##                 566                 375                 486                 559
## Hamilton_fed_67.txt Hamilton_fed_68.txt Hamilton_fed_69.txt  Hamilton_fed_7.txt
##                 401                 390                 712                 542
## Hamilton_fed_70.txt Hamilton_fed_71.txt Hamilton_fed_72.txt Hamilton_fed_73.txt
##                 753                 413                 485                 610
## Hamilton_fed_74.txt Hamilton_fed_75.txt Hamilton_fed_76.txt Hamilton_fed_77.txt
##                 247                 536                 523                 525
## Hamilton_fed_78.txt Hamilton_fed_79.txt  Hamilton_fed_8.txt Hamilton_fed_80.txt
##                 762                 259                 474                 694
## Hamilton_fed_81.txt Hamilton_fed_82.txt Hamilton_fed_83.txt Hamilton_fed_84.txt
##                1059                 448                1450                1086
## Hamilton_fed_85.txt  Hamilton_fed_9.txt       HM_fed_18.txt       HM_fed_19.txt
##                 662                 454                 395                 419
##       HM_fed_20.txt       Jay_fed_2.txt       Jay_fed_3.txt       Jay_fed_4.txt
##                 348                 439                 449                 398
##       Jay_fed_5.txt      Jay_fed_64.txt  Madison_fed_10.txt  Madison_fed_14.txt
##                 361                 604                 767                 472
##  Madison_fed_37.txt  Madison_fed_38.txt  Madison_fed_39.txt  Madison_fed_40.txt
##                 619                 764                 767                 773
##  Madison_fed_41.txt  Madison_fed_42.txt  Madison_fed_43.txt  Madison_fed_44.txt
##                 886                 716                 851                 826
##  Madison_fed_45.txt  Madison_fed_46.txt  Madison_fed_47.txt  Madison_fed_48.txt
##                 631                 718                 804                 496
##  Madison_fed_58.txt
##                 549
```

```
Papers_M2 <- as.matrix(Papers_DTM2)
Papers_M_N12 <- apply(Papers_M2, 1, function(i) round(i/sum(i),3))
Papers_Matrix_Norm2 <- t(Papers_M_N12)
Papers_dtm_matrix = as.matrix(Papers_DTM2)


Papers_DFNoJay <- as.data.frame(as.matrix(Papers_Matrix_Norm2))
```

#remove Jays papers

```
Papers_DFNoHM<-Papers_DFNoJay[-66:-70,]
```

```
# remove Ham Mad papers
Papers_DFHM <- Papers_DFNoJay[63:65]

Papers_DFNoHM <- Papers_DFNoHM%>%add_rownames()

#  Provide row names
names(Papers_DFNoHM)[1]<-"Author"
Papers_DFNoHM[1:11,1] = "dispt"
Papers_DFNoHM[12:62,1] = "hamil"
Papers_DFNoHM[63:80,1] = "madis"
```

##Make Train and Test sets using higher ratio

```
trainRatio <- .75
set.seed(11) # Set Seed so that same sample can be reproduced in future also
sampleNoHM <- sample.int(n = nrow(Papers_DFNoHM), size = floor(trainRatio*nrow(Papers_DFNoHM)), replace
trainNoHM <- Papers_DFNoHM[sampleNoHM, ]
testNoHM <- Papers_DFNoHM[-sampleNoHM, ]
# train / test ratio
length(sampleNoHM)/nrow(Papers_DFNoHM)
```

```
## [1] 0.75
```

```
##Decision Tree Models
#Train Tree Model NoHM
train_treeNoHM <- rpart(Author ~ ., data = trainNoHM, method="class", control=rpart.control(cp=0))
summary(train_treeNoHM)
```

```
## Call:
## rpart(formula = Author ~ ., data = trainNoHM, method = "class",
##     control = rpart.control(cp = 0))
##   n= 60
##
##          CP nsplit rel error    xerror      xstd
## 1 0.5833333      0 1.0000000 1.0000000 0.1581139
## 2 0.2500000      1 0.4166667 0.4583333 0.1248842
## 3 0.0000000      2 0.1666667 0.3750000 0.1152443
##
## Variable importance
##        upon      matter        kind     assembl       among    maintain
##          22          11          10           9           9           9
##       union      branch    confeder confederaci   establish      legisl
##           7           6           4           4           4           4
##
## Node number 1: 60 observations,    complexity param=0.5833333
##   predicted class=hamil  expected loss=0.4  P(node) =1
##     class counts:     9    36    15
##    probabilities: 0.150 0.600 0.250
##   left son=2 (35 obs) right son=3 (25 obs)
##   Primary splits:
##       upon    < 0.0055 to the right, improve=20.580000, (0 missing)
##       matter  < 5e-04  to the right, improve= 8.992992, (0 missing)
```

```
##       kind    < 0.0015 to the right, improve= 8.751429, (0 missing)
##       thing   < 0.0015 to the right, improve= 7.216667, (0 missing)
##       assembl < 0.0025 to the right, improve= 6.894108, (0 missing)
##   Surrogate splits:
##       matter   < 5e-04  to the right, agree=0.800, adj=0.52, (0 split)
##       kind     < 5e-04  to the right, agree=0.783, adj=0.48, (0 split)
##       assembl  < 0.0025 to the left,  agree=0.767, adj=0.44, (0 split)
##       among    < 0.0035 to the left,  agree=0.750, adj=0.40, (0 split)
##       maintain < 0.0015 to the left,  agree=0.750, adj=0.40, (0 split)
##
## Node number 2: 35 observations
##   predicted class=hamil  expected loss=0  P(node) =0.5833333
##     class counts:     0    35     0
##    probabilities: 0.000 1.000 0.000
##
## Node number 3: 25 observations,    complexity param=0.25
##   predicted class=madis  expected loss=0.4  P(node) =0.4166667
##     class counts:     9     1    15
##    probabilities: 0.360 0.040 0.600
##   left son=6 (12 obs) right son=7 (13 obs)
##   Primary splits:
##       union   < 0.0035 to the left,  improve=6.373846, (0 missing)
##       branch  < 0.003  to the right, improve=4.784935, (0 missing)
##       combin  < 0.0015 to the right, improve=4.528824, (0 missing)
##       mani    < 0.0035 to the right, improve=4.483889, (0 missing)
##       absolut < 5e-04  to the left,  improve=4.117436, (0 missing)
##   Surrogate splits:
##       branch      < 0.003  to the right, agree=0.96, adj=0.917, (0 split)
##       confeder    < 0.0015 to the left,  agree=0.84, adj=0.667, (0 split)
##       confederaci < 5e-04  to the left,  agree=0.84, adj=0.667, (0 split)
##       establish   < 0.003  to the left,  agree=0.84, adj=0.667, (0 split)
##       legisl      < 0.0045 to the right, agree=0.80, adj=0.583, (0 split)
##
## Node number 6: 12 observations
##   predicted class=dispt  expected loss=0.25  P(node) =0.2
##     class counts:     9     0     3
##    probabilities: 0.750 0.000 0.250
##
## Node number 7: 13 observations
##   predicted class=madis  expected loss=0.07692308  P(node) =0.2166667
##     class counts:     0     1    12
##    probabilities: 0.000 0.077 0.923
```

#predict the test dataset using the model for train tree No. 1

```
predictedNoHM = predict(train_treeNoHM, testNoHM, type="class")
(ResultsNoHM <- data.frame(Predicted=predictedNoHM,Actual=testNoHM$Author))
```

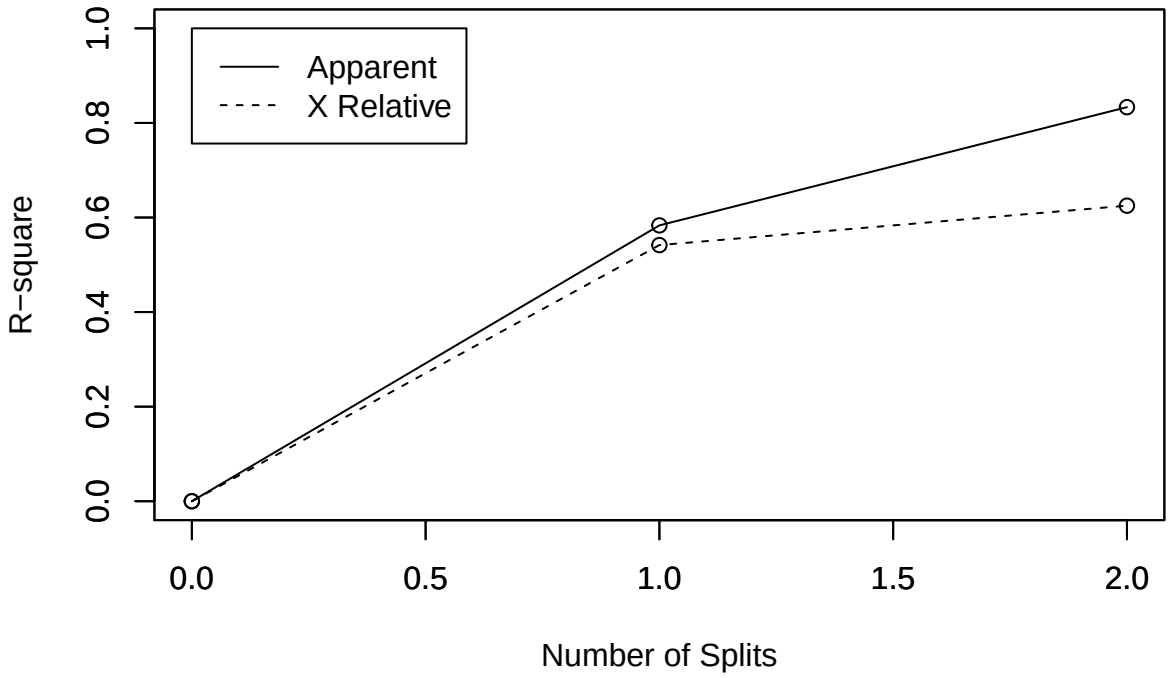
```
##    Predicted Actual
## 1      dispt  dispt
## 2      dispt  dispt
## 3      hamil  hamil
## 4      hamil  hamil
```

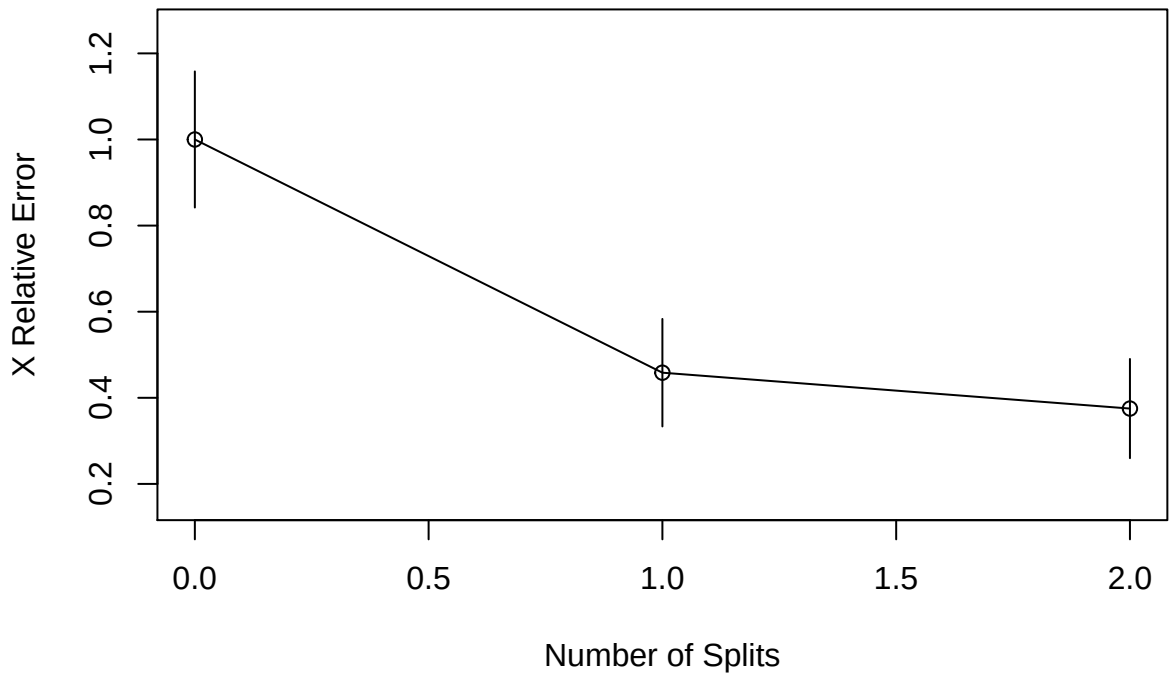
```
## 5      hamil  hamil
## 6      hamil  hamil
## 7      madis  hamil
## 8      hamil  hamil
## 9      hamil  hamil
## 10     hamil  hamil
## 11     hamil  hamil
## 12     madis  hamil
## 13     hamil  hamil
## 14     hamil  hamil
## 15     hamil  hamil
## 16     hamil  hamil
## 17     hamil  hamil
## 18     madis  madis
## 19     dispt  madis
## 20     madis  madis
```

#plot number of splits

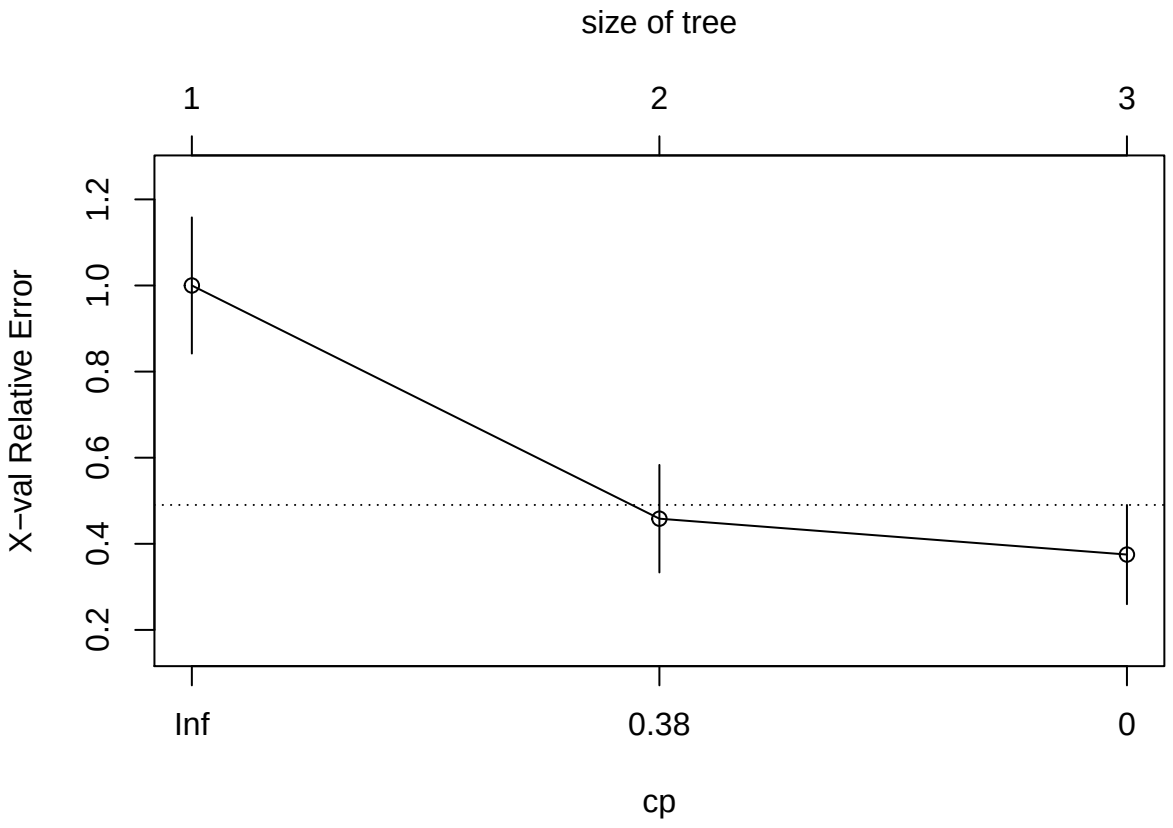
```
rsq.rpart(train_treeNoHM)
```

```
##
## Classification tree:
## rpart(formula = Author ~ ., data = trainNoHM, method = "class",
##     control = rpart.control(cp = 0))
##
## Variables actually used in tree construction:
## [1] union upon
##
## Root node error: 24/60 = 0.4
##
## n= 60
##
##         CP nsplit rel error  xerror    xstd
## 1 0.58333      0   1.00000 1.00000 0.15811
## 2 0.25000      1   0.41667 0.45833 0.12488
## 3 0.00000      2   0.16667 0.37500 0.11524

## Warning in rsq.rpart(train_treeNoHM): may not be applicable for this method
```

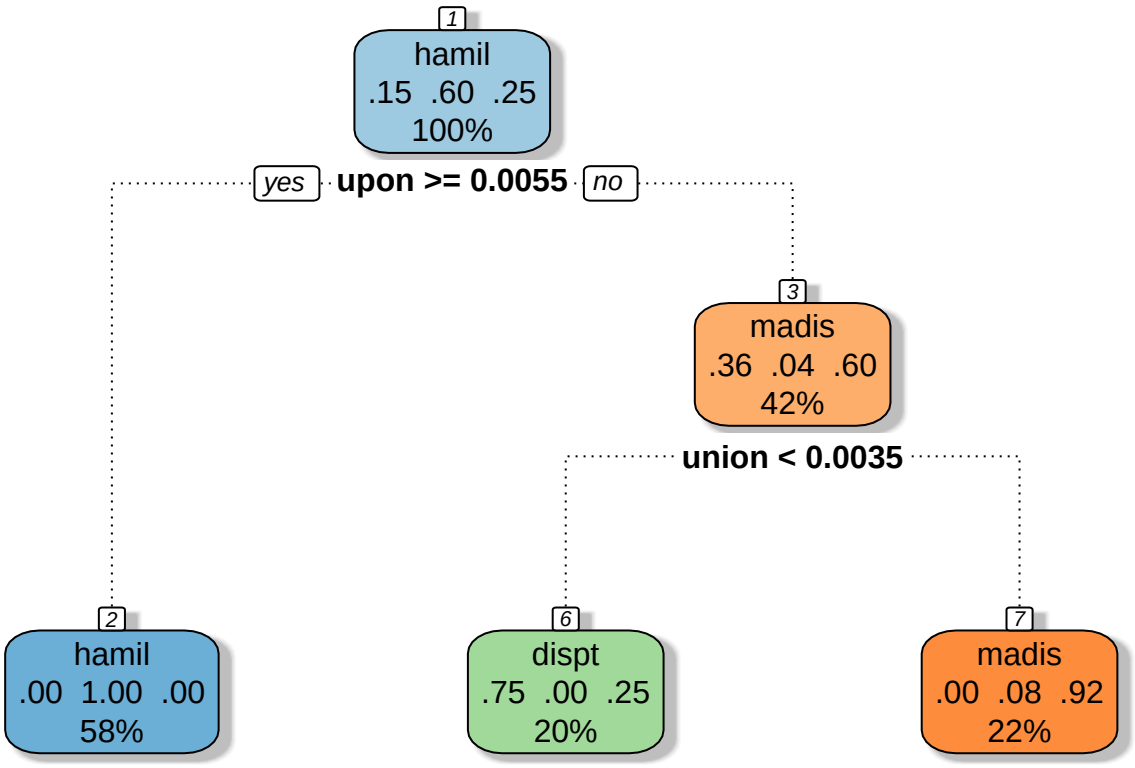

Apparent
X Relative
R−square
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
1.5
2.0
Number of Splits




```
plotcp(train_treeNoHM)
```




```
fancyRpartPlot(train_treeNoHM)
```



#confusion matrix to find correct and incorrect predictions

```
table(Authorship=predictedNoHM, true=testNoHM$Author)
```

```
##           true
## Authorship dispt hamil madis
##      dispt     2     0     1
##      hamil     0    13     0
##      madis     0     2     2
```

#attributed hamilton with disputed

#Train Tree Model 4

```
train_tree4NoHM <- rpart(Author ~ ., data = trainNoHM, method="class", control=rpart.control(cp=0, minsp
summary(train_tree4NoHM)
```

```
## Call:
## rpart(formula = Author ~ ., data = trainNoHM, method = "class",
##     control = rpart.control(cp = 0, minsplit = 2, maxdepth = 5))
##   n= 60
##
##           CP nsplit  rel error    xerror      xstd
## 1 0.58333333      0 1.00000000 1.0000000 0.1581139
## 2 0.25000000      1 0.41666667 0.4583333 0.1248842
```

```
## 3 0.08333333      2 0.16666667 0.2916667 0.1036096
## 4 0.04166667      3 0.08333333 0.3750000 0.1152443
## 5 0.00000000      5 0.00000000 0.3750000 0.1152443
##
## Variable importance
##        upon      matter        kind     assembl       among    maintain
##          18          10           9           8           7           7
##       union      branch    confeder confederaci   establish      legisl
##           6           5           4           4           4           3
##     america       exist         man        seem       addit         act
##           3           2           2           2           2           1
##      answer      affair
##           1           1
##
## Node number 1: 60 observations,    complexity param=0.5833333
##   predicted class=hamil  expected loss=0.4  P(node) =1
##     class counts:     9    36    15
##    probabilities: 0.150 0.600 0.250
##   left son=2 (35 obs) right son=3 (25 obs)
##   Primary splits:
##       upon    < 0.0055 to the right, improve=20.580000, (0 missing)
##       matter  < 5e-04  to the right, improve= 8.992992, (0 missing)
##       kind    < 0.0015 to the right, improve= 8.751429, (0 missing)
##       thing   < 0.0015 to the right, improve= 7.216667, (0 missing)
##       assembl < 0.0025 to the right, improve= 6.894108, (0 missing)
##   Surrogate splits:
##       matter   < 5e-04  to the right, agree=0.800, adj=0.52, (0 split)
##       kind     < 5e-04  to the right, agree=0.783, adj=0.48, (0 split)
##       assembl  < 0.0025 to the left,  agree=0.767, adj=0.44, (0 split)
##       among    < 0.0035 to the left,  agree=0.750, adj=0.40, (0 split)
##       maintain < 0.0015 to the left,  agree=0.750, adj=0.40, (0 split)
##
## Node number 2: 35 observations
##   predicted class=hamil  expected loss=0  P(node) =0.5833333
##     class counts:     0    35     0
##    probabilities: 0.000 1.000 0.000
##
## Node number 3: 25 observations,    complexity param=0.25
##   predicted class=madis  expected loss=0.4  P(node) =0.4166667
##     class counts:     9     1    15
##    probabilities: 0.360 0.040 0.600
##   left son=6 (12 obs) right son=7 (13 obs)
##   Primary splits:
##       union  < 0.0035 to the left,  improve=6.373846, (0 missing)
##       branch < 0.003  to the right, improve=4.784935, (0 missing)
##       calcul < 0.0015 to the right, improve=4.737544, (0 missing)
##       combin < 0.0015 to the right, improve=4.528824, (0 missing)
##       mani   < 0.0035 to the right, improve=4.483889, (0 missing)
##   Surrogate splits:
##       branch      < 0.003  to the right, agree=0.96, adj=0.917, (0 split)
##       confeder    < 0.0015 to the left,  agree=0.84, adj=0.667, (0 split)
##       confederaci < 5e-04  to the left,  agree=0.84, adj=0.667, (0 split)
##       establish   < 0.003  to the left,  agree=0.84, adj=0.667, (0 split)
##       legisl      < 0.0045 to the right, agree=0.80, adj=0.583, (0 split)
```

```
##
## Node number 6: 12 observations,    complexity param=0.08333333
##   predicted class=dispt  expected loss=0.25  P(node) =0.2
##     class counts:     9     0     3
##    probabilities: 0.750 0.000 0.250
##   left son=12 (8 obs) right son=13 (4 obs)
##   Primary splits:
##       america < 0.0015 to the right, improve=3, (0 missing)
##       answer  < 0.001  to the right, improve=3, (0 missing)
##       conclus < 5e-04  to the left,  improve=3, (0 missing)
##       mani    < 0.0025 to the right, improve=3, (0 missing)
##       relat   < 0.001  to the right, improve=3, (0 missing)
##   Surrogate splits:
##       exist  < 0.001  to the right, agree=0.917, adj=0.75, (0 split)
##       man    < 0.001  to the right, agree=0.917, adj=0.75, (0 split)
##       seem   < 0.0035 to the left,  agree=0.917, adj=0.75, (0 split)
##       act    < 0.003  to the left,  agree=0.833, adj=0.50, (0 split)
##       answer < 0.001  to the right, agree=0.833, adj=0.50, (0 split)
##
## Node number 7: 13 observations,    complexity param=0.04166667
##   predicted class=madis  expected loss=0.07692308  P(node) =0.2166667
##     class counts:     0     1    12
##    probabilities: 0.000 0.077 0.923
##   left son=14 (1 obs) right son=15 (12 obs)
##   Primary splits:
##       addit  < 0.0035 to the right, improve=1.846154, (0 missing)
##       afford < 0.0065 to the right, improve=1.846154, (0 missing)
##       alway  < 0.0035 to the right, improve=1.846154, (0 missing)
##       calcul < 0.0015 to the right, improve=1.846154, (0 missing)
##       combin < 0.005  to the right, improve=1.846154, (0 missing)
##
## Node number 12: 8 observations
##   predicted class=dispt  expected loss=0  P(node) =0.1333333
##     class counts:     8     0     0
##    probabilities: 1.000 0.000 0.000
##
## Node number 13: 4 observations,    complexity param=0.04166667
##   predicted class=madis  expected loss=0.25  P(node) =0.06666667
##     class counts:     1     0     3
##    probabilities: 0.250 0.000 0.750
##   left son=26 (1 obs) right son=27 (3 obs)
##   Primary splits:
##       affair < 0.007  to the right, improve=1.5, (0 missing)
##       among  < 0.0055 to the right, improve=1.5, (0 missing)
##       amount < 0.0015 to the right, improve=1.5, (0 missing)
##       answer < 0.001  to the right, improve=1.5, (0 missing)
##       appear < 0.002  to the left,  improve=1.5, (0 missing)
##
## Node number 14: 1 observations
##   predicted class=hamil  expected loss=0  P(node) =0.01666667
##     class counts:     0     1     0
##    probabilities: 0.000 1.000 0.000
##
## Node number 15: 12 observations
```

```
##   predicted class=madis  expected loss=0  P(node) =0.2
##     class counts:     0     0    12
##    probabilities: 0.000 0.000 1.000
##
## Node number 26: 1 observations
##   predicted class=dispt  expected loss=0  P(node) =0.01666667
##     class counts:     1     0     0
##    probabilities: 1.000 0.000 0.000
##
## Node number 27: 3 observations
##   predicted class=madis  expected loss=0  P(node) =0.05
##     class counts:     0     0     3
##    probabilities: 0.000 0.000 1.000
```

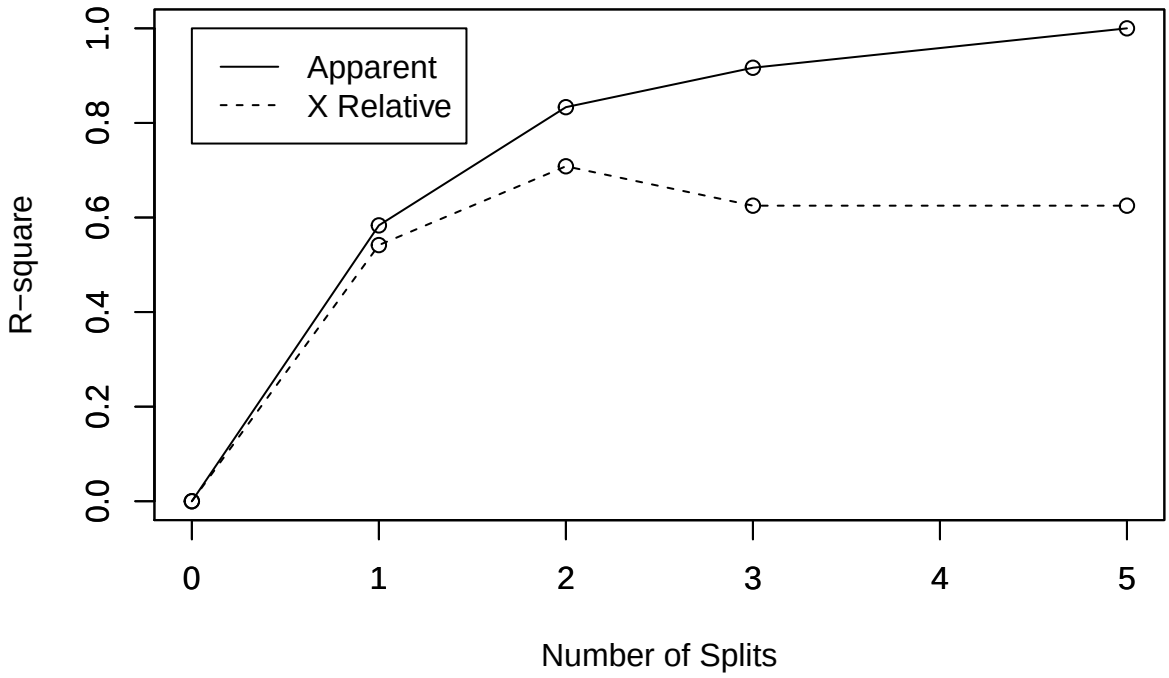
#predict the test dataset using the model for train tree No. 1

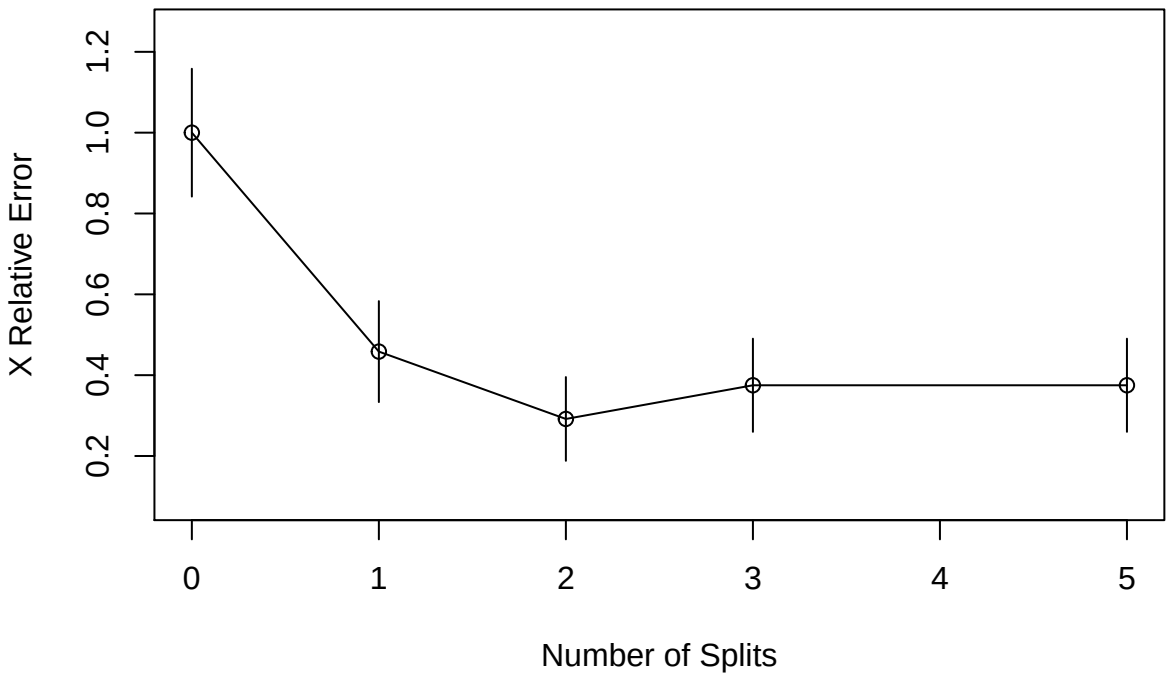
```
predicted4NoHM = predict(train_tree4NoHM, testNoHM, type="class")
```

#plot number of splits

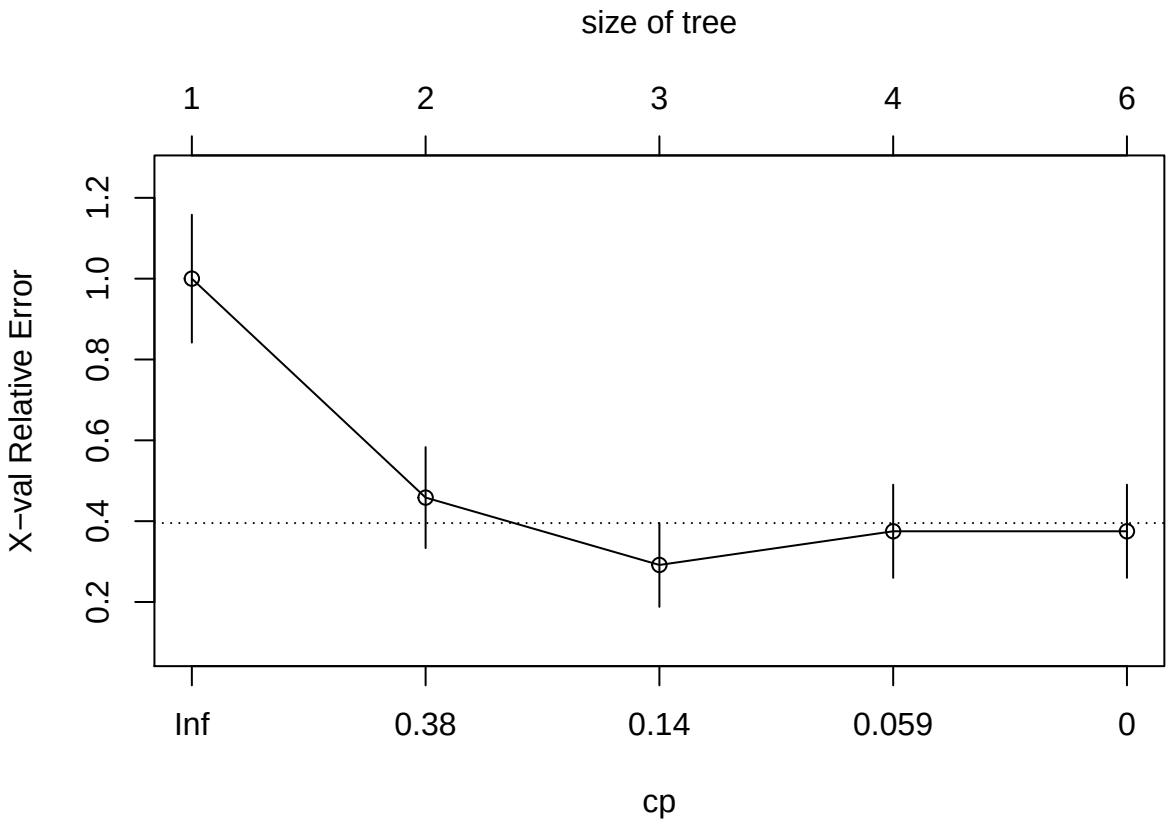
```
rsq.rpart(train_tree4NoHM)
```

```
##
## Classification tree:
## rpart(formula = Author ~ ., data = trainNoHM, method = "class",
##     control = rpart.control(cp = 0, minsplit = 2, maxdepth = 5))
##
## Variables actually used in tree construction:
## [1] addit   affair  america union   upon
##
## Root node error: 24/60 = 0.4
##
## n= 60
##
##         CP nsplit rel error  xerror    xstd
## 1 0.583333      0  1.000000 1.00000 0.15811
## 2 0.250000      1  0.416667 0.45833 0.12488
## 3 0.083333      2  0.166667 0.29167 0.10361
## 4 0.041667      3  0.083333 0.37500 0.11524
## 5 0.000000      5  0.000000 0.37500 0.11524

## Warning in rsq.rpart(train_tree4NoHM): may not be applicable for this method
```

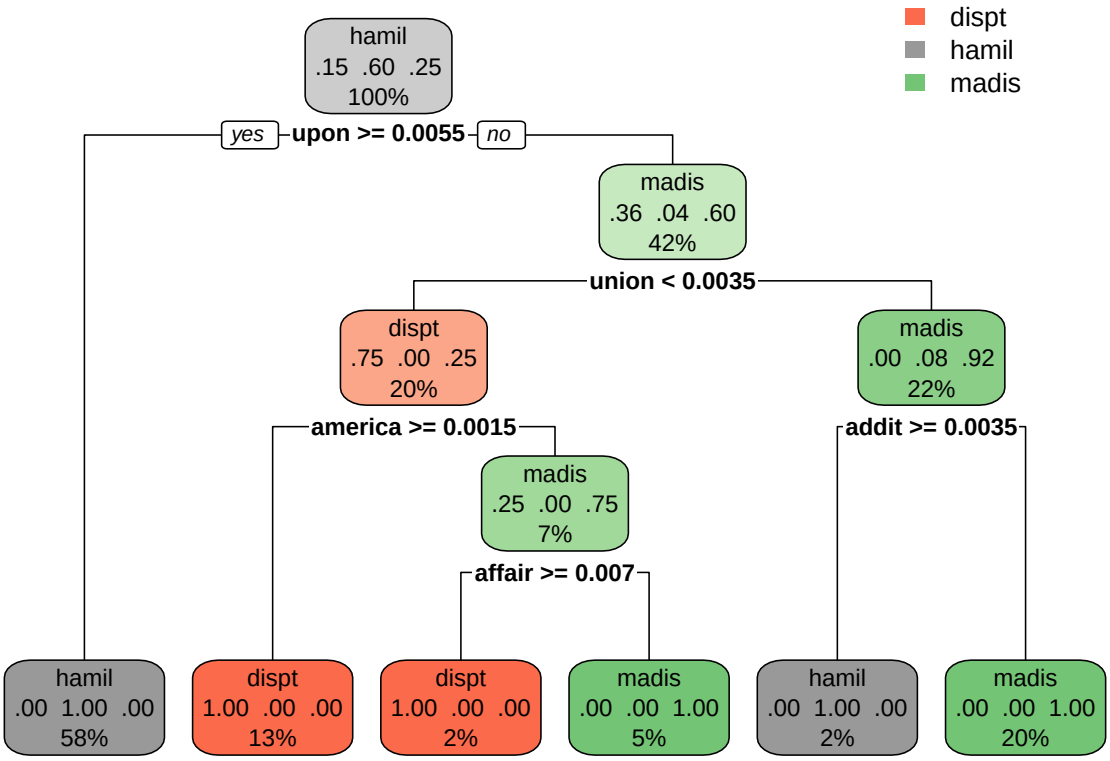

Apparent
X Relative
R−square
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
5
Number of Splits




```
plotcp(train_tree4NoHM)
```




```
rpart.plot(train_tree4NoHM)
```



#confusion matrix to find correct and incorrect predictions

```
table(Authorship=predicted4NoHM, true=testNoHM$Author)
```

```
##           true
## Authorship dispt hamil madis
##      dispt     0     0     0
##      hamil     0    13     0
##      madis     2     2     3
```

```
(Results4NoHM<-data.frame(Predicted=predicted4NoHM, Actual=testNoHM$Author))
```

```
##    Predicted Actual
## 1      madis  dispt
## 2      madis  dispt
## 3      hamil  hamil
## 4      hamil  hamil
## 5      hamil  hamil
## 6      hamil  hamil
## 7      madis  hamil
## 8      hamil  hamil
## 9      hamil  hamil
## 10     hamil  hamil
## 11     hamil  hamil
## 12     madis  hamil
```

```
## 13     hamil  hamil
## 14     hamil  hamil
## 15     hamil  hamil
## 16     hamil  hamil
## 17     hamil  hamil
## 18     madis  madis
## 19     madis  madis
## 20     madis  madis
```

#Train Tree 5

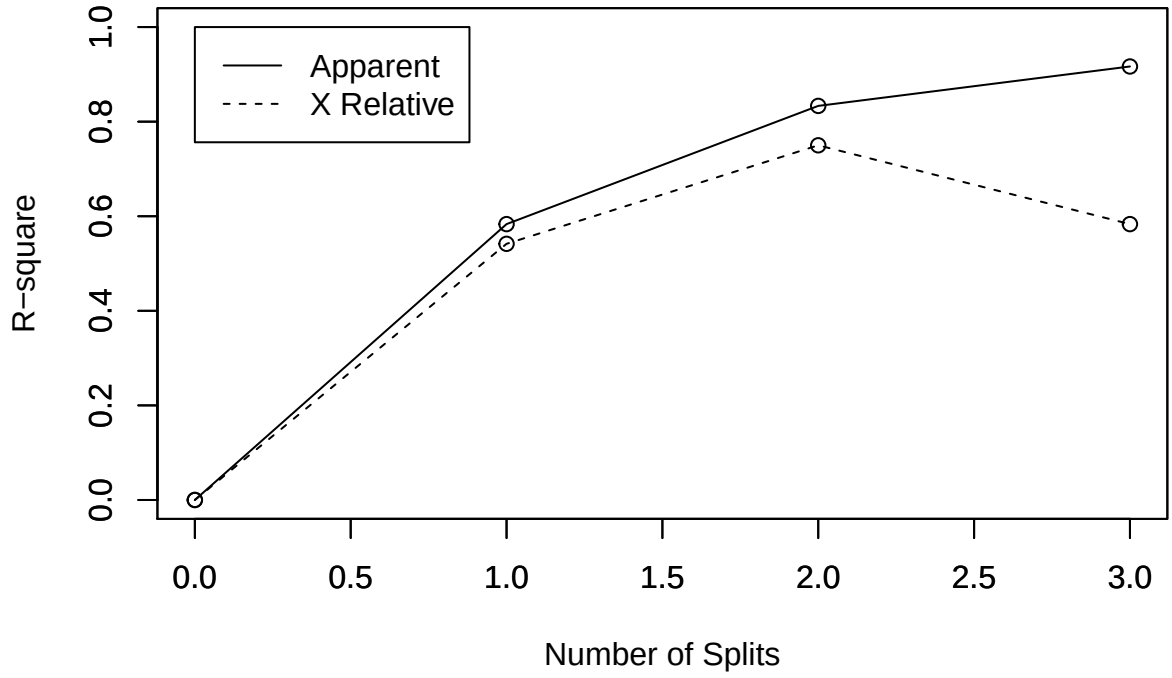
```
train_tree5NoHM <- rpart(Author ~ ., data = trainNoHM, method="class", control=rpart.control(cp=0, minsp
summary(train_tree5NoHM)
```

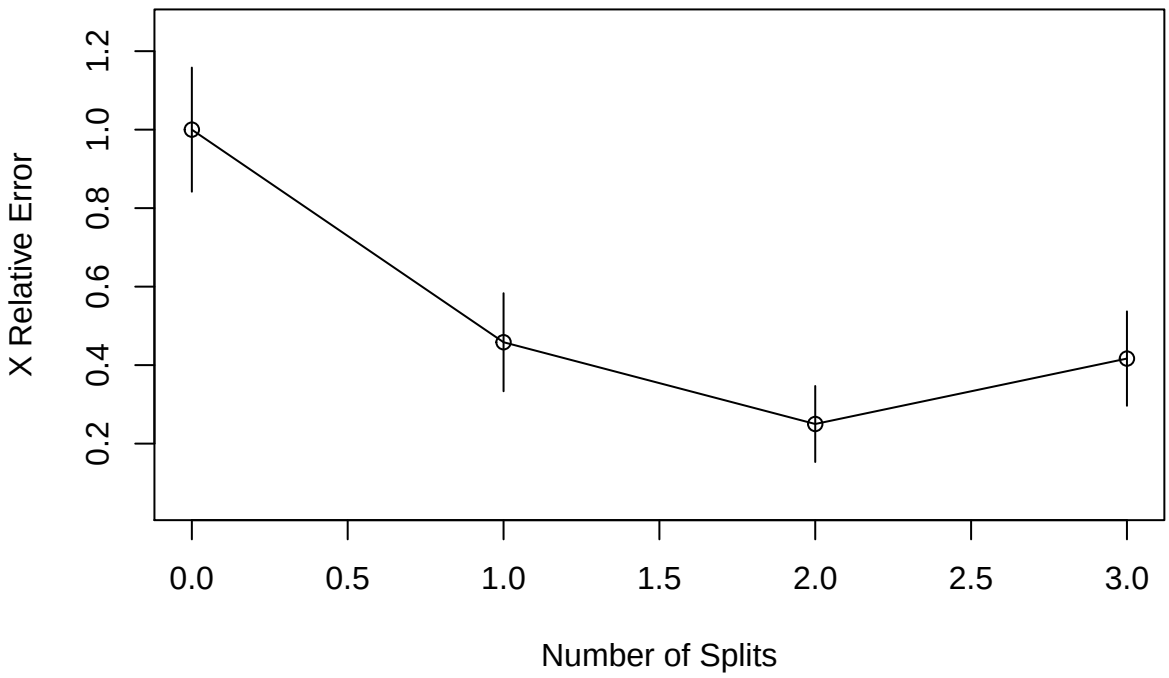
```
## Call:
## rpart(formula = Author ~ ., data = trainNoHM, method = "class",
##     control = rpart.control(cp = 0, minsplit = 5, maxdepth = 7))
##   n= 60
##
##           CP nsplit  rel error    xerror       xstd
## 1 0.58333333      0 1.00000000 1.0000000 0.15811388
## 2 0.25000000      1 0.41666667 0.4583333 0.12488421
## 3 0.08333333      2 0.16666667 0.2500000 0.09682458
## 4 0.00000000      3 0.08333333 0.4166667 0.12028131
##
## Variable importance
##        upon      matter        kind     assembl       among    maintain
##          19          10           9           8           8           8
##       union      branch    confeder confederaci   establish      legisl
##           6           5           4           4           4           3
##     america       exist         man        seem         act      answer
##           3           2           2           2           1           1
##
## Node number 1: 60 observations,    complexity param=0.5833333
##   predicted class=hamil  expected loss=0.4  P(node) =1
##     class counts:     9    36    15
##    probabilities: 0.150 0.600 0.250
##   left son=2 (35 obs) right son=3 (25 obs)
##   Primary splits:
##       upon    < 0.0055 to the right, improve=20.580000, (0 missing)
##       matter  < 5e-04  to the right, improve= 8.992992, (0 missing)
##       kind    < 0.0015 to the right, improve= 8.751429, (0 missing)
##       thing   < 0.0015 to the right, improve= 7.216667, (0 missing)
##       assembl < 0.0025 to the right, improve= 6.894108, (0 missing)
##   Surrogate splits:
##       matter   < 5e-04  to the right, agree=0.800, adj=0.52, (0 split)
##       kind     < 5e-04  to the right, agree=0.783, adj=0.48, (0 split)
##       assembl  < 0.0025 to the left,  agree=0.767, adj=0.44, (0 split)
##       among    < 0.0035 to the left,  agree=0.750, adj=0.40, (0 split)
##       maintain < 0.0015 to the left,  agree=0.750, adj=0.40, (0 split)
##
## Node number 2: 35 observations
##   predicted class=hamil  expected loss=0  P(node) =0.5833333
```

```
##     class counts:     0    35     0
##    probabilities: 0.000 1.000 0.000
##
## Node number 3: 25 observations,    complexity param=0.25
##   predicted class=madis  expected loss=0.4  P(node) =0.4166667
##     class counts:     9     1    15
##    probabilities: 0.360 0.040 0.600
##   left son=6 (12 obs) right son=7 (13 obs)
##   Primary splits:
##       union  < 0.0035 to the left,  improve=6.373846, (0 missing)
##       branch < 0.003  to the right, improve=4.784935, (0 missing)
##       calcul < 0.0015 to the right, improve=4.737544, (0 missing)
##       combin < 0.0015 to the right, improve=4.528824, (0 missing)
##       mani   < 0.0035 to the right, improve=4.483889, (0 missing)
##   Surrogate splits:
##       branch      < 0.003  to the right, agree=0.96, adj=0.917, (0 split)
##       confeder    < 0.0015 to the left,  agree=0.84, adj=0.667, (0 split)
##       confederaci < 5e-04  to the left,  agree=0.84, adj=0.667, (0 split)
##       establish   < 0.003  to the left,  agree=0.84, adj=0.667, (0 split)
##       legisl      < 0.0045 to the right, agree=0.80, adj=0.583, (0 split)
##
## Node number 6: 12 observations,    complexity param=0.08333333
##   predicted class=dispt  expected loss=0.25  P(node) =0.2
##     class counts:     9     0     3
##    probabilities: 0.750 0.000 0.250
##   left son=12 (8 obs) right son=13 (4 obs)
##   Primary splits:
##       america < 0.0015 to the right, improve=3, (0 missing)
##       answer  < 0.001  to the right, improve=3, (0 missing)
##       conclus < 5e-04  to the left,  improve=3, (0 missing)
##       mani    < 0.0025 to the right, improve=3, (0 missing)
##       relat   < 0.001  to the right, improve=3, (0 missing)
##   Surrogate splits:
##       exist  < 0.001  to the right, agree=0.917, adj=0.75, (0 split)
##       man    < 0.001  to the right, agree=0.917, adj=0.75, (0 split)
##       seem   < 0.0035 to the left,  agree=0.917, adj=0.75, (0 split)
##       act    < 0.003  to the left,  agree=0.833, adj=0.50, (0 split)
##       answer < 0.001  to the right, agree=0.833, adj=0.50, (0 split)
##
## Node number 7: 13 observations
##   predicted class=madis  expected loss=0.07692308  P(node) =0.2166667
##     class counts:     0     1    12
##    probabilities: 0.000 0.077 0.923
##
## Node number 12: 8 observations
##   predicted class=dispt  expected loss=0  P(node) =0.1333333
##     class counts:     8     0     0
##    probabilities: 1.000 0.000 0.000
##
## Node number 13: 4 observations
##   predicted class=madis  expected loss=0.25  P(node) =0.06666667
##     class counts:     1     0     3
##    probabilities: 0.250 0.000 0.750
```

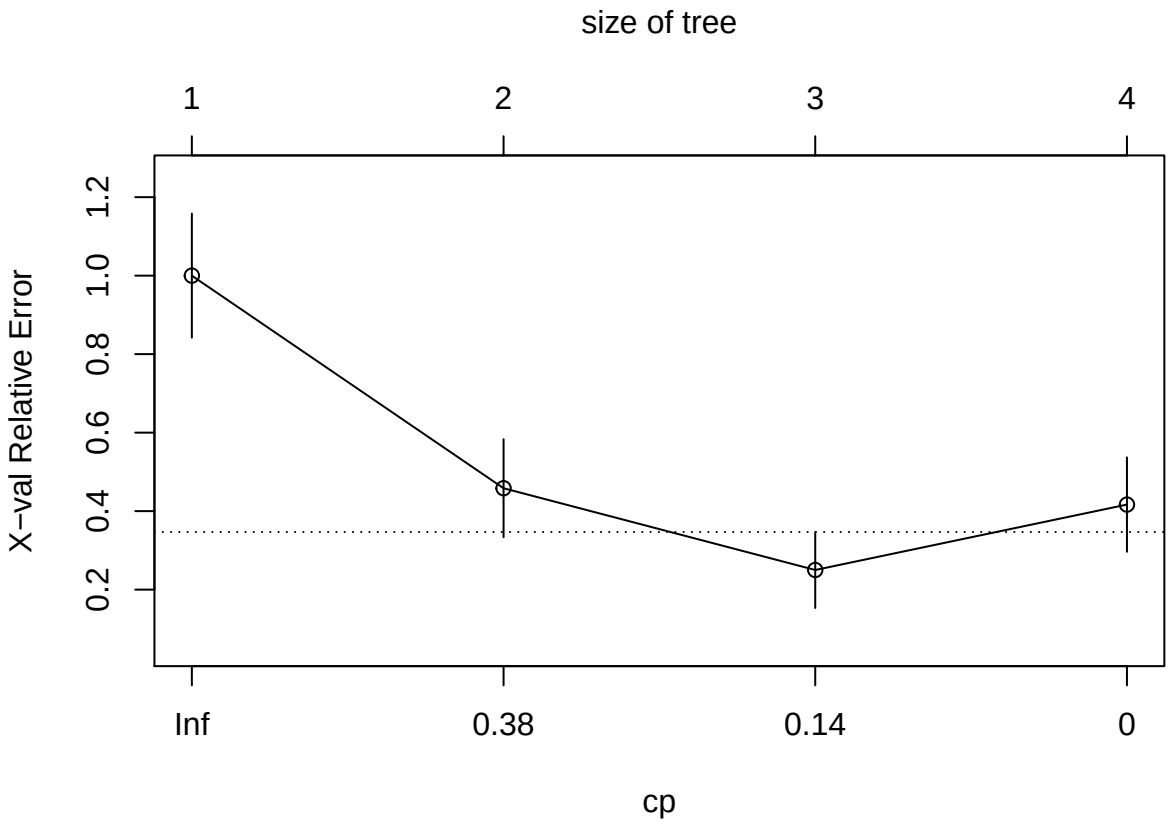
```
predicted5NoHM= predict(train_tree5NoHM, testNoHM, type="class")
rsq.rpart(train_tree5NoHM)
```

```
##
## Classification tree:
## rpart(formula = Author ~ ., data = trainNoHM, method = "class",
##     control = rpart.control(cp = 0, minsplit = 5, maxdepth = 7))
##
## Variables actually used in tree construction:
## [1] america union   upon
##
## Root node error: 24/60 = 0.4
##
## n= 60
##
##         CP nsplit rel error  xerror     xstd
## 1 0.583333      0  1.000000 1.00000 0.158114
## 2 0.250000      1  0.416667 0.45833 0.124884
## 3 0.083333      2  0.166667 0.25000 0.096825
## 4 0.000000      3  0.083333 0.41667 0.120281

## Warning in rsq.rpart(train_tree5NoHM): may not be applicable for this method
```

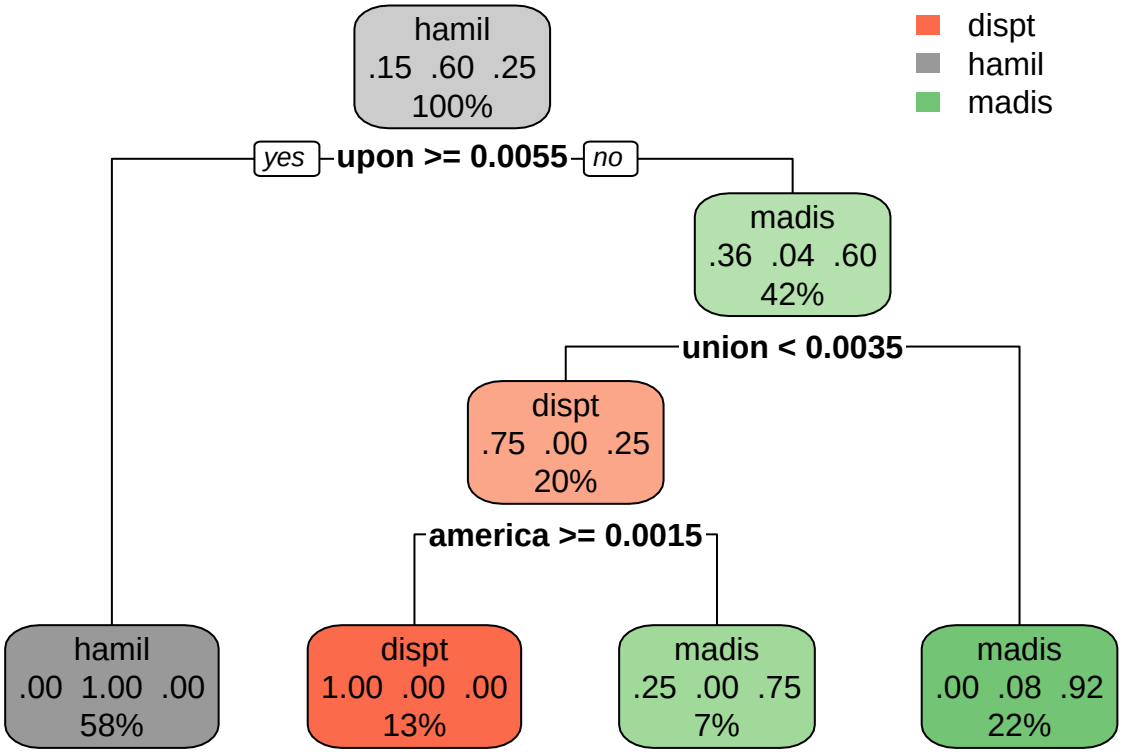





```
plotcp(train_tree5NoHM)
```




```
rpart.plot(train_tree5NoHM)
```




```
table(Authorship=predicted5NoHM, true = testNoHM$Author)
```

```
##           true
## Authorship dispt hamil madis
##      dispt     0     0     0
##      hamil     0    13     0
##      madis     2     2     3
```

```
(Results5NoHM<-data.frame(Predicted=predicted5NoHM, Actual=testNoHM$Author))
```

```
##    Predicted Actual
## 1      madis  dispt
## 2      madis  dispt
## 3      hamil  hamil
## 4      hamil  hamil
## 5      hamil  hamil
## 6      hamil  hamil
## 7      madis  hamil
## 8      hamil  hamil
## 9      hamil  hamil
## 10     hamil  hamil
## 11     hamil  hamil
## 12     madis  hamil
## 13     hamil  hamil
```

```
## 14     hamil  hamil
## 15     hamil  hamil
## 16     hamil  hamil
## 17     hamil  hamil
## 18     madis  madis
## 19     madis  madis
## 20     madis  madis
```

Leaving in HM papers

```
Papers_DFNoJay <- as.data.frame(as.matrix(Papers_Matrix_Norm2))
```

#remove Jays papers

```
Papers_DFNoJay<-Papers_DFNoJay[-66:-70,]
```

remove Ham Mad papers

```
Papers_DFNoJay <- Papers_DFNoJay[63:65]

Papers_DFNoJay<- Papers_DFNoJay%>%add_rownames()

names(Papers_DFNoJay)[1]<-"Author"
Papers_DFNoJay[1:11,1]="dispt"
Papers_DFNoJay[12:65,1]="hamil"
Papers_DFNoJay[66:80,1]="madis"
```

##Make Train and Test sets

```
trainRatio <- .75
set.seed(11) # Set Seed so that same sample can be reproduced in future also
sampleNoJay <- sample.int(n = nrow(Papers_DFNoJay), size = floor(trainRatio*nrow(Papers_DFNoJay)), repla
train2NoJay <- Papers_DFNoJay[sampleNoJay, ]
test2NoJay <- Papers_DFNoJay[-sampleNoJay, ]
```

train / test ratio

```
length(sampleNoJay)/nrow(Papers_DFNoJay)
```

```
## [1] 0.75
```

##Decision Tree Models #Train Tree Model 3 NoJay

```
train_tree3NoJay <- rpart(Author ~ ., data = train2NoJay, method="class", control=rpart.control(cp=0))
summary(train_tree3NoJay)
```

```
## Call:
## rpart(formula = Author ~ ., data = train2NoJay, method = "class",
##     control = rpart.control(cp = 0))
##   n= 60
##
##           CP nsplit rel error   xerror      xstd
## 1 0.04545455      0 1.0000000 1.000000 0.1696699
## 2 0.00000000      2 0.9090909 1.090909 0.1724879
##
## Variable importance
## conclus concern
##      60      40
##
## Node number 1: 60 observations,    complexity param=0.04545455
##   predicted class=hamil  expected loss=0.3666667  P(node) =1
##     class counts:     9    38    13
##    probabilities: 0.150 0.633 0.217
##   left son=2 (22 obs) right son=3 (38 obs)
##   Primary splits:
##       concern < 5e-04  to the left,  improve=1.7427430, (0 missing)
##       conclus < 0.0015 to the left,  improve=1.0111110, (0 missing)
##       compos  < 5e-04  to the left,  improve=0.6943641, (0 missing)
##
## Node number 2: 22 observations,    complexity param=0.04545455
##   predicted class=hamil  expected loss=0.5454545  P(node) =0.3666667
##     class counts:     6    10     6
##    probabilities: 0.273 0.455 0.273
##   left son=4 (14 obs) right son=5 (8 obs)
##   Primary splits:
##       conclus < 5e-04  to the left,  improve=2.5746750, (0 missing)
##       compos  < 0.0015 to the right, improve=0.4294372, (0 missing)
##
## Node number 3: 38 observations
##   predicted class=hamil  expected loss=0.2631579  P(node) =0.6333333
##     class counts:     3    28     7
##    probabilities: 0.079 0.737 0.184
##
## Node number 4: 14 observations
##   predicted class=hamil  expected loss=0.5  P(node) =0.2333333
##     class counts:     6     7     1
##    probabilities: 0.429 0.500 0.071
##
## Node number 5: 8 observations
##   predicted class=madis  expected loss=0.375  P(node) =0.1333333
##     class counts:     0     3     5
##    probabilities: 0.000 0.375 0.625
```

#predict the test dataset using the model for train tree No. 1

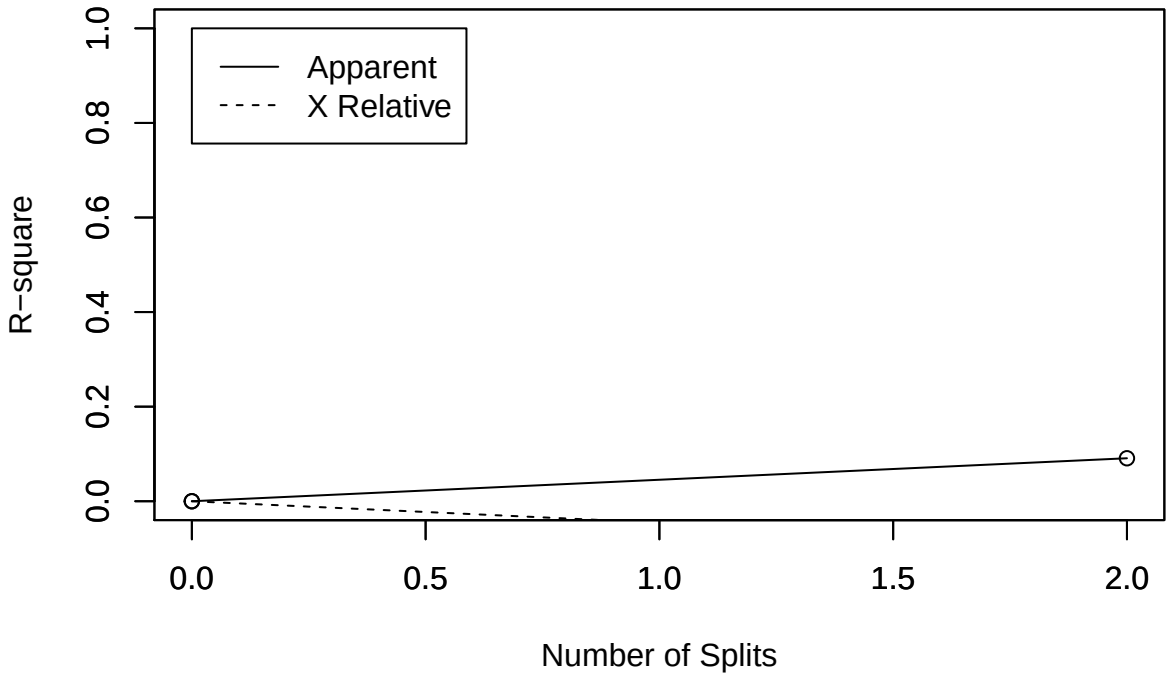
```
predicted3NoJay= predict(train_tree3NoJay, test2NoJay, type="class")
(Results3NoJay <- data.frame(Predicted=predicted3NoJay,Actual=test2NoJay$Author))
```

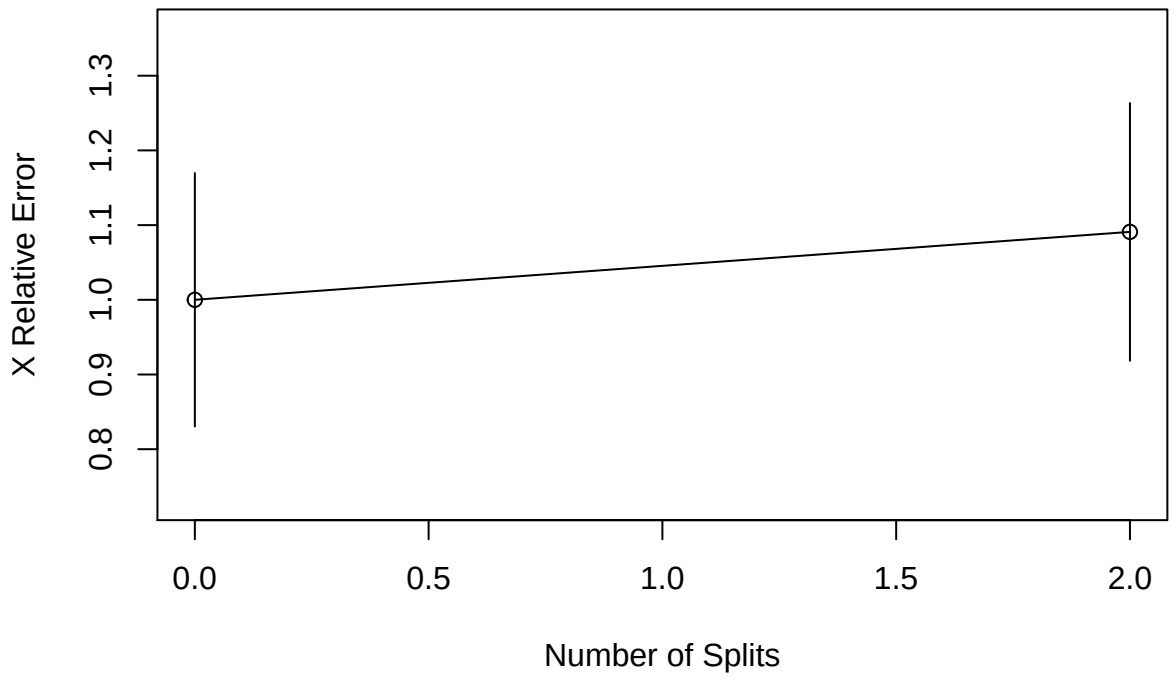
```
##    Predicted Actual
## 1      hamil  dispt
## 2      madis  dispt
## 3      hamil  hamil
## 4      hamil  hamil
## 5      hamil  hamil
## 6      hamil  hamil
## 7      hamil  hamil
## 8      hamil  hamil
## 9      hamil  hamil
## 10     hamil  hamil
## 11     hamil  hamil
## 12     madis  hamil
## 13     hamil  hamil
## 14     hamil  hamil
## 15     hamil  hamil
## 16     madis  hamil
## 17     hamil  hamil
## 18     madis  hamil
## 19     hamil  madis
## 20     hamil  madis
```

#plot number of splits

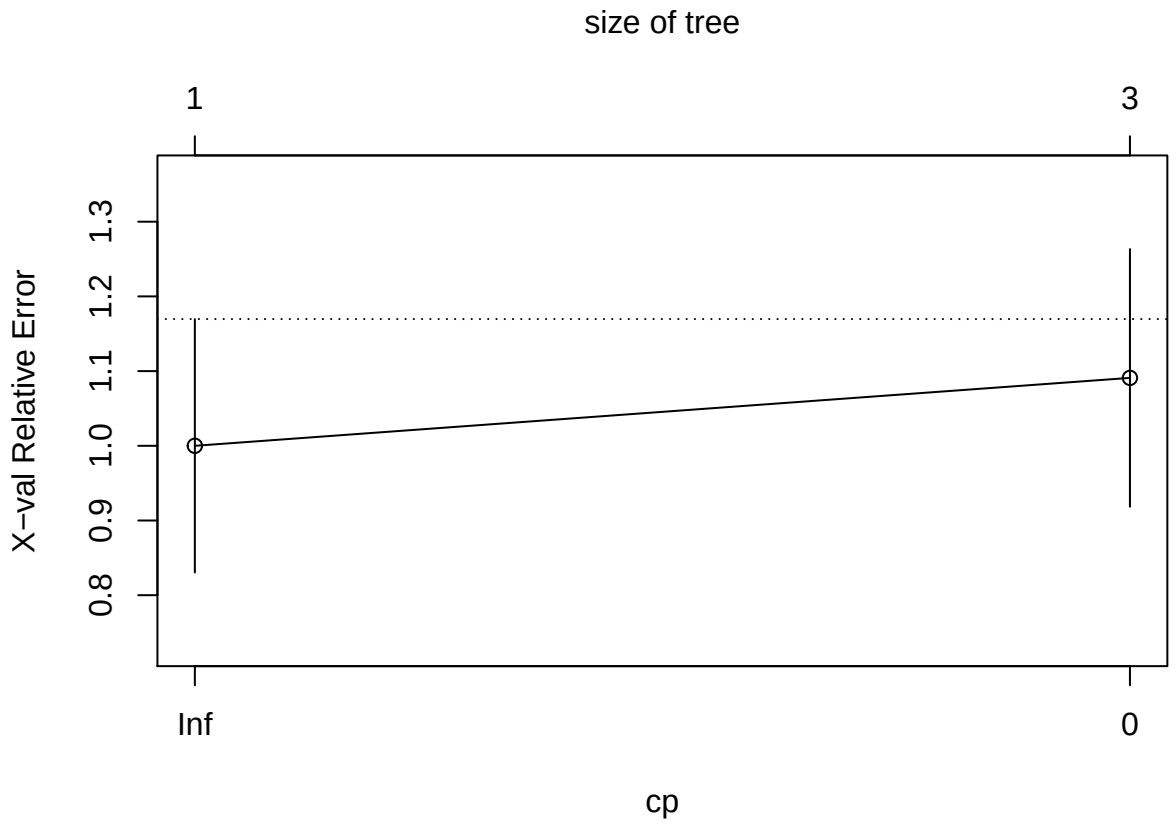
```
rsq.rpart(train_tree3NoJay)
```

```
##
## Classification tree:
## rpart(formula = Author ~ ., data = train2NoJay, method = "class",
##     control = rpart.control(cp = 0))
##
## Variables actually used in tree construction:
## [1] concern conclus
##
## Root node error: 22/60 = 0.36667
##
## n= 60
##
##         CP nsplit rel error xerror    xstd
## 1 0.045455      0   1.00000 1.0000 0.16967
## 2 0.000000      2   0.90909 1.0909 0.17249

## Warning in rsq.rpart(train_tree3NoJay): may not be applicable for this method
```

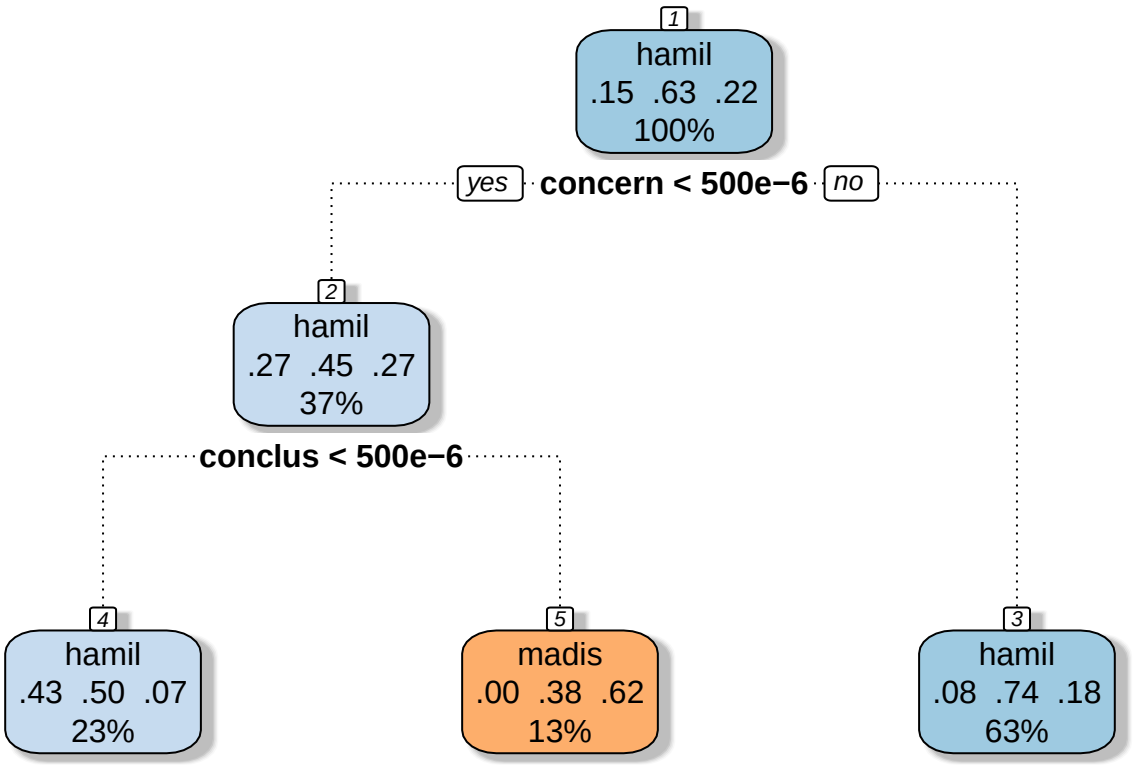

Apparent
X Relative
R−square
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
1.5
2.0
Number of Splits




```
plotcp(train_tree3NoJay)
```




```
fancyRpartPlot(train_tree3NoJay)
```



#confusion matrix to find correct and incorrect predictions

```
table(Authorship=predicted3NoJay, true=test2NoJay$Author)
```

```
##           true
## Authorship dispt hamil madis
##      dispt     0     0     0
##      hamil     1    13     2
##      madis     1     3     0
```

#attributed hamilton with disputed

#Train Tree Model 4

```
train_tree4NoJay <- rpart(Author ~ ., data = train2NoJay, method="class", control=rpart.control(cp=0, mi
summary(train_tree4NoJay)
```

```
## Call:
## rpart(formula = Author ~ ., data = train2NoJay, method = "class",
##     control = rpart.control(cp = 0, minsplit = 5, maxdepth = 5))
##   n= 60
##
##           CP nsplit rel error   xerror      xstd
## 1 0.06060606      0 1.0000000 1.000000 0.1696699
## 2 0.04545455      3 0.8181818 1.318182 0.1759469
## 3 0.03030303      5 0.7272727 1.227273 0.1751623
```

```
## 4 0.00000000      8 0.6363636 1.227273 0.1751623
##
## Variable importance
## conclus  compos concern
##      47      26      26
##
## Node number 1: 60 observations,    complexity param=0.06060606
##   predicted class=hamil  expected loss=0.3666667  P(node) =1
##     class counts:     9    38    13
##    probabilities: 0.150 0.633 0.217
##   left son=2 (22 obs) right son=3 (38 obs)
##   Primary splits:
##       concern < 5e-04  to the left,  improve=1.7427430, (0 missing)
##       conclus < 0.0025 to the right, improve=1.1121210, (0 missing)
##       compos  < 5e-04  to the left,  improve=0.6943641, (0 missing)
##
## Node number 2: 22 observations,    complexity param=0.06060606
##   predicted class=hamil  expected loss=0.5454545  P(node) =0.3666667
##     class counts:     6    10     6
##    probabilities: 0.273 0.455 0.273
##   left son=4 (14 obs) right son=5 (8 obs)
##   Primary splits:
##       conclus < 5e-04  to the left,  improve=2.5746750, (0 missing)
##       compos  < 0.0015 to the right, improve=0.4294372, (0 missing)
##
## Node number 3: 38 observations,    complexity param=0.04545455
##   predicted class=hamil  expected loss=0.2631579  P(node) =0.6333333
##     class counts:     3    28     7
##    probabilities: 0.079 0.737 0.184
##   left son=6 (35 obs) right son=7 (3 obs)
##   Primary splits:
##       compos  < 0.0035 to the left,  improve=1.308772, (0 missing)
##       concern < 0.0055 to the left,  improve=1.308772, (0 missing)
##       conclus < 5e-04  to the right, improve=0.604010, (0 missing)
##
## Node number 4: 14 observations,    complexity param=0.04545455
##   predicted class=hamil  expected loss=0.5  P(node) =0.2333333
##     class counts:     6     7     1
##    probabilities: 0.429 0.500 0.071
##   left son=8 (5 obs) right son=9 (9 obs)
##   Primary splits:
##       compos < 0.001  to the right, improve=0.3460317, (0 missing)
##
## Node number 5: 8 observations,    complexity param=0.06060606
##   predicted class=madis  expected loss=0.375  P(node) =0.1333333
##     class counts:     0     3     5
##    probabilities: 0.000 0.375 0.625
##   left son=10 (2 obs) right son=11 (6 obs)
##   Primary splits:
##       conclus < 0.0025 to the right, improve=2.08333300, (0 missing)
##       compos  < 0.0015 to the right, improve=0.08333333, (0 missing)
##
## Node number 6: 35 observations,    complexity param=0.03030303
##   predicted class=hamil  expected loss=0.2285714  P(node) =0.5833333
```

```
##     class counts:     3    27     5
##    probabilities: 0.086 0.771 0.143
##   left son=12 (18 obs) right son=13 (17 obs)
##   Primary splits:
##       compos  < 5e-04  to the right, improve=1.6640520, (0 missing)
##       conclus < 0.0015 to the right, improve=0.9692308, (0 missing)
##       concern < 0.0055 to the left,  improve=0.4424242, (0 missing)
##   Surrogate splits:
##       concern < 0.0045 to the right, agree=0.571, adj=0.118, (0 split)
##
## Node number 7: 3 observations
##   predicted class=madis  expected loss=0.3333333  P(node) =0.05
##     class counts:     0     1     2
##    probabilities: 0.000 0.333 0.667
##
## Node number 8: 5 observations
##   predicted class=dispt  expected loss=0.4  P(node) =0.08333333
##     class counts:     3     2     0
##    probabilities: 0.600 0.400 0.000
##
## Node number 9: 9 observations
##   predicted class=hamil  expected loss=0.4444444  P(node) =0.15
##     class counts:     3     5     1
##    probabilities: 0.333 0.556 0.111
##
## Node number 10: 2 observations
##   predicted class=hamil  expected loss=0  P(node) =0.03333333
##     class counts:     0     2     0
##    probabilities: 0.000 1.000 0.000
##
## Node number 11: 6 observations
##   predicted class=madis  expected loss=0.1666667  P(node) =0.1
##     class counts:     0     1     5
##    probabilities: 0.000 0.167 0.833
##
## Node number 12: 18 observations
##   predicted class=hamil  expected loss=0.05555556  P(node) =0.3
##     class counts:     0    17     1
##    probabilities: 0.000 0.944 0.056
##
## Node number 13: 17 observations,    complexity param=0.03030303
##   predicted class=hamil  expected loss=0.4117647  P(node) =0.2833333
##     class counts:     3    10     4
##    probabilities: 0.176 0.588 0.235
##   left son=26 (4 obs) right son=27 (13 obs)
##   Primary splits:
##       conclus < 0.0015 to the right, improve=1.339367, (0 missing)
##       concern < 0.0035 to the right, improve=1.071301, (0 missing)
##
## Node number 26: 4 observations
##   predicted class=hamil  expected loss=0  P(node) =0.06666667
##     class counts:     0     4     0
##    probabilities: 0.000 1.000 0.000
##
```

```
## Node number 27: 13 observations,    complexity param=0.03030303
##   predicted class=hamil  expected loss=0.5384615  P(node) =0.2166667
##     class counts:     3     6     4
##    probabilities: 0.231 0.462 0.308
##   left son=54 (10 obs) right son=55 (3 obs)
##   Primary splits:
##       concern < 0.0015 to the right, improve=1.3743590, (0 missing)
##       conclus < 5e-04  to the right, improve=0.3986014, (0 missing)
##
## Node number 54: 10 observations
##   predicted class=hamil  expected loss=0.4  P(node) =0.1666667
##     class counts:     2     6     2
##    probabilities: 0.200 0.600 0.200
##
## Node number 55: 3 observations
##   predicted class=madis  expected loss=0.3333333  P(node) =0.05
##     class counts:     1     0     2
##    probabilities: 0.333 0.000 0.667
```

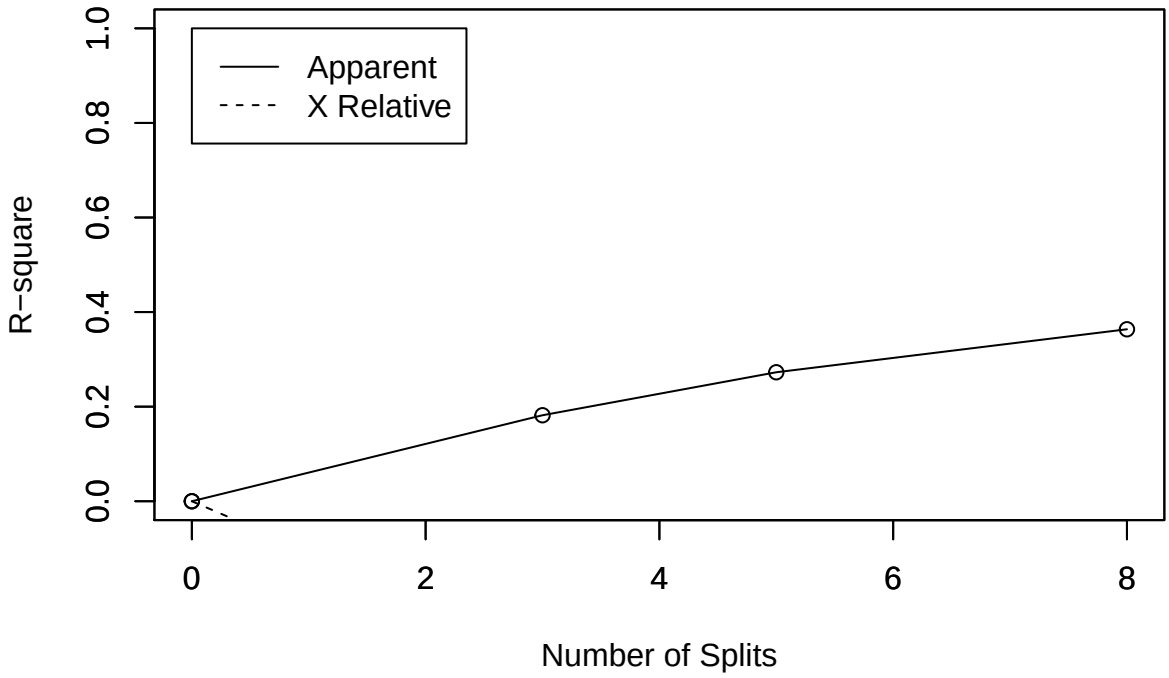
#predict the test dataset using the model for train tree No. 1

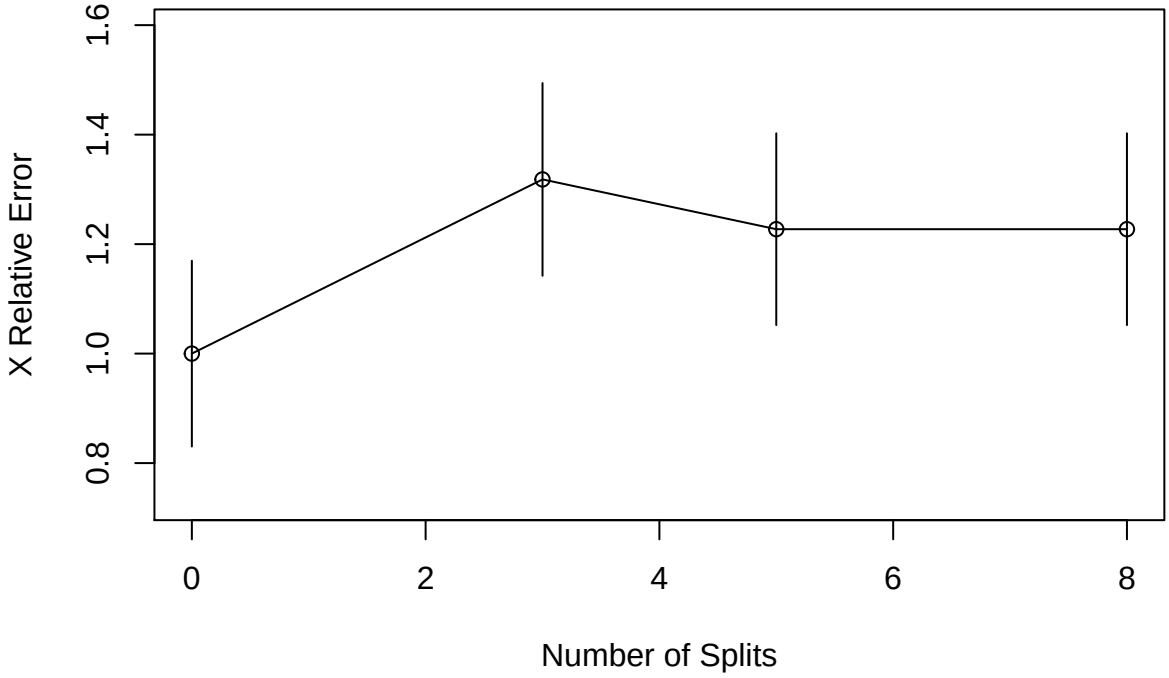
```
predicted4NoJay= predict(train_tree4NoJay, test2NoJay, type="class")
```

#plot number of splits

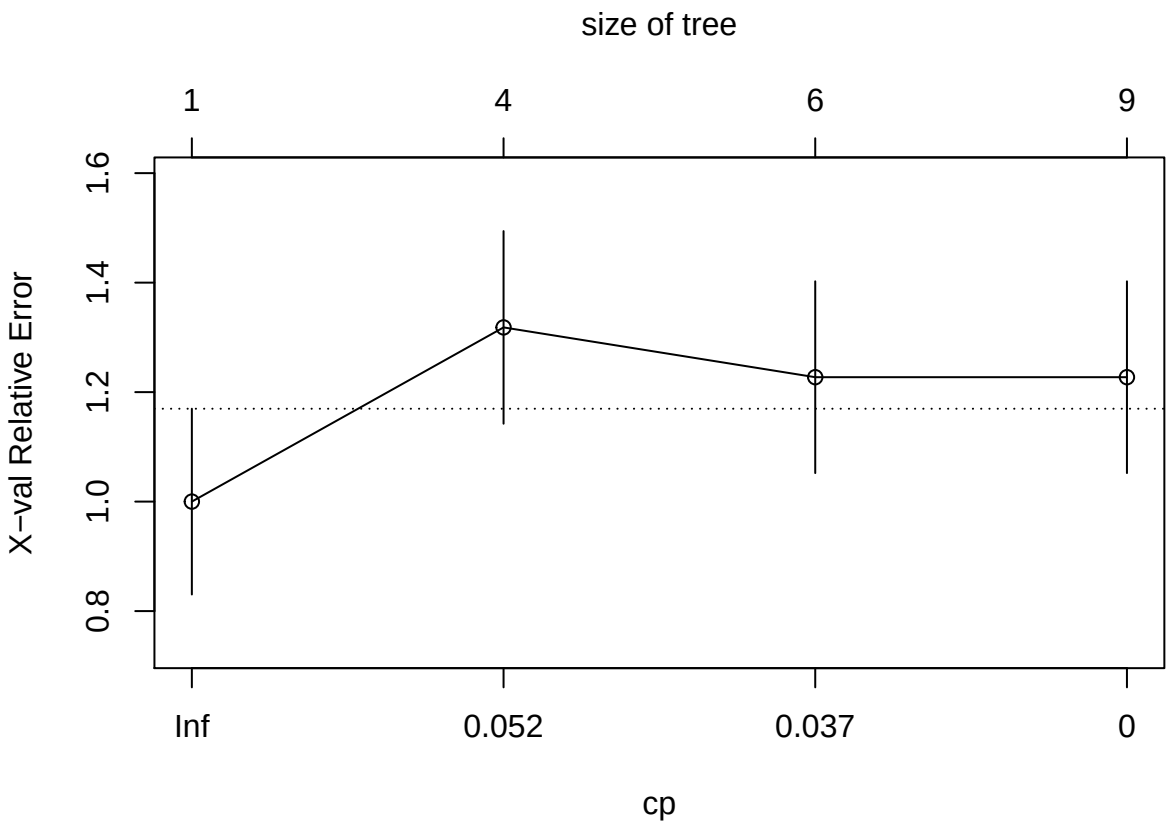
```
rsq.rpart(train_tree4NoJay)
```

```
##
## Classification tree:
## rpart(formula = Author ~ ., data = train2NoJay, method = "class",
##     control = rpart.control(cp = 0, minsplit = 5, maxdepth = 5))
##
## Variables actually used in tree construction:
## [1] compos  concern conclus
##
## Root node error: 22/60 = 0.36667
##
## n= 60
##
##         CP nsplit rel error xerror    xstd
## 1 0.060606      0   1.00000 1.0000 0.16967
## 2 0.045455      3   0.81818 1.3182 0.17595
## 3 0.030303      5   0.72727 1.2273 0.17516
## 4 0.000000      8   0.63636 1.2273 0.17516

## Warning in rsq.rpart(train_tree4NoJay): may not be applicable for this method
```

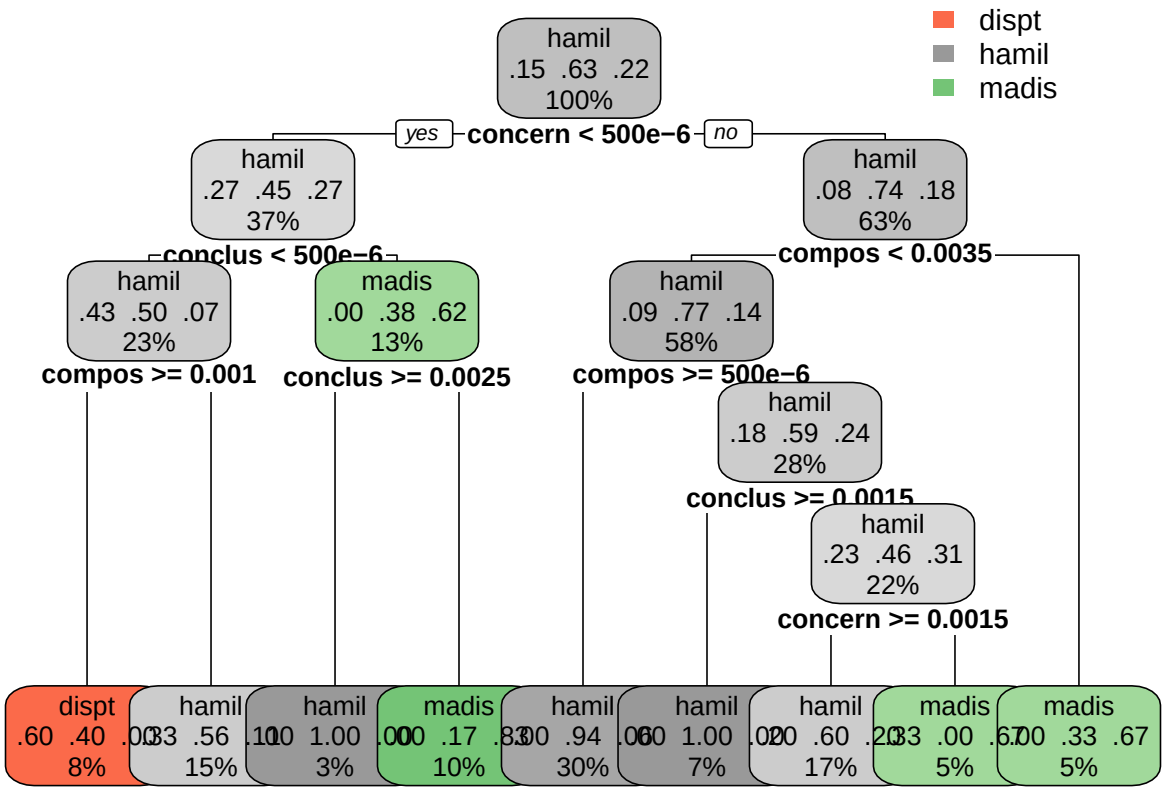

Apparent
X Relative
R−square
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
Number of Splits




```
plotcp(train_tree4NoJay)
```




```
rpart.plot(train_tree4NoJay,cex = .8)
```



#confusion matrix to find correct and incorrect predictions

```
table(Authorship=predicted4NoJay, true=test2NoJay$Author)
```

```
##           true
## Authorship dispt hamil madis
##      dispt     0     2     1
##      hamil     2    11     1
##      madis     0     3     0
```

```
(Results4NoJay<-data.frame(Predicted=predicted4NoJay, Actual=test2NoJay$Author))
```

```
##    Predicted Actual
## 1      hamil  dispt
## 2      hamil  dispt
## 3      hamil  hamil
## 4      hamil  hamil
## 5      madis  hamil
## 6      hamil  hamil
## 7      hamil  hamil
## 8      hamil  hamil
## 9      hamil  hamil
## 10     hamil  hamil
## 11     hamil  hamil
## 12     hamil  hamil
```

```
## 13     madis  hamil
## 14     dispt  hamil
## 15     hamil  hamil
## 16     madis  hamil
## 17     dispt  hamil
## 18     hamil  hamil
## 19     hamil  madis
## 20     dispt  madis
```

#Train Tree 5

```
train_tree5NoJay <- rpart(Author ~ ., data = train2NoJay, method="class", control=rpart.control(cp=0, m
summary(train_tree5NoJay)
```

```
## Call:
## rpart(formula = Author ~ ., data = train2NoJay, method = "class",
##     control = rpart.control(cp = 0, minsplit = 5, maxdepth = 7))
##   n= 60
##
##           CP nsplit rel error   xerror      xstd
## 1 0.06060606      0 1.0000000 1.000000 0.1696699
## 2 0.04545455      3 0.8181818 1.318182 0.1759469
## 3 0.03030303      5 0.7272727 1.272727 0.1756531
## 4 0.00000000      8 0.6363636 1.318182 0.1759469
##
## Variable importance
## conclus  compos concern
##      47      26      26
##
## Node number 1: 60 observations,    complexity param=0.06060606
##   predicted class=hamil  expected loss=0.3666667  P(node) =1
##     class counts:     9    38    13
##    probabilities: 0.150 0.633 0.217
##   left son=2 (22 obs) right son=3 (38 obs)
##   Primary splits:
##       concern < 5e-04  to the left,  improve=1.7427430, (0 missing)
##       conclus < 0.0025 to the right, improve=1.1121210, (0 missing)
##       compos  < 5e-04  to the left,  improve=0.6943641, (0 missing)
##
## Node number 2: 22 observations,    complexity param=0.06060606
##   predicted class=hamil  expected loss=0.5454545  P(node) =0.3666667
##     class counts:     6    10     6
##    probabilities: 0.273 0.455 0.273
##   left son=4 (14 obs) right son=5 (8 obs)
##   Primary splits:
##       conclus < 5e-04  to the left,  improve=2.5746750, (0 missing)
##       compos  < 0.0015 to the right, improve=0.4294372, (0 missing)
##
## Node number 3: 38 observations,    complexity param=0.04545455
##   predicted class=hamil  expected loss=0.2631579  P(node) =0.6333333
##     class counts:     3    28     7
##    probabilities: 0.079 0.737 0.184
##   left son=6 (35 obs) right son=7 (3 obs)
```

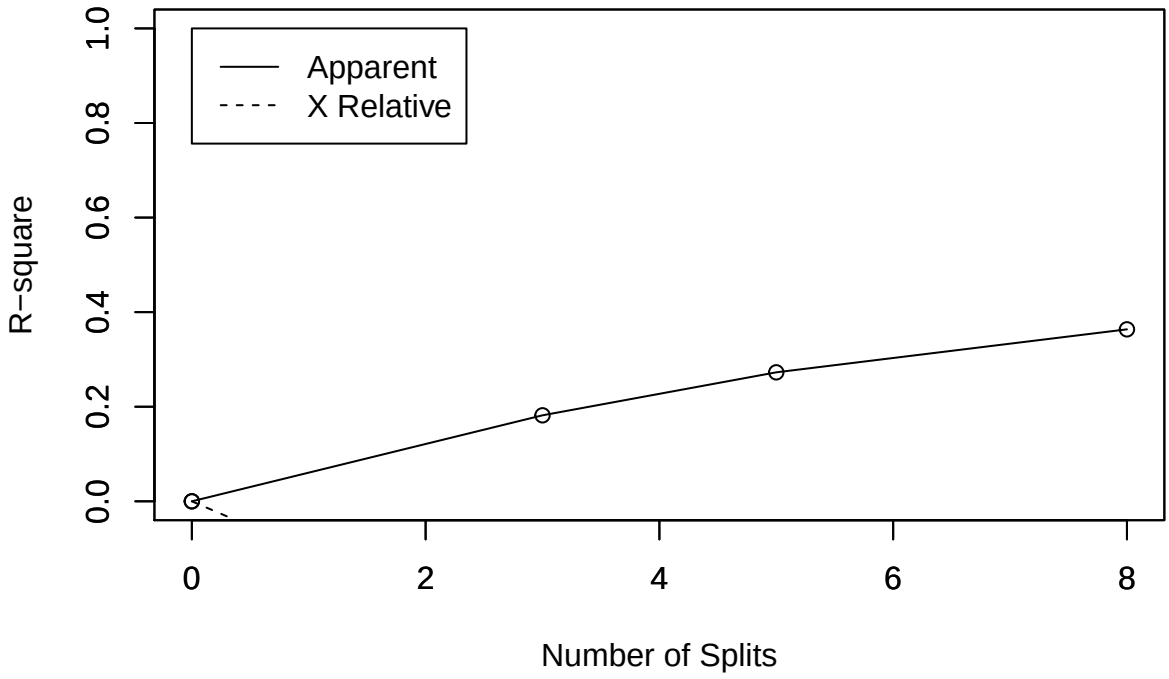
```
##   Primary splits:
##       compos  < 0.0035 to the left,  improve=1.308772, (0 missing)
##       concern < 0.0055 to the left,  improve=1.308772, (0 missing)
##       conclus < 5e-04  to the right, improve=0.604010, (0 missing)
##
## Node number 4: 14 observations,    complexity param=0.04545455
##   predicted class=hamil  expected loss=0.5  P(node) =0.2333333
##     class counts:     6     7     1
##    probabilities: 0.429 0.500 0.071
##   left son=8 (5 obs) right son=9 (9 obs)
##   Primary splits:
##       compos < 0.001  to the right, improve=0.3460317, (0 missing)
##
## Node number 5: 8 observations,    complexity param=0.06060606
##   predicted class=madis  expected loss=0.375  P(node) =0.1333333
##     class counts:     0     3     5
##    probabilities: 0.000 0.375 0.625
##   left son=10 (2 obs) right son=11 (6 obs)
##   Primary splits:
##       conclus < 0.0025 to the right, improve=2.08333300, (0 missing)
##       compos  < 0.0015 to the right, improve=0.08333333, (0 missing)
##
## Node number 6: 35 observations,    complexity param=0.03030303
##   predicted class=hamil  expected loss=0.2285714  P(node) =0.5833333
##     class counts:     3    27     5
##    probabilities: 0.086 0.771 0.143
##   left son=12 (18 obs) right son=13 (17 obs)
##   Primary splits:
##       compos  < 5e-04  to the right, improve=1.6640520, (0 missing)
##       conclus < 0.0015 to the right, improve=0.9692308, (0 missing)
##       concern < 0.0055 to the left,  improve=0.4424242, (0 missing)
##   Surrogate splits:
##       concern < 0.0045 to the right, agree=0.571, adj=0.118, (0 split)
##
## Node number 7: 3 observations
##   predicted class=madis  expected loss=0.3333333  P(node) =0.05
##     class counts:     0     1     2
##    probabilities: 0.000 0.333 0.667
##
## Node number 8: 5 observations
##   predicted class=dispt  expected loss=0.4  P(node) =0.08333333
##     class counts:     3     2     0
##    probabilities: 0.600 0.400 0.000
##
## Node number 9: 9 observations
##   predicted class=hamil  expected loss=0.4444444  P(node) =0.15
##     class counts:     3     5     1
##    probabilities: 0.333 0.556 0.111
##
## Node number 10: 2 observations
##   predicted class=hamil  expected loss=0  P(node) =0.03333333
##     class counts:     0     2     0
##    probabilities: 0.000 1.000 0.000
##
```

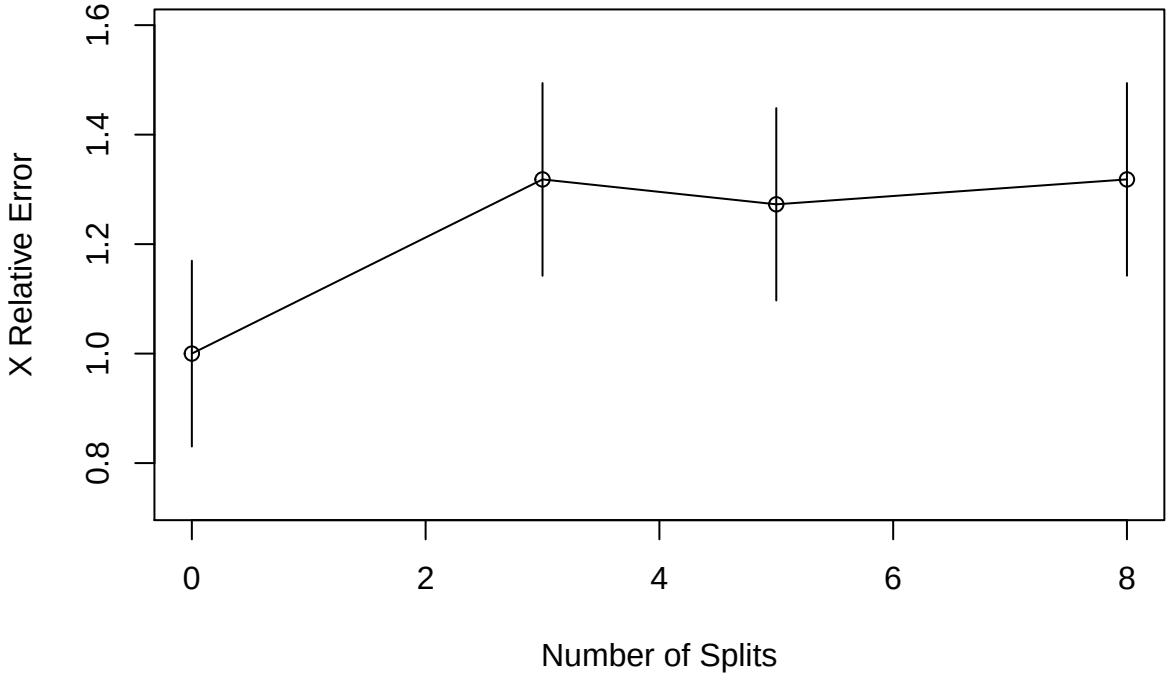
```
## Node number 11: 6 observations
##   predicted class=madis  expected loss=0.1666667  P(node) =0.1
##     class counts:     0     1     5
##    probabilities: 0.000 0.167 0.833
##
## Node number 12: 18 observations
##   predicted class=hamil  expected loss=0.05555556  P(node) =0.3
##     class counts:     0    17     1
##    probabilities: 0.000 0.944 0.056
##
## Node number 13: 17 observations,    complexity param=0.03030303
##   predicted class=hamil  expected loss=0.4117647  P(node) =0.2833333
##     class counts:     3    10     4
##    probabilities: 0.176 0.588 0.235
##   left son=26 (4 obs) right son=27 (13 obs)
##   Primary splits:
##       conclus < 0.0015 to the right, improve=1.339367, (0 missing)
##       concern < 0.0035 to the right, improve=1.071301, (0 missing)
##
## Node number 26: 4 observations
##   predicted class=hamil  expected loss=0  P(node) =0.06666667
##     class counts:     0     4     0
##    probabilities: 0.000 1.000 0.000
##
## Node number 27: 13 observations,    complexity param=0.03030303
##   predicted class=hamil  expected loss=0.5384615  P(node) =0.2166667
##     class counts:     3     6     4
##    probabilities: 0.231 0.462 0.308
##   left son=54 (10 obs) right son=55 (3 obs)
##   Primary splits:
##       concern < 0.0015 to the right, improve=1.3743590, (0 missing)
##       conclus < 5e-04  to the right, improve=0.3986014, (0 missing)
##
## Node number 54: 10 observations
##   predicted class=hamil  expected loss=0.4  P(node) =0.1666667
##     class counts:     2     6     2
##    probabilities: 0.200 0.600 0.200
##
## Node number 55: 3 observations
##   predicted class=madis  expected loss=0.3333333  P(node) =0.05
##     class counts:     1     0     2
##    probabilities: 0.333 0.000 0.667
```

```
predicted5NoJay= predict(train_tree5NoJay, test2NoJay, type="class")
rsq.rpart(train_tree5NoJay)
```

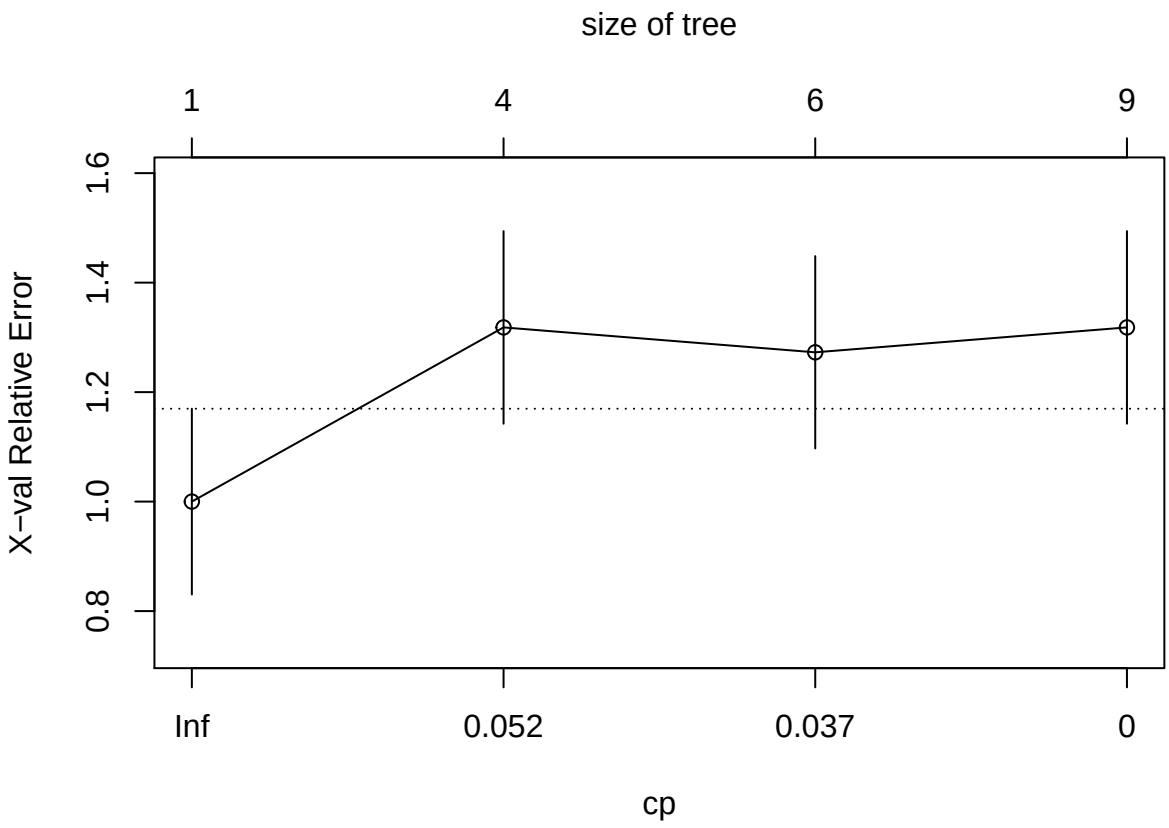
```
##
## Classification tree:
## rpart(formula = Author ~ ., data = train2NoJay, method = "class",
##     control = rpart.control(cp = 0, minsplit = 5, maxdepth = 7))
##
## Variables actually used in tree construction:
## [1] compos  concern conclus
##
```

```
## Root node error: 22/60 = 0.36667
##
## n= 60
##
##         CP nsplit rel error xerror    xstd
## 1 0.060606      0   1.00000 1.0000 0.16967
## 2 0.045455      3   0.81818 1.3182 0.17595
## 3 0.030303      5   0.72727 1.2727 0.17565
## 4 0.000000      8   0.63636 1.3182 0.17595

## Warning in rsq.rpart(train_tree5NoJay): may not be applicable for this method
```

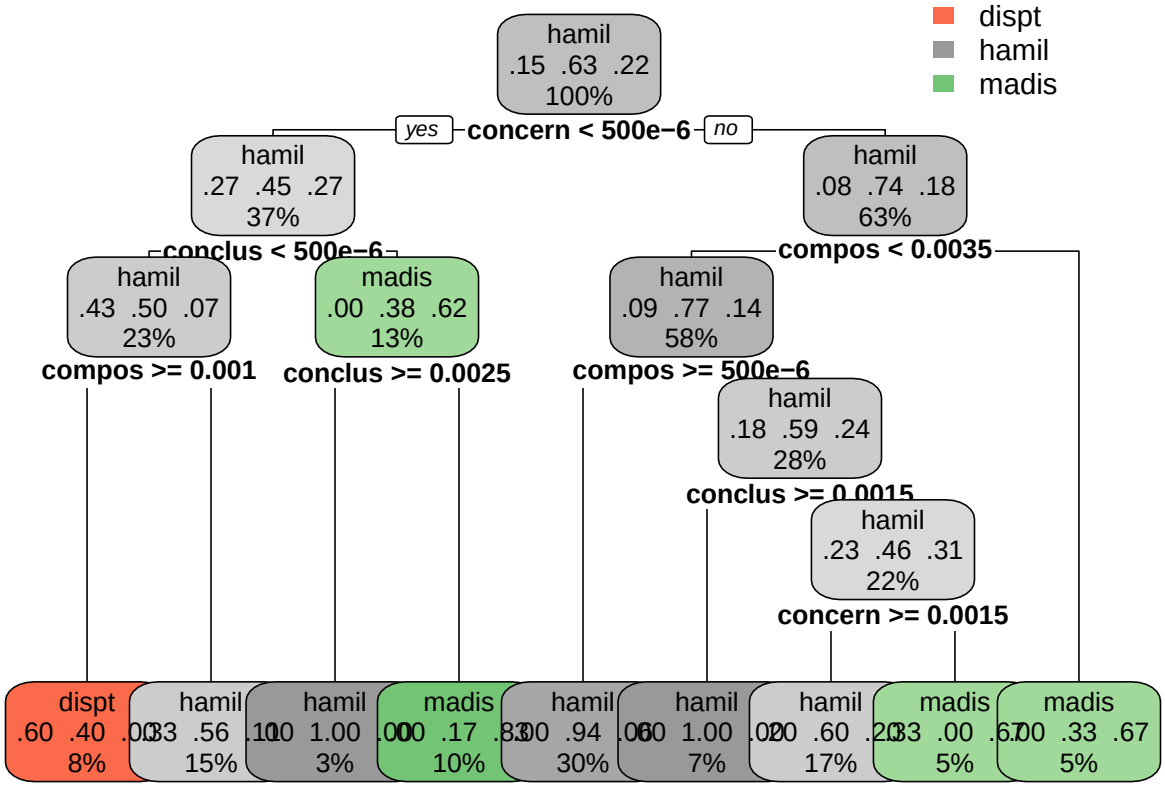





```
plotcp(train_tree5NoJay)
```




```
rpart.plot(train_tree5NoJay, cex = .8)
```




```
table(Authorship=predicted5NoJay, true = test2NoJay$Author)
```

```
##           true
## Authorship dispt hamil madis
##      dispt     0     2     1
##      hamil     2    11     1
##      madis     0     3     0
```

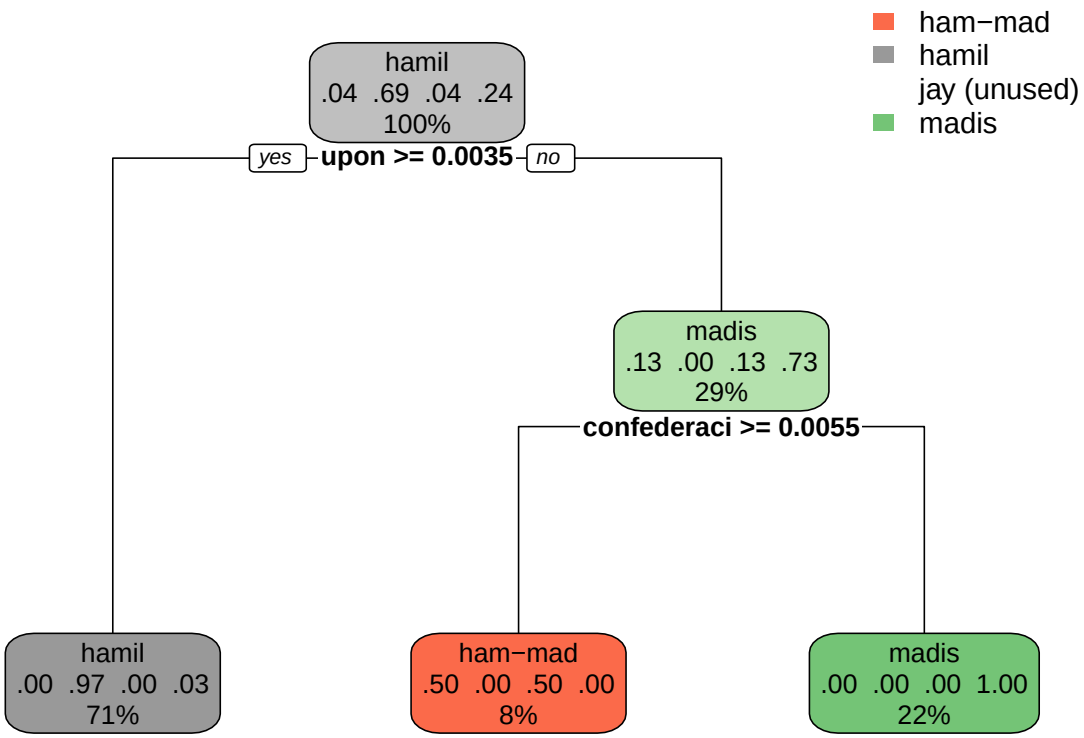
All results

```
(ResultsAll <- data.frame(Actual=testNoHM$Author, Predicted4NoHM=predicted4NoHM, Predicted5NoHM=predicte
                          Predicted4NoJay=predicted4NoJay, Predicted3NoJay=predicted3NoJay, Predicted4N
                          Predicted5NoHM=predicted5NoHM ))
```

```
##    Actual Predicted4NoHM Predicted5NoHM Predicted4NoJay Predicted3NoJay
## 1   dispt          madis          madis           hamil           hamil
## 2   dispt          madis          madis           hamil           madis
## 3   hamil          hamil          hamil           hamil           hamil
## 4   hamil          hamil          hamil           hamil           hamil
## 5   hamil          hamil          hamil           madis           hamil
## 6   hamil          hamil          hamil           hamil           hamil
## 7   hamil          madis          madis           hamil           hamil
```

```
## 8   hamil          hamil          hamil          hamil          hamil
## 9   hamil          hamil          hamil          hamil          hamil
## 10  hamil          hamil          hamil          hamil          hamil
## 11  hamil          hamil          hamil          hamil          hamil
## 12  hamil          madis          madis          hamil          madis
## 13  hamil          hamil          hamil          madis          hamil
## 14  hamil          hamil          hamil          dispt          hamil
## 15  hamil          hamil          hamil          hamil          hamil
## 16  hamil          hamil          hamil          madis          madis
## 17  hamil          hamil          hamil          dispt          hamil
## 18  madis          madis          madis          hamil          madis
## 19  madis          madis          madis          hamil          hamil
## 20  madis          madis          madis          dispt          hamil
##    Predicted4NoJay.1 Predicted5NoHM.1
## 1              hamil            madis
## 2              hamil            madis
## 3              hamil            hamil
## 4              hamil            hamil
## 5              madis            hamil
## 6              hamil            hamil
## 7              hamil            madis
## 8              hamil            hamil
## 9              hamil            hamil
## 10             hamil            hamil
## 11             hamil            hamil
## 12             hamil            madis
## 13             madis            hamil
## 14             dispt            hamil
## 15             hamil            hamil
## 16             madis            hamil
## 17             dispt            hamil
## 18             hamil            madis
## 19             hamil            madis
## 20             dispt            madis
```

```
feds.tree3 <- rpart(Author ~ . , data = train, method = 'class', control = rpart.control(cp = 0, minspli

rpart.plot(feds.tree3, cex = 0.8)
```



########################## #### Saved

```
FedPapersCorpus <- Corpus(DirSource("FedPapersCorpus"))
(numberFedPapers<-length(FedPapersCorpus))
```

Load Fed Papers Corpus

```
## [1] 85
```

#(summary(FedPapersCorpus))

```
(meta(FedPapersCorpus[[1]]))
```

```
##   author       : character(0)
##   datetimestamp: 2021-08-08 21:47:44
##   description  : character(0)
##   heading      : character(0)
##   id           : dispt_fed_49.txt
##   language     : en
##   origin       : character(0)
```

```
(meta(FedPapersCorpus[[1]],5))
```

```
## [1] "dispt_fed_49.txt"
```

```
## Cleaning and Preparing
#Choosing some good stop words can really go a long way to improve modeling results. There are also many
#other parameters one can tweak and tune using the DocumentTermMatrix function. See many below.
#Data Preparation and Transformation on Fed Papers

##Remove punctuation,numbers, and space
(getTransformations())
```

```
## [1] "removeNumbers"     "removePunctuation" "removeWords"
## [4] "stemDocument"      "stripWhitespace"
```

```
(nFedPapersCorpus<-length(FedPapersCorpus))
```

```
## [1] 85
```

```
(minTermFreq <-30)
```

```
## [1] 30
```

```
(maxTermFreq <-1000)
```

```
## [1] 1000
```

```
#  Create a personalized list of stop words
MyStopwords <- c("will","one","two", "may","less","publius","Madison","Alexand", "Alexander", "James",
                 "without", "small", "single" ,"several", "but", "very", "can", "must", "also", "any",
                 "almost", "for", "add", "Author")

STOPS <-stopwords('english')

Papers_DTM <- DocumentTermMatrix(FedPapersCorpus,
                                 control = list(stopwords = TRUE, wordLengths=c(3, 15),
                                                removePunctuation = T, removeNumbers = T,
                                                tolower=T, stemming = T,
                                                remove_separators = T,
                                                stopwords = MyStopwords,
                                                removeWords= c(STOPS,MyStopwords),
                                                removeWords=MyStopwords,
                                                bounds = list(global = c(minTermFreq, maxTermFreq))
                                 ))
```

##inspect FedPapers Document Term Matrix (DTM)

```
DTM <- as.matrix(Papers_DTM)
```

Confirming 1st 11 are disputed

```
(DTM[1:11,1:10])
```

```
##                   Terms
## Docs                abl absolut accord act addit administr admit adopt advantag
##   dispt_fed_49.txt    2       0      0   0     0         1     1     0        4
##   dispt_fed_50.txt    0       2      0   0     0         2     0     0        1
##   dispt_fed_51.txt    1       2      0   0     1         1     3     0        0
##   dispt_fed_52.txt    1       1      0   1     1         0     0     1        2
##   dispt_fed_53.txt    0       0      1   2     0         0     1     0        2
##   dispt_fed_54.txt    0       0      2   1     0         0     5     1        4
##   dispt_fed_55.txt    0       0      2   0     0         0     2     0        0
##   dispt_fed_56.txt    0       0      1   1     0         0     0     0        1
##   dispt_fed_57.txt    0       0      1   0     1         1     1     0        0
##   dispt_fed_62.txt    1       0      0   1     1         0     0     1        7
##   dispt_fed_63.txt    4       0      1   3     1         1     1     0        5
##                   Terms
## Docs                affair
##   dispt_fed_49.txt      0
##   dispt_fed_50.txt      0
##   dispt_fed_51.txt      1
##   dispt_fed_52.txt      0
##   dispt_fed_53.txt      9
##   dispt_fed_54.txt      0
##   dispt_fed_55.txt      1
##   dispt_fed_56.txt      5
##   dispt_fed_57.txt      0
##   dispt_fed_62.txt      4
##   dispt_fed_63.txt      1
```

Vectorization

##Vectorizing words is often done by encoding frequency information. Below we take a peak at the frequency # of the words. Next some normalization techniques are tried. Which works best . . . ?? Try many and assess # the results!!! ##Look at word frequencies

```
WordFreq <- colSums(as.matrix(Papers_DTM))
```

```
(head(WordFreq, 20))
```

```
##       abl   absolut    accord       act     addit administr     admit     adopt
##        74        63        71       139        61        90       107        57
##  advantag    affair    affect    afford   alexand    almost      alon   alreadi
##       142        65        56        64        67        45        70        56
##      also     alway   america     among
##        96        84       114       131
```

```
(length(WordFreq))
```

```
## [1] 427
```

```
ord <- order(WordFreq)
(WordFreq[head(ord, 20)])
```

```
##      jame      expos    furnish       word     unless      bound   descript      drawn
##        30         34         36         36         37         38         38         38
##      leav     design      fulli   tendenc    applic apprehens      avoid   portion
##        38         39         39         39         40         40         40         40
##    preced   foundat    extrem       fall
##        40         41         42         42
```

```
(WordFreq[tail(ord)])
```

```
## constitut       may     power    govern      will     state
##       686       811       937      1040      1263      1662
```

Row Sums per Fed Papers

```
Row_Sum_Per_doc <- rowSums((as.matrix(Papers_DTM)))
```

Create a normalized version of Papers_DTM

```
Papers_M <- as.matrix(Papers_DTM)
Papers_M_N1 <- apply(Papers_M, 1, function(i) round(i/sum(i),3))
Papers_Matrix_Norm <- t(Papers_M_N1)
```

Convert to matrix and view

```
Papers_dtm_matrix = as.matrix(Papers_DTM)
```

#str(Papers_dtm_matrix) #(Papers_dtm_matrix[c(1:11),c(2:10)])

Label the Data

Below we label the data, prepare for modeling, and create some wordclouds for fun.

Also convert to DF

```
Papers_DF <- as.data.frame(as.matrix(Papers_Matrix_Norm))
Papers_DF1<- Papers_DF%>%add_rownames()

names(Papers_DF1)[1]<-"Author"
Papers_DF1[1:11,1] = "dispt"
Papers_DF1[12:62,1] = "hamil"
Papers_DF1[63:65,1] = "ham-mad"
Papers_DF1[66:70,1] = "jay"
Papers_DF1[71:85,1 ]="madis"


head(Papers_DF1, 15)
```

```
## # A tibble: 15 x 428
##    Author   abl absolut accord   act addit administr admit adopt advantag affair
##    <chr>  <dbl>   <dbl>  <dbl> <dbl> <dbl>     <dbl> <dbl> <dbl>    <dbl>  <dbl>
##  1 dispt  0.004   0      0     0     0         0.002 0.002 0        0.008  0
##  2 dispt  0       0.006  0     0     0         0.006 0     0        0.003  0
##  3 dispt  0.002   0.003  0     0     0.002     0.002 0.005 0        0      0.002
##  4 dispt  0.002   0.002  0     0.002 0.002     0     0     0.002    0.004  0
##  5 dispt  0       0      0.001 0.003 0         0     0.001 0        0.003  0.013
##  6 dispt  0       0      0.003 0.002 0         0     0.009 0.002    0.007  0
##  7 dispt  0       0      0.003 0     0         0     0.003 0        0      0.002
##  8 dispt  0       0      0.002 0.002 0         0     0     0        0.002  0.009
##  9 dispt  0       0      0.002 0     0.002     0.002 0.002 0        0      0
## 10 dispt  0.001   0      0     0.001 0.001     0     0     0.001    0.01   0.006
## 11 dispt  0.004   0      0.001 0.003 0.001     0.001 0.001 0        0.005  0.001
## 12 hamil  0.002   0      0.002 0.002 0.002     0     0.002 0.006    0.002  0
## 13 hamil  0.007   0      0     0     0.002     0     0     0        0.009  0.002
## 14 hamil  0.004   0      0.004 0     0.002     0.002 0.004 0        0.002  0
## 15 hamil  0.003   0      0.003 0     0.003     0.003 0.003 0        0.003  0.003
## # ... with 417 more variables: affect <dbl>, afford <dbl>, alexand <dbl>,
## #   almost <dbl>, alon <dbl>, alreadi <dbl>, also <dbl>, alway <dbl>,
## #   america <dbl>, among <dbl>, amount <dbl>, anoth <dbl>, answer <dbl>,
## #   appear <dbl>, appli <dbl>, applic <dbl>, appoint <dbl>, apprehens <dbl>,
## #   argument <dbl>, aris <dbl>, articl <dbl>, assembl <dbl>, attempt <dbl>,
## #   attend <dbl>, attent <dbl>, author <dbl>, avoid <dbl>, becom <dbl>,
## #   best <dbl>, better <dbl>, bodi <dbl>, bound <dbl>, branch <dbl>,
## #   britain <dbl>, calcul <dbl>, call <dbl>, can <dbl>, capac <dbl>,
## #   care <dbl>, carri <dbl>, case <dbl>, caus <dbl>, certain <dbl>,
## #   chang <dbl>, charact <dbl>, circumst <dbl>, citizen <dbl>, civil <dbl>,
## #   class <dbl>, clear <dbl>, collect <dbl>, combin <dbl>, commit <dbl>,
## #   common <dbl>, communiti <dbl>, complet <dbl>, compos <dbl>, concern <dbl>,
## #   conclus <dbl>, conduct <dbl>, confeder <dbl>, confederaci <dbl>,
## #   confid <dbl>, confin <dbl>, congress <dbl>, connect <dbl>, consequ <dbl>,
## #   consid <dbl>, consider <dbl>, consist <dbl>, constitu <dbl>,
## #   constitut <dbl>, contend <dbl>, continu <dbl>, contrari <dbl>,
## #   control <dbl>, convent <dbl>, council <dbl>, countri <dbl>, cours <dbl>,
## #   danger <dbl>, decid <dbl>, decis <dbl>, declar <dbl>, defect <dbl>,
## #   defens <dbl>, degre <dbl>, deliber <dbl>, depart <dbl>, depend <dbl>,
## #   deriv <dbl>, descript <dbl>, design <dbl>, desir <dbl>, determin <dbl>,
## #   differ <dbl>, difficulti <dbl>, direct <dbl>, dispos <dbl>, disposit <dbl>,
## #   ...
```

```
tail(Papers_DF1, 20)
```

```
## # A tibble: 20 x 428
##    Author   abl absolut accord   act addit administr admit adopt advantag affair
##    <chr>  <dbl>   <dbl>  <dbl> <dbl> <dbl>     <dbl> <dbl> <dbl>    <dbl>  <dbl>
##  1 jay    0       0      0     0     0         0     0.002 0.004    0      0
##  2 jay    0.004   0      0.002 0.002 0.002     0.002 0     0.002    0.002  0
##  3 jay    0.002   0.004  0.002 0.002 0         0     0.002 0        0.004  0
##  4 jay    0       0      0     0.002 0         0     0.002 0        0      0.002
##  5 jay    0.007   0.003  0     0.009 0         0     0.001 0        0.006  0.009
##  6 madis  0.002   0      0.003 0.005 0         0.002 0.001 0.001    0.005  0
##  7 madis  0       0.002  0     0.004 0.002     0.004 0     0.002    0.002  0.002
##  8 madis  0.001   0      0.001 0.003 0.001     0.003 0.001 0.001    0.001  0.001
##  9 madis  0.001   0.005  0.003 0.002 0         0.002 0.002 0.001    0      0
## 10 madis  0       0.003  0.009 0.007 0         0     0     0.001    0      0
## 11 madis  0       0.004  0.002 0.013 0         0     0.002 0        0      0
## 12 madis  0.003   0      0     0.001 0.002     0     0.001 0        0.004  0.001
## 13 madis  0       0.001  0.001 0.002 0.001     0.002 0.002 0.001    0      0.002
## 14 madis  0.001   0.002  0     0.001 0         0     0.005 0.002    0.002  0
## 15 madis  0       0.002  0.003 0.004 0.002     0     0.001 0.001    0.001  0
## 16 madis  0       0.003  0.001 0     0.003     0.001 0     0        0.006  0.001
## 17 madis  0.006   0.001  0.002 0.001 0.001     0.004 0.002 0        0.008  0.001
## 18 madis  0       0.001  0.002 0.005 0         0.001 0.003 0.001    0      0
## 19 madis  0       0      0.002 0.005 0         0.004 0     0        0.002  0
## 20 madis  0.002   0.002  0     0.002 0.005     0     0.005 0        0.008  0.003
## # ... with 417 more variables: affect <dbl>, afford <dbl>, alexand <dbl>,
## #   almost <dbl>, alon <dbl>, alreadi <dbl>, also <dbl>, alway <dbl>,
## #   america <dbl>, among <dbl>, amount <dbl>, anoth <dbl>, answer <dbl>,
## #   appear <dbl>, appli <dbl>, applic <dbl>, appoint <dbl>, apprehens <dbl>,
## #   argument <dbl>, aris <dbl>, articl <dbl>, assembl <dbl>, attempt <dbl>,
## #   attend <dbl>, attent <dbl>, author <dbl>, avoid <dbl>, becom <dbl>,
## #   best <dbl>, better <dbl>, bodi <dbl>, bound <dbl>, branch <dbl>,
## #   britain <dbl>, calcul <dbl>, call <dbl>, can <dbl>, capac <dbl>,
## #   care <dbl>, carri <dbl>, case <dbl>, caus <dbl>, certain <dbl>,
## #   chang <dbl>, charact <dbl>, circumst <dbl>, citizen <dbl>, civil <dbl>,
## #   class <dbl>, clear <dbl>, collect <dbl>, combin <dbl>, commit <dbl>,
## #   common <dbl>, communiti <dbl>, complet <dbl>, compos <dbl>, concern <dbl>,
## #   conclus <dbl>, conduct <dbl>, confeder <dbl>, confederaci <dbl>,
## #   confid <dbl>, confin <dbl>, congress <dbl>, connect <dbl>, consequ <dbl>,
## #   consid <dbl>, consider <dbl>, consist <dbl>, constitu <dbl>,
## #   constitut <dbl>, contend <dbl>, continu <dbl>, contrari <dbl>,
## #   control <dbl>, convent <dbl>, council <dbl>, countri <dbl>, cours <dbl>,
## #   danger <dbl>, decid <dbl>, decis <dbl>, declar <dbl>, defect <dbl>,
## #   defens <dbl>, degre <dbl>, deliber <dbl>, depart <dbl>, depend <dbl>,
## #   deriv <dbl>, descript <dbl>, design <dbl>, desir <dbl>, determin <dbl>,
## #   differ <dbl>, difficulti <dbl>, direct <dbl>, dispos <dbl>, disposit <dbl>,
## #   ...
```

```
Papers_DF1[62:71,1]  #  Checking row names
```

```
## # A tibble: 10 x 1
##    Author
```

```
##    <chr>
##  1 hamil
##  2 ham-mad
##  3 ham-mad
##  4 ham-mad
##  5 jay
##  6 jay
##  7 jay
##  8 jay
##  9 jay
## 10 madis
```

Removing both Jay and HM essays

#str(Papers_DF1) #remove Jays papers

```
Papers_DFNoHM<-Papers_DF1[-66:-70,]
```

#str(Papers_DFNoHM)

remove Ham Mad papers

```
Papers_DFNoHM <- Papers_DFNoHM[-63:-65,]
```

#str(Papers_DFNoHM)

```
Papers_DF22 <- Papers_DFNoHM
```

remove disputed papers

```
Papers_DFNoHM <- Papers_DFNoHM[-1:-11,]
```

#str(Papers_DFNoHM)

#head(Papers_DFNoHM, 15) #tail(Papers_DFNoHM, 20) #Papers_DFNoHM[42:61,1]

##Make Train and Test sets # Disputed already removed

```
trainRatio <- .60
set.seed(11) # Set Seed so that same sample can be reproduced in future also

sampleNoHM <- sample.int(n = nrow(Papers_DFNoHM), size = floor(trainRatio*nrow(Papers_DFNoHM)), replace

trainNoHM <- Papers_DFNoHM[sampleNoHM, ]

testNoHM <- Papers_DFNoHM[-sampleNoHM, ]

# train / test ratio
length(sampleNoHM)/nrow(Papers_DFNoHM)
```

```
## [1] 0.5909091
```

```
#        Classification
#        We are now ready to train and test using classifiers. Below we use a few different decision tr
#        different params and prunings to get varied results.
#        Use fancyRpartPlot to visualize the learned tree models. What do these diagrams display???

##Decision Tree Models
#Train Tree Model 1
train_treeNoHM <- rpart(Author ~ ., data = trainNoHM, method="class", control=rpart.control(cp=0))
summary(train_treeNoHM)
```

```
## Call:
## rpart(formula = Author ~ ., data = trainNoHM, method = "class",
##     control = rpart.control(cp = 0))
##   n= 39
##
##   CP nsplit rel error xerror      xstd
## 1  1      0         1      1 0.2923527
## 2  0      1         0      0 0.0000000
##
## Variable importance
##  alexand hamilton     jame     upon     form congress
##       20       20       20       18       11        9
##
## Node number 1: 39 observations,    complexity param=1
##   predicted class=hamil  expected loss=0.2307692  P(node) =1
##     class counts:    30     9
##    probabilities: 0.769 0.231
##   left son=2 (30 obs) right son=3 (9 obs)
##   Primary splits:
##       alexand  < 5e-04  to the right, improve=13.846150, (0 missing)
##       hamilton < 5e-04  to the right, improve=13.846150, (0 missing)
##       jame     < 5e-04  to the left,  improve=13.846150, (0 missing)
##       upon     < 0.003  to the right, improve=11.910670, (0 missing)
##       form     < 0.0065 to the left,  improve= 6.694368, (0 missing)
##   Surrogate splits:
##       hamilton < 5e-04  to the right, agree=1.000, adj=1.000, (0 split)
##       jame     < 5e-04  to the left,  agree=1.000, adj=1.000, (0 split)
##       upon     < 0.003  to the right, agree=0.974, adj=0.889, (0 split)
##       form     < 0.0065 to the left,  agree=0.897, adj=0.556, (0 split)
##       congress < 0.0035 to the left,  agree=0.872, adj=0.444, (0 split)
##
## Node number 2: 30 observations
##   predicted class=hamil  expected loss=0  P(node) =0.7692308
##     class counts:    30     0
##    probabilities: 1.000 0.000
##
## Node number 3: 9 observations
##   predicted class=madis  expected loss=0  P(node) =0.2307692
##     class counts:     0     9
##    probabilities: 0.000 1.000
```

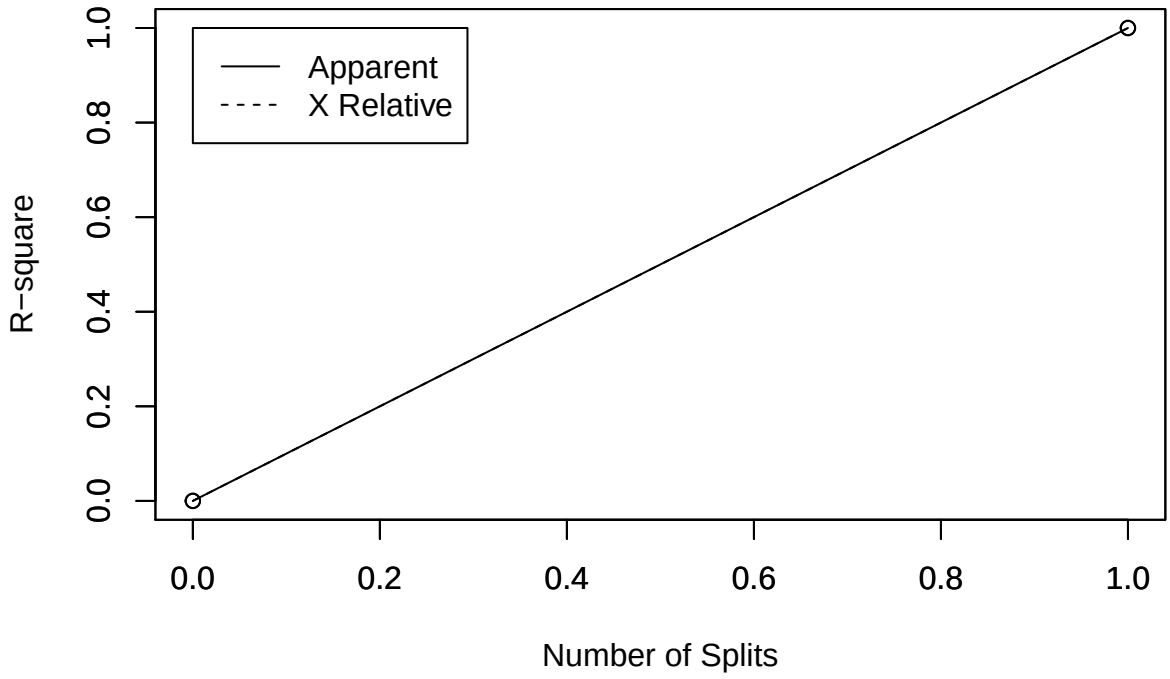
#predict the test dataset using the model for train tree No. 1

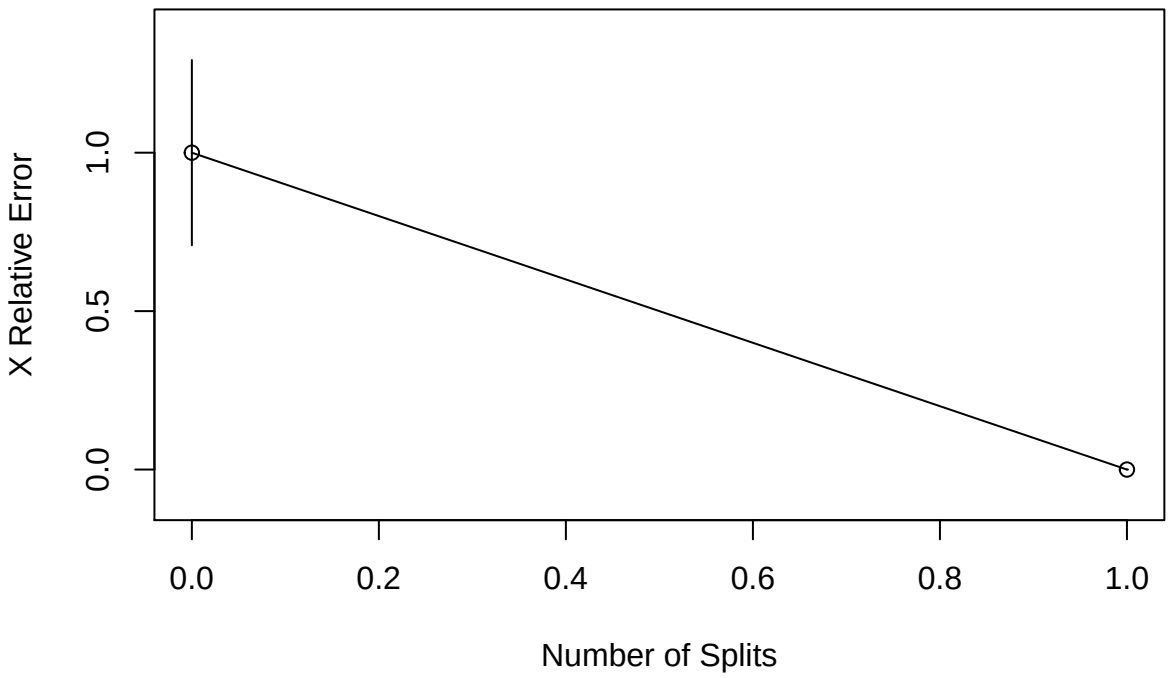
```
predicted1= predict(train_treeNoHM, testNoHM, type="class")
```

#plot number of splits

```
rsq.rpart(train_treeNoHM)
```

```
##
## Classification tree:
## rpart(formula = Author ~ ., data = trainNoHM, method = "class",
##     control = rpart.control(cp = 0))
##
## Variables actually used in tree construction:
## [1] alexand
##
## Root node error: 9/39 = 0.23077
##
## n= 39
##
##   CP nsplit rel error xerror    xstd
## 1  1      0         1      1 0.29235
## 2  0      1         0      0 0.00000

## Warning in rsq.rpart(train_treeNoHM): may not be applicable for this method
```





Classification tree:

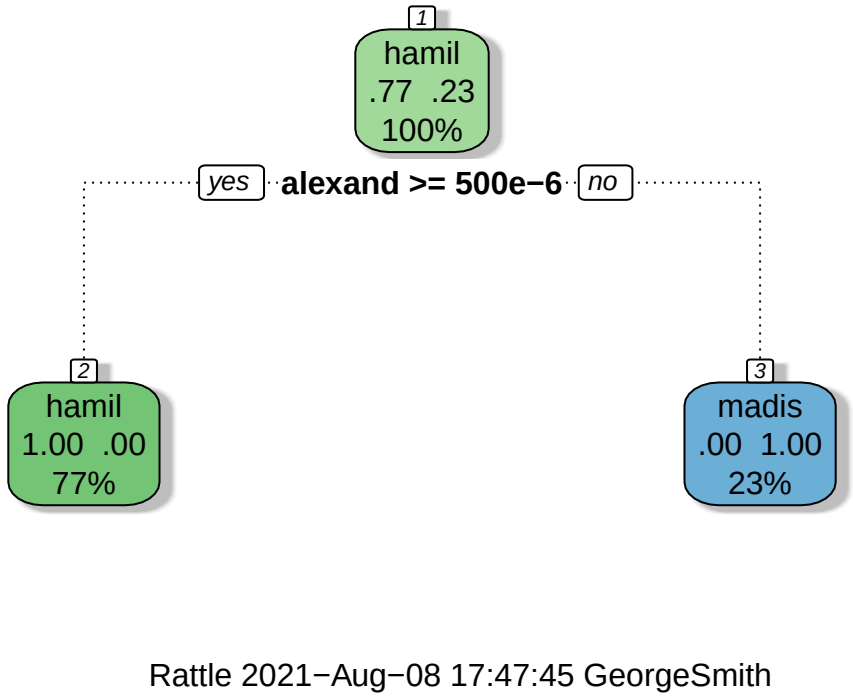
```
classTree <- rpart(formula = Author ~ ., data = trainNoHM, method = "class", control = rpart.control(cp
summary(classTree)
```

```
## Call:
## rpart(formula = Author ~ ., data = trainNoHM, method = "class",
##     control = rpart.control(cp = 0))
##   n= 39
##
##   CP nsplit rel error xerror      xstd
## 1  1      0         1      1 0.2923527
## 2  0      1         0      0 0.0000000
##
## Variable importance
##  alexand hamilton     jame     upon     form congress
##       20       20       20       18       11        9
##
## Node number 1: 39 observations,    complexity param=1
##   predicted class=hamil  expected loss=0.2307692  P(node) =1
##     class counts:    30     9
##    probabilities: 0.769 0.231
##   left son=2 (30 obs) right son=3 (9 obs)
##   Primary splits:
##       alexand  < 5e-04  to the right, improve=13.846150, (0 missing)
##       hamilton < 5e-04  to the right, improve=13.846150, (0 missing)
```

```
##       jame     < 5e-04  to the left,  improve=13.846150, (0 missing)
##       upon     < 0.003  to the right, improve=11.910670, (0 missing)
##       form     < 0.0065 to the left,  improve= 6.694368, (0 missing)
##   Surrogate splits:
##       hamilton < 5e-04  to the right, agree=1.000, adj=1.000, (0 split)
##       jame     < 5e-04  to the left,  agree=1.000, adj=1.000, (0 split)
##       upon     < 0.003  to the right, agree=0.974, adj=0.889, (0 split)
##       form     < 0.0065 to the left,  agree=0.897, adj=0.556, (0 split)
##       congress < 0.0035 to the left,  agree=0.872, adj=0.444, (0 split)
##
## Node number 2: 30 observations
##   predicted class=hamil  expected loss=0  P(node) =0.7692308
##     class counts:    30     0
##    probabilities: 1.000 0.000
##
## Node number 3: 9 observations
##   predicted class=madis  expected loss=0  P(node) =0.2307692
##     class counts:     0     9
##    probabilities: 0.000 1.000
```

#plot the decision tree

```
fancyRpartPlot(train_treeNoHM)
```



#confusion matrix to find correct and incorrect predictions

```
table(Authorship=predicted1, true=testNoHM$Author)
```

```
##           true
## Authorship hamil madis
##      hamil    21     0
##      madis     0     6
```

```
train_treeNoHM2 <- rpart(Author ~ ., data = trainNoHM, method="class", control=rpart.control(cp=0), mins
(summary(train_treeNoHM2))
```

```
## Call:
## rpart(formula = Author ~ ., data = trainNoHM, method = "class",
##     control = rpart.control(cp = 0), minsplit = 2, maxdepth = 5)
##   n= 39
##
##   CP nsplit rel error xerror      xstd
## 1  1      0         1      1 0.2923527
## 2  0      1         0      0 0.0000000
##
## Variable importance
##  alexand hamilton     jame     upon     form congress
##       20       20       20       18       11        9
##
## Node number 1: 39 observations,    complexity param=1
##   predicted class=hamil  expected loss=0.2307692  P(node) =1
##     class counts:    30     9
##    probabilities: 0.769 0.231
##   left son=2 (30 obs) right son=3 (9 obs)
##   Primary splits:
##       alexand  < 5e-04  to the right, improve=13.846150, (0 missing)
##       hamilton < 5e-04  to the right, improve=13.846150, (0 missing)
##       jame     < 5e-04  to the left,  improve=13.846150, (0 missing)
##       upon     < 0.003  to the right, improve=11.910670, (0 missing)
##       form     < 0.0065 to the left,  improve= 6.694368, (0 missing)
##   Surrogate splits:
##       hamilton < 5e-04  to the right, agree=1.000, adj=1.000, (0 split)
##       jame     < 5e-04  to the left,  agree=1.000, adj=1.000, (0 split)
##       upon     < 0.003  to the right, agree=0.974, adj=0.889, (0 split)
##       form     < 0.0065 to the left,  agree=0.897, adj=0.556, (0 split)
##       congress < 0.0035 to the left,  agree=0.872, adj=0.444, (0 split)
##
## Node number 2: 30 observations
##   predicted class=hamil  expected loss=0  P(node) =0.7692308
##     class counts:    30     0
##    probabilities: 1.000 0.000
##
## Node number 3: 9 observations
##   predicted class=madis  expected loss=0  P(node) =0.2307692
##     class counts:     0     9
##    probabilities: 0.000 1.000

## n= 39
```

```
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 39 9 hamil (0.7692308 0.2307692)
##   2) alexand>=0.0005 30 0 hamil (1.0000000 0.0000000) *
##   3) alexand< 0.0005 9 0 madis (0.0000000 1.0000000) *
```

#predict the test dataset using the model for train tree No. 1

```
predicted2= predict(train_treeNoHM2, testNoHM, type="class")
(ResultsP2Disp <- data.frame(Predicted=predicted2,Actual=testNoHM$Author))
```

```
##    Predicted Actual
## 1      hamil  hamil
## 2      hamil  hamil
## 3      hamil  hamil
## 4      hamil  hamil
## 5      hamil  hamil
## 6      hamil  hamil
## 7      hamil  hamil
## 8      hamil  hamil
## 9      hamil  hamil
## 10     hamil  hamil
## 11     hamil  hamil
## 12     hamil  hamil
## 13     hamil  hamil
## 14     hamil  hamil
## 15     hamil  hamil
## 16     hamil  hamil
## 17     hamil  hamil
## 18     hamil  hamil
## 19     hamil  hamil
## 20     hamil  hamil
## 21     hamil  hamil
## 22     madis  madis
## 23     madis  madis
## 24     madis  madis
## 25     madis  madis
## 26     madis  madis
## 27     madis  madis
```

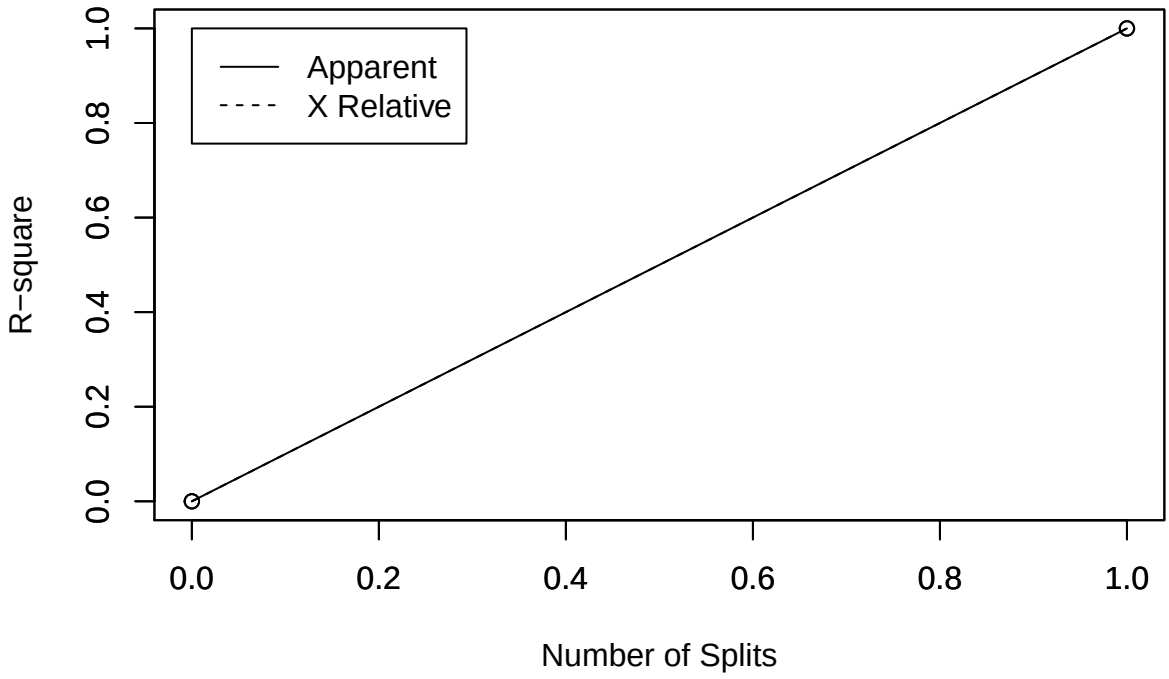
#plot number of splits

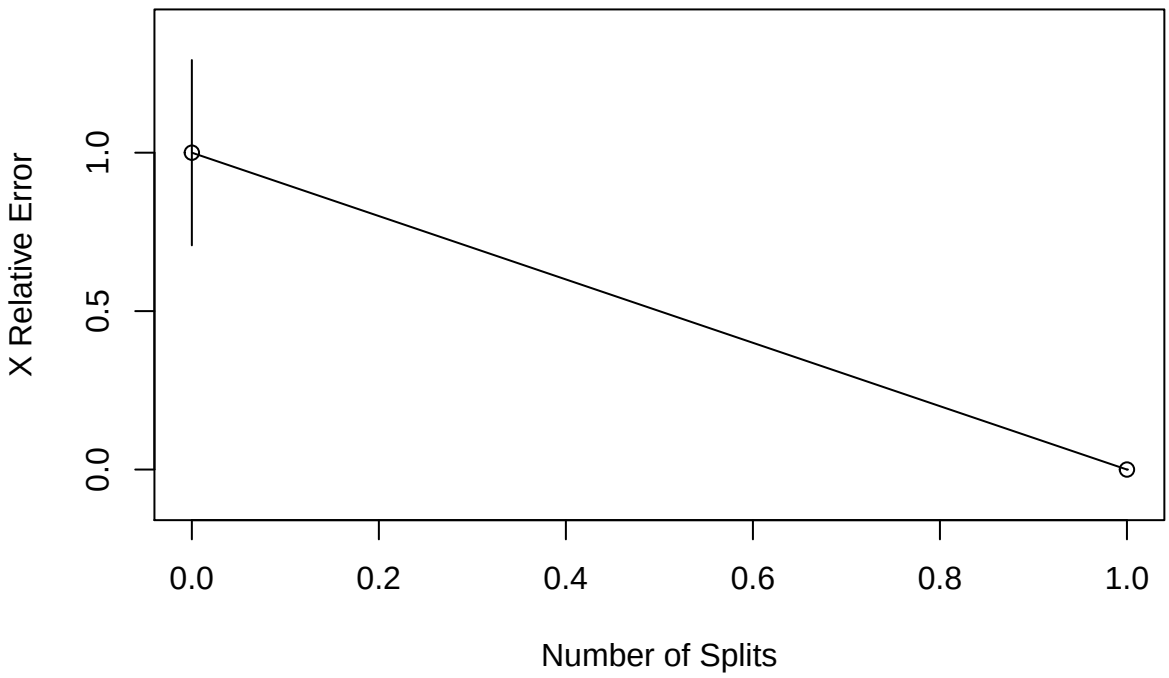
```
rsq.rpart(train_treeNoHM2)
```

```
##
## Classification tree:
## rpart(formula = Author ~ ., data = trainNoHM, method = "class",
##     control = rpart.control(cp = 0), minsplit = 2, maxdepth = 5)
##
## Variables actually used in tree construction:
```

```
## [1] alexand
##
## Root node error: 9/39 = 0.23077
##
## n= 39
##
##   CP nsplit rel error xerror    xstd
## 1  1      0         1      1 0.29235
## 2  0      1         0      0 0.00000
```

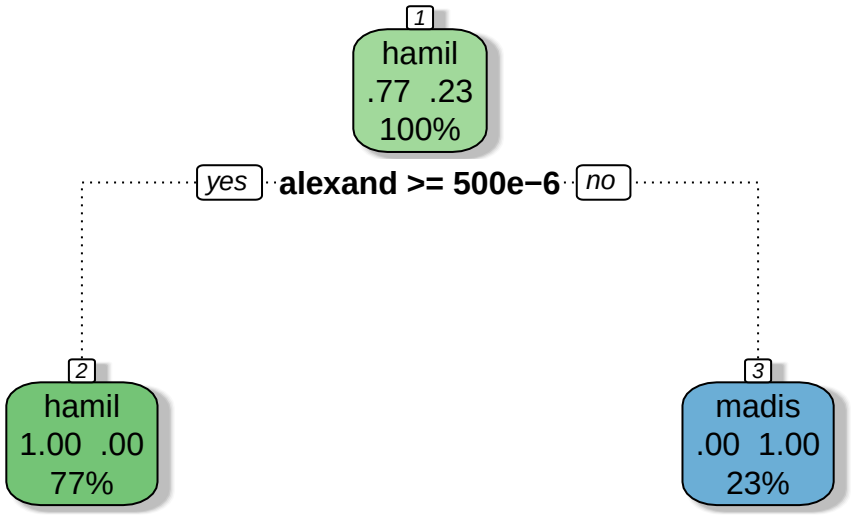
```
## Warning in rsq.rpart(train_treeNoHM2): may not be applicable for this method
```





#plot the decision tree

```
fancyRpartPlot(train_treeNoHM2)
```



#confusion matrix to find correct and incorrect predictions

```
table(Authorship=predicted2, true=testNoHM$Author)
```

```
##           true
## Authorship hamil madis
##      hamil    21     0
##      madis     0     6
```

Comparing disputed against

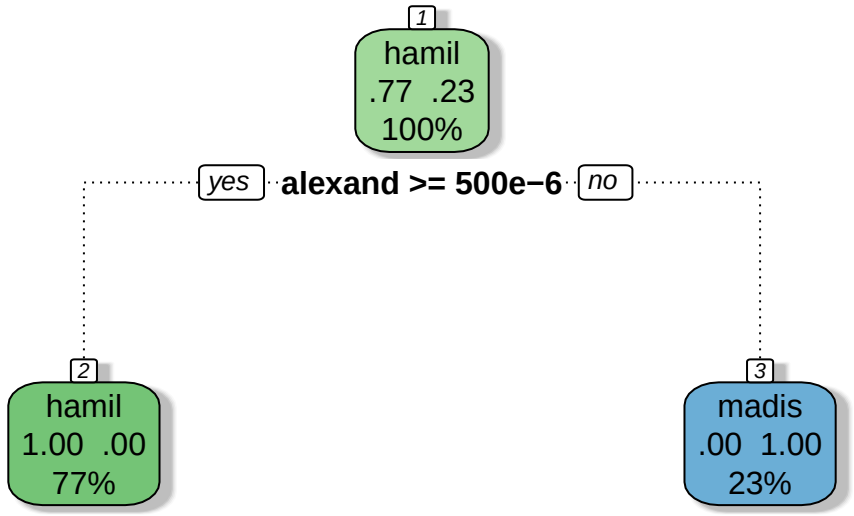
#predict the disputed dataset using the model for train tree No. 1

```
disputed <- Papers_DF1[1:11,]
#str(disputed)
predictedDisp= predict(train_treeNoHM2, Papers_DF22, type="class")
head(ResultsPDisp <- data.frame(Predicted=predictedDisp,Actual=Papers_DF22$Author),20)
```

```
##    Predicted Actual
## 1      hamil  dispt
## 2      hamil  dispt
## 3      hamil  dispt
## 4      hamil  dispt
## 5      hamil  dispt
```

```
## 6      hamil  dispt
## 7      hamil  dispt
## 8      hamil  dispt
## 9      hamil  dispt
## 10     hamil  dispt
## 11     hamil  dispt
## 12     hamil  hamil
## 13     hamil  hamil
## 14     hamil  hamil
## 15     hamil  hamil
## 16     hamil  hamil
## 17     hamil  hamil
## 18     hamil  hamil
## 19     hamil  hamil
## 20     hamil  hamil
```

```
#plot the decision tree
fancyRpartPlot(train_treeNoHM2)
```



Rattle 2021−Aug−08 17:47:45 GeorgeSmith

```
#confusion matrix to find correct and incorrect predictions
table(Authorship=predictedDisp, true=Papers_DF22$Author)
```

```
##           true
## Authorship dispt hamil madis
##      hamil    11    51     0
##      madis     0     0    15
```

Conclusion

Using the Decision Tree Algorithm we were able to create multiple views related to the authors of the federalist papers. By analyzing these views we are able to make informed decisions about which author was responsible for writing specific sections of the federalist papers.