Week 5- Mahaddalkar Shivani

Shivani Sanjay Mahaddalkar

## Load R Packages

#Calling the appropriate packages  
# basic Rweka packages  
library(RWeka) # Weka

library(party) # A computational toolbox for recursive partitioning

library(partykit) # A toolkit with infrastructure for representing, summarizing, and visualizing tree-structured regression and classification models.

# Helper packages  
library(dplyr) # for data wrangling

library(ggplot2) # for awesome plotting  
  
# Modeling packages  
library(rpart) # direct engine for decision tree application  
library(caret) # meta engine for decision tree application

library(AmesHousing) # dataset

# Model interpretability packages  
library(rpart.plot) # for plotting decision trees  
library(vip) # for feature importance

library(pdp) # for feature effects

**Data Preparation**

Load the csv file containing the data

#Loading the data frame  
df <- read.csv("HW4-data-fedPapers85.csv")  
#Removing the filename column  
df <- df[,-2]

Separating data into training, testing and to-predict data. We are partitioning here such that, train and test data are in one dataset called trainTestData and the to-predict data is in another called predictionData

trainTestData <- df[df$author!='dispt',]  
trainTestData$author <- as.factor(trainTestData$author)  
predictionData <- df[df$author=='dispt',]  
predictionData

Splitting the trainTestData into training and testing(validation) data sets. The partition is created such that 75% of the data is used for training and the remaining 25% is used for testing(validating).

set.seed(123)  
trainData <- createDataPartition(trainTestData$author, p=.75, list = FALSE)  
trainingData <- trainTestData[trainData,]  
testingData <- trainTestData[-trainData,]

**Building up decision tree**

**Model 1: No feature engineering**

dt\_model <- train(author ~ ., data = trainingData, metric = "Accuracy", method = "rpart")  
names(dt\_model)

## [1] "method" "modelInfo" "modelType" "results" "pred"   
## [6] "bestTune" "call" "dots" "metric" "control"   
## [11] "finalModel" "preProcess" "trainingData" "resample" "resampledCM"   
## [16] "perfNames" "maximize" "yLimits" "times" "levels"   
## [21] "terms" "coefnames" "xlevels"

print(dt\_model$finalModel)

## n= 58   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 58 19 Hamilton (0.67241379 0.05172414 0.06896552 0.20689655)   
## 2) upon>=0.015 38 0 Hamilton (1.00000000 0.00000000 0.00000000 0.00000000) \*  
## 3) upon< 0.015 20 8 Madison (0.05000000 0.15000000 0.20000000 0.60000000) \*

confusionMatrix(dt\_model,trainingData$authorship)

## Bootstrapped (25 reps) Confusion Matrix   
##   
## (entries are un-normalized aggregated counts)  
##   
## Reference  
## Prediction Hamilton HM Jay Madison  
## Hamilton 346 2 2 3  
## HM 0 0 3 6  
## Jay 1 1 1 12  
## Madison 10 24 29 94  
##   
## Accuracy (average) : 0.8258

Using our model to predict for the test data

testingData$pred <- predict (dt\_model, newdata = testingData, type = c("raw"))  
confusionMatrix(testingData$author,testingData$pred)

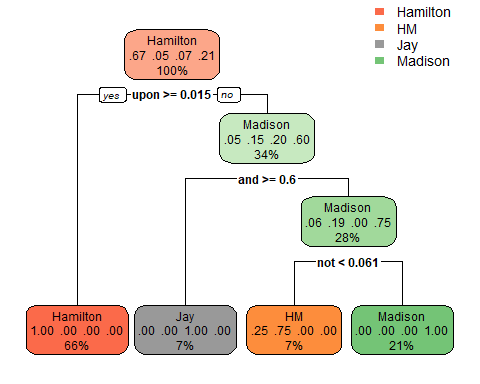
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Hamilton HM Jay Madison  
## Hamilton 12 0 0 0  
## HM 0 0 0 0  
## Jay 0 0 0 1  
## Madison 1 0 0 2  
##   
## Overall Statistics  
##   
## Accuracy : 0.875   
## 95% CI : (0.6165, 0.9845)  
## No Information Rate : 0.8125   
## P-Value [Acc > NIR] : 0.3998   
##   
## Kappa : 0.6484   
##   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: Hamilton Class: HM Class: Jay Class: Madison  
## Sensitivity 0.9231 NA NA 0.6667  
## Specificity 1.0000 1 0.9375 0.9231  
## Pos Pred Value 1.0000 NA NA 0.6667  
## Neg Pred Value 0.7500 NA NA 0.9231  
## Prevalence 0.8125 0 0.0000 0.1875  
## Detection Rate 0.7500 0 0.0000 0.1250  
## Detection Prevalence 0.7500 0 0.0625 0.1875  
## Balanced Accuracy 0.9615 NA NA 0.7949

The model presents an accuracy of 0.875 with the testing data. We will prune and imporve the performance to finally try to predict on our unknown essays list.

## Model 2: Some feature engineering

In this model the number of papers in a split are set to a minimum of 10 papers in a bucket and a maximum depth of 4.

dt\_model2 <- rpart(author~., data = trainingData, method = "class", control = rpart.control(cp=0,minsplit=10, maxdepth=4))  
rpart.plot(dt\_model2)



testingData$pred <- predict(dt\_model2, newdata = testingData, type = c("class"))  
confusionMatrix(testingData$author,testingData$pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Hamilton HM Jay Madison  
## Hamilton 12 0 0 0  
## HM 0 0 0 0  
## Jay 0 0 1 0  
## Madison 1 0 0 2  
##   
## Overall Statistics  
##   
## Accuracy : 0.9375   
## 95% CI : (0.6977, 0.9984)  
## No Information Rate : 0.8125   
## P-Value [Acc > NIR] : 0.1693   
##   
## Kappa : 0.828   
##   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: Hamilton Class: HM Class: Jay Class: Madison  
## Sensitivity 0.9231 NA 1.0000 1.0000  
## Specificity 1.0000 1 1.0000 0.9286  
## Pos Pred Value 1.0000 NA 1.0000 0.6667  
## Neg Pred Value 0.7500 NA 1.0000 1.0000  
## Prevalence 0.8125 0 0.0625 0.1250  
## Detection Rate 0.7500 0 0.0625 0.1250  
## Detection Prevalence 0.7500 0 0.0625 0.1875  
## Balanced Accuracy 0.9615 NA 1.0000 0.9643

By setting a minimum number of papers in a split to 10 and a maximum depth of 10 the accuracy went up to 93.75%.

**Model 3: Looping through the min number of instances**

## set up potential values for confidence factor and minimum number of instances  
C.values <- c(0.01,0.05,0.10,0.15,0.20,0.25,0.30,0.35,0.40,0.45,0.5)  
M.values <- c(2,3,4,5,6,7,8,9,10)  
  
## variable to record the best model  
best\_performance = 0.0  
best\_c <- 0.0  
best\_m <- 0.0  
  
for (i in 1:length(C.values)) {  
   
 for (j in 1:length(M.values)) {  
   
 c\_value = C.values[i]  
   
 m\_value = M.values[j]  
   
 m <- J48(author~., data = trainingData,   
 control = Weka\_control(U=FALSE, C = c\_value, M=m\_value))  
   
 e <- evaluate\_Weka\_classifier(m,  
 numFolds = 3, complexity = TRUE,  
 seed = 9, class = TRUE)  
  
 if (e$details['pctCorrect'] > best\_performance) {  
 best\_performance <- e$details['pctCorrect']  
   
 best\_c <- c\_value  
 best\_m <- m\_value  
 }  
   
 }  
   
}  
  
print(paste("best accuracy: ", best\_performance))

## [1] "best accuracy: 84.4827586206897"

print(paste("best m: ", best\_m))

## [1] "best m: 3"

print(paste("best c: ", best\_c))

## [1] "best c: 0.01"

m=J48(author~., data = trainingData, control=Weka\_control(U=FALSE, M=best\_m, C=best\_c))  
  
testingData$pred<- predict (m, newdata = testingData, type = c("class"))  
confusionMatrix(testingData$author,testingData$pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Hamilton HM Jay Madison  
## Hamilton 12 0 0 0  
## HM 0 0 0 0  
## Jay 0 0 0 1  
## Madison 1 0 0 2  
##   
## Overall Statistics  
##   
## Accuracy : 0.875   
## 95% CI : (0.6165, 0.9845)  
## No Information Rate : 0.8125   
## P-Value [Acc > NIR] : 0.3998   
##   
## Kappa : 0.6484   
##   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: Hamilton Class: HM Class: Jay Class: Madison  
## Sensitivity 0.9231 NA NA 0.6667  
## Specificity 1.0000 1 0.9375 0.9231  
## Pos Pred Value 1.0000 NA NA 0.6667  
## Neg Pred Value 0.7500 NA NA 0.9231  
## Prevalence 0.8125 0 0.0000 0.1875  
## Detection Rate 0.7500 0 0.0000 0.1250  
## Detection Prevalence 0.7500 0 0.0625 0.1875  
## Balanced Accuracy 0.9615 NA NA 0.7949

Here with the test data our accuracy is less than the previous model which is 87.5%. Therefore we go with the dt\_model2 to make our final predicitions.

**Prediction**

We can use the trained model ‘dt\_model2’ to now use to predict for the unknown essay authorship.

predictionData$author <- predict(dt\_model2, newdata = predictionData, type = c("class"))  
table(predictionData$author)

##   
## Hamilton HM Jay Madison   
## 0 0 0 11

Our model assigns all of the 11 articles to Madison. In the cluster assignment, my analysis was that the articles were definitely not written by Hamilton, however the cluster analysis showed that the HM papers were classified in the same clusters as that of Madison’s. However, in this analysis, it can be clearly seen that the papers are classified as Madison. Hence, this can be taken as further evidence that the paperswere infact written by Madison.