***Лабораторні роботи з класифікації та кластеризації даних***

***(лабораторні роботи №5-11 та***

***лабораторна робота з побудови нейронної мережі)***

студентки групи ЕП-61

Рибакової Анастасії Сергіївни

Мета лабораторних робіт – підготовка даних (збір, кодування, обчислення статистик, заповнення пропусків, видалення викидів, шкалювання та/чи логарифмування даних), побудова моделей класифікації (логістичну регресію (модель Бенчмарка з усіма змінними та оптимізовану), нейронну мережу, SVM, наївний Байєсовський класифікатор, дерево рішень та випадковий ліс), побудова моделей кластеризації (ієрархічної, за методом к-середніх та мапи Кохонена), їх аналіз та порівняння.

Для виконання лабораторних робіт було обрано датасет на медичну тему. Розглянемо як смертність від різних хвороб, освіта та витрати на медицину впливають на тривалість життя в різних країнах. Дані зібрані по країнах за 2015 рік.

Обрані показники:

1. Тривалість життя (Life) (перетворену на бінарну)
2. Смертність дорослих (AM)
3. Дитяча смертність (ID)
4. Гепатит Б (HepB)
5. Кір(Measles)
6. ІМТ (BMI)
7. Смертність дітей до 5 років (U5D)
8. Поліомієліт (Polio)
9. Загальні витрати на медицину (TotalExpend)
10. Дифтерія (Diphth)
11. ВІЛ / СНІД,% (HIV)
12. ВВП (GDP)
13. Чисельність населення (P)
14. Виснаження 1-19 років (TH119)
15. Виснаження 5-9 років (TH59)
16. Навчання в школі(S)

Classification

## Download the data

set.seed(123)  
f <- read.csv2('D.csv', header = TRUE, encoding = 'UNICOD')  
f <- f[,-1]  
head (f)

## Life AM ID HepB Measles BMI U5D Polio TotalExpend Diphth HIV GDP  
## 1 59.9 271 64 62 492 18.6 86 58 8.18 62 0.1 613  
## 2 77.5 8 NA 98 NA 57.2 1 98 5.88 98 0.1 4576  
## 3 75.4 11 21 95 NA 58.4 24 95 7.21 95 0.1 548  
## 4 51.7 348 67 64 11699 22.7 101 68 3.31 64 2.0 479  
## 5 76.2 118 8 94 1 62.2 9 92 4.79 94 0.1 12245  
## 6 74.6 12 1 93 13 54.1 1 95 4.48 93 0.1 3995  
## P TH119 TH59 S  
## 1 327582 17.5 0.5 10.0  
## 2 288914 1.3 0.8 14.2  
## 3 39113313 5.8 0.7 14.4  
## 4 2692466 8.3 0.5 11.4  
## 5 42981515 0.9 0.8 17.3  
## 6 29622 2.1 0.7 12.7

library (dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library (psych)  
d=describe(f)  
glimpse(f)

## Observations: 132  
## Variables: 16  
## $ Life <dbl> 59.9, 77.5, 75.4, 51.7, 76.2, 74.6, 73.5, 82.7, 81...  
## $ AM <dbl> 271, 8, 11, 348, 118, 12, 132, 6, 66, 119, 132, 19...  
## $ ID <dbl> 64, NA, 21, 67, 8, 1, 1, 1, NA, 5, 98, NA, NA, NA,...  
## $ HepB <dbl> 62, 98, 95, 64, 94, 93, 94, 91, 98, 94, 97, 97, 98...  
## $ Measles <dbl> 492, NA, NA, 11699, 1, 13, 2, 340, 117, NA, 289, 6...  
## $ BMI <dbl> 18.6, 57.2, 58.4, 22.7, 62.2, 54.1, 51.2, 66.1, 57...  
## $ U5D <dbl> 86, 1, 24, 101, 9, 1, 1, 1, NA, 6, 121, NA, 1, NA,...  
## $ Polio <dbl> 58, 98, 95, 68, 92, 95, 96, 92, 98, 97, 97, 97, 99...  
## $ TotalExpend <dbl> 8.18, 5.88, 7.21, 3.31, 4.79, 4.48, 4.56, 9.42, 11...  
## $ Diphth <dbl> 62, 98, 95, 64, 94, 93, 94, 92, 98, 94, 97, 97, 99...  
## $ HIV <dbl> 0.1, 0.1, 0.1, 2.0, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...  
## $ GDP <dbl> 613, 4576, 548, 479, 12245, 3995, 3218, 62215, 513...  
## $ P <dbl> 327582, 288914, 39113313, 2692466, 42981515, 29622...  
## $ TH119 <dbl> 17.5, 1.3, 5.8, 8.3, 0.9, 2.1, 2.1, 0.6, 2.0, 2.9,...  
## $ TH59 <dbl> 0.5, 0.8, 0.7, 0.5, 0.8, 0.7, 0.7, 0.9, 0.9, 0.8, ...  
## $ S <dbl> 10.0, 14.2, 14.4, 11.4, 17.3, 12.7, 12.3, 20.4, 15...

#Висновок: 132 спостереження та 16 змінних.Факторних змінних серед них немає

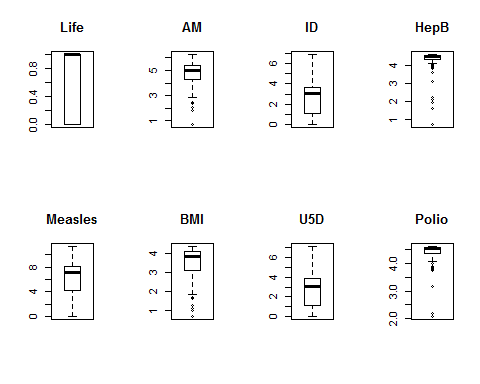
# Fill n/a with average

mLife <- mean(f$Life)  
f$Life <- ifelse(f$Life> mLife, 1, 0)  
#Змінна Life перетворена в бінарну  
f\_fill <- f  
f\_fill$ID <- ifelse(is.na(f$ID),round(mean(f$ID,na.rm = TRUE)),f$ID)  
f\_fill$Measles <- ifelse(is.na(f$Measles),round(mean(f$Measles,na.rm = TRUE)),f$Measles)  
f\_fill$U5D <- ifelse(is.na(f$U5D),round(mean(f$U5D,na.rm = TRUE)),f$U5D)  
f <- f\_fill  
h=describe(f\_fill)  
#Висновок: пропуски заповнені середніми значеннями змінних.

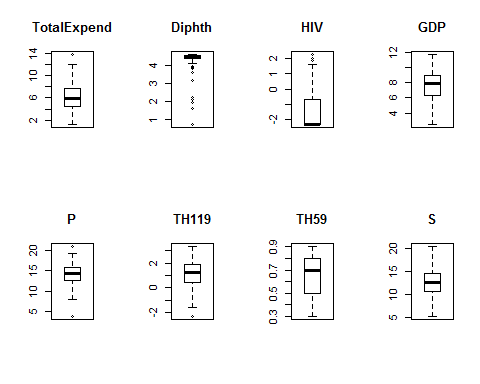
# Log

f\_log <- f  
f\_log$AM <- log(f$AM, )  
f\_log$ID <- log(f$ID, )  
f\_log$HepB <- log(f$HepB, )  
f\_log$Measles <- log(f$Measles, )  
f\_log$BMI <- log(f$BMI, )  
f\_log$U5D <- log(f$U5D, )  
f\_log$Polio <- log(f$Polio, )  
f\_log$Diphth<- log(f$Diphth, )  
f\_log$HIV <- log(f$HIV, )  
f\_log$GDP <- log(f$GDP, )  
f\_log$P <- log(f$P, )  
f\_log$TH119 <- log(f$TH119, )  
f<- f\_log

#Box-plot  
par(mfrow = c(2, 4))  
boxplot(f$Life, main = 'Life')  
boxplot(f$AM, main = 'AM')  
boxplot(f$ID, main = 'ID')  
boxplot(f$HepB, main = 'HepB')  
boxplot(f$Measles, main = 'Measles')  
boxplot(f$BMI, main = 'BMI')  
boxplot(f$U5D, main = 'U5D')  
boxplot(f$Polio, main = 'Polio')



boxplot(f$TotalExpend, main = 'TotalExpend')  
boxplot(f$Diphth, main = 'Diphth')  
boxplot(f$HIV, main = 'HIV')  
boxplot(f$GDP, main = 'GDP')  
boxplot(f$P, main = 'P')  
boxplot(f$TH119, main = 'TH119')  
boxplot(f$TH59, main = 'TH59')  
boxplot(f$S, main = 'S')



#Висновок: змінні AM, HepB, BMI, Polio, TotalExpend, Diphth, HIV, P, TH119 мають викиди. Змінні ID, Measles, U5D та GDP після логарифмування звільнилися від викидів.

# Replace ejections (outside the three sigma) with boundary values.

f\_ej <- f  
f\_ej$AM <- ifelse(f$AM < mean(f$AM) + sd(f$AM)\*3, f$AM,mean(f$AM) + sd(f$AM)\*3)  
f\_ej$HepB <- ifelse(f$HepB < mean(f$HepB) + sd(f$HepB)\*3, f$HepB,mean(f$HepB) + sd(f$HepB)\*3)  
f\_ej$BMI <- ifelse(f$BMI < mean(f$BMI) + sd(f$BMI)\*3, f$BMI,mean(f$BMI) + sd(f$BMI)\*3)  
f\_ej$Polio <- ifelse(f$Polio < mean(f$Polio) + sd(f$Polio)\*3, f$Polio,mean(f$Polio) + sd(f$Polio)\*3)  
f\_ej$TotalExpend <- ifelse(f$TotalExpend < mean(f$TotalExpend) + sd(f$TotalExpend)\*3, f$TotalExpend,mean(f$TotalExpend) + sd(f$TotalExpend)\*3)  
f\_ej$Diphth <- ifelse(f$Diphth < mean(f$Diphth) + sd(f$Diphth)\*3, f$Diphth,mean(f$Diphth) + sd(f$Diphth)\*3)  
f\_ej$HIV <- ifelse(f$HIV < mean(f$HIV) + sd(f$HIV)\*3, f$HIV,mean(f$HIV) + sd(f$HIV)\*3)  
f\_ej$P <- ifelse(f$P < mean(f$P) + sd(f$P)\*3, f$P,mean(f$P) + sd(f$P)\*3)  
f\_ej$TH119 <- ifelse(f$TH119 < mean(f$TH119) + sd(f$TH119)\*3, f$TH119,mean(f$TH119) + sd(f$TH119)\*3)  
f <- f\_ej  
#Висновок: для корекції викидів здійснене заповнення граничними значеннями.

# Splitting the scaled dataset into the TRAIN set and TEST set

set.seed(123)  
library(caTools)  
split = sample.split(f$Life, SplitRatio = 2/3)  
f\_train\_ws = subset(f, split == TRUE)  
f\_test\_ws = subset(f, split == FALSE)  
#Висновок: датасет розподілений на навчальну та тестову вибірки (88 та 44 спостереження)

# Features Scaling

f\_sc=f  
f\_train<-f\_train\_ws  
f\_test<-f\_test\_ws  
  
mAM <- mean(f\_train$AM)  
sAM <- sd(f\_train$AM)  
f\_train$AM <- (f\_train$AM-mAM)/sAM  
f\_test$AM <- (f\_test$AM-mAM)/sAM  
f\_sc$AM <- (f$AM-mAM)/sAM  
  
mID <- mean(f\_train$ID)  
sID <- sd(f\_train$ID)  
f\_train$ID <- (f\_train$ID-mID)/sID  
f\_test$ID <- (f\_test$ID-mID)/sID  
f\_sc$ID <- (f$ID-mID)/sID  
  
mHepB <- mean(f\_train$HepB)  
sHepB <- sd(f\_train$HepB)  
f\_train$HepB <- (f\_train$HepB-mHepB)/sHepB  
f\_test$HepB <- (f\_test$HepB-mHepB)/sHepB  
f\_sc$HepB <- (f$HepB-mHepB)/sHepB  
  
mMeasles <- mean(f\_train$Measles)  
sMeasles <- sd(f\_train$Measles)  
f\_train$Measles <- (f\_train$Measles-mMeasles)/sMeasles  
f\_test$Measles <- (f\_test$Measles-mMeasles)/sMeasles  
f\_sc$Measles <- (f$Measles-mMeasles)/sMeasles  
  
mBMI <- mean(f\_train$BMI)  
sBMI <- sd(f\_train$BMI)  
f\_train$BMI <- (f\_train$BMI-mBMI)/sBMI  
f\_test$BMI <- (f\_test$BMI-mBMI)/sBMI  
f\_sc$BMI <- (f$BMI-mBMI)/sBMI  
  
mU5D <- mean(f\_train$U5D)  
sU5D <- sd(f\_train$U5D)  
f\_train$U5D <- (f\_train$U5D-mU5D)/sU5D  
f\_test$U5D <- (f\_test$U5D-mU5D)/sU5D  
f\_sc$U5D <- (f$U5D-mU5D)/sU5D  
  
mPolio <- mean(f\_train$Polio)  
sPolio <- sd(f\_train$Polio)  
f\_train$Polio <- (f\_train$Polio-mPolio)/sPolio  
f\_test$Polio <- (f\_test$Polio-mPolio)/sPolio  
f\_sc$Polio <- (f$Polio-mPolio)/sPolio  
  
mTotalExpend <- mean(f\_train$TotalExpend)  
sTotalExpend <- sd(f\_train$TotalExpend)  
f\_train$TotalExpend <- (f\_train$TotalExpend-mTotalExpend)/sTotalExpend  
f\_test$TotalExpend <- (f\_test$TotalExpend-mTotalExpend)/sTotalExpend  
f\_sc$TotalExpend <- (f$TotalExpend-mTotalExpend)/sTotalExpend  
  
mDiphth <- mean(f\_train$Diphth)  
sDiphth <- sd(f\_train$Diphth)  
f\_train$Diphth <- (f\_train$Diphth-mDiphth)/sDiphth  
f\_test$Diphth <- (f\_test$Diphth-mDiphth)/sDiphth  
f\_sc$Diphth <- (f$Diphth-mDiphth)/sDiphth  
  
mHIV <- mean(f\_train$HIV)  
sHIV <- sd(f\_train$HIV)  
f\_train$HIV <- (f\_train$HIV-mHIV)/sHIV  
f\_test$HIV <- (f\_test$HIV-mHIV)/sHIV  
f\_sc$HIV <- (f$HIV-mHIV)/sHIV  
  
mGDP <- mean(f\_train$GDP)  
sGDP <- sd(f\_train$GDP)  
f\_train$GDP <- (f\_train$GDP-mGDP)/sGDP  
f\_test$GDP <- (f\_test$GDP-mGDP)/sGDP  
f\_sc$GDP <- (f$GDP-mGDP)/sGDP  
  
mP <- mean(f\_train$P)  
sP <- sd(f\_train$P)  
f\_train$P <- (f\_train$P-mP)/sP  
f\_test$P <- (f\_test$P-mP)/sP  
f\_sc$P <- (f$P-mP)/sP  
  
mTH119 <- mean(f\_train$TH119)  
sTH119 <- sd(f\_train$TH119)  
f\_train$TH119 <- (f\_train$TH119-mTH119)/sTH119  
f\_test$TH119 <- (f\_test$TH119-mTH119)/sTH119  
f\_sc$TH119 <- (f$TH119-mTH119)/sTH119  
  
mTH59 <- mean(f\_train$TH59)  
sTH59 <- sd(f\_train$TH59)  
f\_train$TH59 <- (f\_train$TH59-mTH59)/sTH59  
f\_test$TH59 <- (f\_test$TH59-mTH59)/sTH59  
f\_sc$TH59 <- (f$TH59-mTH59)/sTH59  
  
mS <- mean(f\_train$S)  
sS <- sd(f\_train$S)  
f\_train$S <- (f\_train$S-mS)/sS  
f\_test$S <- (f\_test$S-mS)/sS  
f\_sc$S <- (f$S-mS)/sS  
#Висновки: виконане шкалювання кількісних змінних

# Fitting (Benchmark model)

class\_lr <- glm(Life ~ ., f\_train, family = binomial)  
summary(class\_lr)

##   
## Call:  
## glm(formula = Life ~ ., family = binomial, data = f\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.16469 -0.14851 0.00826 0.29628 2.31271   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.3679 0.5465 0.673 0.5008   
## AM 0.2452 0.5960 0.411 0.6808   
## ID -6.0775 8.3854 -0.725 0.4686   
## HepB 0.8703 2.4262 0.359 0.7198   
## Measles -0.5514 0.5995 -0.920 0.3577   
## BMI 0.2341 0.4760 0.492 0.6228   
## U5D 5.9268 8.6177 0.688 0.4916   
## Polio -0.3434 0.4825 -0.712 0.4766   
## TotalExpend 1.8144 0.7958 2.280 0.0226 \*  
## Diphth -0.3295 2.4361 -0.135 0.8924   
## HIV -1.9246 0.9432 -2.040 0.0413 \*  
## GDP 0.8047 0.6418 1.254 0.2099   
## P -0.2840 0.4722 -0.601 0.5476   
## TH119 -0.6347 0.6131 -1.035 0.3005   
## TH59 2.6571 1.5058 1.765 0.0776 .  
## S 0.2185 1.4726 0.148 0.8821   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 121.266 on 87 degrees of freedom  
## Residual deviance: 37.419 on 72 degrees of freedom  
## AIC: 69.419  
##   
## Number of Fisher Scoring iterations: 8

#Висновки: змінні TotalExpend та HIV значущі. Змінна TH59 менш значуща.

# Optimized model

#Для оптимізованої моделі обрано змінні TotalExpend i HIV, оскільки вони найбільш значущі.  
class\_opt <- glm(Life ~ TotalExpend + HIV, f\_train, family = binomial)  
summary(class\_opt)

##   
## Call:  
## glm(formula = Life ~ TotalExpend + HIV, family = binomial, data = f\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.1930 -0.4264 0.3575 0.6424 1.8536   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.1592 0.3511 -0.453 0.6502   
## TotalExpend 0.8519 0.3329 2.559 0.0105 \*   
## HIV -2.4094 0.5753 -4.188 2.81e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 121.266 on 87 degrees of freedom  
## Residual deviance: 70.961 on 85 degrees of freedom  
## AIC: 76.961  
##   
## Number of Fisher Scoring iterations: 6

#Висновки: усі змінні значущі

# Predicting

p <- predict(class\_opt, f\_test[, c('TotalExpend','HIV')], type = 'response')  
y <- ifelse(p > 0.5, 1, 0)  
#Висновки: розраховано ймовірностівіднесення об'єктів до кожного з 2 класів (ректор р), визначені класи об'єктів (вектор у)

# Confusion Matrix

cm = table(f\_test[, 'Life'], y > 0.7)  
print(cm)

##   
## FALSE TRUE  
## 0 16 4  
## 1 0 24

t11=cm[1]  
t21=cm[2]  
t12=cm[3]  
t22=cm[4]  
sum=t11+t12+t21+t22  
tochn\_opt=(t11+t22)/sum  
nevcl\_opt=(t12+t21)/sum  
chutl\_opt=t22/(t21+t22)  
spec\_opt=t11/(t11+t12)  
tochn\_opt\*100

## [1] 90.90909

nevcl\_opt\*100

## [1] 9.090909

chutl\_opt\*100

## [1] 100

spec\_opt\*100

## [1] 80

#Точність моделі = (24+16)/44=90,9%  
#Частка невірно класифікованих = (4+0)/44=9,1%  
#Чутливість моделі = 24/(0+24)=100%  
#специфічність моделі = 16/(4+16)=80%  
#Отже, модель більш чутлива до виявлення позитивних випадків

# ROCR

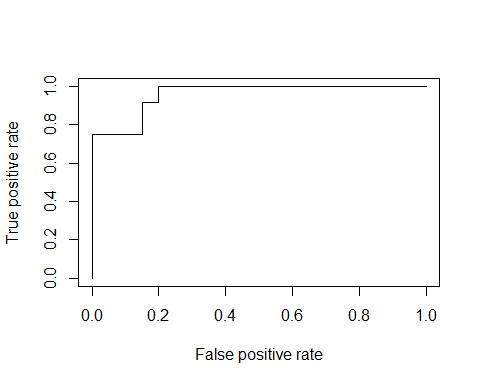
library(ROCR)

## Loading required package: gplots

##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

pref <- prediction(p, f\_test$Life)  
perf <- performance(pref, "tpr", "fpr")  
plot(perf)



#Висновок: Співвідношення істинно-позитивних та хибно негативних випадків свідчить про відносно хорошу якість моделі.

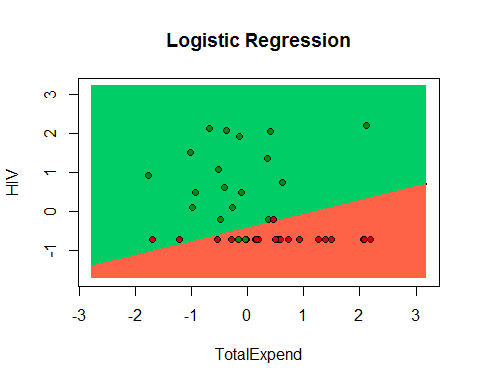
# Visualising the Test set results

library(ggplot2)

##   
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':  
##   
## %+%, alpha

set = f\_test[,c('TotalExpend','HIV','Life')]  
X1 = seq(min(set['TotalExpend']) - 1, max(set['TotalExpend']) + 1, by = 0.01)  
X2 = seq(min(set['HIV']) - 1, max(set['HIV']) + 1, by = 0.01)  
grid\_set = expand.grid(X1, X2)  
colnames(grid\_set) = c('TotalExpend', 'HIV')  
prob\_set = predict(class\_opt, grid\_set, type = 'response')  
y\_grid = ifelse(prob\_set > 0.7, 1, 0)  
plot(set[, -3],  
 main = 'Logistic Regression',  
 xlab = 'TotalExpend', ylab = 'HIV',  
 xlim = range(X1), ylim = range(X2))  
contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE)  
points(grid\_set, pch = '.', col = ifelse(y\_grid == 1, 'tomato', 'springgreen3'))  
points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'red3', 'green4'))



#Червоним позначені країни, тривалість життя в яких нижче за середню, зеленим позначені країни, тривалість життя в яких вище за середню. Червона зона - скоріш за все країни, тривалість життя в яких нижче за середню. У моделі лінійний варіант розподільної кривої

## K-Nearest Neighbors (K-NN)

# Fitting & predicting

library(class)  
y = knn(train = f\_train[,c('TotalExpend','HIV')],  
 test = f\_test[,c('TotalExpend','HIV')],  
 cl = f\_train[, 'Life'],  
 k = 5,  
 prob = TRUE)

# Confusion Matrix

cm = table(f\_test[, 'Life'], y == '1')  
print(cm)

##   
## FALSE TRUE  
## 0 14 6  
## 1 1 23

t11=cm[1]  
t21=cm[2]  
t12=cm[3]  
t22=cm[4]  
sum=t11+t12+t21+t22  
tochn\_knn=(t11+t22)/sum  
nevcl\_knn=(t12+t21)/sum  
chutl\_knn=t22/(t21+t22)  
spec\_knn=t11/(t11+t12)  
tochn\_knn\*100

## [1] 84.09091

nevcl\_knn\*100

## [1] 15.90909

chutl\_knn\*100

## [1] 95.83333

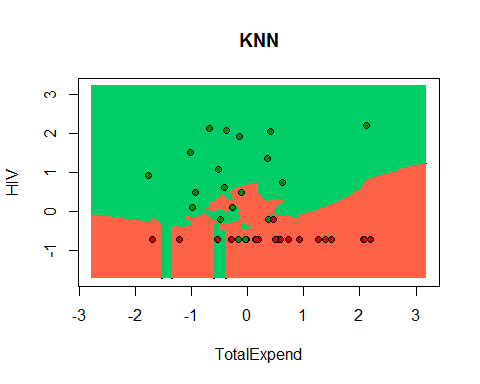
spec\_knn\*100

## [1] 70

#Точність моделі = (23+14)/44=84,1%  
#Частка невірно класифікованих = (1+6)/44=15,9%  
#Чутливість моделі = 23/(1+23)=95,8%  
#специфічність моделі = 14/(6+14)=70%  
#Отже, модель більш чутлива до виявлення позитивних випадків

# Visualising the Test set results

library(ggplot2)  
set = f\_test[,c('TotalExpend','HIV','Life')]  
X1 = seq(min(set['TotalExpend']) - 1, max(set['TotalExpend']) + 1, by = 0.01)  
X2 = seq(min(set['HIV']) - 1, max(set['HIV']) + 1, by = 0.01)  
grid\_set = expand.grid(X1, X2)  
colnames(grid\_set) = c('TotalExpend', 'HIV')  
y\_grid = knn(train = f\_train[,c('TotalExpend','HIV')], test = grid\_set, cl = f\_train[, 'Life'], k = 5)  
plot(set[, -3],  
 main = 'KNN',  
 xlab = 'TotalExpend', ylab = 'HIV',  
 xlim = range(X1), ylim = range(X2))  
contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE)  
points(grid\_set, pch = '.', col = ifelse(y\_grid == 1, 'tomato', 'springgreen3'))  
points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'red3', 'green4'))



#Червоним позначені країни, тривалість життя в яких нижче за середню, зеленим позначені країни, тривалість життя в яких вище за середню. Червона зона - скоріш за все країни, тривалість життя в яких нижче за середню. У моделі нелінійний варіант розподільної кривої

## Support Vector Machine (SVM)

# Fitting SVM model

f\_train2f<-f\_train[,c('TotalExpend','HIV','Life')]  
f\_test2f<-f\_test[,c('TotalExpend','HIV','Life')]  
library(e1071)  
class\_svm\_l = svm(Life ~ TotalExpend + HIV, data = f\_train2f, kernel = 'linear')  
summary(class\_svm\_l)

##   
## Call:  
## svm(formula = Life ~ TotalExpend + HIV, data = f\_train2f, kernel = "linear")  
##   
##   
## Parameters:  
## SVM-Type: eps-regression   
## SVM-Kernel: linear   
## cost: 1   
## gamma: 0.5   
## epsilon: 0.1   
##   
##   
## Number of Support Vectors: 62

#Для навчання базової моделі, заснованої на методі опорних векторів, обрано лінійне ядро

# Predicting

p <- predict(class\_svm\_l, f\_test2f[, c('TotalExpend','HIV')])  
y <- ifelse(p > 0.5, 1, 0)  
#Визначено класи об'єктів (вектор у)

# Confusion Matrix

cm = table(f\_test2f[, 'Life'], y)  
print(cm)

## y  
## 0 1  
## 0 11 9  
## 1 0 24

t11=cm[1]  
t21=cm[2]  
t12=cm[3]  
t22=cm[4]  
sum=t11+t12+t21+t22  
tochn\_svm=(t11+t22)/sum  
nevcl\_svm=(t12+t21)/sum  
chutl\_svm=t22/(t21+t22)  
spec\_svm=t11/(t11+t12)  
tochn\_svm\*100

## [1] 79.54545

nevcl\_svm\*100

## [1] 20.45455

chutl\_svm\*100

## [1] 100

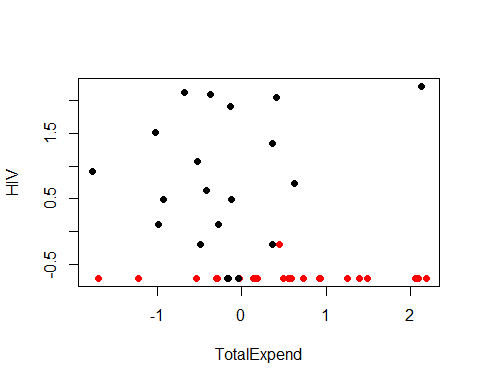
spec\_svm\*100

## [1] 55

#Точність моделі = (24+11)/44=79,5%  
#Частка невірно класифікованих = (9+0)/44=20,5%  
#Чутливість моделі = 24/(0+24)=100%  
#специфічність моделі = 11/(9+11)=55%  
#Отже, модель більш чутлива до виявлення позитивних випадків

# Visualising the Test set results

xgrid = expand.grid(TotalExpend = f\_test2f$TotalExpend, HIV = f\_test2f$HIV)  
ygrid = predict(class\_svm\_l, xgrid)  
#Finally, you plot the points and color them according to the decision boundary. You can see that the decision boundary is linear. You can put the data points in the plot as well to see where they lie.  
plot(xgrid, col = as.numeric(ygrid), pch = 10, cex = .9)  
points(f\_test2f[, c('TotalExpend','HIV')], col = as.factor(f\_test2f$Life), pch = 19)



#Червоним позначені країни, тривалість життя в яких нижче за середню, чорним позначені країни, тривалість життя в яких вище за середню. У моделі лінійний варіант розподільної кривої

# Fitting RBF-kernel model

library(e1071)  
class\_svm\_r = svm(Life ~ TotalExpend + HIV, data = f\_train2f, kernel = 'radial')  
summary(class\_svm\_r)

##   
## Call:  
## svm(formula = Life ~ TotalExpend + HIV, data = f\_train2f, kernel = "radial")  
##   
##   
## Parameters:  
## SVM-Type: eps-regression   
## SVM-Kernel: radial   
## cost: 1   
## gamma: 0.5   
## epsilon: 0.1   
##   
##   
## Number of Support Vectors: 53

# Predicting

p <- predict(class\_svm\_r, f\_test2f[, c('TotalExpend','HIV')])  
y <- ifelse(p > 0.5, 1, 0)

## Confusion Matrix

cm = table(f\_test2f[, 'Life'], y)  
print(cm)

## y  
## 0 1  
## 0 15 5  
## 1 1 23

t11=cm[1]  
t21=cm[2]  
t12=cm[3]  
t22=cm[4]  
sum=t11+t12+t21+t22  
tochn\_svm\_r=(t11+t22)/sum  
nevcl\_svm\_r=(t12+t21)/sum  
chutl\_svm\_r=t22/(t21+t22)  
spec\_svm\_r=t11/(t11+t12)  
tochn\_svm\_r\*100

## [1] 86.36364

nevcl\_svm\_r\*100

## [1] 13.63636

chutl\_svm\_r\*100

## [1] 95.83333

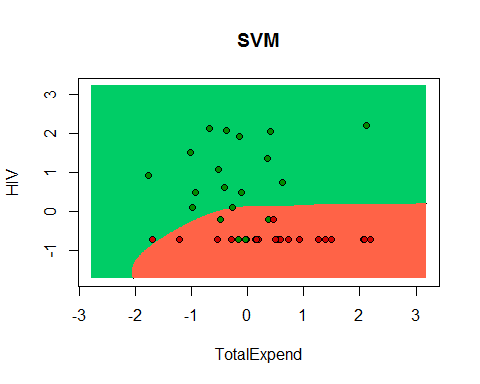
spec\_svm\_r\*100

## [1] 75

#Точність моделі = (23+15)/44=86,4%  
#Частка невірно класифікованих = (5+1)/44=13,6%  
#Чутливість моделі = 23/(1+23)=95,8%  
#специфічність моделі = 15/(5+15)=75%  
#Отже, модель більш чутлива до виявлення позитивних випадків

# Visualising the Test set results

library(ggplot2)  
set = f\_test[,c('TotalExpend','HIV','Life')]  
X1 = seq(min(set['TotalExpend']) - 1, max(set['TotalExpend']) + 1, by = 0.01)  
X2 = seq(min(set['HIV']) - 1, max(set['HIV']) + 1, by = 0.01)  
grid\_set = expand.grid(X1, X2)  
colnames(grid\_set) = c('TotalExpend', 'HIV')  
p\_grid = predict(class\_svm\_r, grid\_set)  
y\_grid <- ifelse(p\_grid > 0.5, 1, 0)  
plot(set[, -3],  
 main = 'SVM',  
 xlab = 'TotalExpend', ylab = 'HIV',  
 xlim = range(X1), ylim = range(X2))  
contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE)  
points(grid\_set, pch = '.', col = ifelse(y\_grid == 1, 'tomato', 'springgreen3'))  
points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'red3', 'green4'))



#Червоним позначені країни, тривалість життя в яких нижче за середню, зеленим позначені країни, тривалість життя в яких вище за середню. Червона зона - скоріш за все країни, тривалість життя в яких нижче за середню. У моделі нелінійний варіант розподільної кривої

## Naive Bayes

# Fitting

library(e1071)  
f\_train$Life <- as.factor(f\_train$Life)  
f\_test$Life <- as.factor(f\_test$Life)  
class\_nb = naiveBayes(Life ~ TotalExpend + HIV, data = f\_train)

# Predicting

y <- predict(class\_nb, f\_test[, c('TotalExpend','HIV')])

# Confusion Matrix

cm = table(f\_test[, 'Life'], y)  
print(cm)

## y  
## 0 1  
## 0 15 5  
## 1 0 24

t11=cm[1]  
t21=cm[2]  
t12=cm[3]  
t22=cm[4]  
sum=t11+t12+t21+t22  
tochn\_nb=(t11+t22)/sum  
nevcl\_nb=(t12+t21)/sum  
chutl\_nb=t22/(t21+t22)  
spec\_nb=t11/(t11+t12)  
tochn\_nb\*100

## [1] 88.63636

nevcl\_nb\*100

## [1] 11.36364

chutl\_nb\*100

## [1] 100

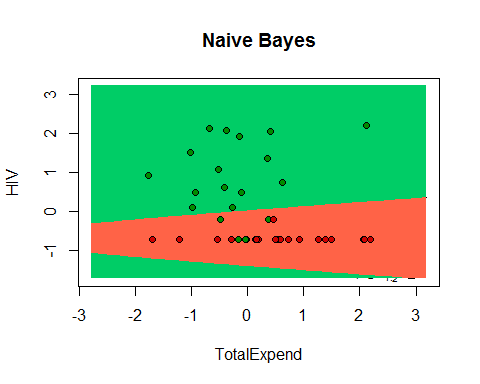
spec\_nb\*100

## [1] 75

#Точність моделі = (24+15)/44=88,6%  
#Частка невірно класифікованих = (5+0)/44=11,4%  
#Чутливість моделі = 24/(0+24)=100%  
#специфічність моделі = 15/(5+15)=75%  
#Отже, модель більш чутлива до виявлення позитивних випадків

# Visualising the Test set results

library(ggplot2)  
set = f\_test[,c('TotalExpend','HIV','Life')]  
X1 = seq(min(set['TotalExpend']) - 1, max(set['TotalExpend']) + 1, by = 0.01)  
X2 = seq(min(set['HIV']) - 1, max(set['HIV']) + 1, by = 0.01)  
grid\_set = expand.grid(X1, X2)  
colnames(grid\_set) = c('TotalExpend', 'HIV')  
y\_grid = predict(class\_nb, grid\_set)  
plot(set[, -3],  
 main = 'Naive Bayes',  
 xlab = 'TotalExpend', ylab = 'HIV',  
 xlim = range(X1), ylim = range(X2))  
contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE)  
points(grid\_set, pch = '.', col = ifelse(y\_grid == 1, 'tomato', 'springgreen3'))  
points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'red3', 'green4'))



#Червоним позначені країни, тривалість життя в яких нижче за середню, зеленим позначені країни, тривалість життя в яких вище за середню. Червона зона - скоріш за все країни, тривалість життя в яких нижче за середню. У моделі нелінійний варіант розподільної кривої. Гарно видно, що модель дуже близька до реальності.

## Classification Tree with scaling

# Fitting

library(rpart)  
f\_train$Life <- as.factor(f\_train$Life)  
f\_test$Life <- as.factor(f\_test$Life)  
class\_dt\_sc = rpart(Life ~ ., data = f\_train)

# Predicting

y <- predict(class\_dt\_sc, f\_test[-1], type = 'class')

# Confusion Matrix

cm = table(f\_test[, 'Life'], y)  
print(cm)

## y  
## 0 1  
## 0 18 2  
## 1 1 23

t11=cm[1]  
t21=cm[2]  
t12=cm[3]  
t22=cm[4]  
sum=t11+t12+t21+t22  
tochn\_dt\_sc=(t11+t22)/sum  
nevcl\_dt\_sc=(t12+t21)/sum  
chutl\_dt\_sc=t22/(t21+t22)  
spec\_dt\_sc=t11/(t11+t12)  
tochn\_dt\_sc\*100

## [1] 93.18182

nevcl\_dt\_sc\*100

## [1] 6.818182

chutl\_dt\_sc\*100

## [1] 95.83333

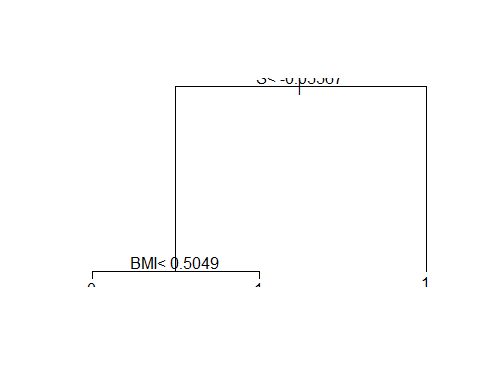
spec\_dt\_sc\*100

## [1] 90

#Точність моделі = (23+18)/44=93,2%  
#Частка невірно класифікованих = (1+2)/44=6,8%  
#Чутливість моделі = 23/(1+23)=95,8%  
#специфічність моделі = 18/(2+18)=90%  
#Отже, модель більш чутлива до виявлення позитивних випадків

# Plotting the tree

library(rpart.plot)  
plot(class\_dt\_sc)  
text(class\_dt\_sc)



#Візуалізація дозволяє проаналізувати логіку побудови дерева.На шкальованих даних інтерпритація результату є досить умовною, але країни, S яких не більше за -0,56, у яких ВМІ не більше 0,51 скоріш за все мають тривалість життя населення нижче, за середню.

# Fitting 2 factors

class\_ct\_sc = rpart(Life ~ TotalExpend + HIV, data = f\_train)

# Predicting

y <- predict(class\_ct\_sc, f\_test[, c('TotalExpend','HIV')], type = 'class')

# Confusion Matrix

cm = table(f\_test[, 'Life'], y)  
print(cm)

## y  
## 0 1  
## 0 14 6  
## 1 1 23

t11=cm[1]  
t21=cm[2]  
t12=cm[3]  
t22=cm[4]  
sum=t11+t12+t21+t22  
tochn\_ct\_sc=(t11+t22)/sum  
nevcl\_ct\_sc=(t12+t21)/sum  
chutl\_ct\_sc=t22/(t21+t22)  
spec\_ct\_sc=t11/(t11+t12)  
tochn\_ct\_sc\*100

## [1] 84.09091

nevcl\_ct\_sc\*100

## [1] 15.90909

chutl\_ct\_sc\*100

## [1] 95.83333

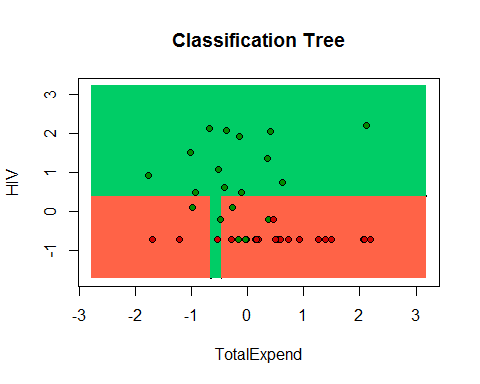
spec\_ct\_sc\*100

## [1] 70

#Точність моделі = (23+14)/44=84,1%  
#Частка невірно класифікованих = (1+6)/44=15,9%  
#Чутливість моделі = 23/(1+23)=95,8%  
#специфічність моделі = 14/(14+6)=70%  
#Отже, модель більш чутлива до виявлення позитивних випадків

# Visualising the Test set results

library(ggplot2)  
set = f\_test[,c('TotalExpend','HIV','Life')]  
X1 = seq(min(set['TotalExpend']) - 1, max(set['TotalExpend']) + 1, by = 0.01)  
X2 = seq(min(set['HIV']) - 1, max(set['HIV']) + 1, by = 0.01)  
grid\_set = expand.grid(X1, X2)  
colnames(grid\_set) = c('TotalExpend', 'HIV')  
y\_grid = predict(class\_ct\_sc, grid\_set, type = 'class')  
plot(set[, -3],  
 main = 'Classification Tree',  
 xlab = 'TotalExpend', ylab = 'HIV',  
 xlim = range(X1), ylim = range(X2))  
contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE)  
points(grid\_set, pch = '.', col = ifelse(y\_grid == 1, 'tomato', 'springgreen3'))  
points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'red3', 'green4'))



#Червоним позначені країни, тривалість життя в яких нижче за середню, зеленим позначені країни, тривалість життя в яких вище за середню. Червона зона - скоріш за все країни, тривалість життя в яких нижче за середню. У моделі лінійний варіант розподільної кривої

## Fitting Random Forest Classification to the Training set with scaling

library(randomForest)

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':  
##   
## margin

## The following object is masked from 'package:psych':  
##   
## outlier

## The following object is masked from 'package:dplyr':  
##   
## combine

set.seed(123)  
class\_rf\_sc = randomForest(Life ~ TotalExpend + HIV, data = f\_train, ntree = 5)

# Predicting

y <- predict(class\_rf\_sc, f\_test[, c('TotalExpend','HIV')])

# Confusion Matrix

cm = table(f\_test[, 'Life'], y)  
print(cm)

## y  
## 0 1  
## 0 14 6  
## 1 3 21

t11=cm[1]  
t21=cm[2]  
t12=cm[3]  
t22=cm[4]  
sum=t11+t12+t21+t22  
tochn\_rf\_sc=(t11+t22)/sum  
nevcl\_rf\_sc=(t12+t21)/sum  
chutl\_rf\_sc=t22/(t21+t22)  
spec\_rf\_sc=t11/(t11+t12)  
tochn\_rf\_sc\*100

## [1] 79.54545

nevcl\_rf\_sc\*100

## [1] 20.45455

chutl\_rf\_sc\*100

## [1] 87.5

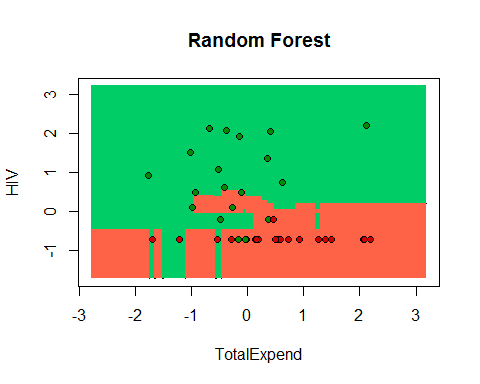
spec\_rf\_sc\*100

## [1] 70

#Точність моделі = (21+14)/44=79,5%  
#Частка невірно класифікованих = (3+6)/44=20,5%  
#Чутливість моделі = 21/(3+21)=87,5%  
#специфічність моделі = 14/(6+14)=70%  
#Отже, модель більш чутлива до виявлення позитивних випадків

# Visualising the Test set results

set = f\_test[,c('TotalExpend','HIV','Life')]  
X1 = seq(min(set['TotalExpend']) - 1, max(set['TotalExpend']) + 1, by = 0.01)  
X2 = seq(min(set['HIV']) - 1, max(set['HIV']) + 1, by = 0.01)  
grid\_set = expand.grid(X1, X2)  
colnames(grid\_set) = c('TotalExpend', 'HIV')  
y\_grid = predict(class\_rf\_sc, grid\_set)  
plot(set[, -3],  
 main = 'Random Forest',  
 xlab = 'TotalExpend', ylab = 'HIV',  
 xlim = range(X1), ylim = range(X2))  
contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE)  
points(grid\_set, pch = '.', col = ifelse(y\_grid == 1, 'tomato', 'springgreen3'))  
points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'red3', 'green4'))



#Червоним позначені країни, тривалість життя в яких нижче за середню, зеленим позначені країни, тривалість життя в яких вище за середню. Червона зона - скоріш за все країни, тривалість життя в яких нижче за середню. У моделі нелінійний варіант розподільної кривої

## Classification Tree without scaling

set.seed(123)  
library(caTools)  
split = sample.split(f\_fill$Life, SplitRatio = 2/3)  
f\_train\_ws\_wl = subset(f\_fill, split == TRUE)  
f\_test\_ws\_wl = subset(f\_fill, split == FALSE)  
f\_test <- f\_test\_ws\_wl  
f\_train <- f\_train\_ws\_wl

# Fitting

library(rpart)  
f\_train$Life<- as.factor(f\_train$Life)  
f\_test$Life <- as.factor(f\_test$Life)  
class\_dt = rpart(Life ~ ., data = f\_train)

# Predicting

y <- predict(class\_dt, f\_test[-1], type = 'class')

# Confusion Matrix

cm = table(f\_test[, 'Life'], y)  
print(cm)

## y  
## 0 1  
## 0 18 2  
## 1 1 23

t11=cm[1]  
t21=cm[2]  
t12=cm[3]  
t22=cm[4]  
sum=t11+t12+t21+t22  
tochn\_dt=(t11+t22)/sum  
nevcl\_dt=(t12+t21)/sum  
chutl\_dt=t22/(t21+t22)  
spec\_dt=t11/(t11+t12)  
tochn\_dt\*100

## [1] 93.18182

nevcl\_dt\*100

## [1] 6.818182

chutl\_dt\*100

## [1] 95.83333

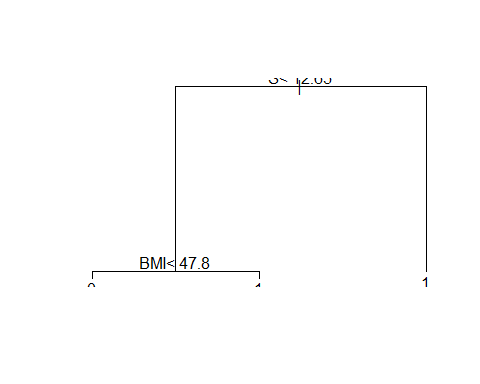
spec\_dt\*100

## [1] 90

#Точність моделі = (23+18)/44=93,2%  
#Частка невірно класифікованих = (1+2)/44=6,8%  
#Чутливість моделі = 23/(1+23)=95,8%  
#специфічність моделі = 18/(2+18)=90%  
#Отже, модель більш чутлива до виявлення позитивних випадків

# Plotting the tree

library(rpart.plot)  
plot(class\_dt)  
text(class\_dt)



#Візуалізація дозволяє проаналізувати логіку побудови дерева.Країни, S яких не більше за 12,65, у яких ВМІ не більше 47,8 скоріш за все мають тривалість життя населення нижче, за середню.

# Fitting 2 factors

class\_ct = rpart(Life ~ TotalExpend + HIV, data = f\_train)

# Predicting

y <- predict(class\_ct, f\_test[, c('TotalExpend','HIV')], type = 'class')

# Confusion Matrix

cm = table(f\_test[, 'Life'], y)  
print(cm)

## y  
## 0 1  
## 0 14 6  
## 1 1 23

t11=cm[1]  
t21=cm[2]  
t12=cm[3]  
t22=cm[4]  
sum=t11+t12+t21+t22  
tochn\_ct=(t11+t22)/sum  
nevcl\_ct=(t12+t21)/sum  
chutl\_ct=t22/(t21+t22)  
spec\_ct=t11/(t11+t12)  
tochn\_ct\*100

## [1] 84.09091

nevcl\_ct\*100

## [1] 15.90909

chutl\_ct\*100

## [1] 95.83333

spec\_ct\*100

## [1] 70

#Точність моделі = (23+14)/44=84,1%  
#Частка невірно класифікованих = (1+6)/44=15,9%  
#Чутливість моделі = 23/(1+23)=95,8%  
#специфічність моделі = 14/(6+14)=70%  
#Отже, модель більш чутлива до виявлення позитивних випадків

# Visualising the Test set results #Вилітає RStudio!

library(ggplot2) set = f\_test[,c(‘AST’,‘ALB’,‘Category’)] X1 = seq(min(set[‘AST’]) - 1, max(set[‘AST’]) + 1, by = 0.01) X2 = seq(min(set[‘ALB’]) - 1, max(set[‘ALB’]) + 1, by = 0.01) grid\_set = expand.grid(X1, X2) colnames(grid\_set) = c(‘AST’, ‘ALB’) y\_grid = predict(class\_ct, grid\_set, type = ‘class’) plot(set[, -3], main = ‘Classification Tree’, xlab = ‘AST’, ylab = ‘ALB’, xlim = range(X1), ylim = range(X2)) contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE) points(grid\_set, pch = ‘.’, col = ifelse(y\_grid == 1, ‘tomato’, ‘springgreen3’)) points(set, pch = 21, bg = ifelse(set[, 3] == 1, ‘red3’, ‘green4’))

## Fitting Random Forest Classification to the Training set without scaling

library(randomForest)  
set.seed(123)  
class\_rf = randomForest(Life ~ TotalExpend + HIV, data = f\_train, ntree = 5)

# Predicting

y <- predict(class\_rf, f\_test[, c('TotalExpend','HIV')])

# Confusion Matrix

cm = table(f\_test[, 'Life'], y)  
print(cm)

## y  
## 0 1  
## 0 14 6  
## 1 1 23

t11=cm[1]  
t21=cm[2]  
t12=cm[3]  
t22=cm[4]  
sum=t11+t12+t21+t22  
tochn\_rf=(t11+t22)/sum  
nevcl\_rf=(t12+t21)/sum  
chutl\_rf=t22/(t21+t22)  
spec\_rf=t11/(t11+t12)  
tochn\_rf\*100

## [1] 84.09091

nevcl\_rf\*100

## [1] 15.90909

chutl\_rf\*100

## [1] 95.83333

spec\_rf\*100

## [1] 70

#Точність моделі = (23+14)/44=84,1%  
#Частка невірно класифікованих = (1+6)/44=15,9%  
#Чутливість моделі = 23/(1+23)=95,8%  
#специфічність моделі = 14/(6+14)=70%  
#Отже, модель більш чутлива до виявлення позитивних випадків

# Visualising the Test set results #Вилітає RStudio!

set = f\_test[,c(‘AST’,‘ALB’,‘Category’)] X1 = seq(min(set[‘AST’]) - 1, max(set[‘AST’]) + 1, by = 0.01) X2 = seq(min(set[‘ALB’]) - 1, max(set[‘ALB’]) + 1, by = 0.01) grid\_set = expand.grid(X1, X2) colnames(grid\_set) = c(‘AST’, ‘ALB’) y\_grid = predict(class\_rf, grid\_set) plot(set[, -3], main = ‘Random Forest’, xlab = ‘AST’, ylab = ‘ALB’, xlim = range(X1), ylim = range(X2)) contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE) points(grid\_set, pch = ‘.’, col = ifelse(y\_grid == 1, ‘tomato’, ‘springgreen3’)) points(set, pch = 21, bg = ifelse(set[, 3] == 1, ‘red3’, ‘green4’))

## NEUORAL NETWORKS FOR CLASSIFICATION

library(neuralnet)

##   
## Attaching package: 'neuralnet'

## The following object is masked from 'package:ROCR':  
##   
## prediction

## The following object is masked from 'package:dplyr':  
##   
## compute

# fit neural network  
nn = neuralnet(Life ~ TotalExpend + HIV, f\_train\_ws, hidden = , linear.output = T)  
# plot neural network  
plot(nn)  
#Побудовано двошарову нейронну мережу

# Fitting the NN

library(nnet)  
library(scales)

##   
## Attaching package: 'scales'

## The following objects are masked from 'package:psych':  
##   
## alpha, rescale

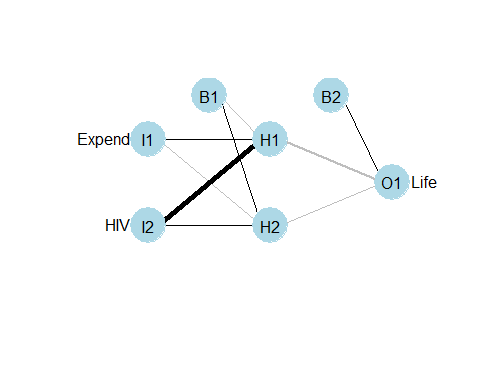
set.seed(11)  
ff\_cl <- nnet(data = f\_train\_ws, Life ~ TotalExpend + HIV, size = 2, maxit = 1000)  
library(graphics)  
source(file = 'plot.nnet.R')  
plot.nnet(ff\_cl)

## Loading required package: reshape

##   
## Attaching package: 'reshape'

## The following object is masked from 'package:class':  
##   
## condense

## The following object is masked from 'package:dplyr':  
##   
## rename



#Візуалізовано двошарову нейронну мережу

# Compare models

#Точності моделей:  
tochn\_opt

## [1] 0.9090909091

tochn\_knn

## [1] 0.8409090909

tochn\_svm

## [1] 0.7954545455

tochn\_svm\_r

## [1] 0.8636363636

tochn\_nb

## [1] 0.8863636364

tochn\_dt\_sc

## [1] 0.9318181818

tochn\_ct\_sc

## [1] 0.8409090909

tochn\_rf\_sc

## [1] 0.7954545455

tochn\_dt

## [1] 0.9318181818

tochn\_ct

## [1] 0.8409090909

tochn\_rf

## [1] 0.8409090909

#Частки невірно класифікованих:  
nevcl\_opt

## [1] 0.09090909091

nevcl\_knn

## [1] 0.1590909091

nevcl\_svm

## [1] 0.2045454545

nevcl\_svm\_r

## [1] 0.1363636364

nevcl\_nb

## [1] 0.1136363636

nevcl\_dt\_sc

## [1] 0.06818181818

nevcl\_ct\_sc

## [1] 0.1590909091

nevcl\_rf\_sc

## [1] 0.2045454545

nevcl\_dt

## [1] 0.06818181818

nevcl\_ct

## [1] 0.1590909091

nevcl\_rf

## [1] 0.1590909091

#Чутливості моделей:  
chutl\_opt

## [1] 1

chutl\_knn

## [1] 0.9583333333

chutl\_svm

## [1] 1

chutl\_svm\_r

## [1] 0.9583333333

chutl\_nb

## [1] 1

chutl\_dt\_sc

## [1] 0.9583333333

chutl\_ct\_sc

## [1] 0.9583333333

chutl\_rf\_sc

## [1] 0.875

chutl\_dt

## [1] 0.9583333333

chutl\_ct

## [1] 0.9583333333

chutl\_rf

## [1] 0.9583333333

#Специфічності моделей:  
spec\_opt

## [1] 0.8

spec\_knn

## [1] 0.7

spec\_svm

## [1] 0.55

spec\_svm\_r

## [1] 0.75

spec\_nb

## [1] 0.75

spec\_dt\_sc

## [1] 0.9

spec\_ct\_sc

## [1] 0.7

spec\_rf\_sc

## [1] 0.7

spec\_dt

## [1] 0.9

spec\_ct

## [1] 0.7

spec\_rf

## [1] 0.7

*#Зібрано характеристики матриць неточностей усіх розглянутих моделей класифікації*

# Висновки

У ході роботи з класифікації побудовано логістичну регресію (модель Бенчмарка з усіма змінними та оптимізовану), нейронну мережу, SVM, наївний Байєсовський класифікатор, дерево рішень та випадковий ліс (останні дві моделі побудовано і на вихідних, і на прошкальованих даних, але візуалізовано лише на прошкальованих (оскільки не вистачає пам’яті).

Розглянемо характеристики матриць неточностей моделей.

Найнижча точність у лінійної SVM та випадкового лісу на шкальованих даних (79,5%), найвища – 93,2% у обох дерев рішень.

Відповідно, найвища частка невірно класифікованих у лінійної SVM та випадкового лісу на шкальованих даних (20,5%), а найнижча – у обох дерев рішень 6,8%.

Найнижча чутливість у випадкового лісу на шкальованих даних 87,5%%, найвища – 100% у оптимізованій моделі Бенчмарка, SVM, наївному Байєсовському класифікаторі, у решти 95,8%. Отже, оптимізована модель Бенчмарка, SVM та наївний Байєсовський класифікатор є перенавченими.

Найнижча специфічність у SVM (55%), найвища – у обох дерев рішень (90%).

Серед розглянутих, найякіснішими моделями є дерева рішень на вихідних даних та на прошкальованих даних. Крім того, їх матриці неточностей най збалансованіші (невірна класифікація: 21=1 та 12=2).

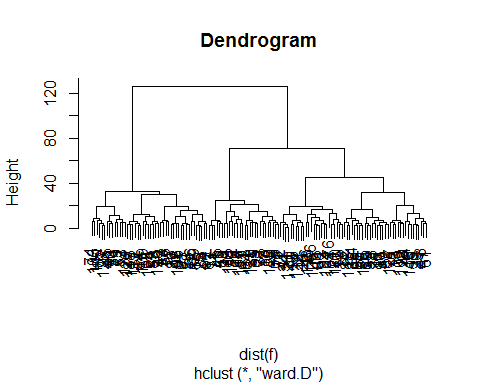
Оскільки дерево рішень на нешкальованих даний має більш точну інтерпретацію, то саме цю модель обрано.

## 

Clusterization

## Hierarchical clustering

model\_hc <- hclust(dist(f), method = "ward.D" )  
plot(model\_hc, main = paste('Dendrogram'))



#Побудовано дендрограму на основі навчальної вибірки з використанням методу ward.D

# Fitting HC to the dataset

y\_hc <- cutree(model\_hc, k = 2)  
#cluster cores  
aggregate(f,by=list(y\_hc),FUN=mean)

## Group.1 Life AM ID HepB Measles  
## 1 1 0.02173913043 5.319493325 3.349902527 4.113626945 6.000017906  
## 2 2 0.82558139535 4.395323117 2.129283978 4.377535095 6.256212703  
## BMI U5D Polio TotalExpend Diphth HIV  
## 1 2.993695809 3.592708540 4.215135476 5.265217391 4.165133814 -0.184827901  
## 2 3.743724696 2.154706079 4.427747055 6.538213781 4.423219388 -1.985196894  
## GDP P TH119 TH59 S  
## 1 6.058847311 15.07570857 1.9100150917 0.5086956522 9.967391304  
## 2 8.353434808 13.59811626 0.7056514433 0.7534883721 14.120930233

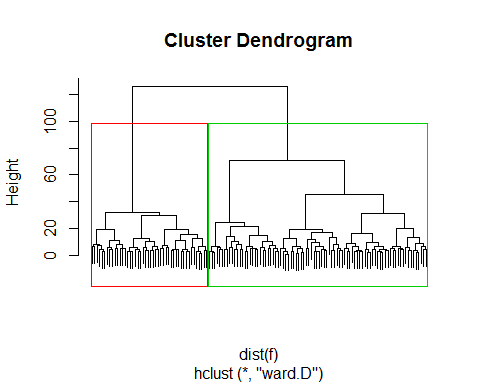
#Cluster stat  
f$hc <- y\_hc  
table(f$hc)

##   
## 1 2   
## 46 86

#На основі аналізу дендрограми виділено 2 кластера. HepB, Measles, Polio, Diphth та TH59 фактично не впливають на розподіл. До 1 потрапили країни в яких: коротка тривалість життя, трохи вищі AM, ID, U5D, HIV, нижчі BMI, TotalExpend, GDP, значно вищі Р та TH119 I значно нижчий S. До 2 потрапили країни в яких: довга тривалість життя, трохи нижчі AM, ID, U5D, HIV, вищі BMI, TotalExpend, GDP, значно нижчі Р та TH119 I значно вищий S.

# Plotting the dendrogram

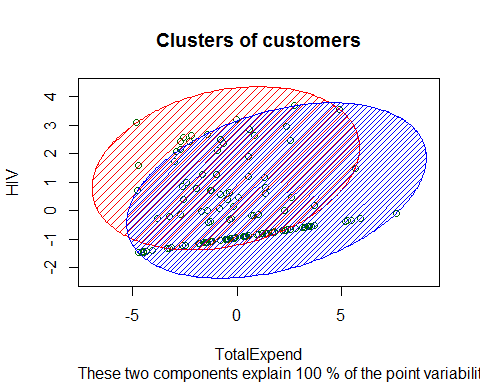
plot(model\_hc, cex = 0.7, labels = FALSE)  
rect.hclust(model\_hc, k = 2, border = 2:5)



#візуалізовано поділ на кластери на дендрограмі

# Visualising the clusters

library(cluster)  
clusplot(f[,c('TotalExpend','HIV')],  
 y\_hc,  
 lines = 0,  
 shade = TRUE,  
 color = TRUE,  
 labels= 0,  
 plotchar = FALSE,  
 span = TRUE,  
 main = paste('Clusters of customers'),  
 xlab = 'TotalExpend',  
 ylab = 'HIV')



#візуалізовано кластери на датасеті. Є накладання областей, тому є об'єкти, що знаходяться на межі.

## K-Means

# Historgram for each attribute

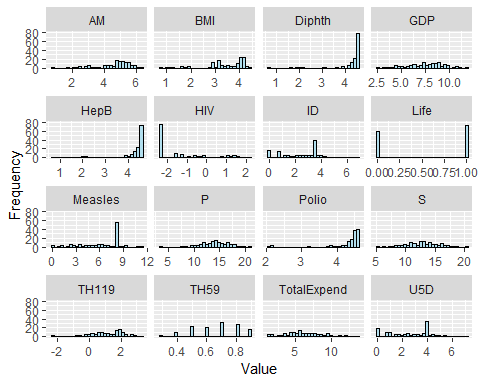
library(tidyr)

##   
## Attaching package: 'tidyr'

## The following objects are masked from 'package:reshape':  
##   
## expand, smiths

library(ggplot2)  
f %>%   
 gather(Attributes, value, 1:16) %>%   
 ggplot(aes(x=value)) +  
 geom\_histogram(fill = "lightblue2", color = "black") +   
 facet\_wrap(~Attributes, scales = "free\_x") +  
 labs(x = "Value", y = "Frequency")

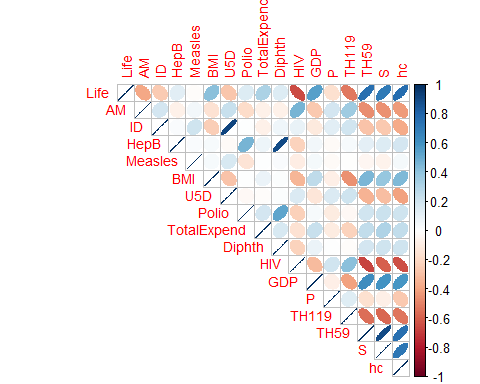
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

 # Correlation

library(corrplot)

## corrplot 0.84 loaded

corrplot(cor(f), type = "upper", method = "ellipse", tl.cex = 0.9)



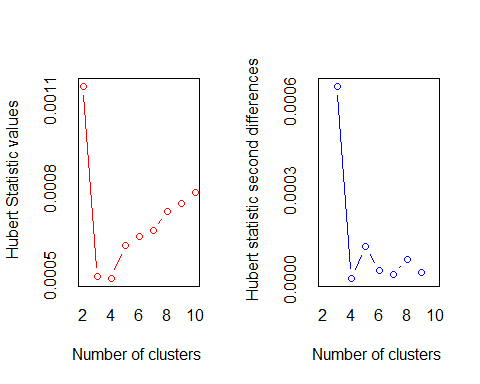
#Гарна кореляція з Life у HIV, TH119, TH59, S; з ID у U5D; з HepB у Diphth; з HIV у TH59; з TH59 у S.

# NbCLust

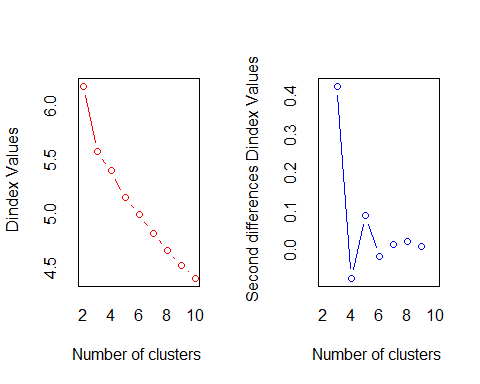
library(factoextra)

## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ

library(NbClust)  
res.nbclust <- NbClust(f, distance = "euclidean",  
 min.nc = 2, max.nc = 10,   
 method = "complete", index ="all")



## \*\*\* : The Hubert index is a graphical method of determining the number of clusters.  
## In the plot of Hubert index, we seek a significant knee that corresponds to a   
## significant increase of the value of the measure i.e the significant peak in Hubert  
## index second differences plot.   
##

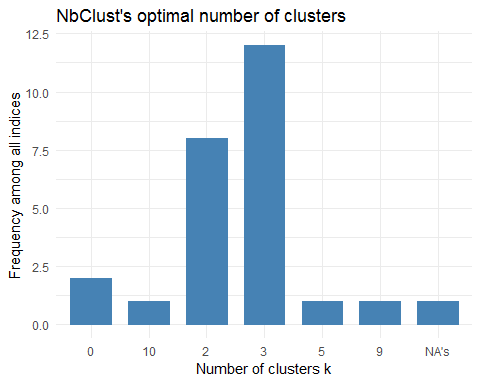


## \*\*\* : The D index is a graphical method of determining the number of clusters.   
## In the plot of D index, we seek a significant knee (the significant peak in Dindex  
## second differences plot) that corresponds to a significant increase of the value of  
## the measure.   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
## \* Among all indices:   
## \* 8 proposed 2 as the best number of clusters   
## \* 12 proposed 3 as the best number of clusters   
## \* 1 proposed 5 as the best number of clusters   
## \* 1 proposed 9 as the best number of clusters   
## \* 1 proposed 10 as the best number of clusters   
##   
## \*\*\*\*\* Conclusion \*\*\*\*\*   
##   
## \* According to the majority rule, the best number of clusters is 3   
##   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

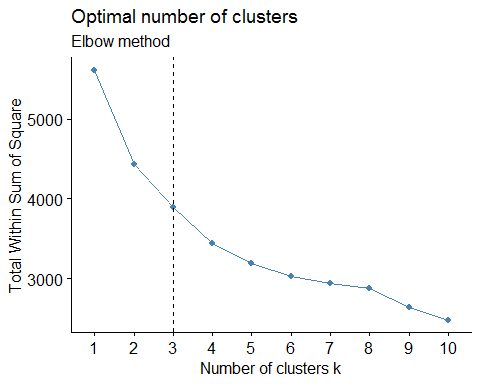
**#найбільше методів обрали 3 кластери, трохи менше методів – 2. Мною обрано 2 кластери.**

fviz\_nbclust(res.nbclust) + theme\_minimal() + ggtitle("NbClust's optimal number of clusters")

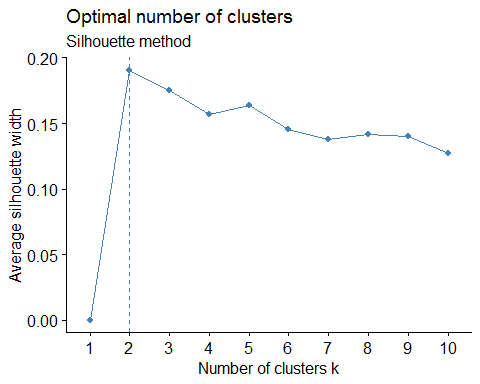
## Among all indices:   
## ===================  
## \* 2 proposed 0 as the best number of clusters  
## \* 8 proposed 2 as the best number of clusters  
## \* 12 proposed 3 as the best number of clusters  
## \* 1 proposed 5 as the best number of clusters  
## \* 1 proposed 9 as the best number of clusters  
## \* 1 proposed 10 as the best number of clusters  
## \* 1 proposed NA's as the best number of clusters  
##   
## Conclusion  
## =========================  
## \* According to the majority rule, the best number of clusters is 3 .



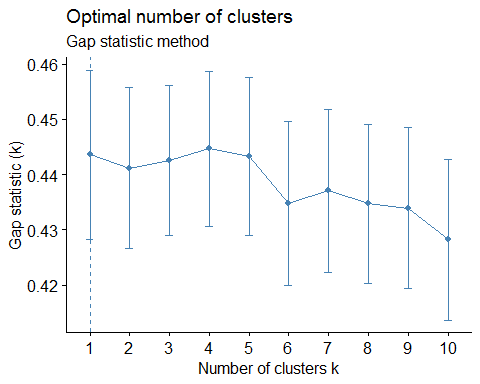
# Elbow method  
# The sum of squares at each number of clusters is calculated and graphed, and the user looks for a change of slope from steep to shallow (an elbow) to determine the optimal number of clusters.  
fviz\_nbclust(f, kmeans, method = "wss") +  
 geom\_vline(xintercept = 2, linetype = 2)+  
 labs(subtitle = "Elbow method")



# Silhouette method  
# The optimal number of clusters k is the one that maximize the average silhouette over a range of possible values for k.  
fviz\_nbclust(f, kmeans, method = "silhouette")+  
 labs(subtitle = "Silhouette method")



# Gap statistic  
# The gap statistic compares the total within intra-cluster variation for different values of k with their expected values under null reference distribution of the data. The estimate of the optimal clusters will be value that maximize the gap statistic. This means that the clustering structure is far away from the random uniform distribution of points.  
fviz\_nbclust(f, kmeans, nstart = 25, method = "gap\_stat", nboot = 50)+  
 labs(subtitle = "Gap statistic method")

 # Clustree

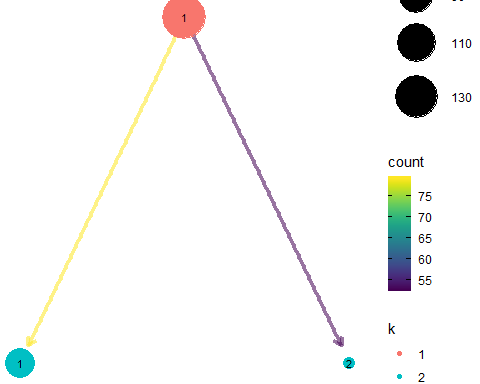
library(clustree)

## Loading required package: ggraph

library(dplyr)  
library(ggraph)  
tmp <- NULL  
for (k in 1:2){  
 tmp[k] <- kmeans(f, k, nstart = 30)  
}

## Warning in tmp[k] <- kmeans(f, k, nstart = 30): число единиц для замены не  
## является произведением длины замены  
  
## Warning in tmp[k] <- kmeans(f, k, nstart = 30): число единиц для замены не  
## является произведением длины замены

df <- data.frame(tmp)  
# add a prefix to the column names  
colnames(df) <- seq(1:2  
 )  
colnames(df) <- paste0("k",colnames(df))  
# get individual PCA  
df.pca <- prcomp(df, center = TRUE, scale. = FALSE)  
ind.coord <- df.pca$x  
ind.coord <- ind.coord[,1:2]  
df <- bind\_cols(as.data.frame(df), as.data.frame(ind.coord))  
clustree(df, prefix = "k")

 # Fitting K-Means to the dataset

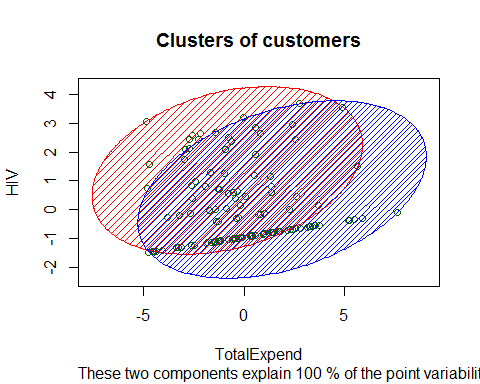
set.seed(29)  
model\_km = kmeans(f, 2)  
#cluster cores  
y\_km = model\_km$cluster  
aggregate(f,by=list(y\_km),FUN=mean)

## Group.1 Life AM ID HepB Measles  
## 1 1 0.03773584906 5.319527081 3.240356655 4.151612080 6.132780012  
## 2 2 0.88607594937 4.313411971 2.094620704 4.375435663 6.189845260  
## BMI U5D Polio TotalExpend Diphth  
## 1 3.063043267 3.401452071 4.239039544 5.112452830 4.198127916  
## 2 3.763658708 2.155599442 4.430549150 6.753498546 4.423952446  
## HIV GDP P TH119 TH59  
## 1 -0.3712931448 6.219743228 14.92787768 1.8508450350 0.5245283019  
## 2 -2.0196263247 8.448809857 13.56636804 0.6386320442 0.7645569620  
## S hc  
## 1 10.16603774 1.132075472  
## 2 14.35569620 2.000000000

# виділено 2 кластера. HepB, Measles, Polio, Diphth та TH59 фактично не впливають на розподіл. До 1 потрапили країни в яких: коротка тривалість життя, трохи вищі AM, ID, U5D, HIV, нижчі BMI, TotalExpend, GDP, значно вищі Р та TH119 I значно нижчий S. До 2 потрапили країни в яких: довга тривалість життя, трохи нижчі AM, ID, U5D, HIV, вищі BMI, TotalExpend, GDP, значно нижчі Р та TH119 I значно вищий S.

# Visualising the clusters

library(cluster)  
clusplot(f[,c('TotalExpend','HIV')],  
 y\_km,  
 lines = 0,  
 shade = TRUE,  
 color = TRUE,  
 labels= 0,  
 plotchar = FALSE,  
 span = TRUE,  
 main = paste('Clusters of customers'),  
 xlab = 'TotalExpend',  
 ylab = 'HIV')



#візуалізовано поділ. Видно, що є перетин.

# Comparing to HC

library(clusteval)  
cluster\_similarity(y\_hc,y\_km)

## [1] 0.8254189944

## Kohonen maps

f=f\_sc  
f\_matrix <- as.matrix(f)  
head (f)

## Life AM ID HepB Measles  
## 1 0 0.8806888244 1.1271834127 -0.3484149812 -0.06312162227  
## 2 1 -2.4759467559 0.7933779692 0.3986865356 0.71337319495  
## 3 1 -2.1725033690 0.4136176259 0.3479522722 0.71337319495  
## 4 0 1.1189847683 1.1565169303 -0.2966067983 1.21208975734  
## 5 1 0.0884443509 -0.2043588897 0.3306841672 -2.55757398841  
## 6 1 -2.0895932808 -1.5359011920 0.3132313731 -1.52536200763  
## BMI U5D Polio TotalExpend Diphth  
## 1 -0.6421584143 1.1193993029 -0.5580872217 0.87480127142 -0.4471498849  
## 2 0.7231276112 -1.5830966258 0.4448573800 -0.02155136946 0.3683732066  
## 3 0.7483601647 0.3450596149 0.3854089875 0.49677428809 0.3129925623  
## 4 -0.4000617423 1.2169419831 -0.2539392048 -1.02312801602 -0.3905969687  
## 5 0.8249731293 -0.2500190547 0.3240528377 -0.44634457753 0.2941429976  
## 6 0.6554100884 -1.5830966258 0.3854089875 -0.56715732478 0.2750918295  
## HIV GDP P TH119 TH59  
## 1 -0.7155143329 -0.6743649413 -0.6645919776 1.6216271680 -1.2082762544  
## 2 -0.7155143329 0.3978946647 -0.7143943530 -0.9296355587 0.8630544674  
## 3 -0.7155143329 -0.7341541515 1.2315954703 0.5379171294 0.1726108935  
## 4 1.5200407236 -0.8059370010 0.1705961390 0.8896186269 -1.2082762544  
## 5 -0.7155143329 0.9229214437 1.2689869956 -1.2904899752 0.8630544674  
## 6 -0.7155143329 0.3254680029 -1.6174360796 -0.4590225530 0.1726108935  
## S  
## 1 -1.00330449105  
## 2 0.49860454051  
## 3 0.57012401820  
## 4 -0.50266814720  
## 5 1.60715644475  
## 6 -0.03779154219

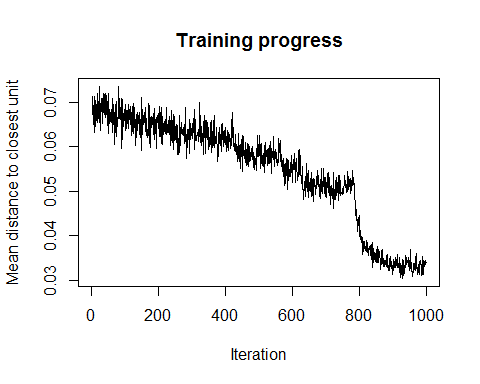
# Fitting the NN

set.seed(123)  
library(kohonen)

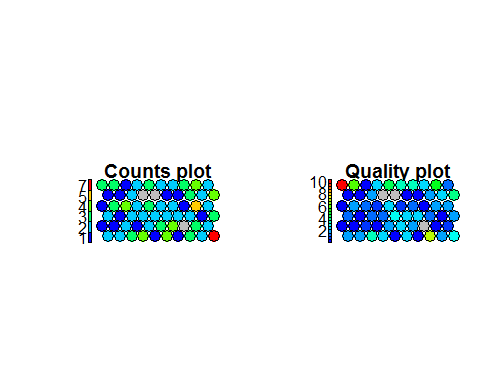
##   
## Attaching package: 'kohonen'

## The following object is masked from 'package:class':  
##   
## somgrid

som\_grid <- somgrid(xdim = 10, ydim = 6, topo = "hexagonal")   
som\_model <- som(f\_matrix, grid = som\_grid, rlen = 1000,  
 alpha = c(0.05,0.01), keep.data = TRUE)  
plot(som\_model, type = "changes")

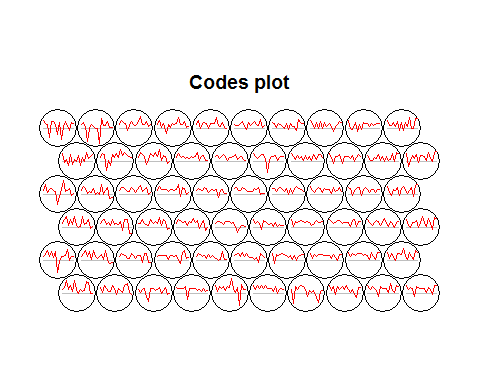
 # Visualization

#Palette  
coolBlueHotRed <- function(n, alpha = 1) {  
 rainbow(n, end = 4/6, alpha = alpha)[n:1]   
}  
par(mfrow = c(1, 2))  
#Number of objects at sells  
plot(som\_model, type = "counts", palette.name = coolBlueHotRed)  
#Distance to core  
plot(som\_model, type = "quality", palette.name = coolBlueHotRed)

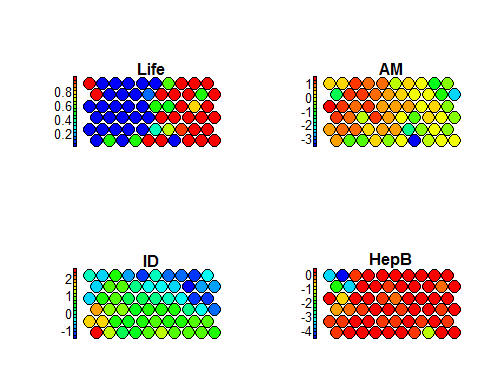


# Maps of the factors

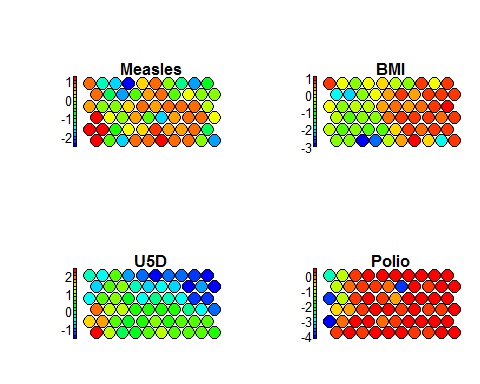
plot(som\_model, type = "codes")



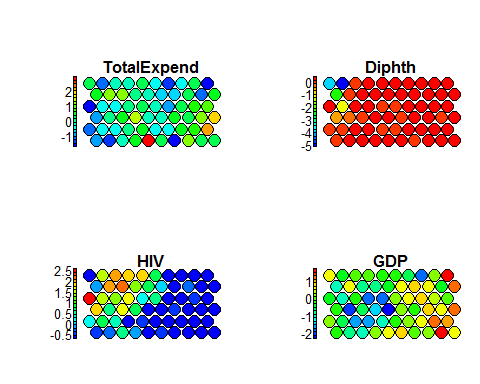
par(mfrow = c(2, 2))  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,1],   
 main = "Life",  
 palette.name = coolBlueHotRed)  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,2],  
 main = "AM",  
 palette.name = coolBlueHotRed)  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,3],   
 main = "ID",  
 palette.name = coolBlueHotRed)  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,4],   
 main = "HepB",  
 palette.name = coolBlueHotRed)



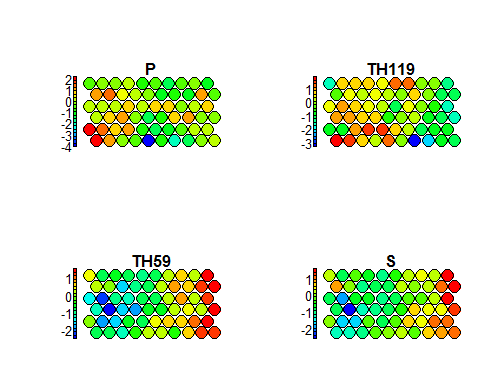
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,5],   
 main = "Measles",  
 palette.name = coolBlueHotRed)  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,6],   
 main = "BMI",  
 palette.name = coolBlueHotRed)  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,7],   
 main = "U5D",  
 palette.name = coolBlueHotRed)  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,8],   
 main = "Polio",  
 palette.name = coolBlueHotRed)



plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,9],   
 main = "TotalExpend",  
 palette.name = coolBlueHotRed)  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,10],   
 main = "Diphth",  
 palette.name = coolBlueHotRed)  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,11],   
 main = "HIV",  
 palette.name = coolBlueHotRed)  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,12],   
 main = "GDP",  
 palette.name = coolBlueHotRed)

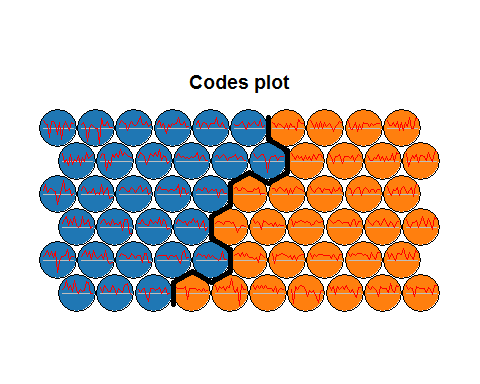


plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,13],   
 main = "P",  
 palette.name = coolBlueHotRed)  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,14],   
 main = "TH119",  
 palette.name = coolBlueHotRed)  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,15],   
 main = "TH59",  
 palette.name = coolBlueHotRed)  
plot(som\_model, type = "property",   
 property = som\_model$codes[[1]][,16],   
 main = "S",  
 palette.name = coolBlueHotRed)



# Clusters description

mydata <- as.matrix(som\_model$codes[[1]])  
#Use hierarchical clustering, k=4  
som\_cluster <- cutree(hclust(dist(mydata)), 2)  
#Palette  
pretty\_palette <- c("#1f77b4", '#ff7f0e', '#2ca02c',  
 '#d62728', '#9467bd', '#8c564b', '#e377c2')  
#Colored clusters  
plot(som\_model, type = "codes",   
 bgcol = pretty\_palette[som\_cluster])  
add.cluster.boundaries(som\_model, som\_cluster)



aggregate(mydata,by=list(som\_cluster),FUN=mean)

## Group.1 Life AM ID HepB  
## 1 1 0.1249828985 0.5452425679 0.40210587346 -0.4148610402  
## 2 2 0.8197600112 -0.3171562064 -0.05275277347 0.1820801833  
## Measles BMI U5D Polio TotalExpend  
## 1 -0.01822793881 -0.4430725319 0.3538413932 -0.5545998883 -0.1823053050  
## 2 0.08058300459 0.3618039132 -0.1399038245 0.3001741331 0.1939710195  
## Diphth HIV GDP P TH119  
## 1 -0.3781011809 0.8518599433 -0.6340046323 0.3846835396 0.5392745797  
## 2 0.2361355401 -0.6192223676 0.3172208346 -0.4561339418 -0.5192602235  
## TH59 S  
## 1 -0.9563048291 -0.7696867474  
## 2 0.5444575031 0.3956802799

Висновки

У ході роботи з кластеризації побудовано ієрархічну кластеризацію, модель за методом к-середніх та мапу Кохонена.

За допомогою усіх розглянутих методів виконано поділ прошкальованих даних на 2 кластери