Clasificación Monótona

Francisco Pérez

15/2/2017

Lectura del Data set lev

Leemos el data set lev

```
library("foreign")
datos <- read.arff("lev.arff")
summary(datos)
```

```
##       In1             In2             In3             In4
##  Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000
##  1st Qu.:0.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000
##  Median :2.000   Median :2.000   Median :2.000   Median :2.000
##  Mean   :1.722   Mean   :1.985   Mean   :2.127   Mean   :1.985
##  3rd Qu.:3.000   3rd Qu.:3.000   3rd Qu.:3.000   3rd Qu.:3.000
##  Max.   :4.000   Max.   :4.000   Max.   :4.000   Max.   :4.000
##       Out1
##  Min.   :0.000
##  1st Qu.:1.000
##  Median :2.000
##  Mean   :1.785
##  3rd Qu.:2.000
##  Max.   :4.000
```

Veamos cuantas clases tiene este dataset

```
clases = as.integer(unique(datos$Out1))
clases
```

```
## [1] 3 2 0 4 1
```

```
length(clases)
```

```
## [1] 5
```

Teniendo un total de 5 clases Pasemos la clase a tipo factor:

```
datos$Out1 = as.factor(datos$Out1)
```

Trabajo con el Data set

Vamos a seleccionar los índices de las clases, una a una

```
indices.1 <- which(datos$Out1==clases[1])
indices.2 <- c(indices.1, which(datos$Out1==clases[2]))
indices.3 <- c(indices.2, which(datos$Out1==clases[3]))
indices.4 <- c(indices.3, which(datos$Out1==clases[4]))
```

Vamos a selecionar a 0 solo la clase primera, y el resto a 1

```
p1 <- as.integer(datos$Out1)
p1[indices.1]<-0
p1 = ifelse(p1==0,0,1)
p2 <- as.integer(datos$Out1)
p2[indices.2]<-0
p2 = ifelse(p2==0,0,1)
p3 <- as.integer(datos$Out1)
p3[indices.3]<-0
p3 = ifelse(p3==0,0,1)
p4 <- as.integer(datos$Out1)
p4[indices.4]<-0
p4 = ifelse(p4==0,0,1)
```

Con lo que ya tenemos casi listo el primer data frame derivado, nos queda por juntar el resto del dataset con la neuva clase binaria.

```
data1 = cbind(datos[,1:4],target1=as.factor(p1))
data2 = cbind(datos[,1:4],target2=as.factor(p2))
data3 = cbind(datos[,1:4],target3=as.factor(p3))
data4 = cbind(datos[,1:4],target4=as.factor(p4))
```

Creación del modelo de clasificación

Vamos a usar el C4.5, implementado en el paquete de RWeka como J48.

```
library(RWeka)
```

```
##
## Attaching package: 'RWeka'
```

```
## The following objects are masked from 'package:foreign':
##
##     read.arff, write.arff
```

```
modelo1 <- J48(target1 ~., data = data1)
modelo1
```

```
## J48 pruned tree
## ------------------
##
## In2 <= 2
## |   In1 <= 3: 1 (542.0/16.0)
## |   In1 > 3
## |   |   In2 <= 1: 1 (89.0/11.0)
## |   |   In2 > 1
## |   |   |   In3 <= 3: 1 (16.0/5.0)
## |   |   |   In3 > 3: 0 (16.0/3.0)
## In2 > 2
## |   In1 <= 2
## |   |   In2 <= 3: 1 (80.0/20.0)
## |   |   In2 > 3
## |   |   |   In3 <= 2: 1 (95.0/34.0)
## |   |   |   In3 > 2: 0 (63.0/28.0)
## |   In1 > 2
## |   |   In1 <= 3: 0 (79.0/24.0)
## |   |   In1 > 3: 1 (20.0/8.0)
##
## Number of Leaves  :  9
##
## Size of the tree :   17
```

```r
modelo2 <- J48(target2 ~., data = data2)
modelo2
```

```
## J48 pruned tree
## ------------------
##
## In2 <= 1
## |   In1 <= 1: 1 (157.0/16.0)
## |   In1 > 1
## |   |   In3 <= 2
## |   |   |   In4 <= 2: 1 (100.0/29.0)
## |   |   |   In4 > 2: 0 (41.0/16.0)
## |   |   In3 > 2
## |   |   |   In4 <= 2: 0 (42.0/9.0)
## |   |   |   In4 > 2: 1 (25.0/10.0)
## In2 > 1
## |   In1 <= 1
## |   |   In3 <= 1
## |   |   |   In2 <= 3
## |   |   |   |   In4 <= 3: 1 (78.0/16.0)
## |   |   |   |   In4 > 3: 0 (12.0/4.0)
## |   |   |   In2 > 3: 0 (16.0/2.0)
## |   |   In3 > 1
## |   |   |   In4 <= 0
## |   |   |   |   In1 <= 0: 0 (21.0/7.0)
## |   |   |   |   In1 > 0: 1 (7.0/2.0)
## |   |   |   In4 > 0: 0 (146.0/26.0)
## |   In1 > 1: 0 (355.0/42.0)
##
## Number of Leaves  :  12
##
## Size of the tree :   23
```

```r
modelo3 <- J48(target3 ~., data = data3)
modelo3
```

```
## J48 pruned tree
## ------------------
##
## In2 <= 1
## |   In4 <= 1
## |   |   In1 <= 0: 0 (35.0/5.0)
## |   |   In1 > 0
## |   |   |   In1 <= 3
## |   |   |   |   In2 <= 0: 0 (33.0/12.0)
## |   |   |   |   In2 > 0: 1 (46.0/15.0)
## |   |   |   In1 > 3: 0 (42.0/7.0)
## |   In4 > 1
## |   |   In4 <= 2
## |   |   |   In1 <= 2: 1 (65.0/14.0)
## |   |   |   In1 > 2
## |   |   |   |   In1 <= 3: 0 (10.0/3.0)
## |   |   |   |   In1 > 3: 1 (23.0/8.0)
## |   |   In4 > 2: 0 (111.0/53.0)
## In2 > 1
## |   In1 <= 1
## |   |   In3 <= 1
## |   |   |   In2 <= 3
## |   |   |   |   In4 <= 3: 1 (78.0/25.0)
## |   |   |   |   In4 > 3: 0 (12.0/4.0)
## |   |   |   In2 > 3: 0 (16.0/1.0)
## |   |   In3 > 1: 0 (174.0/33.0)
## |   In1 > 1: 0 (355.0/39.0)
##
## Number of Leaves  :  13
##
## Size of the tree :   25
```

```r
modelo4 <- J48(target4 ~., data = data4)
modelo4
```

```
## J48 pruned tree
## ------------------
##
## In2 <= 2
## |   In3 <= 2
## |   |   In4 <= 2
## |   |   |   In4 <= 1
## |   |   |   |   In2 <= 0
## |   |   |   |   |   In3 <= 1: 0 (14.0/3.0)
## |   |   |   |   |   In3 > 1: 1 (6.0/2.0)
## |   |   |   |   In2 > 0
## |   |   |   |   |   In3 <= 0: 1 (42.0/15.0)
## |   |   |   |   |   In3 > 0
## |   |   |   |   |   |   In1 <= 0: 0 (7.0)
## |   |   |   |   |   |   In1 > 0
## |   |   |   |   |   |   |   In2 <= 1: 1 (35.0/12.0)
## |   |   |   |   |   |   |   In2 > 1: 0 (16.0/7.0)
## |   |   |   In4 > 1
## |   |   |   |   In1 <= 2: 1 (76.0/20.0)
## |   |   |   |   In1 > 2
## |   |   |   |   |   In1 <= 3: 0 (16.0/3.0)
## |   |   |   |   |   In1 > 3: 1 (23.0/8.0)
## |   |   In4 > 2
## |   |   |   In1 <= 0: 1 (35.0/13.0)
## |   |   |   In1 > 0: 0 (118.0/25.0)
## |   In3 > 2
## |   |   In2 <= 1
## |   |   |   In1 <= 3
## |   |   |   |   In2 <= 0: 0 (68.0/22.0)
## |   |   |   |   In2 > 0: 1 (33.0/10.0)
## |   |   |   In1 > 3: 0 (48.0/8.0)
## |   |   In2 > 1
## |   |   |   In4 <= 1
## |   |   |   |   In1 <= 1: 1 (7.0/2.0)
## |   |   |   |   In1 > 1: 0 (26.0/5.0)
## |   |   |   In4 > 1: 0 (93.0/6.0)
## In2 > 2: 0 (337.0/26.0)
##
## Number of Leaves  :  18
##
## Size of the tree :   35
```

Hagamos un estudio más detallado de los modelos con la función “evaluate_Weka_classifier”:

```
evaluacion.modelo.1 <- evaluate_Weka_classifier(modelo1, numFolds = 10, complexity = FALSE, class = TRU
evaluacion.modelo.1
```

```
## === 10 Fold Cross Validation ===
##
## === Summary ===
##
## Correctly Classified Instances         834               83.4    %
## Incorrectly Classified Instances       166               16.6    %
## Kappa statistic                          0.4246
## Mean absolute error                      0.2175
## Root mean squared error                  0.3355
## Relative absolute error                 68.6477 %
## Root relative squared error             84.3489 %
## Total Number of Instances             1000
##
## === Detailed Accuracy By Class ===
##
##                  TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
##                  0,462    0,075    0,603      0,462    0,523      0,430    0,836     0,544     0
##                  0,925    0,538    0,875      0,925    0,900      0,430    0,836     0,938     1
## Weighted Avg.    0,834    0,447    0,821      0,834    0,825      0,430    0,836     0,861
##
## === Confusion Matrix ===
##
```

```
##    a   b   <-- classified as
##   91 106 |   a = 0
##   60 743 |   b = 1
```

```
evaluacion.modelo.2 <- evaluate_Weka_classifier(modelo1, numFolds = 10, complexity = FALSE, class = TRU
evaluacion.modelo.2
```

```
## === 10 Fold Cross Validation ===
##
## === Summary ===
##
## Correctly Classified Instances         828               82.8    %
## Incorrectly Classified Instances       172               17.2    %
## Kappa statistic                          0.4038
## Mean absolute error                      0.2198
## Root mean squared error                  0.341
## Relative absolute error                 69.3783 %
## Root relative squared error             85.7333 %
## Total Number of Instances             1000
##
## === Detailed Accuracy By Class ===
##
##                  TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
##                  0,447    0,078    0,583      0,447    0,506      0,409    0,835     0,529     0
##                  0,922    0,553    0,872      0,922    0,896      0,409    0,835     0,941     1
## Weighted Avg.    0,828    0,460    0,815      0,828    0,819      0,409    0,835     0,860
##
## === Confusion Matrix ===
##
##    a   b   <-- classified as
##   88 109 |   a = 0
##   63 740 |   b = 1
```

```
evaluacion.modelo.3 <- evaluate_Weka_classifier(modelo1, numFolds = 10, complexity = FALSE, class = TRU
evaluacion.modelo.3
```

```
## === 10 Fold Cross Validation ===
##
## === Summary ===
##
## Correctly Classified Instances         832               83.2    %
## Incorrectly Classified Instances       168               16.8    %
## Kappa statistic                          0.4273
## Mean absolute error                      0.2184
## Root mean squared error                  0.3374
## Relative absolute error                 68.9277 %
## Root relative squared error             84.8365 %
## Total Number of Instances             1000
##
## === Detailed Accuracy By Class ===
##
##                  TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
##                  0,477    0,081    0,591      0,477    0,528      0,431    0,836     0,546     0
##                  0,919    0,523    0,878      0,919    0,898      0,431    0,836     0,938     1
## Weighted Avg.    0,832    0,436    0,821      0,832    0,825      0,431    0,836     0,861
##
## === Confusion Matrix ===
##
##    a   b   <-- classified as
##   94 103 |   a = 0
##   65 738 |   b = 1
```

```
evaluacion.modelo.4 <- evaluate_Weka_classifier(modelo1, numFolds = 10, complexity = FALSE, class = TRU
evaluacion.modelo.4
```

```
## === 10 Fold Cross Validation ===
##
## === Summary ===
##
## Correctly Classified Instances         827               82.7    %
## Incorrectly Classified Instances       173               17.3    %
## Kappa statistic                          0.3966
## Mean absolute error                      0.2206
## Root mean squared error                  0.3412
## Relative absolute error                 69.6264 %
## Root relative squared error             85.7958 %
## Total Number of Instances             1000
##
## === Detailed Accuracy By Class ===
##
##                  TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
##                  0,437    0,077    0,581      0,437    0,499      0,402    0,827     0,525     0
##                  0,923    0,563    0,870      0,923    0,895      0,402    0,827     0,935     1
## Weighted Avg.    0,827    0,468    0,813      0,827    0,817      0,402    0,827     0,854
##
## === Confusion Matrix ===
##
##    a   b   <-- classified as
##   86 111 |   a = 0
##   62 741 |   b = 1
```

Necesitamos conocer las probabilidades generadas por nuestros modelos, para ello probaremos a predecir la instancia 500 de nuestro dataset, sabiendo de por si que pertenece a la clase:

```
datos[500,3]
```

```
## [1] 2
```

```
prediccion1 <- predict(modelo1,datos[500,1:4],type="probability")
prediccion1
```

```
##             0         1
## 500 0.0295203 0.9704797
```

```r
prediccion2 <- predict(modelo2,datos[500,1:4],type="probability")
prediccion2
```

```
##             0         1
## 500 0.6097561 0.3902439
```

```r
prediccion3 <- predict(modelo3,datos[500,1:4],type="probability")
prediccion3
```

```
##             0         1
## 500 0.5225225 0.4774775
```

```r
prediccion4 <- predict(modelo4,datos[500,1:4],type="probability")
prediccion4
```

```
##             0         1
## 500 0.7881356 0.2118644
```