

# CARET

Wenjun ZHAO

12/9/2020

Nous allons désormais apprendre à utiliser le package **Caret**. Cette librairie permet de faire de l'analyse prédictive.

Tout comme les autres librairies abordées plus tôt, il faut d'abord installer le package pour pouvoir l'utiliser:

```
install.packages("caret")
```

Pour mieux comprendre cette librairie, nous allons présenter une exemple:

## I. `var_seq`: Séquences De Variables Pour Le Réglage

Cette fonction génère une séquence de mtry valeurs pour les forêts aléatoires

### Arguments

**p**: Le nombre de prédicteurs

**classification**: Le résultat est-il un facteur ( `classification = TRUE` ou numérique?)

**len**: Le nombre de mtryvaleurs à générer.

### Détails

Si le nombre de prédicteurs est inférieur à 500, une simple séquence de valeurs de longueur len est générée entre 2 et p. Pour un plus grand nombre de prédicteurs, la séquence est créée à l'aide d' log2 étapes

Si len = 1, les valeurs par défaut du randomForest package sont utilisées

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
var_seq(p = 100, len = 10)
```

```
## [1] 2 12 23 34 45 56 67 78 89 100
```

```
var_seq(p = 600, len = 10)
```

```
## [1] 2 3 7 13 25 47 89 168 318 600
```

## II. `as.matrix.confusionMatrix`

### Arguments

**X**: un objet de classe confusionMatrix

**what**: données à convertir en matrice. Soit "xtabs", "overall" soit "classes"

...: non utilisé actuellement

```

library(caret)
lvs <- c("normal", "abnormal")
truth <- factor(rep(lvs, times = c(86, 258)),
                levels = rev(lvs))
pred <- factor(
  c(
    rep(lvs, times = c(54, 32)),
    rep(lvs, times = c(27, 231))),
  levels = rev(lvs))

xtab <- table(pred, truth)

results <- confusionMatrix(xtab)
as.table(results)

```

```

##           truth
## pred      abnormal normal
## abnormal    231     32
## normal      27     54

```

```
as.matrix(results)
```

```

##           abnormal normal
## abnormal    231     32
## normal      27     54

```

```
as.matrix(results, what = "overall")
```

```

##           [,1]
## Accuracy      0.8284883721
## Kappa          0.5335968379
## AccuracyLower  0.7844134380
## AccuracyUpper  0.8667985207
## AccuracyNull   0.7500000000
## AccuracyPValue 0.0003096983
## McNemarPValue  0.6025370061

```

```
as.matrix(results, what = "classes")
```

```

##           [,1]
## Sensitivity    0.8953488
## Specificity    0.6279070
## Pos Pred Value 0.8783270
## Neg Pred Value 0.6666667
## Precision      0.8783270
## Recall         0.8953488
## F1             0.8867562
## Prevalence     0.7500000
## Detection Rate 0.6715116
## Detection Prevalence 0.7645349
## Balanced Accuracy 0.7616279

```