Cluster Analysis and Decision Tree Induction

# Section 1: Data preparation

## Dependencies

#### loading required libraries

library(caret)  
library(rpart)  
library(rattle)  
library(factoextra)  
library(dendextend)

## Data

#### Having look at the data

df <- read.csv('Disputed\_Essay\_data.csv')  
sum(!complete.cases(df))

## [1] 0

str(df)

## 'data.frame': 85 obs. of 72 variables:  
## $ author : Factor w/ 5 levels "dispt","Hamilton",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ filename: Factor w/ 85 levels "dispt\_fed\_49.txt",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ 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 1.42 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 ...

## Data Exploration and Preprocessing

### Decision Tree:

##### Training data will be data for essays with authors hamilton and madison as we want to find out who was the person who wrote those disputed essays and testing data will be disputed author essays

##### Also removing filenames from the data

training\_data = df[df$author== 'Hamilton'|df$author=='Madison',]  
testing\_data = df[df$author=='dispt',]  
training\_data <- training\_data[,-2]  
testing\_data <- testing\_data[,c(-1,-2)]  
charArr<-as.character(training\_data$author)  
training\_data$author <- as.factor(charArr)

### K-means:

##### Input data will be data for essays with authors hamilton, madison and disputed.

##### Also removing filenames from the data and also authornames can’t be an input to the kmeans algorithm

df <- read.csv("Disputed\_Essay\_data.csv")  
df <- df[df$author!="HM",]  
df <- df[df$author!="Jay",]  
df1<-df  
authors<-df$author  
charArr<-as.character(authors)  
authors <- as.factor(charArr)  
k\_authorlist <- as.factor(charArr)  
df <- df[,-1:-2]  
df <- scale(df, center = T, scale = T)

### HAC:

##### Input will be same as k-means.

# Section 2: Build and tune cluster analysis and decision tree models

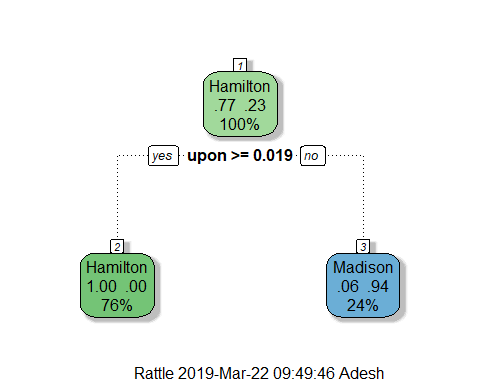
### Decision Tree:

#### Default setting:

dt\_model <- train(author~.,data=training\_data, metric= 'Accuracy',method= 'rpart')  
print(dt\_model$finalModel)

## n= 66   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 66 15 Hamilton (0.7727273 0.2272727)   
## 2) upon>=0.019 50 0 Hamilton (1.0000000 0.0000000) \*  
## 3) upon< 0.019 16 1 Madison (0.0625000 0.9375000) \*

#### Visualization:

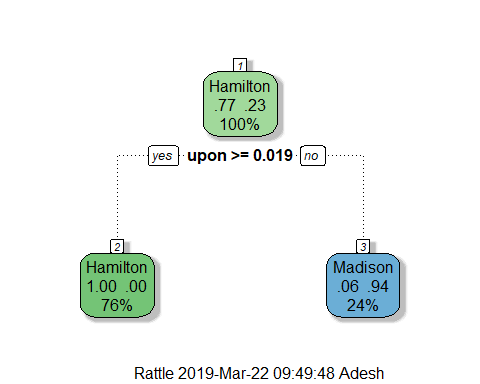
fancyRpartPlot(dt\_model$finalModel)

#### Model Tuning:

dt\_model2 <- train(author~.,data=training\_data, metric= 'ROC',method='rpart',trControl=trainControl(method="cv",number=3,classProbs=T,summaryFunction = twoClassSummary),  
 tuneGrid= expand.grid(cp=seq(0,0.01,0.001)))  
print(dt\_model2)

## CART   
##   
## 66 samples  
## 70 predictors  
## 2 classes: 'Hamilton', 'Madison'   
##   
## No pre-processing  
## Resampling: Cross-Validated (3 fold)   
## Summary of sample sizes: 44, 44, 44   
## Resampling results across tuning parameters:  
##   
## cp ROC Sens Spec   
## 0.000 0.827451 0.9215686 0.7333333  
## 0.001 0.827451 0.9215686 0.7333333  
## 0.002 0.827451 0.9215686 0.7333333  
## 0.003 0.827451 0.9215686 0.7333333  
## 0.004 0.827451 0.9215686 0.7333333  
## 0.005 0.827451 0.9215686 0.7333333  
## 0.006 0.827451 0.9215686 0.7333333  
## 0.007 0.827451 0.9215686 0.7333333  
## 0.008 0.827451 0.9215686 0.7333333  
## 0.009 0.827451 0.9215686 0.7333333  
## 0.010 0.827451 0.9215686 0.7333333  
##   
## ROC was used to select the optimal model using the largest value.  
## The final value used for the model was cp = 0.01.

#### Visualization:

fancyRpartPlot(dt\_model2$finalModel)

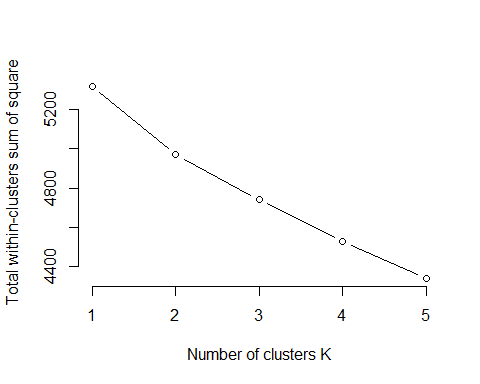
#### Observation:

##### Even after fine tuning the model learned was the default one. The model is simple and works accurately. Only one factor(upon) is enough for the finding out who wrote essays based on the available data

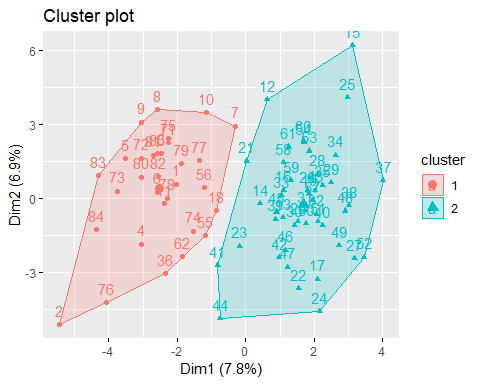
### K-means:

#### Visualization for default setting

wss <- function(k){  
 return(kmeans(df, k, nstart = 25)$tot.withinss)  
}  
k\_values <- 1:5  
  
wss\_values <- purrr::map\_dbl(k\_values, wss)  
  
plot(x = k\_values, y = wss\_values,   
 type = "b", frame = F,  
 xlab = "Number of clusters K",  
 ylab = "Total within-clusters sum of square")



km\_output <- kmeans(df, centers = 2, nstart = 25, iter.max = 100, algorithm = "Hartigan-Wong")  
fviz\_cluster(km\_output, data = df)



### HAC:

HACdist = dist(as.matrix(df))  
HAC <- as.dendrogram(hclust(HACdist, method = "ward.D2"))  
labels(HAC) <- df1$author[order.dendrogram(HAC)]

# Section 3: Prediction and interpretation

### Prediction:

#### Decision Tree Default:

dt\_predict <- predict(dt\_model, newdata = testing\_data , type = "prob")  
dt\_predict

## Hamilton Madison  
## 1 0.0625 0.9375  
## 2 0.0625 0.9375  
## 3 0.0625 0.9375  
## 4 0.0625 0.9375  
## 5 0.0625 0.9375  
## 6 0.0625 0.9375  
## 7 0.0625 0.9375  
## 8 0.0625 0.9375  
## 9 0.0625 0.9375  
## 10 0.0625 0.9375  
## 11 0.0625 0.9375

#### Decision Tree Tuned:

dt\_predict <- predict(dt\_model2, newdata = testing\_data , type = "prob")  
dt\_predict

## Hamilton Madison  
## 1 0.0625 0.9375  
## 2 0.0625 0.9375  
## 3 0.0625 0.9375  
## 4 0.0625 0.9375  
## 5 0.0625 0.9375  
## 6 0.0625 0.9375  
## 7 0.0625 0.9375  
## 8 0.0625 0.9375  
## 9 0.0625 0.9375  
## 10 0.0625 0.9375  
## 11 0.0625 0.9375

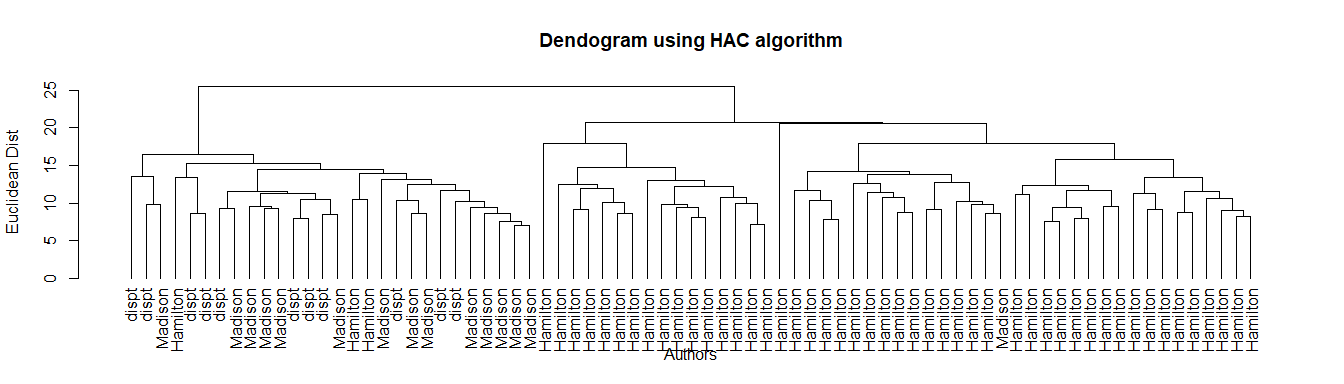
#### K-means output:

table(authors,km\_output$cluster)

##   
## authors 1 2  
## dispt 11 0  
## Hamilton 5 46  
## Madison 15 0

#### HAC output:

plot(HAC, main="Dendogram using HAC algorithm",xlab = "Authors", ylab = "Euclidean Dist",cex=0.05)



### Interpretation:

### It was MADISON !!

##### It can be seen from the Decision tree models that all the disputed essay work was predicted to be done by **Madison**. And it can be seen in the table that k-means estimated the same and HAC produced the same result. Hence all algorithms concluded the same result.

### Where are the papers with joint authorships located?

##### Lets use k-means to answer this question

df2 <- read.csv("Disputed\_Essay\_data.csv")  
df2 <- df2[df2$author!="Jay",]  
df3<-df2  
authors2<-df2$author  
charArr<-as.character(authors2)  
authors2 <- as.factor(charArr)  
df2 <- df2[,-1:-2]  
df2 <- scale(df2, center = T, scale = T)

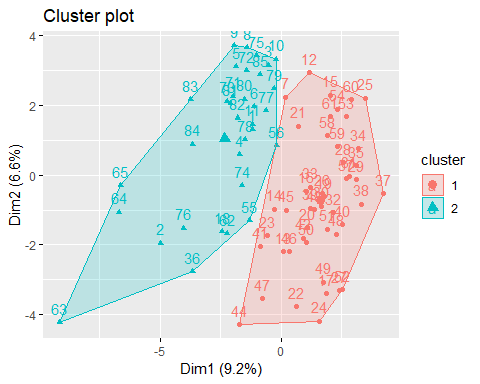
##### Locating HM data points in the dataframe

row.names(df3[df3$author=="HM",])

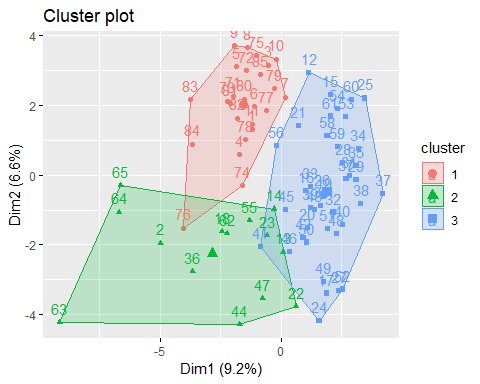
## [1] "63" "64" "65"

##### Now lets locate where HM lies in the 2 clusters.

km\_output2 <- kmeans(df2, centers = 2, nstart = 25, iter.max = 100, algorithm = "Hartigan-Wong")  
fviz\_cluster(km\_output2, data = df2)



km\_output3 <- kmeans(df2, centers = 3, nstart = 25, iter.max = 100, algorithm = "Hartigan-Wong")  
fviz\_cluster(km\_output3, data = df2)



##### As it can be seen in the plot points 63,64,65 are located to the extreme left side of the Madison(majority) cluster. Hence it can be inferenced that joint essays could have more Madison influence. Or could form their own cluster.