HW 5: Use Decision Tree to Solve a Mystery in History

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

Organize your report using the following template:

Section 1: Data preparation

You will need to separate the original data set to training and testing data for classification experiments. Describe what examples in your training and what in your test data.

Section 2: Build and tune decision tree models

First build a DT model using default setting, and then tune the parameters to see if better model can be generated. Compare these models using appropriate evaluation measures. Describe and compare the patterns learned in these models.

Section 3: Prediction

After building the classification model, apply it to the disputed papers to find out the authorship. Does the DT model reach the same conclusion as the clustering algorithms did?

```
#Name: Hrishikesh Telang
```

I am loading the required packages

```
library(factoextra)
```

```
## Loading required package: ggplot2
```

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

```
library(stringr)
library(rpart)
library(caret)
```

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

```
library(gridExtra)
library(tidyr)
```

Now I am loading the dataset

```
df <- read.csv("HW4-data-fedPapers85.csv")
```

I am viewing the dataset

```
View(df)
```

I am trying to check the structure of the dataframe

```
str(df)
```

```
## 'data.frame':    85 obs. of  72 variables:
##  $ author  : chr  "dispt" "dispt" "dispt" "dispt" ...
##  $ filename: chr  "dispt_fed_49.txt" "dispt_fed_50.txt" "dispt_fed_51.txt" "dispt_fed_52.txt" ...
##  $ a       : num  0.28 0.177 0.339 0.27 0.303 0.245 0.349 0.414 0.248 0.442 ...
##  $ all     : num  0.052 0.063 0.09 0.024 0.054 0.059 0.036 0.083 0.04 0.062 ...
##  $ also    : num  0.009 0.013 0.008 0.016 0.027 0.007 0.007 0.009 0.007 0.006 ...
##  $ an      : num  0.096 0.038 0.03 0.024 0.034 0.067 0.029 0.018 0.04 0.075 ...
##  $ and     : num  0.358 0.393 0.301 0.262 0.404 0.282 0.335 0.478 0.356 0.423 ...
##  $ any     : num  0.026 0.063 0.008 0.056 0.04 0.052 0.058 0.046 0.034 0.037 ...
##  $ are     : num  0.131 0.051 0.068 0.064 0.128 0.111 0.087 0.11 0.154 0.093 ...
##  $ as      : num  0.122 0.139 0.203 0.111 0.148 0.252 0.073 0.074 0.161 0.1 ...
##  $ at      : num  0.017 0.114 0.023 0.056 0.013 0.015 0.116 0.037 0.047 0.031 ...
##  $ be      : num  0.411 0.393 0.474 0.365 0.344 0.297 0.378 0.331 0.289 0.379 ...
##  $ been    : num  0.026 0.165 0.015 0.127 0.047 0.03 0.044 0.046 0.027 0.025 ...
##  $ but     : num  0.009 0 0.038 0.032 0.061 0.037 0.007 0.055 0.027 0.037 ...
##  $ by      : num  0.14 0.139 0.173 0.167 0.209 0.186 0.102 0.092 0.168 0.174 ...
##  $ can     : num  0.035 0 0.023 0.056 0.088 0 0.058 0.037 0.047 0.056 ...
##  $ do      : num  0.026 0.013 0 0 0 0 0.015 0.028 0 0 ...
##  $ down    : num  0 0 0.008 0 0 0.007 0 0 0 0 ...
##  $ even    : num  0.009 0.025 0.015 0.024 0.02 0.007 0.007 0.018 0 0.006 ...
##  $ every   : num  0.044 0 0.023 0.04 0.027 0.007 0.087 0.064 0.081 0.05 ...
##  $ for.    : num  0.096 0.076 0.098 0.103 0.141 0.067 0.116 0.055 0.127 0.1 ...
##  $ from    : num  0.044 0.101 0.053 0.079 0.074 0.096 0.08 0.083 0.074 0.124 ...
##  $ had     : num  0.035 0.101 0.008 0.016 0 0.022 0.015 0.009 0.007 0 ...
##  $ has     : num  0.017 0.013 0.015 0.024 0.054 0.015 0.036 0.037 0.02 0.019 ...
##  $ have    : num  0.044 0.152 0.023 0.143 0.047 0.119 0.044 0.074 0.074 0.044 ...
##  $ her     : num  0 0 0 0 0 0 0.007 0 0.034 0.025 ...
##  $ his     : num  0.017 0 0 0.024 0.02 0.067 0 0.018 0.02 0.05 ...
##  $ if.     : num  0 0.025 0.023 0.04 0.034 0.03 0.029 0 0 0.025 ...
##  $ in.     : num  0.262 0.291 0.308 0.238 0.263 0.401 0.189 0.267 0.248 0.274 ...
##  $ into    : num  0.009 0.025 0.038 0.008 0.013 0.037 0 0.037 0.013 0.037 ...
##  $ is      : num  0.157 0.038 0.15 0.151 0.189 0.26 0.167 0.083 0.208 0.23 ...
##  $ it      : num  0.175 0.127 0.173 0.222 0.108 0.156 0.102 0.165 0.134 0.131 ...
##  $ its     : num  0.07 0.038 0.03 0.048 0.013 0.015 0 0.046 0.02 0.019 ...
##  $ may     : num  0.035 0.038 0.12 0.056 0.047 0.074 0.08 0.092 0.027 0.106 ...
##  $ more    : num  0.026 0 0.038 0.056 0.067 0.045 0.08 0.064 0.06 0.081 ...
##  $ must    : num  0.026 0.013 0.083 0.071 0.013 0.015 0.044 0.018 0.027 0.068 ...
##  $ my      : num  0 0 0 0 0 0 0.007 0 0 0 ...
##  $ no      : num  0.035 0 0.03 0.032 0.047 0.059 0.022 0.018 0.02 0.044 ...
##  $ not     : num  0.114 0.127 0.068 0.087 0.128 0.134 0.102 0.101 0.094 0.106 ...
##  $ now     : num  0 0 0 0 0 0 0.007 0 0.007 0.012 ...
##  $ of      : num  0.9 0.747 0.858 0.802 0.869 ...
```

```
##  $ on      : num  0.14 0.139 0.15 0.143 0.054 0.141 0.051 0.083 0.127 0.118 ...
##  $ one     : num  0.026 0.025 0.03 0.032 0.047 0.052 0.073 0.046 0.06 0.031 ...
##  $ only    : num  0.035 0 0.023 0.048 0.027 0.022 0.007 0.046 0.02 0.012 ...
##  $ or      : num  0.096 0.114 0.06 0.064 0.081 0.074 0.153 0.037 0.154 0.081 ...
##  $ our     : num  0.017 0 0 0.016 0.027 0.03 0.051 0 0.007 0.025 ...
##  $ shall   : num  0.017 0 0.008 0.016 0 0.015 0.007 0 0.02 0 ...
##  $ should  : num  0.017 0.013 0.068 0.032 0 0.03 0.007 0 0 0.012 ...
##  $ so      : num  0.035 0.013 0.038 0.04 0.027 0.007 0.051 0.018 0.04 0.05 ...
##  $ some    : num  0.009 0.063 0.03 0.024 0.067 0.045 0.007 0.028 0.027 0.025 ...
##  $ such    : num  0.026 0 0.045 0.008 0.027 0.015 0.015 0 0.013 0.031 ...
##  $ than    : num  0.009 0 0.023 0 0.047 0.03 0.109 0.055 0.067 0.044 ...
##  $ that    : num  0.184 0.152 0.188 0.238 0.162 0.208 0.233 0.165 0.208 0.218 ...
##  $ the     : num  1.43 1.25 1.49 1.33 1.19 ...
##  $ their   : num  0.114 0.165 0.053 0.071 0.027 0.089 0.109 0.083 0.154 0.081 ...
##  $ then    : num  0 0 0.015 0.008 0.007 0.007 0.015 0.009 0.007 0.012 ...
##  $ there   : num  0.009 0 0.015 0 0.007 0.007 0.036 0.028 0.02 0 ...
##  $ things  : num  0.009 0 0 0 0 0 0 0 0 0.012 ...
##  $ this    : num  0.044 0.051 0.075 0.103 0.094 0.126 0.08 0.11 0.067 0.093 ...
##  $ to      : num  0.507 0.355 0.361 0.532 0.485 0.445 0.56 0.34 0.49 0.498 ...
##  $ up      : num  0 0 0 0 0 0 0.007 0 0 0 ...
##  $ upon    : num  0 0.013 0 0 0 0 0 0 0 0 ...
##  $ was     : num  0.009 0.051 0.008 0.087 0.027 0.007 0.015 0.018 0.027 0 ...
##  $ were    : num  0.017 0 0.015 0.079 0.02 0.03 0.029 0.009 0.007 0 ...
##  $ what    : num  0 0 0.008 0.008 0.02 0.015 0.015 0.009 0.02 0.025 ...
##  $ when    : num  0.009 0 0 0.024 0.007 0.037 0.007 0 0.02 0.012 ...
##  $ which   : num  0.175 0.114 0.105 0.167 0.155 0.186 0.211 0.175 0.201 0.199 ...
##  $ who     : num  0.044 0.038 0.008 0 0.027 0.045 0.022 0.018 0.04 0.031 ...
##  $ will    : num  0.009 0.089 0.173 0.079 0.168 0.111 0.145 0.267 0.154 0.106 ...
##  $ with    : num  0.087 0.063 0.045 0.079 0.074 0.089 0.073 0.129 0.027 0.081 ...
##  $ would   : num  0.192 0.139 0.068 0.064 0.04 0.037 0.073 0.037 0.04 0.031 ...
##  $ your    : num  0 0 0 0 0 0 0 0 0 0 ...
```

I am trying to check the structure of the dataframe

```
View(df)
```

I remove all the useless columns

```
df <- df[-c(2)]
head(df, 5)
```

```
##   author     a   all  also    an   and   any   are    as    at    be  been
## 1  dispt 0.280 0.052 0.009 0.096 0.358 0.026 0.131 0.122 0.017 0.411 0.026
## 2  dispt 0.177 0.063 0.013 0.038 0.393 0.063 0.051 0.139 0.114 0.393 0.165
## 3  dispt 0.339 0.090 0.008 0.030 0.301 0.008 0.068 0.203 0.023 0.474 0.015
## 4  dispt 0.270 0.024 0.016 0.024 0.262 0.056 0.064 0.111 0.056 0.365 0.127
## 5  dispt 0.303 0.054 0.027 0.034 0.404 0.040 0.128 0.148 0.013 0.344 0.047
##      but    by   can    do  down  even every  for.  from   had   has  have her
## 1 0.009 0.140 0.035 0.026 0.000 0.009 0.044 0.096 0.044 0.035 0.017 0.044   0
## 2 0.000 0.139 0.000 0.013 0.000 0.025 0.000 0.076 0.101 0.101 0.013 0.152   0
```

```
## 3 0.038 0.173 0.023 0.000 0.008 0.015 0.023 0.098 0.053 0.008 0.015 0.023   0
## 4 0.032 0.167 0.056 0.000 0.000 0.024 0.040 0.103 0.079 0.016 0.024 0.143   0
## 5 0.061 0.209 0.088 0.000 0.000 0.020 0.027 0.141 0.074 0.000 0.054 0.047   0
##     his   if.   in.  into    is    it   its   may  more  must my    no   not
## 1 0.017 0.000 0.262 0.009 0.157 0.175 0.070 0.035 0.026 0.026  0 0.035 0.114
## 2 0.000 0.025 0.291 0.025 0.038 0.127 0.038 0.038 0.000 0.013  0 0.000 0.127
## 3 0.000 0.023 0.308 0.038 0.150 0.173 0.030 0.120 0.038 0.083  0 0.030 0.068
## 4 0.024 0.040 0.238 0.008 0.151 0.222 0.048 0.056 0.056 0.071  0 0.032 0.087
## 5 0.020 0.034 0.263 0.013 0.189 0.108 0.013 0.047 0.067 0.013  0 0.047 0.128
##   now    of    on   one  only    or   our shall should    so  some  such  than
## 1   0 0.900 0.140 0.026 0.035 0.096 0.017 0.017  0.017 0.035 0.009 0.026 0.009
## 2   0 0.747 0.139 0.025 0.000 0.114 0.000 0.000  0.013 0.013 0.063 0.000 0.000
## 3   0 0.858 0.150 0.030 0.023 0.060 0.000 0.008  0.068 0.038 0.030 0.045 0.023
## 4   0 0.802 0.143 0.032 0.048 0.064 0.016 0.016  0.032 0.040 0.024 0.008 0.000
## 5   0 0.869 0.054 0.047 0.027 0.081 0.027 0.000  0.000 0.027 0.067 0.027 0.047
##    that   the their  then there things  this    to up  upon   was  were  what
## 1 0.184 1.425 0.114 0.000 0.009  0.009 0.044 0.507  0 0.000 0.009 0.017 0.000
## 2 0.152 1.254 0.165 0.000 0.000  0.000 0.051 0.355  0 0.013 0.051 0.000 0.000
## 3 0.188 1.490 0.053 0.015 0.015  0.000 0.075 0.361  0 0.000 0.008 0.015 0.008
## 4 0.238 1.326 0.071 0.008 0.000  0.000 0.103 0.532  0 0.000 0.087 0.079 0.008
## 5 0.162 1.193 0.027 0.007 0.007  0.000 0.094 0.485  0 0.000 0.027 0.020 0.020
##    when which   who  will  with would your
## 1 0.009 0.175 0.044 0.009 0.087 0.192    0
## 2 0.000 0.114 0.038 0.089 0.063 0.139    0
## 3 0.000 0.105 0.008 0.173 0.045 0.068    0
## 4 0.024 0.167 0.000 0.079 0.079 0.064    0
## 5 0.007 0.155 0.027 0.168 0.074 0.040    0
```

Decision Tree Algorithm

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, decision tree algorithm can be used for solving regression and classification problems too.

The general motive of using Decision Tree is to create a training model which can use to predict class or value of target variables by learning decision rules inferred from prior data(training data).

I am splitting the dataset into training and testing sets.

```r
#Training Set
training_set <- subset(df, author != "dispt")

# drop the levels information in original df, it will create troubles in prediction
training_set <- droplevels(training_set)

#Training Set
testing_set <- subset(df, author == "dispt")

# drop the levels information in original df, it will create troubles in prediction
testing_set <- droplevels(testing_set)
```

I am using cross validation to select the best model

Cross-validation is a statistical method used to estimate the skill of machine learning models. It is commonly used in applied machine learning to compare and select a model for a given predictive modeling problem because it is easy to understand, easy to implement, and results in skill estimates that generally have a lower bias than other methods.

```
#install.packages("RWeka")
#install.packages("rJava",type = "source")
library('rJava')
library('RWeka')
grid <- expand.grid(.cp=c(0.01,0.05,0.10,0.15,0.20,0.25,0.30,0.35,0.40,0.45))

grid <- expand.grid(.M=c(2,3,4,5,6,7,8,9,10),
                    .C=c(0.01,0.05,0.10,0.15,0.20,0.25,0.30,0.35,0.40,0.45))

# fit the model
optimal_model = train(author ~ .,
                  data=training_set,
                  method="J48",
                  trControl = trainControl(method = "cv",number = 10),
                  tuneGrid = grid)
```

I am checking the performance on training data

```
training_pred = predict(optimal_model, newdata = training_set)

# get the confusion matrix between groundtruth and prediction for training data
table(training_pred, training_set$author)
```

```
##
## training_pred Hamilton HM Jay Madison
##      Hamilton       50  0   0       0
##      HM              1  3   0       0
##      Jay             0  0   5       0
##      Madison         0  0   0      15
```

```
table(training_pred)
```

```
## training_pred
## Hamilton       HM      Jay  Madison
##       50        4        5       15
```

```
table(training_set$author)
```

```
##
## Hamilton       HM      Jay  Madison
##       51        3        5       15
```

```r
training_set$author <- as.factor(training_set$author)

confusionMatrix(data = training_pred, reference = training_set$author)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction Hamilton HM Jay Madison
##   Hamilton       50  0   0       0
##   HM              1  3   0       0
##   Jay             0  0   5       0
##   Madison         0  0   0      15
##
## Overall Statistics
##
##                Accuracy : 0.9865
##                  95% CI : (0.927, 0.9997)
##     No Information Rate : 0.6892
##     P-Value [Acc > NIR] : 3.743e-11
##
##                   Kappa : 0.9722
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Hamilton Class: HM Class: Jay Class: Madison
## Sensitivity                   0.9804   1.00000    1.00000         1.0000
## Specificity                   1.0000   0.98592    1.00000         1.0000
## Pos Pred Value                1.0000   0.75000    1.00000         1.0000
## Neg Pred Value                0.9583   1.00000    1.00000         1.0000
## Prevalence                    0.6892   0.04054    0.06757         0.2027
## Detection Rate                0.6757   0.04054    0.06757         0.2027
## Detection Prevalence          0.6757   0.05405    0.06757         0.2027
## Balanced Accuracy             0.9902   0.99296    1.00000         1.0000
```

```r
confusionMatrix(data = training_pred, reference = training_set$author, mode = "everything")
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction Hamilton HM Jay Madison
##   Hamilton       50  0   0       0
##   HM              1  3   0       0
##   Jay             0  0   5       0
##   Madison         0  0   0      15
##
## Overall Statistics
##
##                Accuracy : 0.9865
##                  95% CI : (0.927, 0.9997)
##     No Information Rate : 0.6892
```

```
##     P-Value [Acc > NIR] : 3.743e-11
##
##                   Kappa : 0.9722
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Hamilton Class: HM Class: Jay Class: Madison
## Sensitivity                   0.9804   1.00000    1.00000         1.0000
## Specificity                   1.0000   0.98592    1.00000         1.0000
## Pos Pred Value                1.0000   0.75000    1.00000         1.0000
## Neg Pred Value                0.9583   1.00000    1.00000         1.0000
## Precision                     1.0000   0.75000    1.00000         1.0000
## Recall                        0.9804   1.00000    1.00000         1.0000
## F1                            0.9901   0.85714    1.00000         1.0000
## Prevalence                    0.6892   0.04054    0.06757         0.2027
## Detection Rate                0.6757   0.04054    0.06757         0.2027
## Detection Prevalence          0.6757   0.05405    0.06757         0.2027
## Balanced Accuracy             0.9902   0.99296    1.00000         1.0000
```

Thus, with 94.59% probability the disputed articles belong to Madison.

I am predicting the testing data

```r
# predicted labels for the testing data
testing_pred = predict(optimal_model, newdata = testing_set)

## create a new dataframe to store prediction results
testing_result <- testing_set

## create a new column for the predictions
testing_result['prediction'] <- testing_pred

head(testing_result, 5)
```

```
##   author     a   all  also    an   and   any   are    as    at    be  been
## 1  dispt 0.280 0.052 0.009 0.096 0.358 0.026 0.131 0.122 0.017 0.411 0.026
## 2  dispt 0.177 0.063 0.013 0.038 0.393 0.063 0.051 0.139 0.114 0.393 0.165
## 3  dispt 0.339 0.090 0.008 0.030 0.301 0.008 0.068 0.203 0.023 0.474 0.015
## 4  dispt 0.270 0.024 0.016 0.024 0.262 0.056 0.064 0.111 0.056 0.365 0.127
## 5  dispt 0.303 0.054 0.027 0.034 0.404 0.040 0.128 0.148 0.013 0.344 0.047
##     but    by   can    do  down  even every  for.  from   had   has  have her
## 1 0.009 0.140 0.035 0.026 0.000 0.009 0.044 0.096 0.044 0.035 0.017 0.044   0
## 2 0.000 0.139 0.000 0.013 0.000 0.025 0.000 0.076 0.101 0.101 0.013 0.152   0
## 3 0.038 0.173 0.023 0.000 0.008 0.015 0.023 0.098 0.053 0.008 0.015 0.023   0
## 4 0.032 0.167 0.056 0.000 0.000 0.024 0.040 0.103 0.079 0.016 0.024 0.143   0
## 5 0.061 0.209 0.088 0.000 0.000 0.020 0.027 0.141 0.074 0.000 0.054 0.047   0
##     his   if.   in.  into    is    it   its   may  more  must my    no   not
## 1 0.017 0.000 0.262 0.009 0.157 0.175 0.070 0.035 0.026 0.026  0 0.035 0.114
## 2 0.000 0.025 0.291 0.025 0.038 0.127 0.038 0.038 0.000 0.013  0 0.000 0.127
```

```
## 3 0.000 0.023 0.308 0.038 0.150 0.173 0.030 0.120 0.038 0.083  0 0.030 0.068
## 4 0.024 0.040 0.238 0.008 0.151 0.222 0.048 0.056 0.056 0.071  0 0.032 0.087
## 5 0.020 0.034 0.263 0.013 0.189 0.108 0.013 0.047 0.067 0.013  0 0.047 0.128
##   now    of    on   one  only    or   our shall should    so  some  such  than
## 1   0 0.900 0.140 0.026 0.035 0.096 0.017 0.017  0.017 0.035 0.009 0.026 0.009
## 2   0 0.747 0.139 0.025 0.000 0.114 0.000 0.000  0.013 0.013 0.063 0.000 0.000
## 3   0 0.858 0.150 0.030 0.023 0.060 0.000 0.008  0.068 0.038 0.030 0.045 0.023
## 4   0 0.802 0.143 0.032 0.048 0.064 0.016 0.016  0.032 0.040 0.024 0.008 0.000
## 5   0 0.869 0.054 0.047 0.027 0.081 0.027 0.000  0.000 0.027 0.067 0.027 0.047
##    that   the their  then there things  this    to up  upon   was  were  what
## 1 0.184 1.425 0.114 0.000 0.009  0.009 0.044 0.507  0 0.000 0.009 0.017 0.000
## 2 0.152 1.254 0.165 0.000 0.000  0.000 0.051 0.355  0 0.013 0.051 0.000 0.000
## 3 0.188 1.490 0.053 0.015 0.015  0.000 0.075 0.361  0 0.000 0.008 0.015 0.008
## 4 0.238 1.326 0.071 0.008 0.000  0.000 0.103 0.532  0 0.000 0.087 0.079 0.008
## 5 0.162 1.193 0.027 0.007 0.007  0.000 0.094 0.485  0 0.000 0.027 0.020 0.020
##    when which   who  will  with would your prediction
## 1 0.009 0.175 0.044 0.009 0.087 0.192    0    Madison
## 2 0.000 0.114 0.038 0.089 0.063 0.139    0    Madison
## 3 0.000 0.105 0.008 0.173 0.045 0.068    0    Madison
## 4 0.024 0.167 0.000 0.079 0.079 0.064    0    Madison
## 5 0.007 0.155 0.027 0.168 0.074 0.040    0    Madison
```

We can finally see in the dataset that most of the disputed articles belong to Madison

Dropping the rows of files authored by Jay and Hamilton+Madison:

As we are only concerned about the authorship of the disputed articles, of Hamilton and of Madison, we are not concerned about those 3 articles written by Hamilton and Madison and 5 written by Jay. Thus, we can go ahead and remove 'Jay' and 'HM' from the dataframe and store it in the dataframe 'alt_training_set'.

```r
#Alternative Training Set
alt_training_set <- subset(df, author == "Hamilton" | author == "Madison")

# drop the levels information in original df, it will create troubles in prediction
alt_training_set <- droplevels(alt_training_set)
```

I am using cross validation to select the best model

```r
grid <- expand.grid(.cp=c(0.01,0.05,0.10,0.15,0.20,0.25,0.30,0.35,0.40,0.45))

# fit the model for alternative training data
alt_optimal_model = train(author ~ .,
                  data=alt_training_set,
                  method="rpart",
                  trControl = trainControl(method = "cv",number = 10),
                  tuneGrid = grid)
```

I am checking the performance on training data

```r
# extract predicted labels
alt_training_pred = predict(alt_optimal_model, newdata = alt_training_set)
# extract the probability of class
alt_training_prob = predict(alt_optimal_model, newdata = alt_training_set, type="prob")

# get the confusion matrix between groundtruth and prediction for training data
table(alt_training_pred, alt_training_set$author)
```

```
##
## alt_training_pred Hamilton Madison
##          Hamilton       50       0
##          Madison         1      15
```

```r
alt_training_set$author <- as.factor(alt_training_set$author)
confusionMatrix(data = alt_training_pred, reference = alt_training_set$author)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Hamilton Madison
##   Hamilton       50       0
##   Madison         1      15
##
##                Accuracy : 0.9848
##                  95% CI : (0.9184, 0.9996)
##     No Information Rate : 0.7727
##     P-Value [Acc > NIR] : 8.31e-07
##
##                   Kappa : 0.9579
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9804
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9375
##              Prevalence : 0.7727
##          Detection Rate : 0.7576
##    Detection Prevalence : 0.7576
##       Balanced Accuracy : 0.9902
##
##        'Positive' Class : Hamilton
##
```

```r
confusionMatrix(data = alt_training_pred, reference = alt_training_set$author, mode = "everything")
```

```
## Confusion Matrix and Statistics
##
##           Reference
```

```
## Prediction Hamilton Madison
##   Hamilton       50       0
##   Madison         1      15
##
##                Accuracy : 0.9848
##                  95% CI : (0.9184, 0.9996)
##     No Information Rate : 0.7727
##     P-Value [Acc > NIR] : 8.31e-07
##
##                   Kappa : 0.9579
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9804
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9375
##               Precision : 1.0000
##                  Recall : 0.9804
##                      F1 : 0.9901
##              Prevalence : 0.7727
##          Detection Rate : 0.7576
##    Detection Prevalence : 0.7576
##       Balanced Accuracy : 0.9902
##
##        'Positive' Class : Hamilton
##
```

```
## compute AUC and plot ROC curve
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

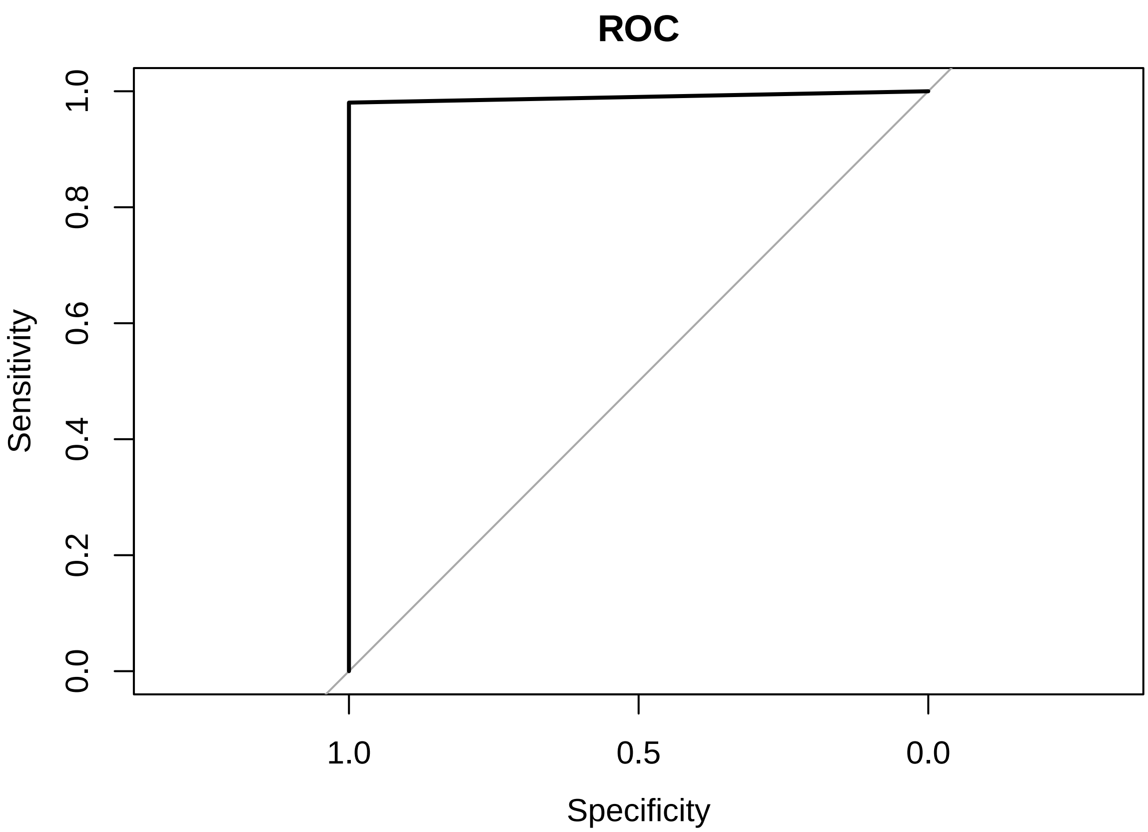
```
# plot ROC and get AUC
roc <- roc(predictor=alt_training_prob$Hamilton,
               response=alt_training_set$author,
               levels=rev(levels(alt_training_set$author)))
```

```
## Setting direction: controls < cases
```

```
roc$auc
```

```
## Area under the curve: 0.9902
```

```r
#Area under the curve: 0.9902
plot(roc,main="ROC")
```




```r
# output the important features in predicting each class
varImp(alt_optimal_model)
```

```
## rpart variable importance
##
##   only 20 most important variables shown (out of 70)
##
##         Overall
## upon     100.00
## there     65.03
## on        59.77
## to        44.85
## by        37.42
## into       0.00
## from       0.00
## had        0.00
## your       0.00
## so         0.00
## is         0.00
## were       0.00
## such       0.00
## their      0.00
```

```
## then      0.00
## our       0.00
## but       0.00
## what      0.00
## will      0.00
## things    0.00
```

Thus, with 98.48% probability the disputed articles belong to Madison.

I am predicting the alternative testing data

```r
# predicted labels for the testing data
alt_testing_pred = predict(alt_optimal_model, newdata = testing_set)

## create a new dataframe to store prediction results
alt_testing_result <- testing_set

## create a new column for the predictions
alt_testing_result['prediction'] <- alt_testing_pred

head(alt_testing_result, 5)
```

```
##   author     a   all  also    an   and   any   are    as    at    be  been
## 1  dispt 0.280 0.052 0.009 0.096 0.358 0.026 0.131 0.122 0.017 0.411 0.026
## 2  dispt 0.177 0.063 0.013 0.038 0.393 0.063 0.051 0.139 0.114 0.393 0.165
## 3  dispt 0.339 0.090 0.008 0.030 0.301 0.008 0.068 0.203 0.023 0.474 0.015
## 4  dispt 0.270 0.024 0.016 0.024 0.262 0.056 0.064 0.111 0.056 0.365 0.127
## 5  dispt 0.303 0.054 0.027 0.034 0.404 0.040 0.128 0.148 0.013 0.344 0.047
##     but    by   can    do  down  even every  for.  from   had   has  have her
## 1 0.009 0.140 0.035 0.026 0.000 0.009 0.044 0.096 0.044 0.035 0.017 0.044   0
## 2 0.000 0.139 0.000 0.013 0.000 0.025 0.000 0.076 0.101 0.101 0.013 0.152   0
## 3 0.038 0.173 0.023 0.000 0.008 0.015 0.023 0.098 0.053 0.008 0.015 0.023   0
## 4 0.032 0.167 0.056 0.000 0.000 0.024 0.040 0.103 0.079 0.016 0.024 0.143   0
## 5 0.061 0.209 0.088 0.000 0.000 0.020 0.027 0.141 0.074 0.000 0.054 0.047   0
##     his   if.   in.  into    is    it   its   may  more  must my    no   not
## 1 0.017 0.000 0.262 0.009 0.157 0.175 0.070 0.035 0.026 0.026  0 0.035 0.114
## 2 0.000 0.025 0.291 0.025 0.038 0.127 0.038 0.038 0.000 0.013  0 0.000 0.127
## 3 0.000 0.023 0.308 0.038 0.150 0.173 0.030 0.120 0.038 0.083  0 0.030 0.068
## 4 0.024 0.040 0.238 0.008 0.151 0.222 0.048 0.056 0.056 0.071  0 0.032 0.087
## 5 0.020 0.034 0.263 0.013 0.189 0.108 0.013 0.047 0.067 0.013  0 0.047 0.128
##   now    of    on   one  only    or   our shall should    so  some  such  than
## 1   0 0.900 0.140 0.026 0.035 0.096 0.017 0.017  0.017 0.035 0.009 0.026 0.009
## 2   0 0.747 0.139 0.025 0.000 0.114 0.000 0.000  0.013 0.013 0.063 0.000 0.000
## 3   0 0.858 0.150 0.030 0.023 0.060 0.000 0.008  0.068 0.038 0.030 0.045 0.023
## 4   0 0.802 0.143 0.032 0.048 0.064 0.016 0.016  0.032 0.040 0.024 0.008 0.000
## 5   0 0.869 0.054 0.047 0.027 0.081 0.027 0.000  0.000 0.027 0.067 0.027 0.047
##    that   the their  then there things  this    to up  upon   was  were  what
## 1 0.184 1.425 0.114 0.000 0.009  0.009 0.044 0.507  0 0.000 0.009 0.017 0.000
## 2 0.152 1.254 0.165 0.000 0.000  0.000 0.051 0.355  0 0.013 0.051 0.000 0.000
## 3 0.188 1.490 0.053 0.015 0.015  0.000 0.075 0.361  0 0.000 0.008 0.015 0.008
## 4 0.238 1.326 0.071 0.008 0.000  0.000 0.103 0.532  0 0.000 0.087 0.079 0.008
```

```
## 5 0.162 1.193 0.027 0.007 0.007  0.000 0.094 0.485  0 0.000 0.027 0.020 0.020
##    when which   who  will  with would your prediction
## 1 0.009 0.175 0.044 0.009 0.087 0.192    0    Madison
## 2 0.000 0.114 0.038 0.089 0.063 0.139    0    Madison
## 3 0.000 0.105 0.008 0.173 0.045 0.068    0    Madison
## 4 0.024 0.167 0.000 0.079 0.079 0.064    0    Madison
## 5 0.007 0.155 0.027 0.168 0.074 0.040    0    Madison
```

We can also finally see in the dataset that most of the disputed articles belong to Madison

Conclusion: So we can hereby conclude that, the disputed articles were authored by Madison.