IST 707 Homework 5

Due Date: 5/19/2021

Introduction

As a continuation of the previous report, this report will attempt to solve the mystery of the federalist papers using decision tree methodology. To recap, the previous report concluded that Madison was likely to be the author. There was still conflicting evidence, however.

Hopefully the decision tree model will help to clear up the uncertainty. The following report will train a decision tree model based on the papers that where the authors are known. After the model is trained, it will test on a smaller portion of the papers where the author is known. The model will be put into production and used on the disputed papers.

Data Preparation

The following code was used to split the data into training and testing. Essentially my logic was that the ratio of the papers by Hamilton and Madison needs to remain the same in both the training and testing sets. This is due to the fact that there were more papers written by Hamilton, so without taking that into consideration the training data could be biased.

```
# split off the testing and training
# keeping an even ratio of each author

set.seed(266)

# split hamilton papers

hamtemp    <- fedpaperssplit[which(fedpaperssplit$author == '0'), ]
hamsamp    <- createDataPartition(hamtemp$author, p = 0.80, list= FALSE)

hamtrain   <- hamtemp[hamsamp, ]
hamtest    <- hamtemp[-hamsamp, ]

# split madison papers

madtemp    <- fedpaperssplit[which(fedpaperssplit$author == '1'), ]
madsamp    <- createDataPartition(madtemp$author, p = 0.80, list= FALSE)

madtrain   <- madtemp[madsamp, ]
madtest    <- madtemp[-madsamp, ]
```

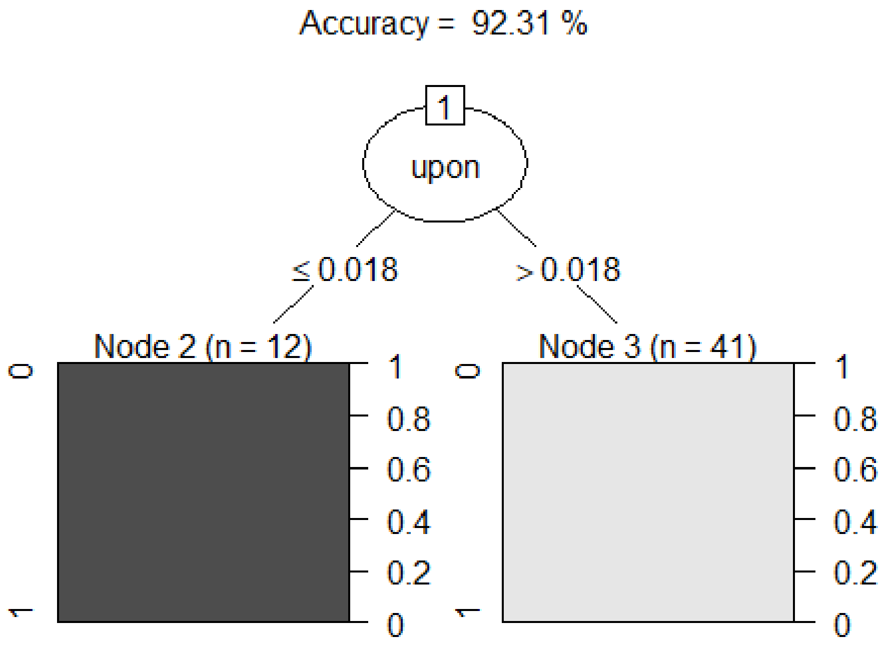
```
# join them back together

fedpaperstrain   <- rbind(hamtrain, madtrain)
fedpaperstest    <- rbind(hamtest, madtest)

# clean up the environment

rm(hamsamp, hamtemp, hamtest, hamtrain,
   madsamp, madtemp, madtest, madtrain)
```

Model 1

```
# implement the C5.0 decision tree model

fedpaperstree1 <- C5.0(author ~ ., data = fedpaperstrain)

fedpaperspredictions1 <- predict(fedpaperstree1, fedpaperstest)

# show the results of the tree model

summary(fedpaperstree1)

##
## Call:
## C5.0.formula(formula = author ~ ., data = fedpaperstrain)
##
##
## C5.0 [Release 2.07 GPL Edition]      Wed May 19 21:31:31 2021
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 53 cases (71 attributes) from undefined.data
##
## Decision tree:
##
## upon <= 0.018: 1 (12)
## upon > 0.018: 0 (41)
##
##
## Evaluation on training data (53 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##       2    0( 0.0%)   <<
##
```

```
##
##     (a)   (b)    <-classified as
##    ----  ----
##      41          (a): class 0
##            12    (b): class 1
##
##
##  Attribute usage:
##
##  100.00% upon
##
##
## Time: 0.0 secs
```

```
plot(fedpaperstree1, main = paste('Accuracy = ',
round(mean(fedpaperstest$author == fedpaperspredictions1) * 100, 2), '%'))
```

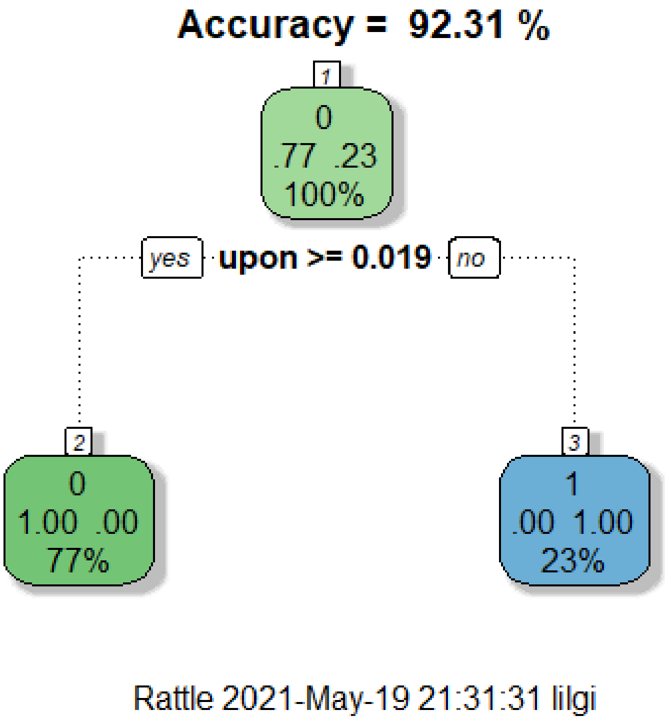


According to the first decision tree model, the word upon is the key distinguisher of which author wrote the paper. If the frequency was less than 0.018, then the author would be Madison, otherwise the author would be Hamilton. This is in line with the results from the previous report.

Model 2

```
# implement the rpart decision tree model

fedpaperstree2 <- rpart(author ~ ., data = fedpaperstrain, method = "class")

fedpaperspredictions2 <- predict(fedpaperstree2, fedpaperstest, type =
"class")

# show the results of the tree model

summary(fedpaperstree2)
```

```
## Call:
## rpart(formula = author ~ ., data = fedpaperstrain, method = "class")
##   n= 53
##
##     CP nsplit rel error     xerror       xstd
## 1 1.00      0         1 1.00000000 0.25390039
## 2 0.01      1         0 0.08333333 0.08254343
##
## Variable importance
##  upon there    on    to    by   and
##    27    22    20    13    11     7
##
## Node number 1: 53 observations,    complexity param=1
##   predicted class=0  expected loss=0.2264151  P(node) =1
##     class counts:    41    12
##    probabilities: 0.774 0.226
##   left son=2 (41 obs) right son=3 (12 obs)
##   Primary splits:
##       upon  < 0.019  to the right, improve=18.566040, (0 missing)
##       there < 0.0115 to the right, improve=15.137470, (0 missing)
##       on    < 0.0915 to the left,  improve=12.938330, (0 missing)
##       to    < 0.4745 to the right, improve= 8.110224, (0 missing)
##       by    < 0.141  to the left,  improve= 8.109897, (0 missing)
##   Surrogate splits:
##       there < 0.0115 to the right, agree=0.962, adj=0.833, (0 split)
##       on    < 0.0915 to the left,  agree=0.943, adj=0.750, (0 split)
##       to    < 0.4745 to the right, agree=0.887, adj=0.500, (0 split)
##       by    < 0.141  to the left,  agree=0.868, adj=0.417, (0 split)
##       and   < 0.428  to the left,  agree=0.830, adj=0.250, (0 split)
##
## Node number 2: 41 observations
##   predicted class=0  expected loss=0  P(node) =0.7735849
##     class counts:    41     0
##    probabilities: 1.000 0.000
##
## Node number 3: 12 observations
##   predicted class=1  expected loss=0  P(node) =0.2264151
##     class counts:     0    12
##    probabilities: 0.000 1.000

fancyRpartPlot(fedpaperstree2, main = paste('Accuracy = ',
round(mean(fedpaperstest$author == fedpaperspredictions2) * 100, 2), '%'))
```



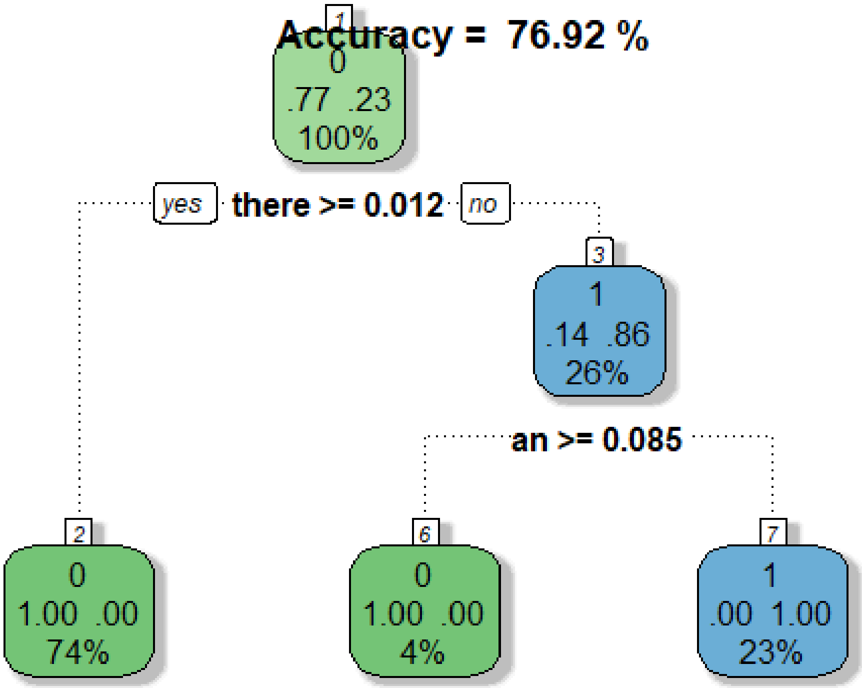
Using a different package to create a second decision tree model, I got the same results as the first package. The model seems to be having a difficult time getting past the word upon. I would be interested to see what would happen if this was removed from the training data and try to run the model again.

Model 3

```
# create new train / test data without upon

nouponfedtrain <- fedpaperstrain[, -which(colnames(fedpaperstrain) ==
'upon')]

nouponfedtest <- fedpaperstest[, -which(colnames(fedpaperstest) == 'upon')]

# implement the rpart decision tree model

fedpaperstree3 <- rpart(author ~ ., data = nouponfedtrain, method = "class",
control=rpart.control(minsplit=3, cp=0))
```

```
fedpaperspredictions3 <- predict(fedpaperstree3, nouponfedtest, type =
"class")

# show the results of the tree model

summary(fedpaperstree3)

## Call:
## rpart(formula = author ~ ., data = nouponfedtrain, method = "class",
##     control = rpart.control(minsplit = 3, cp = 0))
##   n= 53
##
##          CP nsplit rel error    xerror      xstd
## 1 0.8333333      0 1.0000000 1.0000000 0.2539004
## 2 0.1666667      1 0.1666667 0.1666667 0.1156061
## 3 0.0000000      2 0.0000000 0.1666667 0.1156061
##
## Variable importance
## there    on    by    to    an   and   any    or   was   who
##    27    17    16    14     6     6     6     3     3     3
##
## Node number 1: 53 observations,    complexity param=0.8333333
##   predicted class=0  expected loss=0.2264151  P(node) =1
##     class counts:    41    12
##    probabilities: 0.774 0.226
##   left son=2 (39 obs) right son=3 (14 obs)
##   Primary splits:
##       there < 0.0115 to the right, improve=15.137470, (0 missing)
##       on    < 0.0915 to the left,  improve=12.938330, (0 missing)
##       to    < 0.4745 to the right, improve= 8.110224, (0 missing)
##       by    < 0.141  to the left,  improve= 8.109897, (0 missing)
##       an    < 0.064  to the right, improve= 5.177524, (0 missing)
##   Surrogate splits:
##       on  < 0.0915 to the left,  agree=0.906, adj=0.643, (0 split)
##       to  < 0.4745 to the right, agree=0.849, adj=0.429, (0 split)
##       by  < 0.141  to the left,  agree=0.830, adj=0.357, (0 split)
##       and < 0.428  to the left,  agree=0.792, adj=0.214, (0 split)
##       any < 0.0125 to the right, agree=0.792, adj=0.214, (0 split)
##
## Node number 2: 39 observations
##   predicted class=0  expected loss=0  P(node) =0.7358491
##     class counts:    39     0
##    probabilities: 1.000 0.000
##
## Node number 3: 14 observations,    complexity param=0.1666667
##   predicted class=1  expected loss=0.1428571  P(node) =0.2641509
##     class counts:     2    12
##    probabilities: 0.143 0.857
##   left son=6 (2 obs) right son=7 (12 obs)
##   Primary splits:
##       an  < 0.0845 to the right, improve=3.428571, (0 missing)
##       by  < 0.095  to the left,  improve=3.428571, (0 missing)
##       or  < 0.105  to the right, improve=2.095238, (0 missing)
##       to  < 0.5365 to the right, improve=2.095238, (0 missing)
##       was < 0.0105 to the left,  improve=2.095238, (0 missing)
##   Surrogate splits:
##       by  < 0.095  to the left,  agree=1.000, adj=1.0, (0 split)
##       or  < 0.105  to the right, agree=0.929, adj=0.5, (0 split)
##       to  < 0.5365 to the right, agree=0.929, adj=0.5, (0 split)
##       was < 0.0105 to the left,  agree=0.929, adj=0.5, (0 split)
##       who < 0.0035 to the left,  agree=0.929, adj=0.5, (0 split)
##
## Node number 6: 2 observations
##   predicted class=0  expected loss=0  P(node) =0.03773585
##     class counts:     2     0
##    probabilities: 1.000 0.000
##
## Node number 7: 12 observations
##   predicted class=1  expected loss=0  P(node) =0.2264151
##     class counts:     0    12
##    probabilities: 0.000 1.000
```

```
fancyRpartPlot(fedpaperstree3, main = paste('Accuracy = ',
round(mean(nouponfedtest$author == fedpaperspredictions3) * 100, 2), '%'))
```


Accuracy = 76.92 %

Rattle 2021-May-19 21:31:32 lilgi

The model accuracy went down a lot, but there is some useful insight here. The next most important word that distinguishes the author is ‘there’. Some other words that are important are ‘on’, ‘to’, ‘by’, ‘and’, ‘any’, It is very interesting that the word ‘an’ was not as important however it is used as the third node in the decision tree model.

Deploy

```
# apply the model to the disputed papers

fedpaperspredictionsDISP <- predict(fedpaperstree1, fedpapersdisp, type =
"class")

# substitute in the predicted author

fedpapersdisp$author <- fedpaperspredictionsDISP

fedpapersdisp$author <- ifelse(fedpapersdisp$author == '1', 'Madison',
'Hamilton')

# here are the disputed papers with key words

fedpapersdisp[, c(1, which(colnames(fedpapersdisp) %in% c('upon', 'there',
'an')))]

## # A tibble: 11 x 4
##    author     an there  upon
##    <chr>   <dbl> <dbl> <dbl>
##  1 Madison 0.096 0.009 0
##  2 Madison 0.038 0     0.013
##  3 Madison 0.03  0.015 0
##  4 Madison 0.024 0     0
##  5 Madison 0.034 0.007 0
##  6 Madison 0.067 0.007 0
##  7 Madison 0.029 0.036 0
##  8 Madison 0.018 0.028 0
##  9 Madison 0.04  0.02  0
## 10 Madison 0.075 0     0
## 11 Madison 0.082 0.029 0
```

Conclusion

Overall, the results were in line with my conclusions from the previous report, which was that Madison wrote the papers because the word upon is not frequently used in the disputed papers. Although this provides some additional support, it does not give the full clarity needed to confirm who the author of the disputed papers is.

Even the best decision tree model was not able to be 100% accurate on the training data. There was one paper in the testing data that was not correctly classified. Without the model being 100% accurate, there is still some uncertainty.

There also should not be any issue with over fitting with the decision tree model. This is because the data is a population dataset and not a sample dataset (i.e the entire collection of fed papers), it is probably ok that the model is over fitted. There will be no need to generalize the model to any future papers.

In conclusion, I am more confident that Madison has written the disputed federalist papers, but again I am not certain if that is the case. To continue to try and solve this mystery, it would be interesting to gather outside data to see if would improve the models.