Aggregation Methods

Mathias Bourel

IMERL - Facultad de Ingeniería, Universidad de la República, Uruguay

April 3, 2019

Aggregation methods

Aggregation methods

$\mathcal{L}$ the data base and $\widehat{g_1}, \ldots, \widehat{g_M}$ several predictors built over $\mathcal{L}$

$$\widehat{f} = g(\widehat{g_1}, \ldots, \widehat{g_M})$$

Aggregation methods

Aggregation methods

$\mathcal{L}$ the data base and $\widehat{g_1}, \ldots, \widehat{g_M}$ several predictors built over $\mathcal{L}$

$$\widehat{f} = g(\widehat{g_1}, \ldots, \widehat{g_M})$$

- Homogeneous aggregation methods (sequential and not sequential).
- Not homogeneous aggregation methods (consensus methods).

Plan

Bagging, Breiman (1996)

1. $\mathcal{L} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ where $X_i \in \mathcal{X}$ and $Y_i \in \mathcal{Y}$
2. For $m = 1$ to M:
   1. We consider a bootstrap sample $\mathcal{L}_m^*$ of size n from $\mathcal{L}$.
   2. We build the estimator $g_m : \mathcal{X} \to \mathcal{Y}$ frmo $\mathcal{L}_m^*$.
3. Output: $f_M(x) = \text{Argmax}_y \#\{m : g_m(x) = y\}$ (classification) or $f_M(x) = \frac{1}{M} \sum_{m=1}^{M} g_m(x)$ (regression).

Figure: Bagging

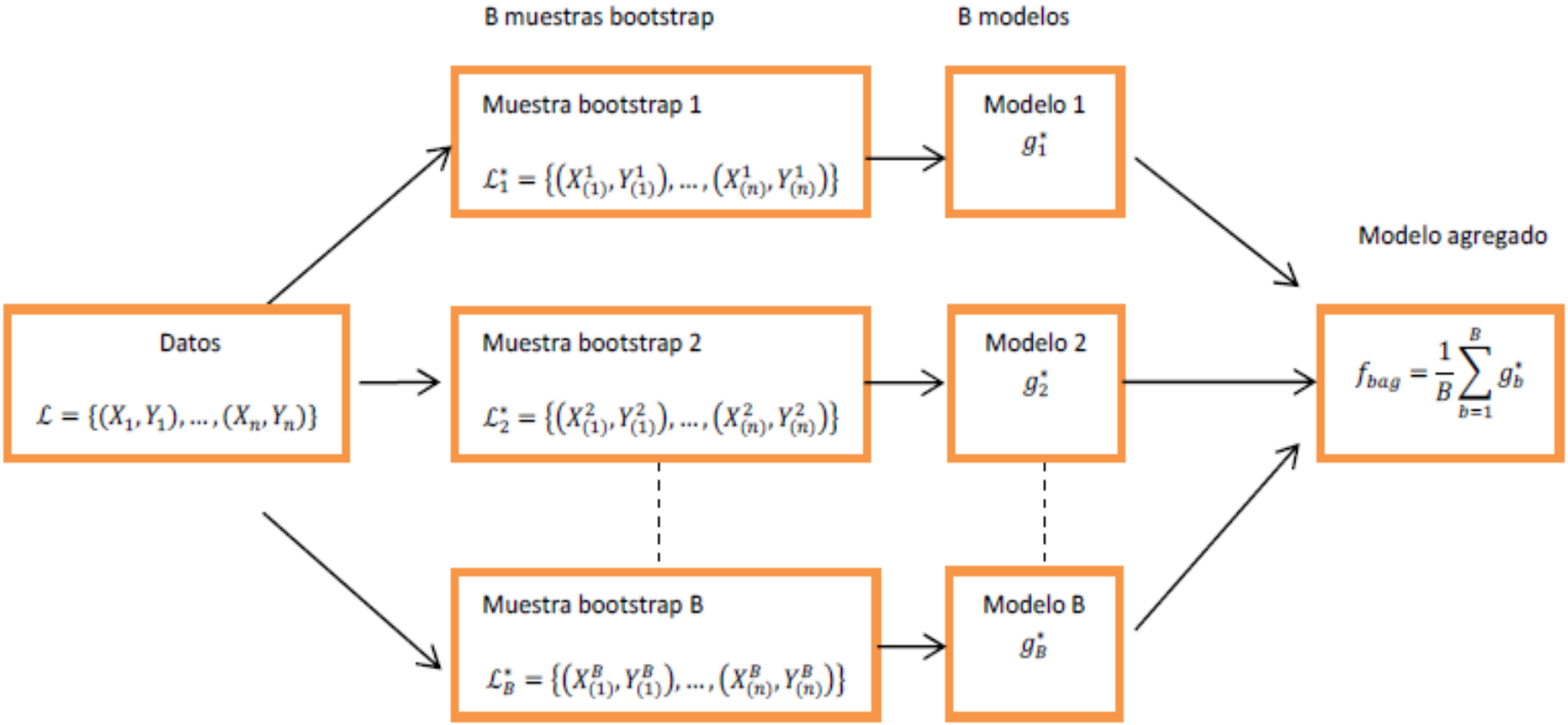
Bagging, Breiman (1996)

1. $\mathcal{L} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ where $X_i \in \mathcal{X}$ and $Y_i \in \mathcal{Y}$
2. For $m = 1$ to M:
   1. We consider a bootstrap sample $\mathcal{L}_m^*$ of size n from $\mathcal{L}$.
   2. We build the estimator $g_m : \mathcal{X} \to \mathcal{Y}$ frmo $\mathcal{L}_m^*$.
3. Output: $f_M(x) = \text{Argmax}_y \#\{m : g_m(x) = y\}$ (classification) or $f_M(x) = \frac{1}{M} \sum_{m=1}^{M} g_m(x)$ (regression).

Figure: Bagging

- The Bagging estimator generally improves the result of any unstable algorithm. In many cases the reduction of the error is important.
- It loose interpretability.
- The observations that are not drawn in the bootstrap sample are called "out of bag" (OOB).
- OOB error is a good approximation of the test error.

Bagging, Breiman (1996)



Plan

Random Forest, Breiman 2001

1. This method combines the predictions of several trees obtained from bootstrap samples of the data set.
2. In each node, only a small number (e.g $\sqrt{p}$ or $\log(p)$ of the total number p) of randomly chosen variables is taken into account to determine the best partition. This value suggested by Breiman in classification, has been confirmed by several works which showed its optimality in terms of performance of forests on OOB samples.
3. There is no pruning.

Random Forest, Breiman 2001

1. This method combines the predictions of several trees obtained from bootstrap samples of the data set.
2. In each node, only a small number (e.g $\sqrt{p}$ or $\log(p)$ of the total number p) of randomly chosen variables is taken into account to determine the best partition. This value suggested by Breiman in classification, has been confirmed by several works which showed its optimality in terms of performance of forests on OOB samples.
3. There is no pruning.

1. $\mathcal{L} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ where $X_i \in \mathcal{X}$ and $Y_i \in \mathcal{Y}$
2. For $m = 1$ to M:
   1. We consider a bootstrap sample $\mathcal{L}_m^*$ of size n from $\mathcal{L}$.
   2. We build a maximal tree T_m from $\mathcal{L}_m^*$ (without pruning).
3. Output: $f_M(x) = \text{Argmax}_y \#\{m : T_m(x) = y\}$ (classification) or $f_M(x) = \frac{1}{M} \sum_{m=1}^{M} g_m(x)$ (regression).

Figure: Random Forest

Random Forest, Breiman 2001

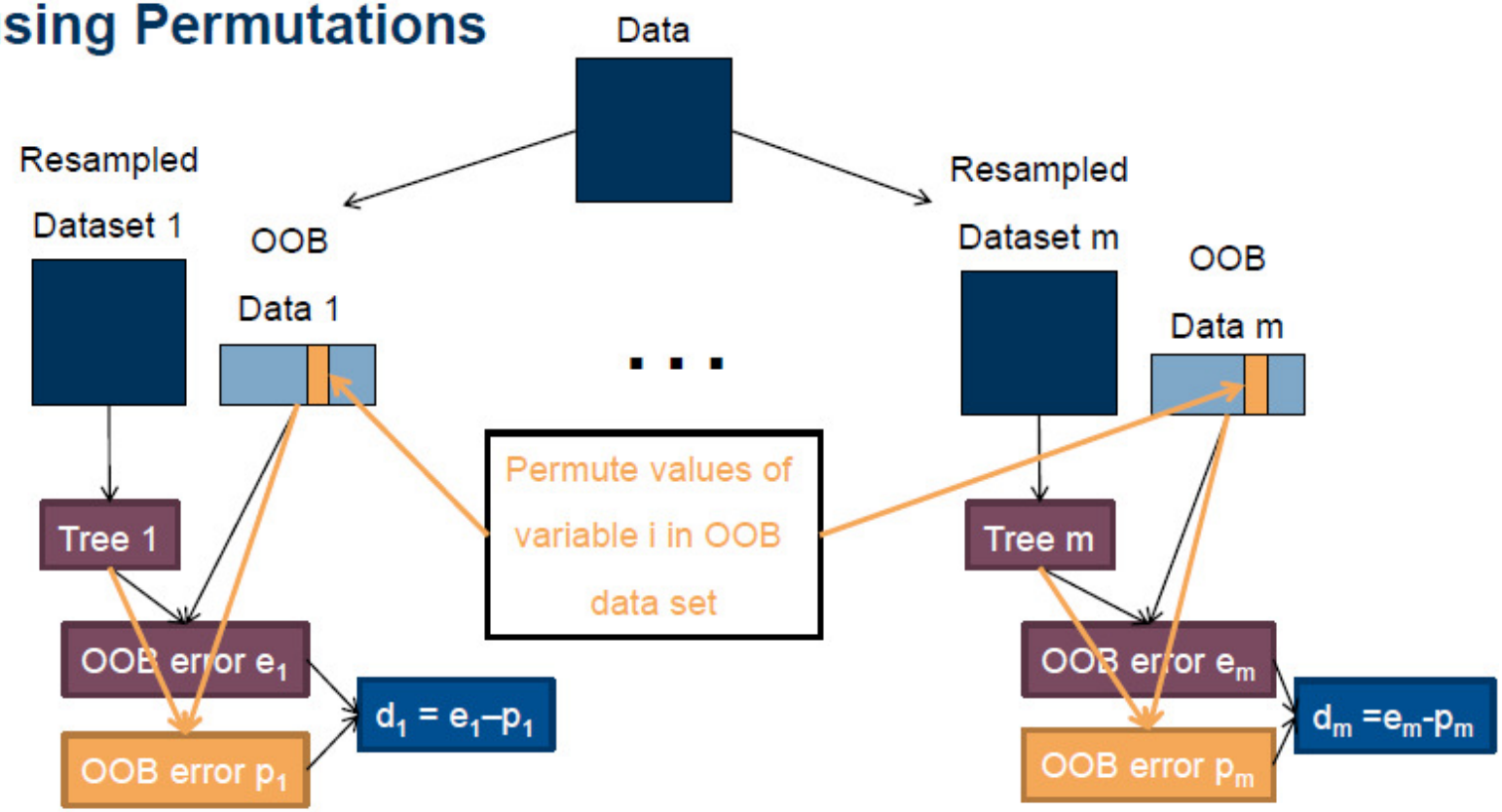
RF has a technique to determine the importance of a predictor variable that use the *out of bag* observations.

1.
   - The OOB error of tree b is:
   $$e_b = \underset{i:x_i \text{ is OOB for b}}{\text{mean}} \left(\text{error}\left(x_i, y_i\right)\right)$$
   - The OOB error is defined as
   $$\underset{i}{\text{mean}} \left(\underset{b:x_i \text{ is OOB for b}}{\text{mean}} \left(error(x_i)\right) \right)$$
   and is an estimation of the test error.
2. Consider the variable $x^{(j)}$. We permute the value for this variable in the OOB sample of tree b, and recompute :
   $$\widehat{e}_b = \underset{i:x_i \text{ is OOB for b}}{\text{mean}} \left(\text{error}\left(x_i, y_i\right)\right)$$
3. The difference between the original OOB error and the latter give an index of the importance of variable j:
   $$VI\left(x^{(j)}\right) = \frac{1}{B}\sum_{b=1}^{B}\left(\widehat{e}_b - e_b\right)$$

This index can also be based on the average decrease of another criterion, as example the Gini criterion used in the construction of trees

Random Forest, Breiman 2001

Variable Importance for variable i using Permutations



$$\left.\begin{array}{l} \overline{d} = \frac{1}{m}\sum_{i=1}^{m} d_i \\ s_d^2 = \frac{1}{m-1}\sum_{i=1}^{m}(d_i - \overline{d})^2 \end{array}\right\} \quad v_i = \frac{\overline{d}}{s_d}$$

Random Forest, Breiman 2001

Another method that is also used is the *Mean Decrease in Gini coefficient* that gives a measure of how each variable contributes to the homogeneity of the nodes. For variable $x^{(j)}$, we average on all the trees of the forest the change in impurity across all the nodes that are splitted by $x^{(j)}$:

$$VI_G(x_j) = \frac{1}{B} \sum_{b=1}^{B} \sum_{\substack{t \in b \text{ s.t.} \\ v(s_t) = x_j}} p(t) \Delta i(s_t, t)$$

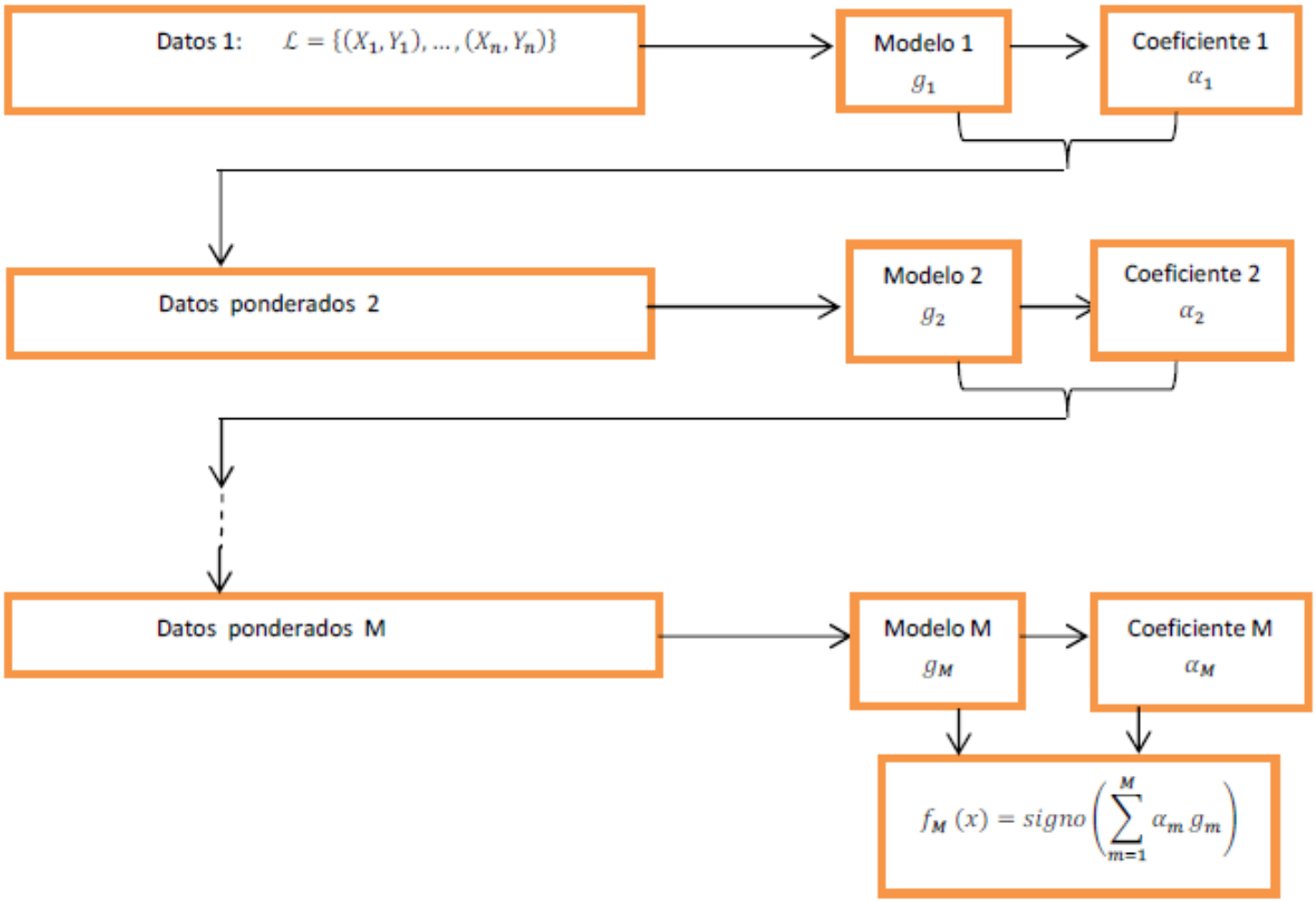
where B is the total number of trees, $p(t)$ is the proportion of observations in node t, $v(s_t)$ is the variable used in split t and $\Delta i(s_t, t)$ is the change in impurity between node t and its two child nodes for the split s_t.

Random Forest, Breiman 2001

1. By using few variables in each partition, overfitting is avoided.
2. In addition, compared to large databases with a high number of variables, the model trains more quickly than for other techniques, such as Bagging or Boosting.
3. As with Bagging, the disadvantage of this method versus CART is the loss of interpretability.
4. But clearly, much is gained in terms of the predictive power of the model.

Plan

Adaboost


Datos 1: $\mathcal{L} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$
Modelo 1
g_1
Coeficiente 1
α_1
Datos ponderados 2
Modelo 2
g_2
Coeficiente 2
α_2
Datos ponderados M
Modelo M
g_M
Coeficiente M
α_M
$f_M(x) = signo\left(\sum_{m=1}^{M} \alpha_m g_m\right)$

Adaboost (Freund and Schapire, 1997)

1. $\mathcal{L} = \{(X_1, Y_1), \ldots, (X_N, Y_N)\}$ where $X_i \in \mathcal{X}$ and $Y_i \in \{-1, 1\}$
2. Initialization of the weights: $w_1(i) = \frac{1}{N} \qquad i = 1, \ldots, N.$
3. For $t = 1$ to T:
   - From $\mathcal{L}$ and weights $w_t(i)$, we build a predictor $h_t : \mathcal{X} \to \mathcal{Y}$ that minimizes the error $\varepsilon_t = \sum_{i=1}^{N} w_t(i) 1_{\{h_t(X_i) \neq Y_i\}}$
   - Calculate $\alpha_t = \frac{1}{2} \log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$.
   - Update the weights: $w_{t+1}(i) = \frac{w_t(i)}{Z_t} \exp\left(-\alpha_t Y_i h_t(X_i)\right)$ for all $i = 1, \ldots, N$, where $Z_t = \sum_{i=1}^{n} w_t(i) \exp\left(-\alpha_t Y_i h_t(X_i)\right)$
4. Output: $f_T(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$

Figure: Adaboost, Freund and Schapire, 1997

Toy Example Schapire

	Yi	W1	h1	W2	h2	W3	h3
X1	+	0,1	0	0,07	0	0,05	1
X2	+	0,1	0	0,07	0	0,05	1
X3	-	0,1	0	0,07	1	0,17	0
X4	-	0,1	0	0,07	1	0,17	0
X5	+	0,1	1	0,166	0	0,11	0
X6	+	0,1	1	0,166	0	0,11	0
X7	-	0,1	0	0,07	1	0,17	0
X8	+	0,1	1	0,166	0	0,11	0
X9	-	0,1	0	0,07	0	0,05	0
X10	-	0,1	0	0,07	0	0,05	1

e1=0,3	e2=0,21	e3=0,15
b1=(1-0,3)/0,3=7/3	b2=(1-0,2)/0,2=4	b3=(1-0,15)/0,15
ln(b1)= 0,84	ln(b2)= 1,38	ln(b3)= 1,72

Figure: Freund and Schapire, 1997

Toy Example Schapire

Final classifier is

$$H(x) = sgn\big(0.84 \times h_1(x) + 1.38 \times h_2(x) + 1.72 \times h_3(x)\big)$$

$$= \underset{y \in \{-1,1\}}{\text{Argmax}} \left(0.84 \times \mathbb{1}_{\{h_1(x)=y\}} + 1.38 \times \mathbb{1}_{\{h_2(x)=y\}} + 1.72 \times \mathbb{1}_{\{h_3(x)=y\}}\right)$$

Final considerations for Adaboost

- Simple and easy to implement
- Single parameter: number of iterations
- It can be extended to cases in which the output variable Y is multiclass and not only for trees (any unstable algorithm).
- Detection of outliers: observations with higher weights are generally outliers.
- It is proved that the classification error on the training sample decays exponentially with the number of iterations.

Bibliography

1. Breiman, Freidman, Stone. Classification and Regression Trees, Chapman & Hall/CRC. 1984.
2. James, Witten, Hastie, Tibshirani. An introduction to Statistical Learning with application in R, Springer, 2013.
3. Hastie, Tibshirani, Friedman. The Elements of Statistical Learning, Springer, 2003.
4. Schapire, R.E and Freund, Y., *Boosting : Foundations and Algorithms*. Adaptive Computation and Machine Learning Series. Mit Press, 2012.
5. Freund, Y. and Schapire, E., A decision-theoretic generalization of on-line learning and application to boosting, *Journal of Computer and System Sciences*, 55(1): p 119-13, 1997.
6. Breiman, L., *Bagging predictors.*, Machine Learning 24, 123?140, 1996
7. Bourel, M., *Métodos de Agregación de modelos y aplicaciones*, Mémorias de trabajos de difusión científica y técnica, Vol. 10, p. 19-32, 2012.
8. Bourel, M., *Agrégation de modèles en apprentissage statistique pour l'estimation de la densité et la classification multiclasse*, Tesis de doctorado, Université Aix-Marseille, 2013.
9. Diaz, J., apuntes del curso 2013 de Aprendizaje Automático y Aplicaciones, FING, UdelaR.
10. Loh W.-Y. (2014) Fifty Years of Classification and Regression Trees, International Statistical Review, 82, pages 329348,