DM1

Cong Bang Huynh

12/11/2020

#devtools::install\_github("clepadellec/ClustersAnalysis")

library(ClustersAnalysis)

library(openxlsx)

## Warning: package 'openxlsx' was built under R version 4.0.3

library(ClustersAnalysis)

df=read.csv('C:/Users/DELL/Desktop/Master SISE/Github/DataMining/Train\_data.csv')

head(df,10)

## duration protocol\_type service flag src\_bytes dst\_bytes land  
## 1 0 tcp ftp\_data SF 491 0 0  
## 2 0 udp other SF 146 0 0  
## 3 0 tcp private S0 0 0 0  
## 4 0 tcp http SF 232 8153 0  
## 5 0 tcp http SF 199 420 0  
## 6 0 tcp private REJ 0 0 0  
## 7 0 tcp private S0 0 0 0  
## 8 0 tcp private S0 0 0 0  
## 9 0 tcp remote\_job S0 0 0 0  
## 10 0 tcp private S0 0 0 0  
## wrong\_fragment urgent hot num\_failed\_logins logged\_in num\_compromised  
## 1 0 0 0 0 0 0  
## 2 0 0 0 0 0 0  
## 3 0 0 0 0 0 0  
## 4 0 0 0 0 1 0  
## 5 0 0 0 0 1 0  
## 6 0 0 0 0 0 0  
## 7 0 0 0 0 0 0  
## 8 0 0 0 0 0 0  
## 9 0 0 0 0 0 0  
## 10 0 0 0 0 0 0  
## root\_shell su\_attempted num\_root num\_file\_creations num\_shells  
## 1 0 0 0 0 0  
## 2 0 0 0 0 0  
## 3 0 0 0 0 0  
## 4 0 0 0 0 0  
## 5 0 0 0 0 0  
## 6 0 0 0 0 0  
## 7 0 0 0 0 0  
## 8 0 0 0 0 0  
## 9 0 0 0 0 0  
## 10 0 0 0 0 0  
## num\_access\_files num\_outbound\_cmds is\_host\_login is\_guest\_login count  
## 1 0 0 0 0 2  
## 2 0 0 0 0 13  
## 3 0 0 0 0 123  
## 4 0 0 0 0 5  
## 5 0 0 0 0 30  
## 6 0 0 0 0 121  
## 7 0 0 0 0 166  
## 8 0 0 0 0 117  
## 9 0 0 0 0 270  
## 10 0 0 0 0 133  
## srv\_count serror\_rate srv\_serror\_rate rerror\_rate srv\_rerror\_rate  
## 1 2 0.0 0.0 0 0  
## 2 1 0.0 0.0 0 0  
## 3 6 1.0 1.0 0 0  
## 4 5 0.2 0.2 0 0  
## 5 32 0.0 0.0 0 0  
## 6 19 0.0 0.0 1 1  
## 7 9 1.0 1.0 0 0  
## 8 16 1.0 1.0 0 0  
## 9 23 1.0 1.0 0 0  
## 10 8 1.0 1.0 0 0  
## same\_srv\_rate diff\_srv\_rate srv\_diff\_host\_rate dst\_host\_count  
## 1 1.00 0.00 0.00 150  
## 2 0.08 0.15 0.00 255  
## 3 0.05 0.07 0.00 255  
## 4 1.00 0.00 0.00 30  
## 5 1.00 0.00 0.09 255  
## 6 0.16 0.06 0.00 255  
## 7 0.05 0.06 0.00 255  
## 8 0.14 0.06 0.00 255  
## 9 0.09 0.05 0.00 255  
## 10 0.06 0.06 0.00 255  
## dst\_host\_srv\_count dst\_host\_same\_srv\_rate dst\_host\_diff\_srv\_rate  
## 1 25 0.17 0.03  
## 2 1 0.00 0.60  
## 3 26 0.10 0.05  
## 4 255 1.00 0.00  
## 5 255 1.00 0.00  
## 6 19 0.07 0.07  
## 7 9 0.04 0.05  
## 8 15 0.06 0.07  
## 9 23 0.09 0.05  
## 10 13 0.05 0.06  
## dst\_host\_same\_src\_port\_rate dst\_host\_srv\_diff\_host\_rate dst\_host\_serror\_rate  
## 1 0.17 0.00 0.00  
## 2 0.88 0.00 0.00  
## 3 0.00 0.00 1.00  
## 4 0.03 0.04 0.03  
## 5 0.00 0.00 0.00  
## 6 0.00 0.00 0.00  
## 7 0.00 0.00 1.00  
## 8 0.00 0.00 1.00  
## 9 0.00 0.00 1.00  
## 10 0.00 0.00 1.00  
## dst\_host\_srv\_serror\_rate dst\_host\_rerror\_rate dst\_host\_srv\_rerror\_rate  
## 1 0.00 0.05 0.00  
## 2 0.00 0.00 0.00  
## 3 1.00 0.00 0.00  
## 4 0.01 0.00 0.01  
## 5 0.00 0.00 0.00  
## 6 0.00 1.00 1.00  
## 7 1.00 0.00 0.00  
## 8 1.00 0.00 0.00  
## 9 1.00 0.00 0.00  
## 10 1.00 0.00 0.00  
## class  
## 1 normal  
## 2 normal  
## 3 anomaly  
## 4 normal  
## 5 normal  
## 6 anomaly  
## 7 anomaly  
## 8 anomaly  
## 9 anomaly  
## 10 anomaly

df\_not\_class=df[,-42]  
head(df\_not\_class)

## duration protocol\_type service flag src\_bytes dst\_bytes land wrong\_fragment  
## 1 0 tcp ftp\_data SF 491 0 0 0  
## 2 0 udp other SF 146 0 0 0  
## 3 0 tcp private S0 0 0 0 0  
## 4 0 tcp http SF 232 8153 0 0  
## 5 0 tcp http SF 199 420 0 0  
## 6 0 tcp private REJ 0 0 0 0  
## urgent hot num\_failed\_logins logged\_in num\_compromised root\_shell  
## 1 0 0 0 0 0 0  
## 2 0 0 0 0 0 0  
## 3 0 0 0 0 0 0  
## 4 0 0 0 1 0 0  
## 5 0 0 0 1 0 0  
## 6 0 0 0 0 0 0  
## su\_attempted num\_root num\_file\_creations num\_shells num\_access\_files  
## 1 0 0 0 0 0  
## 2 0 0 0 0 0  
## 3 0 0 0 0 0  
## 4 0 0 0 0 0  
## 5 0 0 0 0 0  
## 6 0 0 0 0 0  
## num\_outbound\_cmds is\_host\_login is\_guest\_login count srv\_count serror\_rate  
## 1 0 0 0 2 2 0.0  
## 2 0 0 0 13 1 0.0  
## 3 0 0 0 123 6 1.0  
## 4 0 0 0 5 5 0.2  
## 5 0 0 0 30 32 0.0  
## 6 0 0 0 121 19 0.0  
## srv\_serror\_rate rerror\_rate srv\_rerror\_rate same\_srv\_rate diff\_srv\_rate  
## 1 0.0 0 0 1.00 0.00  
## 2 0.0 0 0 0.08 0.15  
## 3 1.0 0 0 0.05 0.07  
## 4 0.2 0 0 1.00 0.00  
## 5 0.0 0 0 1.00 0.00  
## 6 0.0 1 1 0.16 0.06  
## srv\_diff\_host\_rate dst\_host\_count dst\_host\_srv\_count dst\_host\_same\_srv\_rate  
## 1 0.00 150 25 0.17  
## 2 0.00 255 1 0.00  
## 3 0.00 255 26 0.10  
## 4 0.00 30 255 1.00  
## 5 0.09 255 255 1.00  
## 6 0.00 255 19 0.07  
## dst\_host\_diff\_srv\_rate dst\_host\_same\_src\_port\_rate  
## 1 0.03 0.17  
## 2 0.60 0.88  
## 3 0.05 0.00  
## 4 0.00 0.03  
## 5 0.00 0.00  
## 6 0.07 0.00  
## dst\_host\_srv\_diff\_host\_rate dst\_host\_serror\_rate dst\_host\_srv\_serror\_rate  
## 1 0.00 0.00 0.00  
## 2 0.00 0.00 0.00  
## 3 0.00 1.00 1.00  
## 4 0.04 0.03 0.01  
## 5 0.00 0.00 0.00  
## 6 0.00 0.00 0.00  
## dst\_host\_rerror\_rate dst\_host\_srv\_rerror\_rate  
## 1 0.05 0.00  
## 2 0.00 0.00  
## 3 0.00 0.00  
## 4 0.00 0.01  
## 5 0.00 0.00  
## 6 1.00 1.00

###### variable qualitative

var\_quanti=sapply(df\_not\_class, function(x) is.factor(x)| is.character(x)|length(unique(x))<50)

var=var\_quanti==FALSE

df\_not\_class\_quanti=df\_not\_class[,var]

head(df\_not\_class\_quanti,10)

## duration src\_bytes dst\_bytes count srv\_count serror\_rate srv\_serror\_rate  
## 1 0 491 0 2 2 0.0 0.0  
## 2 0 146 0 13 1 0.0 0.0  
## 3 0 0 0 123 6 1.0 1.0  
## 4 0 232 8153 5 5 0.2 0.2  
## 5 0 199 420 30 32 0.0 0.0  
## 6 0 0 0 121 19 0.0 0.0  
## 7 0 0 0 166 9 1.0 1.0  
## 8 0 0 0 117 16 1.0 1.0  
## 9 0 0 0 270 23 1.0 1.0  
## 10 0 0 0 133 8 1.0 1.0  
## rerror\_rate same\_srv\_rate diff\_srv\_rate srv\_diff\_host\_rate dst\_host\_count  
## 1 0 1.00 0.00 0.00 150  
## 2 0 0.08 0.15 0.00 255  
## 3 0 0.05 0.07 0.00 255  
## 4 0 1.00 0.00 0.00 30  
## 5 0 1.00 0.00 0.09 255  
## 6 1 0.16 0.06 0.00 255  
## 7 0 0.05 0.06 0.00 255  
## 8 0 0.14 0.06 0.00 255  
## 9 0 0.09 0.05 0.00 255  
## 10 0 0.06 0.06 0.00 255  
## dst\_host\_srv\_count dst\_host\_same\_srv\_rate dst\_host\_diff\_srv\_rate  
## 1 25 0.17 0.03  
## 2 1 0.00 0.60  
## 3 26 0.10 0.05  
## 4 255 1.00 0.00  
## 5 255 1.00 0.00  
## 6 19 0.07 0.07  
## 7 9 0.04 0.05  
## 8 15 0.06 0.07  
## 9 23 0.09 0.05  
## 10 13 0.05 0.06  
## dst\_host\_same\_src\_port\_rate dst\_host\_srv\_diff\_host\_rate dst\_host\_serror\_rate  
## 1 0.17 0.00 0.00  
## 2 0.88 0.00 0.00  
## 3 0.00 0.00 1.00  
## 4 0.03 0.04 0.03  
## 5 0.00 0.00 0.00  
## 6 0.00 0.00 0.00  
## 7 0.00 0.00 1.00  
## 8 0.00 0.00 1.00  
## 9 0.00 0.00 1.00  
## 10 0.00 0.00 1.00  
## dst\_host\_srv\_serror\_rate dst\_host\_rerror\_rate dst\_host\_srv\_rerror\_rate  
## 1 0.00 0.05 0.00  
## 2 0.00 0.00 0.00  
## 3 1.00 0.00 0.00  
## 4 0.01 0.00 0.01  
## 5 0.00 0.00 0.00  
## 6 0.00 1.00 1.00  
## 7 1.00 0.00 0.00  
## 8 1.00 0.00 0.00  
## 9 1.00 0.00 0.00  
## 10 1.00 0.00 0.00

data=cbind(df\_not\_class\_quanti,df$class)

head(data,10)

## duration src\_bytes dst\_bytes count srv\_count serror\_rate srv\_serror\_rate  
## 1 0 491 0 2 2 0.0 0.0  
## 2 0 146 0 13 1 0.0 0.0  
## 3 0 0 0 123 6 1.0 1.0  
## 4 0 232 8153 5 5 0.2 0.2  
## 5 0 199 420 30 32 0.0 0.0  
## 6 0 0 0 121 19 0.0 0.0  
## 7 0 0 0 166 9 1.0 1.0  
## 8 0 0 0 117 16 1.0 1.0  
## 9 0 0 0 270 23 1.0 1.0  
## 10 0 0 0 133 8 1.0 1.0  
## rerror\_rate same\_srv\_rate diff\_srv\_rate srv\_diff\_host\_rate dst\_host\_count  
## 1 0 1.00 0.00 0.00 150  
## 2 0 0.08 0.15 0.00 255  
## 3 0 0.05 0.07 0.00 255  
## 4 0 1.00 0.00 0.00 30  
## 5 0 1.00 0.00 0.09 255  
## 6 1 0.16 0.06 0.00 255  
## 7 0 0.05 0.06 0.00 255  
## 8 0 0.14 0.06 0.00 255  
## 9 0 0.09 0.05 0.00 255  
## 10 0 0.06 0.06 0.00 255  
## dst\_host\_srv\_count dst\_host\_same\_srv\_rate dst\_host\_diff\_srv\_rate  
## 1 25 0.17 0.03  
## 2 1 0.00 0.60  
## 3 26 0.10 0.05  
## 4 255 1.00 0.00  
## 5 255 1.00 0.00  
## 6 19 0.07 0.07  
## 7 9 0.04 0.05  
## 8 15 0.06 0.07  
## 9 23 0.09 0.05  
## 10 13 0.05 0.06  
## dst\_host\_same\_src\_port\_rate dst\_host\_srv\_diff\_host\_rate dst\_host\_serror\_rate  
## 1 0.17 0.00 0.00  
## 2 0.88 0.00 0.00  
## 3 0.00 0.00 1.00  
## 4 0.03 0.04 0.03  
## 5 0.00 0.00 0.00  
## 6 0.00 0.00 0.00  
## 7 0.00 0.00 1.00  
## 8 0.00 0.00 1.00  
## 9 0.00 0.00 1.00  
## 10 0.00 0.00 1.00  
## dst\_host\_srv\_serror\_rate dst\_host\_rerror\_rate dst\_host\_srv\_rerror\_rate  
## 1 0.00 0.05 0.00  
## 2 0.00 0.00 0.00  
## 3 1.00 0.00 0.00  
## 4 0.01 0.00 0.01  
## 5 0.00 0.00 0.00  
## 6 0.00 1.00 1.00  
## 7 1.00 0.00 0.00  
## 8 1.00 0.00 0.00  
## 9 1.00 0.00 0.00  
## 10 1.00 0.00 0.00  
## df$class  
## 1 normal  
## 2 normal  
## 3 anomaly  
## 4 normal  
## 5 normal  
## 6 anomaly  
## 7 anomaly  
## 8 anomaly  
## 9 anomaly  
## 10 anomaly

Test

object=multivariate\_object(data,22)

m\_test.value(object = object,i=1)

## normal pvalue  
## count 91.8636270 0.000000e+00  
## serror\_rate 103.1582793 0.000000e+00  
## srv\_serror\_rate 102.8194628 0.000000e+00  
## rerror\_rate 40.7676820 0.000000e+00  
## same\_srv\_rate -118.9164731 0.000000e+00  
## diff\_srv\_rate 30.7161982 0.000000e+00  
## dst\_host\_count 58.5391863 0.000000e+00  
## dst\_host\_srv\_count -114.1636266 0.000000e+00  
## dst\_host\_same\_srv\_rate -109.8655952 0.000000e+00  
## dst\_host\_diff\_srv\_rate 37.8015149 0.000000e+00  
## dst\_host\_same\_src\_port\_rate 14.7565334 0.000000e+00  
## dst\_host\_srv\_diff\_host\_rate 9.9877045 0.000000e+00  
## dst\_host\_serror\_rate 103.3325121 0.000000e+00  
## dst\_host\_srv\_serror\_rate 103.7625242 0.000000e+00  
## dst\_host\_rerror\_rate 40.6809729 0.000000e+00  
## dst\_host\_srv\_rerror\_rate 40.7900353 0.000000e+00  
## srv\_diff\_host\_rate -19.1490744 9.852227e-82  
## duration 8.0788895 6.661338e-16  
## dst\_bytes -1.7378326 8.224032e-02  
## src\_bytes 0.9114479 3.620594e-01  
## srv\_count 0.3761911 7.067748e-01

#les variables hot/dst\_bytes/src\_bytes/srv\_count caracterise la classe ‘normal’ le plus mauvais

m\_test.value(object = object,i=2)

## anomaly pvalue  
## count -91.8636270 0.000000e+00  
## serror\_rate -103.1582793 0.000000e+00  
## srv\_serror\_rate -102.8194628 0.000000e+00  
## rerror\_rate -40.7676820 0.000000e+00  
## same\_srv\_rate 118.9164731 0.000000e+00  
## srv\_diff\_host\_rate 19.1490744 0.000000e+00  
## dst\_host\_count -58.5391863 0.000000e+00  
## dst\_host\_srv\_count 114.1636266 0.000000e+00  
## dst\_host\_same\_srv\_rate 109.8655952 0.000000e+00  
## dst\_host\_diff\_srv\_rate -37.8015149 0.000000e+00  
## dst\_host\_serror\_rate -103.3325121 0.000000e+00  
## dst\_host\_srv\_serror\_rate -103.7625242 0.000000e+00  
## dst\_host\_rerror\_rate -40.6809729 0.000000e+00  
## dst\_host\_srv\_rerror\_rate -40.7900353 0.000000e+00  
## diff\_srv\_rate -30.7161982 3.459282e-207  
## dst\_host\_same\_src\_port\_rate -14.7565334 2.792711e-49  
## dst\_host\_srv\_diff\_host\_rate -9.9877045 1.725308e-23  
## duration -8.0788895 6.535921e-16  
## dst\_bytes 1.7378326 8.224032e-02  
## src\_bytes -0.9114479 3.620594e-01  
## srv\_count -0.3761911 7.067748e-01

# les variables hot/dst\_bytes/src\_bytes/srv\_count caracterise la classe ‘normal’ le plus mauvais:

data\_final=data[,-c(2,3,5)]

object=multivariate\_object(data\_final,19)

# Rapport de correlation

### R^2  
  
m\_R2\_multivariate(object, rescale = TRUE)

## [1] 0.004242981

colnames(data\_final)

## [1] "duration" "count"   
## [3] "serror\_rate" "srv\_serror\_rate"   
## [5] "rerror\_rate" "same\_srv\_rate"   
## [7] "diff\_srv\_rate" "srv\_diff\_host\_rate"   
## [9] "dst\_host\_count" "dst\_host\_srv\_count"   
## [11] "dst\_host\_same\_srv\_rate" "dst\_host\_diff\_srv\_rate"   
## [13] "dst\_host\_same\_src\_port\_rate" "dst\_host\_srv\_diff\_host\_rate"  
## [15] "dst\_host\_serror\_rate" "dst\_host\_srv\_serror\_rate"   
## [17] "dst\_host\_rerror\_rate" "dst\_host\_srv\_rerror\_rate"   
## [19] "df$class"

# fisher test

u\_object=Univariate\_object(data\_final,19)

u\_fisher\_test\_all(u\_object)

## Eta2 Test\_value p\_value  
## duration 0.002590943 65.4354 6.661338e-16  
## count 0.334997657 12689.5658 0.000000e+00  
## serror\_rate 0.422437799 18424.3500 0.000000e+00  
## srv\_serror\_rate 0.419667418 18216.1447 0.000000e+00  
## rerror\_rate 0.065976099 1779.3313 0.000000e+00  
## same\_srv\_rate 0.561356340 32237.0241 0.000000e+00  
## diff\_srv\_rate 0.037453250 980.1575 0.000000e+00  
## srv\_diff\_host\_rate 0.014556272 372.0887 0.000000e+00  
## dst\_host\_count 0.136034152 3966.2451 0.000000e+00  
## dst\_host\_srv\_count 0.517380558 27004.3333 0.000000e+00  
## dst\_host\_same\_srv\_rate 0.479157200 23173.9209 0.000000e+00  
## dst\_host\_diff\_srv\_rate 0.056724804 1514.8260 0.000000e+00  
## dst\_host\_same\_src\_port\_rate 0.008644170 219.6453 0.000000e+00  
## dst\_host\_srv\_diff\_host\_rate 0.003959916 100.1469 0.000000e+00  
## dst\_host\_serror\_rate 0.423865986 18532.4663 0.000000e+00  
## dst\_host\_srv\_serror\_rate 0.427401113 18802.4012 0.000000e+00  
## dst\_host\_rerror\_rate 0.065695747 1771.2387 0.000000e+00  
## dst\_host\_srv\_rerror\_rate 0.066048469 1781.4211 0.000000e+00

# Le nombre des valeurs differentes pour chaque colonne

for (i in 1:18){  
 a=length(unique(data\_final[,i]))  
 print(paste(colnames(data\_final[i]), a))  
}

## [1] "duration 758"  
## [1] "count 466"  
## [1] "serror\_rate 70"  
## [1] "srv\_serror\_rate 56"  
## [1] "rerror\_rate 72"  
## [1] "same\_srv\_rate 97"  
## [1] "diff\_srv\_rate 79"  
## [1] "srv\_diff\_host\_rate 57"  
## [1] "dst\_host\_count 256"  
## [1] "dst\_host\_srv\_count 256"  
## [1] "dst\_host\_same\_srv\_rate 101"  
## [1] "dst\_host\_diff\_srv\_rate 101"  
## [1] "dst\_host\_same\_src\_port\_rate 101"  
## [1] "dst\_host\_srv\_diff\_host\_rate 63"  
## [1] "dst\_host\_serror\_rate 100"  
## [1] "dst\_host\_srv\_serror\_rate 88"  
## [1] "dst\_host\_rerror\_rate 101"  
## [1] "dst\_host\_srv\_rerror\_rate 100"

TEST AVEC ARBRE DE DECISION

#TEST AVEC ARBRE DE DECISION

##Séparer en train-test (0.75-0.25)

colnames(data\_final)[19]='class'

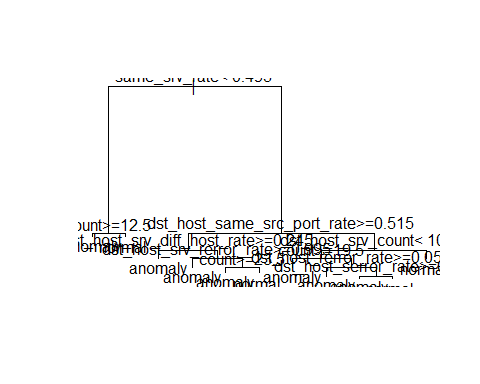
n=nrow(data\_final)  
ind\_test=sample(1:n, n\*0.25, replace = FALSE, prob = NULL)  
data\_train=data\_final[-ind\_test,]  
data\_test=data\_final[ind\_test,]

head(data\_train)

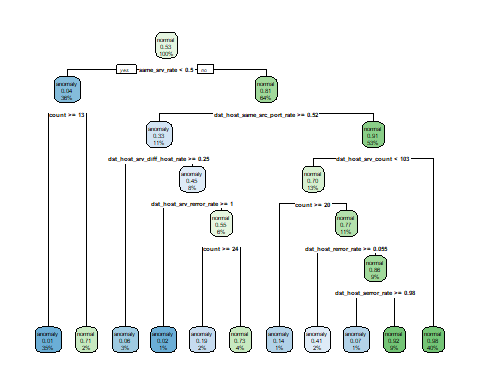
## duration count serror\_rate srv\_serror\_rate rerror\_rate same\_srv\_rate  
## 1 0 2 0.0 0.0 0 1.00  
## 2 0 13 0.0 0.0 0 0.08  
## 3 0 123 1.0 1.0 0 0.05  
## 4 0 5 0.2 0.2 0 1.00  
## 5 0 30 0.0 0.0 0 1.00  
## 6 0 121 0.0 0.0 1 0.16  
## diff\_srv\_rate srv\_diff\_host\_rate dst\_host\_count dst\_host\_srv\_count  
## 1 0.00 0.00 150 25  
## 2 0.15 0.00 255 1  
## 3 0.07 0.00 255 26  
## 4 0.00 0.00 30 255  
## 5 0.00 0.09 255 255  
## 6 0.06 0.00 255 19  
## dst\_host\_same\_srv\_rate dst\_host\_diff\_srv\_rate dst\_host\_same\_src\_port\_rate  
## 1 0.17 0.03 0.17  
## 2 0.00 0.60 0.88  
## 3 0.10 0.05 0.00  
## 4 1.00 0.00 0.03  
## 5 1.00 0.00 0.00  
## 6 0.07 0.07 0.00  
## dst\_host\_srv\_diff\_host\_rate dst\_host\_serror\_rate dst\_host\_srv\_serror\_rate  
## 1 0.00 0.00 0.00  
## 2 0.00 0.00 0.00  
## 3 0.00 1.00 1.00  
## 4 0.04 0.03 0.01  
## 5 0.00 0.00 0.00  
## 6 0.00 0.00 0.00  
## dst\_host\_rerror\_rate dst\_host\_srv\_rerror\_rate class  
## 1 0.05 0.00 normal  
## 2 0.00 0.00 normal  
## 3 0.00 0.00 anomaly  
## 4 0.00 0.01 normal  
## 5 0.00 0.00 normal  
## 6 1.00 1.00 anomaly

##Lancer Arbre de decision

library(rpart)  
data\_train\_arbre=rpart(class~.,data=data\_train, method = "class")  
plot(data\_train\_arbre)  
text(data\_train\_arbre)



library(rpart.plot)  
  
rpart.plot(data\_train\_arbre)



## Predire avec arbre de decision

predict=predict(data\_train\_arbre, data\_test,"class")

## Calculer les metriques

# Matrice de confusion  
  
conf1=table(data\_test$class,predict)  
print(conf1)

## predict  
## anomaly normal  
## anomaly 2704 207  
## normal 143 3244

# Taux d'erreur  
  
print(1-sum(diag(conf1))/sum(conf1))

## [1] 0.0555732

# Taux rappel  
  
print(conf1[2,2]/sum(conf1["normal",]))

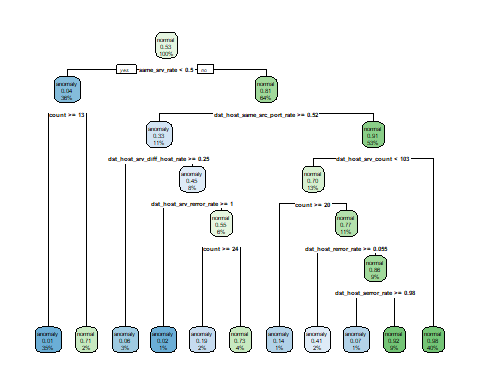
## [1] 0.9577797

# Taux precision  
  
print(conf1[2,2]/sum(conf1[,"normal"]))

## [1] 0.9400174

## tuning les hyperparametres

para=rpart.control(minsplit =10, minbucket = 2)  
data\_train\_arbre2=rpart(class~., data\_train, method = "class", control = para)  
  
#print(DFApp\_arb2)  
  
rpart.plot(data\_train\_arbre2)



predict2=predict(data\_train\_arbre2, data\_test, type="class")

# Matrice de confusion  
  
conf2=table(data\_test$class,predict2)  
print(conf2)

## predict2  
## anomaly normal  
## anomaly 2704 207  
## normal 143 3244

# Taux d'erreur  
  
print(1-sum(diag(conf2))/sum(conf2))

## [1] 0.0555732

# Taux rappel  
  
print(conf2[2,2]/sum(conf2["normal",]))

## [1] 0.9577797

# Taux precision  
  
print(conf2[2,2]/sum(conf2[,"normal"]))

## [1] 0.9400174