



NB_MAX_CLASSES 20

```

○────────────────────────────────○ ↓ realClasses[], estimatedClasses[], nbTests
| displayConfusionMatrice |
○────────────────────────────────○

*
○────────────────────────────────○ ↓ realClasses[], estimatedClasses[], nbTests
| maxMovement |
○────────────────────────────────○ ↓ maxClasses
○────────────────────────────────○ ↓ realClasses[], estimatedClasses[], nbTestsn,
maxClasses
| createMatrix |
○────────────────────────────────○ ↓ matrix[]

○────────────────────────────────○ ↓ matrix[0]], NULL
| displayRowMatrix |
○────────────────────────────────○
i = 1
while (i < NB_MAX_CLASSES AND i < nbClasses)
  ○────────────────────────────────○ ↓ matrix[i]], i+1
  | displayRowMatrix |
  ○────────────────────────────────○

```

```

○──────────○ ↓  realClasses[], estimatedClasses[], nbTests, nbClasses
| createMatrix |
○──────────○ ↓  matrix[],

```

```

*
matrix[][] = new ARRAY( nbClasses)( nbClasses)
// titre mouvement
iTitre = 0
┌ while (iTitre < nbClasses)
│   matrix[0][iTitre] = iTitre +1
└
// init matrix full à 0
i = 1
┌ while ( i < nbClasses)
│   matrix[i][0] = i
│   j = 1
│   ┌ while (j < nbClasses)
│   │   matrix[i][j] = 0
│   └
└
i = 0
┌ while (i < nbTest)
│   matrix[realClasses[i] -1][estimatedClasses[i] -1]++
└

```

```

○──────────○ ↓  realClasses[], estimatedClasses[], nbTests
| maxMouvement |
○──────────○ ↓  maxClasses

```

```

*
maxClasses = 0
i = 0
┌ while (i < nbTests)
│   ┌ if ( realClasses[i] > maxClasses OR estimatedClasses[i] > maxClasses
│   │   )
│   │   || maxClasses =  realClasses[i] > estimatedClasses[i] ? realClasses[i] :
│   │   estimatedClasses[i]
│   │   ┌
│   │   │ i++
│   │   └
└

```

```

○──────────○ ↓  ligne[], mouvement
| displayRowMatrix |
○──────────○

```

```

*
// sort le type du mouvement
sortir mouvement +'\t'
int i = 0
┌ while (i < NB_MAX_CLASSES AND ligne[i] ≠ NULL)
│   // sort le nombres d'hestimation
│   sortir ligne[i] + '\t'
└

```

.=====.

```

○──────────○ ↓ realClasses[],estimatedClasses[],nbTests
| displayAccuracy |
○──────────○

*
NB_CLASSES = 6

// boucler sur les deux tableaux en incrémentant le total et
potentiellement totalCorrect
○──────────○ ↓ realClasses[],estimatedClasses[],nbTest
| countTotCorrect |
○──────────○ ↓ totalCorrect, total

// calculer et sortir l'accuracy
accuracy = totalCorrect/total100
sortir "L'accuracy est de " + accuracy + "%"

○──────────○ ↓ realClasses[],estimatedClasses[],nbTest
| countTotCorrect |
○──────────○ ↓ totalCorrect, total
---
totalCorrect = 0
total = 0
i = 0
while (i < nbTest)
  if (realClasses[i] == estimatedClasses[i])
    totalCorrect++
  total++
  i++

.=====
○──────────○ ↓ realClasses[],estimatedClasses[],nbTests)
| displayResultsByClass |
○──────────○

*
NB_CLASSES = 6

// creation d'un tableau de structures de NB_CLASSES cellules (ici 6)
classes[] = new ARRAY (NB_CLASSES)
movement { number
.         { totalCorrect = 0
.         { total = 0

iClasses = 0
while (iClasses < NB_CLASSES)
  classes[iClasses].number = iClasses+1
  iClasses++

// parcourir et comparer les deux tableaux realClasses et
estimatedClasses
○──────────○ ↓ realClasses[],estimatedClasses[], classes, nbTest
| countResult |
○──────────○ ↓ classes[]

// afficher resultByClass
○──────────○ ↓ classes, NB_CLASSES
| displayRow |
○──────────○

```

```

○──────────○ ↓ realClasses[],estimatedClasses[], classes, nbTest
| countResult |
○──────────○ ↓ classes[]

```

```

*
i = 0
while(i < nbTest)
    iClasses = realClasses[i]-1
    if (realClasses[i] == estimatedClasses[i])
        classes[iClasses].totalCorrect++
    classes[iClasses].total++
    i++

```

```

○──────────○ ↓ classes, NB_CLASSES
| displayRow |
○──────────○

```

```

*
sortir "classe | bien classes | total | Pourcentage\n"
iClasses = 0
while (iClasses < NB_CLASSES)
    if (classes[iClasses].total ≠ 0)
        sortir classes[iClasses].number + " | "
        sortir classes[iClasses].totalCorrect + " | "
        sortir classes[iClasses].total + " | "
        pourcentage = classes[iClasses].totalCorrect /
                        classes[iClasses].total*100
        sortir pourcentage + " %\n"
    i++

```

```

○──────────○ ↓ realClasses[],estimatedClasses[],nbTests
| displayAccuracy |
○──────────○

```

```

*
NB_CLASSES = 6

// boucler sur les deux tableaux en incrémentant le total et
potentiellement totalCorrect
○──────────○ ↓ realClasses[],estimatedClasses[],nbTest
| countTotCorrect |
○──────────○ ↓ totalCorrect, total

// calculer et sortir l'accuracy
accuracy = totalCorrect/total*100
sortir "L'accuracy est de " + accuracy + "%"

```

```

○-----○ ↓ realClasses[],estimatedClasses[],nbTest
| countTotCorrect |
○-----○ ↓ totalCorrect, total
*
totalCorrect = 0
total = 0
i = 0
while (i < nbTest)
  if (realClasses[i] == estimatedClasses[i])
    totalCorrect++
  total++
  i++

```