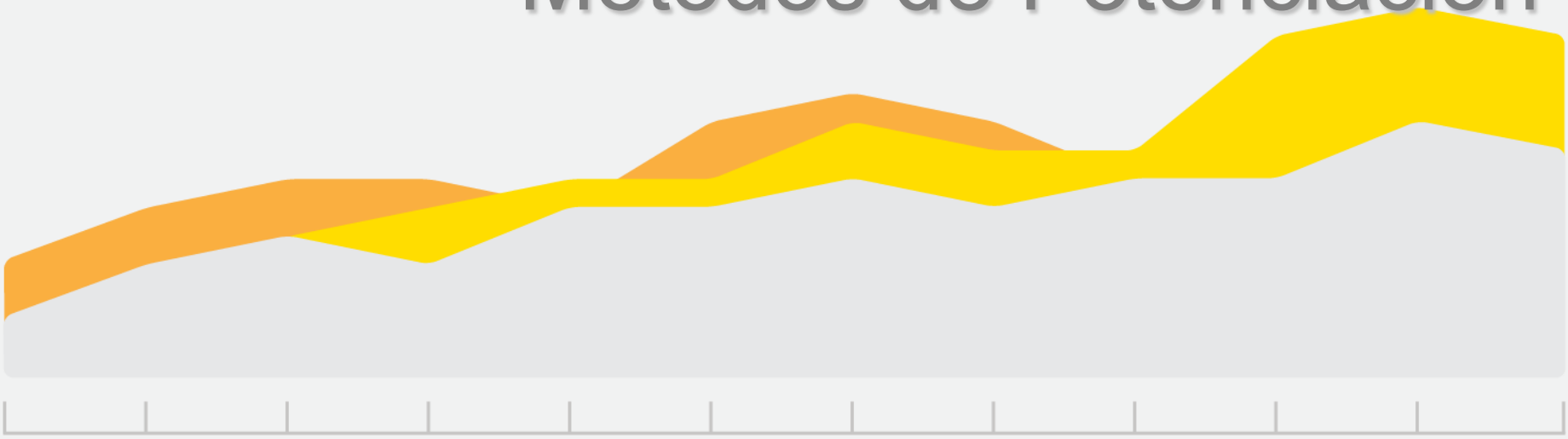
oldemar rodríguez
CONSULTOR en MINERIA DE DATOS

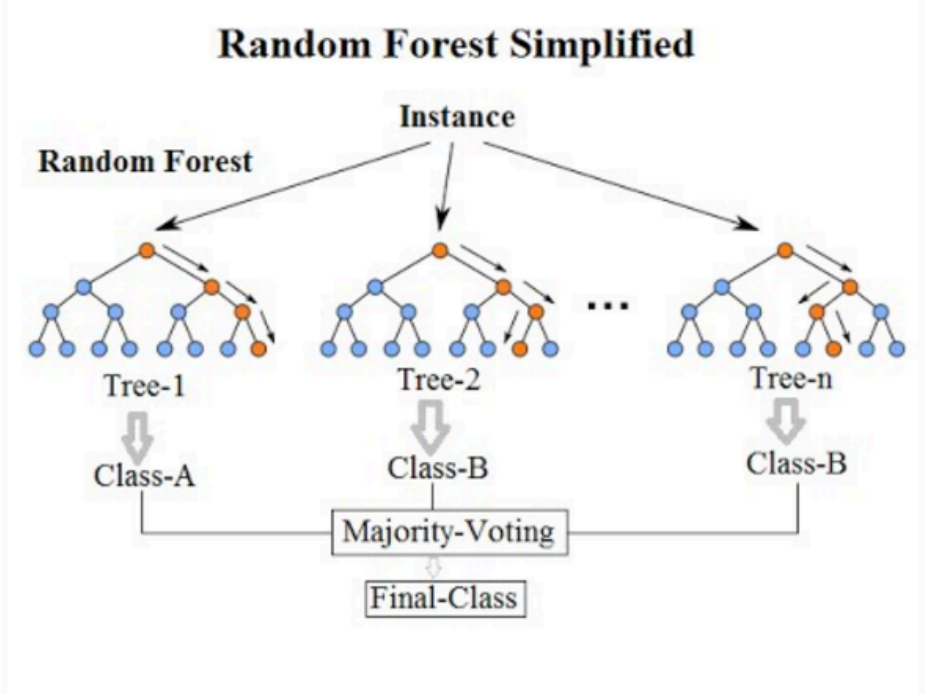
Aprendizaje Supervisado

(Ensemble Methods – Classifier Combination)

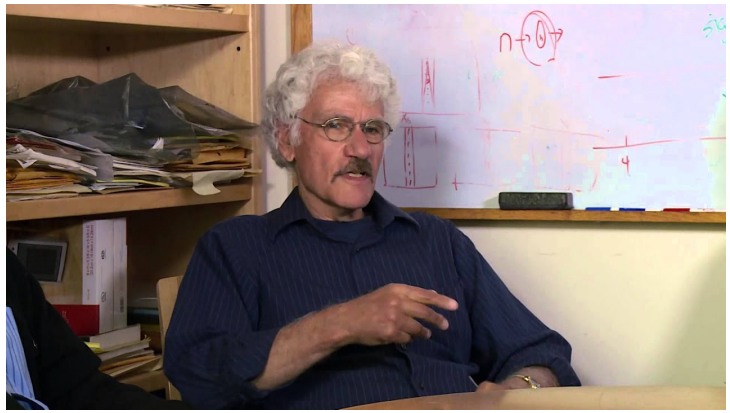
Métodos de Consenso
y
Métodos de Potenciación



Bosques Aleatorios
Random Forests



L. Brieman



J. Friedman

Bagging

Breiman 1996 Hastie et al ch 8.7

Lección de:
J. Friedman

Goal: improve performance of unstable procedures (trees, neural nets, MARS) by stabilizing them.

usual procedure:

$$\hat{F}(\underline{x}) = \underset{F(\underline{x}) \in \mathcal{F}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} L(y_i, F(\underline{x}_i))$$

unstable: small change in training data

$$T = \{y_i, \underline{x}_i\}_1^N$$

$\Rightarrow$ big change in $\hat{F}(\underline{x})$

$\Rightarrow$ big $\text{var}_T \hat{F}(\underline{x})$

$\Rightarrow$ big error

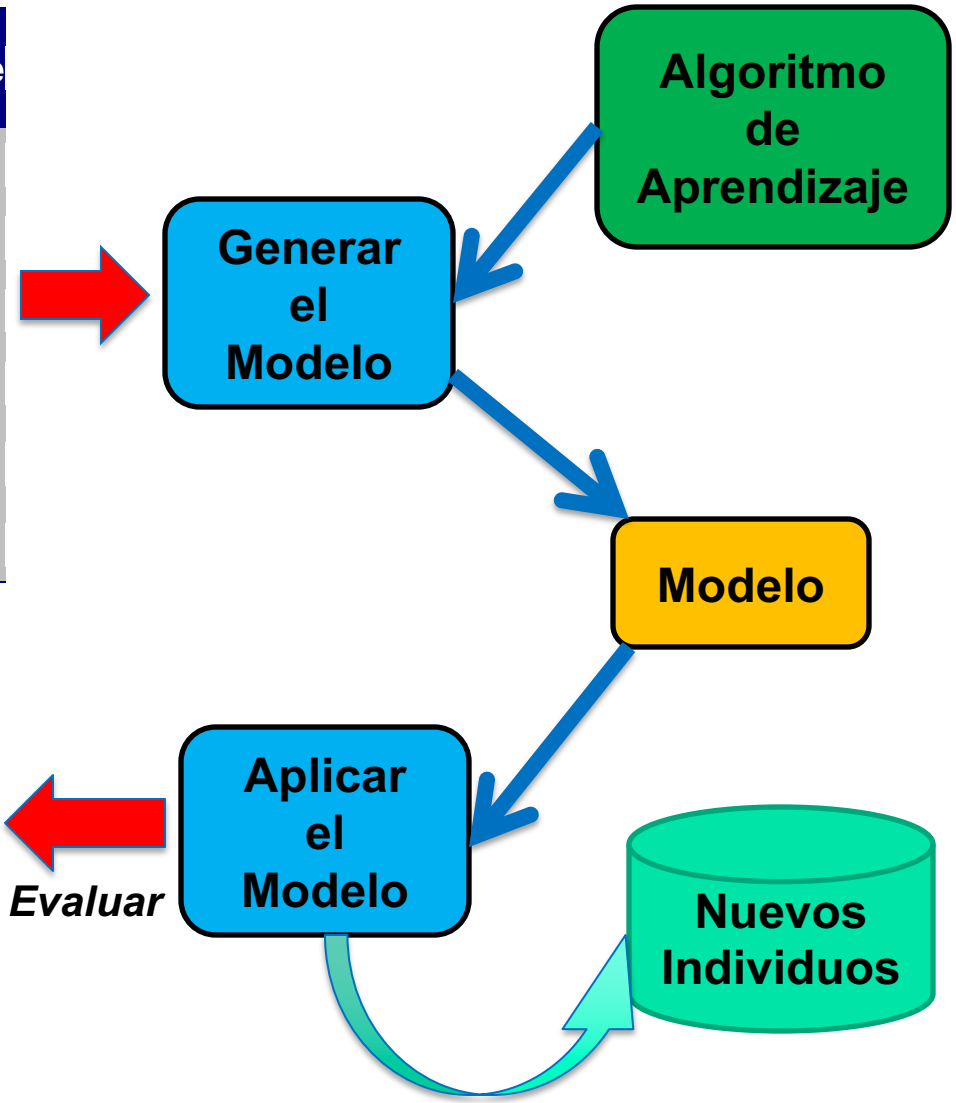
Modelo general de los métodos de Clasificación

Id	Reembolso	Estado Civil	Ingresos Anuales	Fraude
1	Sí	Soltero	125K	No
2	No	Casado	100K	No
3	No	Soltero	70K	No
4	Sí	Casado	120K	No
5	No	Divorciado	95K	Sí
6	No	Casado	60K	No

Tabla de Aprendizaje

Id	Reembolso	Estado Civil	Ingresos Anuales	Fraude
7	No	Soltero	80K	No
8	Si	Casado	100K	No
9	No	Soltero	70K	No

Tabla de Testing



Ejemplo: Créditos en un Banco

Tabla de Aprendizaje

Variable Discriminante

OLDEMARRR.DMEx...ditoViviendaPeq

Id	MontoCredito	IngresoNeto	CoeficienteCre...	MontoCuota	GradoAcademico	BuenPagador
1	2	4	3	1	4	1
2	2	3	2	1	4	1
3	4	1	1	4	2	2
4	1	4	3	1	4	1
5	3	3	1	3	2	2
6	3	4	3	1	4	1
7	4	2	1	3	2	2
8	4	1	3	3	2	2
9	3	4	3	1	3	1
10	1	3	2	2	4	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Con la Tabla de Aprendizaje se entrena (aprende) el modelo matemático de predicción, es decir, a partir de esta tabla se calcula la función ***f*** de la definición anterior.

Ejemplo: Créditos en un Banco

Tabla de Testing

Variable Discriminante

OLDEMARRR.DME...iviendaPeqPRED | OLDEMARRR.DMEx...ditoViviendaPeq

	Id	MontoCredito	IngresoNeto	CoeficienteCre...	MontoCuota	GradoAcademico	BuenPagador
▶	11	3	3	3	3	1	2
	12	2	2	2	2	1	1
	13	2	2	3	2	1	1
	14	1	3	4	3	2	2
	15	1	2	4	2	1	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- Con la Tabla de Testing se valida el modelo matemático de predicción, es decir, se verifica que los resultados en individuos que no participaron en la construcción del modelo es bueno o aceptable.
- Algunas veces, sobre todo cuando hay pocos datos, se utiliza la Tabla de Aprendizaje también como de Tabla Testing.

Ejemplo: Créditos en un Banco

Nuevos Individuos

Variable Discriminante

OLDEMARRR.DMEx ...editoViviendaNI

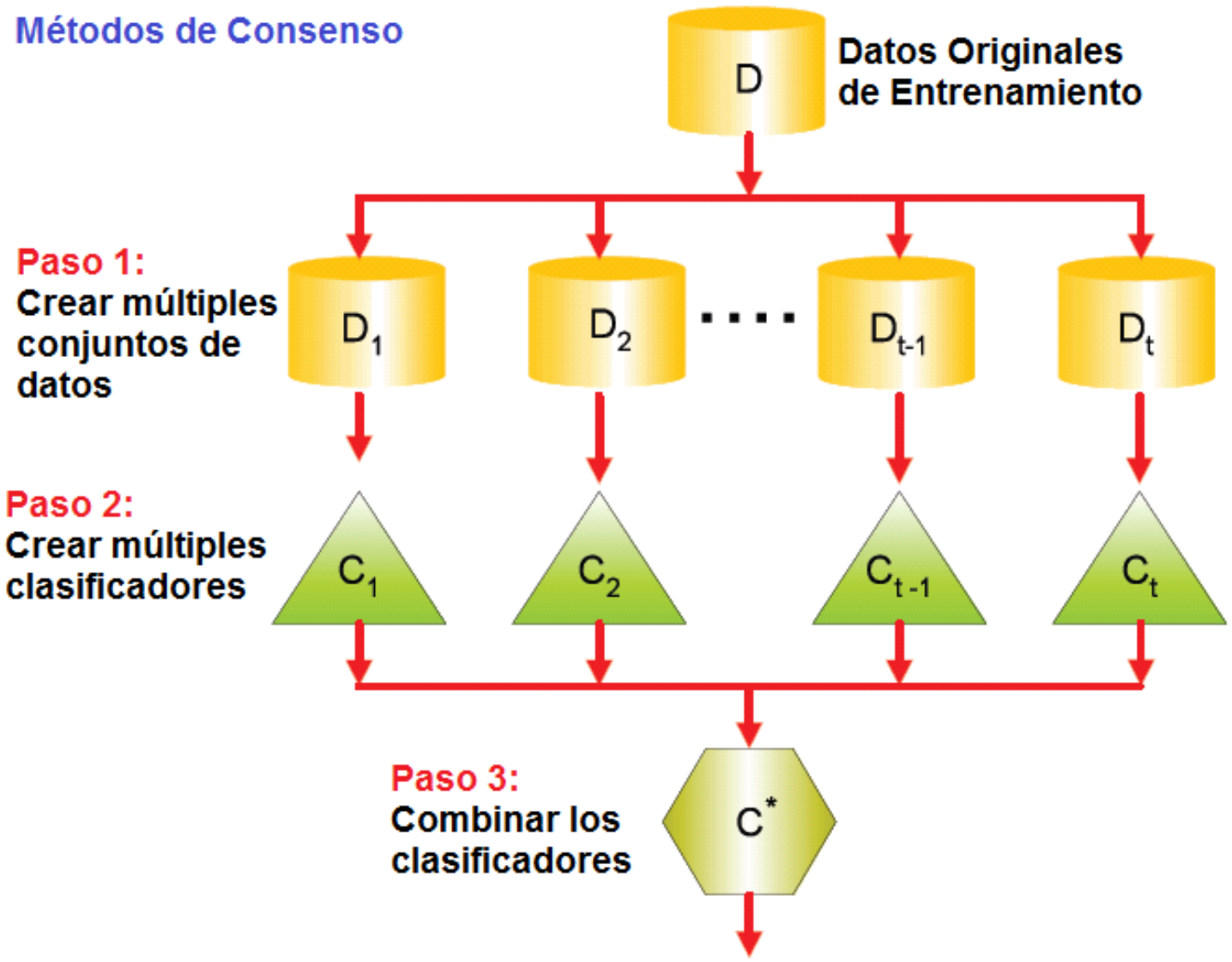
	Id	MontoCredito	IngresoNeto	CoeficienteCre...	MontoCuota	GradoAcademico	BuenPagador
	100	4	4	2	2	3	?
	101	1	4	3	2	4	?
	102	3	2	3	4	2	?
▶*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Con la Tabla de Nuevos Individuos se predice si estos serán o no buenos pagadores.

Métodos de Consenso (Bagging)

- La idea es tomar *m* muestras aleatorias con reemplazo (Boostraps) de los datos originales y luego aplicar en cada una de ellas un método predictivo para luego con algún criterio establecer un consenso de todos los resultados
- El consenso podría ser un promedio, un promedio ponderado basado en cuál método obtuvo los mejores resultados
- El consenso más usado es el que obtenga la "***mayor cantidad de votos***"

Métodos de Consenso



Métodos de Consenso (Bagging)

- Cada muestra de arranque (bootstrap) tiene el mismo tamaño que los datos originales. Debido a que el muestreo se realiza con reemplazo, algunos individuos pueden aparecer varias veces en el mismo conjunto de entrenamiento, mientras que otros podrían no parecer nunca.
- ***Teorema:*** En promedio, una muestra de arranque D_i (bootstrap) contiene aproximadamente 63% de los datos de entrenamiento originales.
- ***Prueba:*** Cada individuo tiene una probabilidad de $1-(1-1/n)^n$ de ser seleccionados en cada D_i. Si n es suficientemente grande, esta probabilidad converge a $1-\frac{1}{e}=0.632$.

Algoritmo General: Métodos de Consenso (Bagging)

Algorithm 5.6 Bagging algorithm.

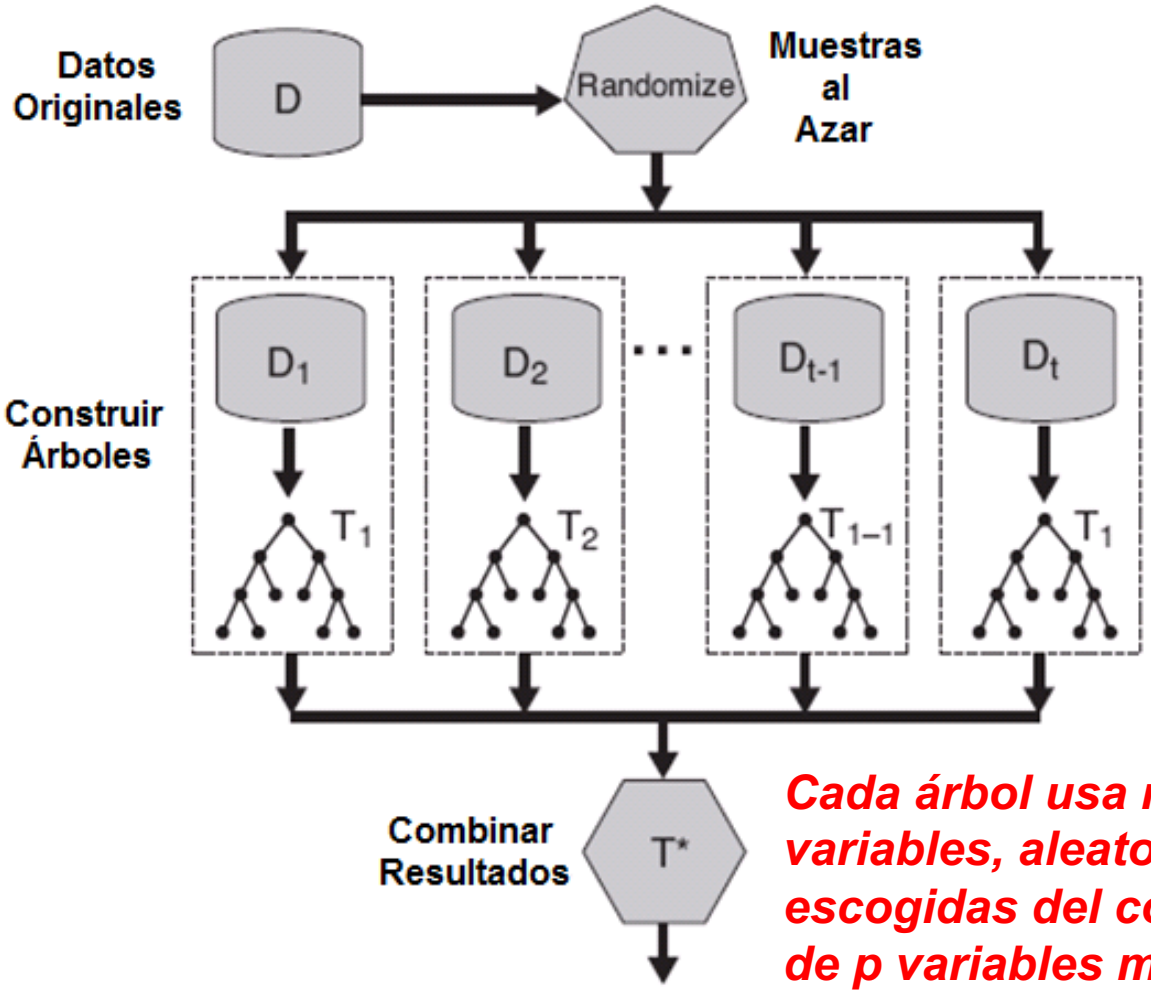
1: Let k be the number of bootstrap samples.
2: **for** $i = 1$ to k **do**
3: Create a bootstrap sample of size N, D_i.
4: Train a base classifier C_i on the bootstrap sample D_i.
5: **end for**
6: $C^*(x) = \underset{y}{\operatorname{argmax}} \sum_i \delta\big(C_i(x) = y\big)$.

$\{\delta(\cdot) = 1$ if its argument is true and 0 otherwise$\}$.

Bosques Aleatorios (Random Forest)

- El caso en el que todos los clasificadores del Método de Consenso son Árboles dicho método se denomina Bosques Aleatorios (Random Forest)

Bosques Aleatorios (Random Forest)



Cada árbol usa m diferentes variables, aleatoriamente escogidas del conjunto de p variables m<p (m=mtry en R)

Ejemplo 1: IRIS.CSV

Ejemplo con la tabla de datos IRIS
IRIS Información de variables:

1.sepal largo en cm
2.sepal ancho en cm
3.petal largo en cm
4.petal ancho en cm
5.clase:

- Iris Setosa
- Iris Versicolor
- Iris Virginica

	A	B	C	D	E
1	s.largo	s.ancho	p.largo	p.ancho	tipo
2	5.1	3.5	1.4	0.2	setosa
3	4.9	3.0	1.4	0.2	setosa
4	4.7	3.2	1.3	0.2	setosa
5	4.6	3.1	1.5	0.2	setosa
6	5.0	3.6	1.4	0.2	setosa
7	5.4	3.9	1.7	0.4	setosa
8	4.6	3.4	1.4	0.3	setosa
9	5.0	3.4	1.5	0.2	setosa
10	4.4	2.9	1.4	0.2	setosa
11	4.9	3.1	1.5	0.1	setosa
12	5.4	3.7	1.5	0.2	setosa
13	4.8	3.4	1.6	0.2	setosa
14	4.8	3.0	1.4	0.1	setosa
15	4.3	3.0	1.1	0.1	setosa
16	5.8	4.0	1.2	0.2	setosa
17	5.7	4.4	1.5	0.4	setosa
18	5.4	3.9	1.3	0.4	setosa
19	5.1	3.5	1.4	0.3	setosa
20	5.7	3.8	1.7	0.3	setosa
21	5.1	3.8	1.5	0.3	setosa
22	5.4	3.4	1.7	0.2	setosa
23	5.1	3.7	1.5	0.4	setosa
24	4.6	3.6	1.0	0.2	setosa
25	....	...	...	...	....

Ejemplo 2: Credit-Scoring

MuestraAprendizajeCredito2500.csv
MuestraTestCredito2500.csv

```
> setwd("C:/Users/Oldemar/Google Drive/Curso Mineria Datos II - Optativo/Datos")
> taprendizaje<-read.csv("MuestraAprendizajeCredito2500.csv",sep = ";",header=T)
> taprendizaje
```

	MontoCredito	IngresoNeto	CoefCreditoAvaluo	MontoCuota	GradoAcademico	BuenPagador
1	1	1	1	1	1	Si
2	3	1	1	1	1	Si
3	2	1	1	1	1	Si
4	1	2	1	1	1	Si
5	1	1	1	1	1	Si
6	2	1	1	1	1	Si
7	4	1	1	1	1	Si
8	1	2	1	1	1	Si
9	1	2	1	1	1	Si
10	3	2	1	1	1	Si
11	1	1	1	1	1	Si
12	1	2	1	1	1	Si
13	3	1	1	1	1	Si
14	3	1	1	1	1	Si
15	2	1	1	1	1	Si
16	3	1	1	1	1	Si
17	3	1	1	1	1	Si

Classification and Regression with Random Forest

Description

`randomForest` implements Breiman's random forest algorithm (based on Breiman and Cutler's original Fortran code) for classification and regression. It can also be used in unsupervised mode for assessing proximities among data points.

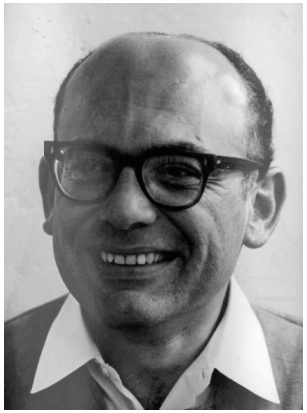
Usage

```
## S3 method for class 'formula'
randomForest(formula, data=NULL, ..., subset, na.action=na.fail)
## Default S3 method:
randomForest(x, y=NULL,  xtest=NULL, ytest=NULL, ntree=500,
             mtry=if (!is.null(y) && !is.factor(y))
             max(floor(ncol(x)/3), 1) else floor(sqrt(ncol(x))),
             replace=TRUE, classwt=NULL, cutoff, strata,
             sampsize = if (replace) nrow(x) else ceiling(.632*nrow(x)),
             nodesize = if (!is.null(y) && !is.factor(y)) 5 else 1,
             maxnodes = NULL,
             importance=FALSE, localImp=FALSE, nPerm=1,
             proximity, oob.prox=proximity,
             norm.votes=TRUE, do.trace=FALSE,
             keep.forest=!is.null(y) && is.null(xtest), corr.bias=FALSE,
             keep.inbag=FALSE, ...)
## S3 method for class 'randomForest'
print(x, ...)
```

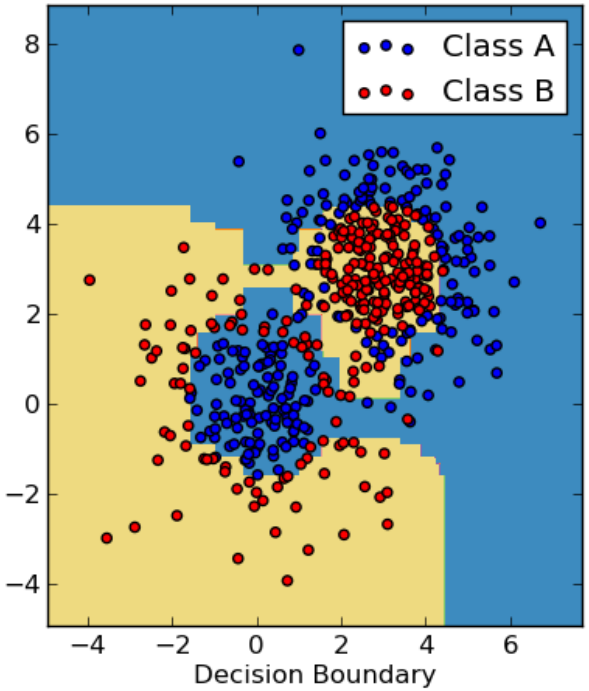
Métodos de Potenciación

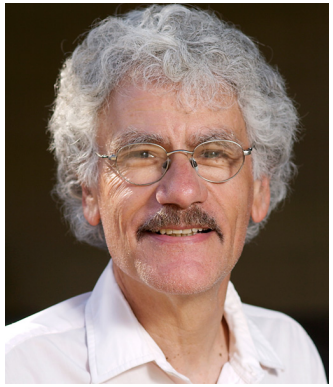
"Best off-the-shelf classifier in the world"

[Breiman, NIPS Workshop, 1996]



Breiman



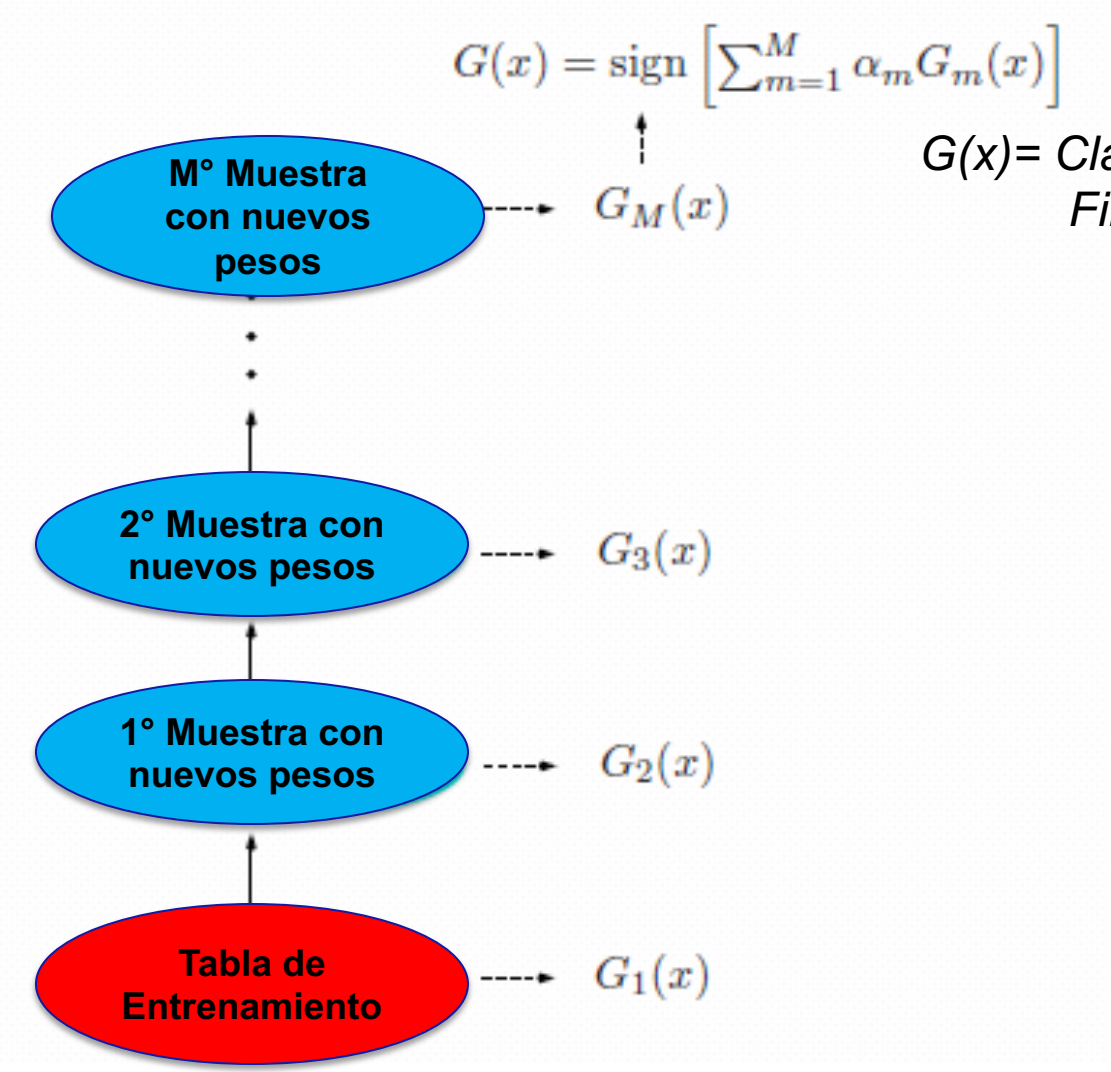


Friedman

Métodos de Potenciación

- La idea es tomar una muestra aleatoria de los datos originales y aplicar sobre esta un método clasificatorio luego *aumentar el peso (potenciar)* a los individuos mal clasificados para que en la siguiente aplicación del método clasificatorio se enfoque más en estos individuos mal clasificados, mejorando su clasificación, y así sucesivamente …
- **Observación:** Solo funciona para problemas de clasificación **binarios** (de 2 clases).

Métodos de Potenciación



G(x)= Clasificador Final

Métodos de Potenciación

- Al inicio el algoritmo de Potenciación utiliza los mismos pesos (*1/n*) en todos los individuos para determinar la distribución de muestreo de la muestra de entrenamiento, es decir, todos los individuos tienen la misma probabilidad de ser elegido.
- A continuación, un clasificador se aplica en la muestra de entrenamiento y se utiliza para clasificar todos los datos originales.
- Luego los pesos de los individuos de entrenamiento se actualizan para "impulsarlos" en la siguiente ejecución del clasificador. Así los individuos que se clasificaron incorrectamente tendrán un incremento es sus pesos, mientras que aquellos que se clasificaron correctamente tendrán una disminución en su peso.
- Esto obliga, en iteraciones posteriores, al clasificador a concentrarse en los individuos que son más difíciles de clasificar.

Métodos de Potenciación
Algoritmo:AdaBoost.M1

1. Initialize the observation weights $w_i = 1/N,\ i = 1, 2, \ldots, N$.
2. For $m = 1$ to M:
   (a) Fit a classifier $G_m(x)$ to the training data using weights w_i.
   (b) Compute
   $$\text{err}_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}.$$
   (c) Compute $\alpha_m = \log((1 - \text{err}_m)/\text{err}_m)$.
   (d) Set $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))],\ i = 1, 2, \ldots, N$.
3. Output $G(x) = \text{sign}\left[\sum_{m=1}^{M} \alpha_m G_m(x)\right]$.

Ejemplo: Algoritmo:AdaBoost.M1

Boosting Round 1:

x	0.1	0.4	0.5	0.6	0.6	0.7	0.7	0.7	0.8	1
y	1	-1	-1	-1	-1	-1	-1	-1	1	1

Boosting Round 2:

x	0.1	0.1	0.2	0.2	0.2	0.2	0.3	0.3	0.3	0.3
y	1	1	1	1	1	1	1	1	1	1

Boosting Round 3:

x	0.2	0.2	0.4	0.4	0.4	0.4	0.5	0.6	0.6	0.7
y	1	1	-1	-1	-1	-1	-1	-1	-1	-1

(a) Training records chosen during boosting

Round	x=0.1	x=0.2	x=0.3	x=0.4	x=0.5	x=0.6	x=0.7	x=0.8	x=0.9	x=1.0
1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
2	0.311	0.311	0.311	0.01	0.01	0.01	0.01	0.01	0.01	0.01
3	0.029	0.029	0.029	0.228	0.228	0.228	0.228	0.009	0.009	0.009

(b) Weights of training records

Ejemplo 1: Credit-Scoring

MuestraAprendizajeCredito2500.csv
MuestraTestCredito2500.csv

```
> setwd("C:/Users/Oldemar/Google Drive/Curso Mineria Datos II - Optativo/Datos")
> taprendizaje<-read.csv("MuestraAprendizajeCredito2500.csv",sep = ";",header=T)
> taprendizaje
```

	MontoCredito	IngresoNeto	CoefCreditoAvaluo	MontoCuota	GradoAcademico	BuenPagador
1	1	1	1	1	1	Si
2	3	1	1	1	1	Si
3	2	1	1	1	1	Si
4	1	2	1	1	1	Si
5	1	1	1	1	1	Si
6	2	1	1	1	1	Si
7	4	1	1	1	1	Si
8	1	2	1	1	1	Si
9	1	2	1	1	1	Si
10	3	2	1	1	1	Si
11	1	1	1	1	1	Si
12	1	2	1	1	1	Si
13	3	1	1	1	1	Si
14	3	1	1	1	1	Si
15	2	1	1	1	1	Si
16	3	1	1	1	1	Si
17	3	1	1	1	1	Si

Descripción de Variables

MontoCredito

1=Muy Bajo
2=Bajo
3=Medio
4=Alto

MontoCuota

1=Muy Bajo
2=Bajo
3=Medio
4=Alto

IngresoNeto

1=Muy Bajo
2=Bajo
3=Medio
4=Alto

GradoAcademico

1=Bachiller
2=Licenciatura
3=Maestría
4=Doctorado

CoeficienteCreditoAvaluo

1=Muy Bajo
2=Bajo
3=Medio
4=Alto

BuenPagador

1=NO
2=Si

Fitting Stochastic Boosting Models

Description

'ada' is used to fit a variety stochastic boosting models for a binary response as described in *Additive Logistic Regression: A Statistical View of Boosting* by Friedman, et al. (2000).

Usage

```
ada(x,...)
## Default S3 method:
ada(x, y,test.x,test.y=NULL, loss=c("exponential","logistic"),
                      type=c("discrete","real","gentle"),iter=50, nu=0.1, bag.frac=0.5,
                      model.coef=TRUE,bag.shift=FALSE,max.iter=20,delta=10^(-10),verbose=FA
                      ...,na.action=na.rpart)

## S3 method for class 'formula'
ada(formula, data, ..., subset, na.action=na.rpart)
```

Evolución Métodos Potenciación

Algoritmo: Extreme Gradient Boosting

- Decision Tree
- Random Forest
- Gradient Boosting Tree
- Extreme Gradient Boosting(xgboost)

Evolución Métodos Potenciación
Algoritmo: Gradient Boosting

In 2000, Friedman et al. developed a statistical view of the AdaBoost algorithm. They interpreted AdaBoost as stagewise estimation procedures for fitting an additive logistic regression model. They showed that AdaBoost was actually minimizing the exponential loss function

$$L(y, F(x)) = E(e^{-yF(x)}).$$

It is minimized at

$$\frac{\partial E[e^{-yF(x)}]}{\partial F(x)} = 0.$$

Evolución Métodos Potenciación
Algoritmo: Gradient Boosting

Stagewise Additive Modeling

Boosting builds an additive model

$$f(x) = \sum_{m=1}^{M} \beta_m b(x; \gamma_m).$$

Here $b(x, \gamma_m)$ is a tree, and γ_m parametrizes the splits.

Evolución Métodos Potenciación
Algoritmo: Gradient Boosting

Adaboost: Stagewise Modeling

- AdaBoost builds an additive logistic regression model

$$f(x) = \log \frac{\Pr(Y = 1|x)}{\Pr(Y = -1|x)} = \sum_{m=1}^{M} \alpha_m G_m(x)$$

by stagewise fitting using the loss function

$$L(y, f(x)) = \exp(-y\, f(x)).$$

Gradient Boosting

Algoritmo: Gradient Boosting

1. Initialize $f_0(x) = \arg\min_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$.
2. For $m = 1$ to M:
   (a) For $i = 1, 2, \ldots, N$ compute

$$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}.$$

   (b) Fit a regression tree to the targets r_{im} giving terminal regions $R_{jm},\ j = 1, 2, \ldots, J_m$.
   (c) For $j = 1, 2, \ldots, J_m$ compute

$$\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{jm}} L\left(y_i, f_{m-1}(x_i) + \gamma\right).$$

   (d) Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$.
3. Output $\hat{f}(x) = f_M(x)$.

Evolución Métodos Potenciación
Algoritmo: Gradient Boosting

Adaboost: Stagewise Modeling

- AdaBoost builds an additive logistic regression model

$$f(x) = \log \frac{\Pr(Y = 1|x)}{\Pr(Y = -1|x)} = \sum_{m=1}^{M} \alpha_m G_m(x)$$

by stagewise fitting using the loss function

$$L(y, f(x)) = \exp(-y\, f(x)).$$

Evolución Métodos Potenciación
Algoritmo: Extreme Gradient Boosting

3.4 Extreme Gradient Boosting

Después de hacer una búsqueda intensiva de los modelos y algoritmos predictivos que se usan en la actualidad, se encuentra que la técnica más empleada y que está cosechando más éxitos es el Extreme Gradient Boosting. Además, este algoritmo está dominando las resoluciones de Machine Learning de la plataforma Kaggle. [12] [13]
Tal y como su nombre indica, este método se basa en el Gradient Boosting. El término "Extreme" se refiere al objetivo de llevar al límite los recursos computacionales de la técnica de Gradient Boosting para obtener mejores resultados. [14] [15]

Este algoritmo creado por Tianqi Chen [16] (estudiante de la Universidad de Washington) nació con la idea de crear un sistema escalable del algoritmo Gradient Boosting.

Evolución Métodos Potenciación

Algoritmo: Extreme Gradient Boosting

AdaBoost	GradientBoost
Both AdaBoost and Gradient Boost use a base weak learner and they try to boost the performance of a weak learner by iteratively shifting the focus towards problematic observations that were difficult to predict. At the end, a strong learner is formed by addition (or weighted addition) of the weak learners.	
In AdaBoost, shift is done by up-weighting observations that were misclassified before.	Gradient boost identifies difficult observations by large residuals computed in the previous iterations.
In AdaBoost "shortcomings" are identified by high-weight data points.	In Gradientboost "shortcomings" are identified by gradients.
Exponential loss of AdaBoost gives more weights for those samples fitted worse.	Gradient boost further dissect error components to bring in more explanation.
AdaBoost is considered as a special case of Gradient boost in terms of loss function, in which exponential losses.	Concepts of gradients are more general in nature.

oldemar rodríguez

CONSULTOR en M1NƐR14 DƐ D4T0S

Gracias....