Random Forest

Rodrigo Assar

Paquete de R

- randomForest, en repositorio CRAN.
- Funcion randomForest():

  randomForest(formula, data=NULL, …)

  randomForest(x, y=NULL, xtest=NULL, ytest=NULL, ntree=500,…)

  - Para clasificacion y regresion:
    - Se fija lo pedido segun clase de componente y.
  - Componentes:
    - ntree: numero de arboles a construir
    - cutoff: numero de arboles necesarios para lograr clase (ej:1/clases para mayoria relativa)

Ejemplo: juguemos en el bosque

```
Rf_fit <- randomForest(Kyphosis ~ Age + Number + Start,
data=kyphosis, importance=TRUE, proximity=TRUE)
prediccionRF=predict(Rf_fit,kyphosis[valid,],type='class')
table(prediccionRF,kyphosis[valid,1])
```

prediccionRF	absent	present
absent	17	1
present	0	3

Ejecutando print(Rf_fit)

Call:

randomForest(formula = Kyphosis ~ Age + Number + Start, data = kyphosis, importance = TRUE, proximity = TRUE)

Type of random forest: classification

Number of trees: 500

No. of variables tried at each split: 1

OOB estimate of error rate: 19.75%

Confusion matrix:

	absent	present	class.error
absent	60	4	0.0625000
present	12	5	0.7058824

Revisando importancia variables

round(importance(Rf_fit), 2)

	absent	present	MeanDecreaseAccuracy	MeanDecreaseGini
Age	0.23	2.25	0.72	8.83
Number	-0.38	1.06	-0.04	5.37
Start	1.88	5.47	2.33	9.88

Validando con datos no usados

```
Rf_fitsub <- randomForest(Kyphosis ~ Age + Number + Start,
data=kyphosis, importance=TRUE, proximity=TRUE,subset=sub)
prediccionRFvalid=predict(Rf_fitsub,kyphosis[valid,],type='class')
table(prediccionRFvalid,kyphosis[valid,1])
```

prediccionRFvalid	absent	present
absent	15	3
present	2	1

Viendo el clasificador

summary(Rf_fit2)

	Length	Class	Mode
call	6	-none-	call
type	1	-none-	character
predicted	60	factor	numeric
err.rate	1500	-none-	numeric
confusion	6	-none-	numeric
votes	120	matrix	numeric
oob.times	60	-none-	numeric
classes	2	-none-	character
importance	12	-none-	numeric
importanceSD	9	-none-	numeric
localImportance	0	-none-	NULL

…

Con funcion summary()

```
summary(fit2)
Call:
rpart(formula = Kyphosis ~ Age + Number + Start, data = kyphosis,
    parms = list(prior = c(0.65, 0.35), split = "information"))
  n= 81

          CP nsplit rel error    xerror      xstd
1 0.3019958      0 1.0000000 1.0000000 0.2155872
2 0.2023372      1 0.6980042 0.9960609 0.1941501
3 0.0100000      2 0.4956670 0.7293855 0.1528517
```

Los campos del clasificador

Rf_fit2$call

randomForest(formula = Kyphosis ~ Age + Number + Start, data = kyphosis,
importance = TRUE, proximity = TRUE, subset = sub)

Rf_fit2$confusion

	absent	present	class.error
absent	44	3	0.06382979
present	10	3	0.76923077

Rf_fit2$importance

	absent	present	MeanDecreaseAccuracy
Age	0.005021869	0.05443413	0.01327076
Number	0.004198105	0.06467937	0.01536023
Start	0.031456498	0.14577857	0.05291519

	MeanDecreaseGini
Age	6.248575
Number	4.094898
Start	7.990978

Revisando campos: votos

Rf_fit2$votes

	absent	present
24	0.5268817	0.473118280
61	0.7674419	0.232558140
44	0.5506329	0.449367089
40	0.5454545	0.454545455
35	0.8507463	0.149253731
48	0.8795181	0.120481928
36	0.8633880	0.136612022
69	0.9661017	0.033898305
27	0.6648045	0.335195531
80	0.4921466	0.507853403
20	0.9774011	0.022598870
57	0.9834254	0.016574586
62	0.3697917	0.630208333
60	0.8715084	0.128491620

…

Extrayendo un arbol del bosque

tree8=getTree(Rf_fit2, k=8)

print(tree8) # arbol se guarda como matriz

left daughter	right daughter	split var	split point
1	2	3	1	130.5
2	4	5	3	8.5
3	0	0	0	0.0
4	6	7	3	5.5
5	8	9	1	55.0
6	10	11	2	3.5
7	0	0	0	0.0
8	0	0	0	0.0

…