AGENDA:

Sesión 1 (3h):

- Introducción a la asignatura
- Teoría: Bagging
- Working Lab: EDA & Bagging

Sesión 2 (1.5h):

- Teoría: One-hot encoding & Random Forest
- Working Lab: Random Forest

Sesión 3 (3h):

- Teoría: Boosting y Ensemble methods
- Working Lab: Boosting y optimización de los modelos mediante stacking

Sesión 4 (1.5h):

- Teoría: SHAP values
- Working Lab: SHAP Analisis sobre los modelos creados

BAGGING
Bootstrap **agg**regat**ing**

Elisabet Golobardes y Angel Berian
La Salle – Universitat Ramon Llull

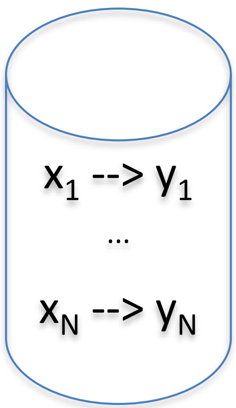
v2021.02.22

BAGGING. Concept (1/2)

- Proposed by Leo Breiman (1994)
- **Bagging** = **B**ootstrap **agg**regat**ing**
- A machine learning ensemble meta-algorithm
- Designed to improve the **stability** and **accuracy** of machine learning algorithms used in statistical classification and regression
- It also reduces **variance** and helps to avoid **overfitting**
- **Supervised learning**

BAGGING. Idea (1/2)

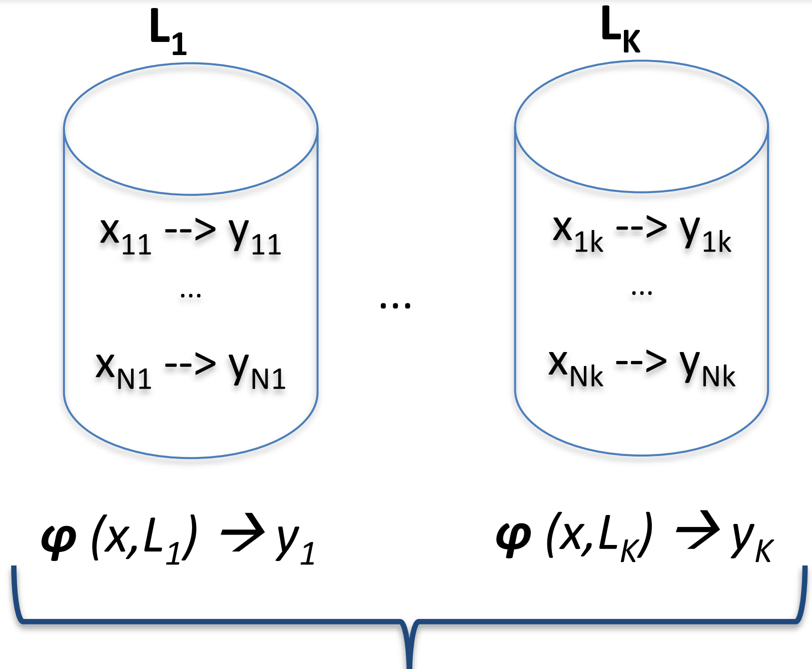
Learning Set
L



Predictor $\boldsymbol{\varphi}\ (x,L) \rightarrow y$

x_i **-->** y_i (data); where

- x_i is the input;
- y_i is the class label or a numerical response



Predictor $\boldsymbol{\varphi}\ (x,L) \rightarrow$ *"average"*

L_i **: the multiple versions** are formed by making **bootstrap** replicates of the learning set L and using these $\{L_i\}$ as new learning sets.

BAGGING. Idea (2/2)

- Are given a sequence of learning sets $\{L_K\}$ each consisting of N independents observations from the underlying distribution as L.

- **If *y* is numerical**,

  An obvious procedure is to replace $\boldsymbol{\varphi}(x,L)$ by the average of $\boldsymbol{\varphi}(x,L_K)$ over K.

- **If $\boldsymbol{\varphi}(x,L)$ predicts a class j in {1, … , J}**,

  Then one method of aggregating the $\boldsymbol{\varphi}(x,L_K)$ is by voting. Let $N_j = \#\{k; \boldsymbol{\varphi}(x,L_K) = j\}$ and take $\boldsymbol{\varphi}_{\boldsymbol{A}}(x) = argmax_j N_j$

Bagging **Classification** Trees. Computations (1/2)

In all runs [Breiman, 1994]:

1. The data set D is randomly divided into a test set T and learning set L.

1. A classification tree is constructed from L, with selection done by 10-fold cross-validation. Running the test set T down this tree gives the missclassification rate $e_S(L,T)$.

1. A bootstrap sample L_B is selected from L, and a tree grown using L_B and 10-fold cross-validation. This is repeated Z times giving tree classifiers $\boldsymbol{\varphi_1}(x)$, … , $\boldsymbol{\varphi_Z}(x)$.

Bagging **Classification** Trees. Computations (2/2)

In all runs [Breiman, 1994] (continuation):

4. If (j_n,x_n) in T, then the estimated class of x_n is that class having the plurality in $\boldsymbol{\varphi_1}(x_n), \ldots, \boldsymbol{\varphi_Z}(x_n)$. The proportion of times the estimated class differs from the true class is the bagging missclassification rate $e_B(L,T)$.

5. The random division of the data is repeated M times and the reported $\bar{e}_S$, $\bar{e}_B$ are the average over the M iterations.

Note: for instance, Z=50 and M=100.

Bagging **Regression** Trees. Computations

In all runs [Breiman, 1994]:

1. The data set D is randomly divided into a test set T and learning set L.

1. A regression tree is constructed from L, with selection done by 10-fold cross-validation. Running the test set T down this tree gives mean-squared-error $e_S(L,T)$.

1. A bootstrap sample L_B is selected from L, and a tree grown using L_B and 10-fold cross-validation. This is repeated Z times giving predictors (regression trees) $\boldsymbol{\varphi}_{\mathbf{1}}(x)$, … , $\boldsymbol{\varphi}_{\mathbf{Z}}(x)$.

Bagging **Regression** Trees. Computations (2/2)

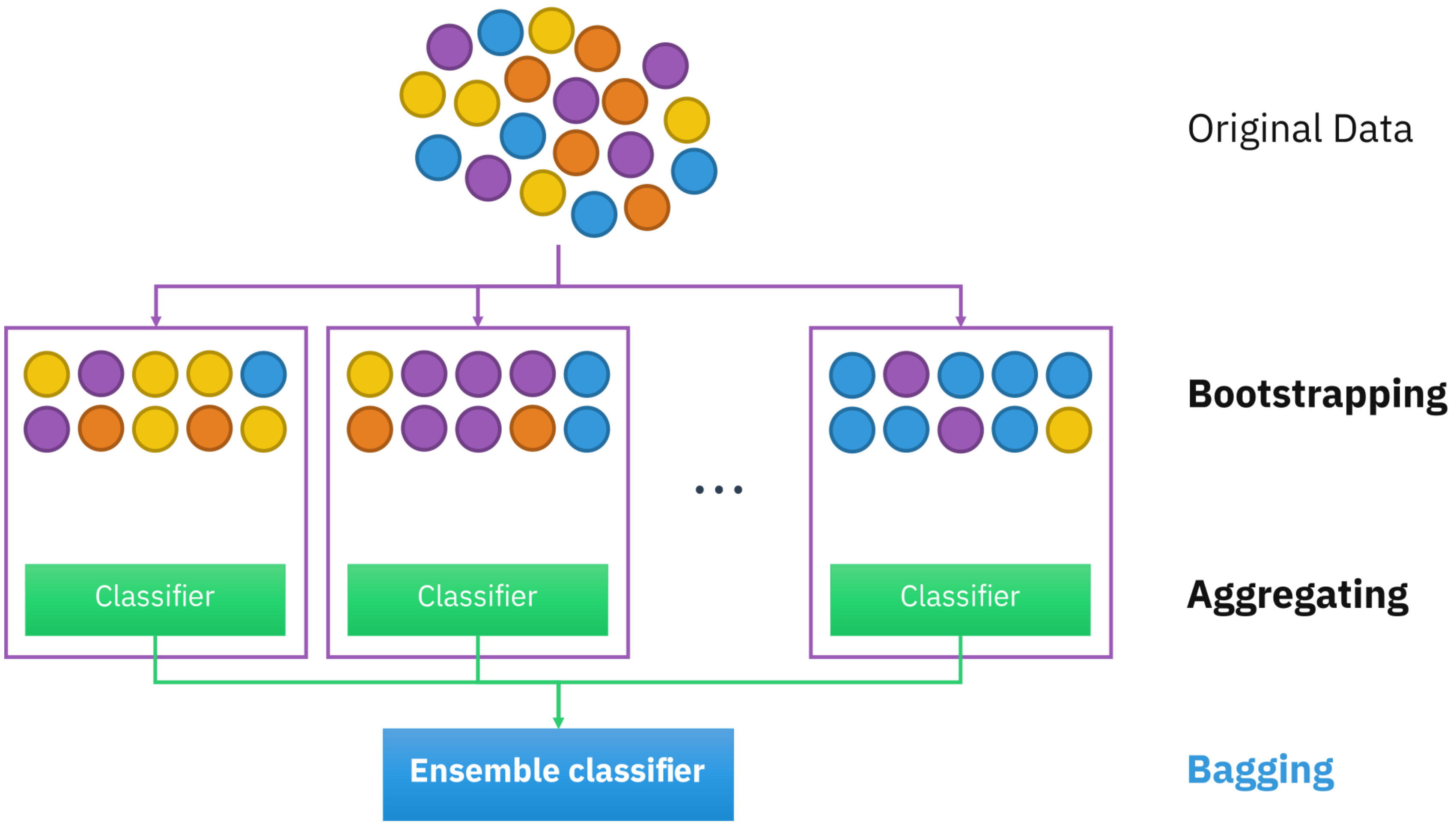
In all runs [Breiman, 1994] (continuation):

4. For each (y_n,x_n) in T, the predicted $\hat{y}_B$ value is taken as $av_k \varphi_k(x_n)$. Then $e_B(L,T)$ is the mean-squared-error between $\hat{y}_B$ and the true *y*-values in T ($= av_k(y_B - \hat{y}_B)^2$).

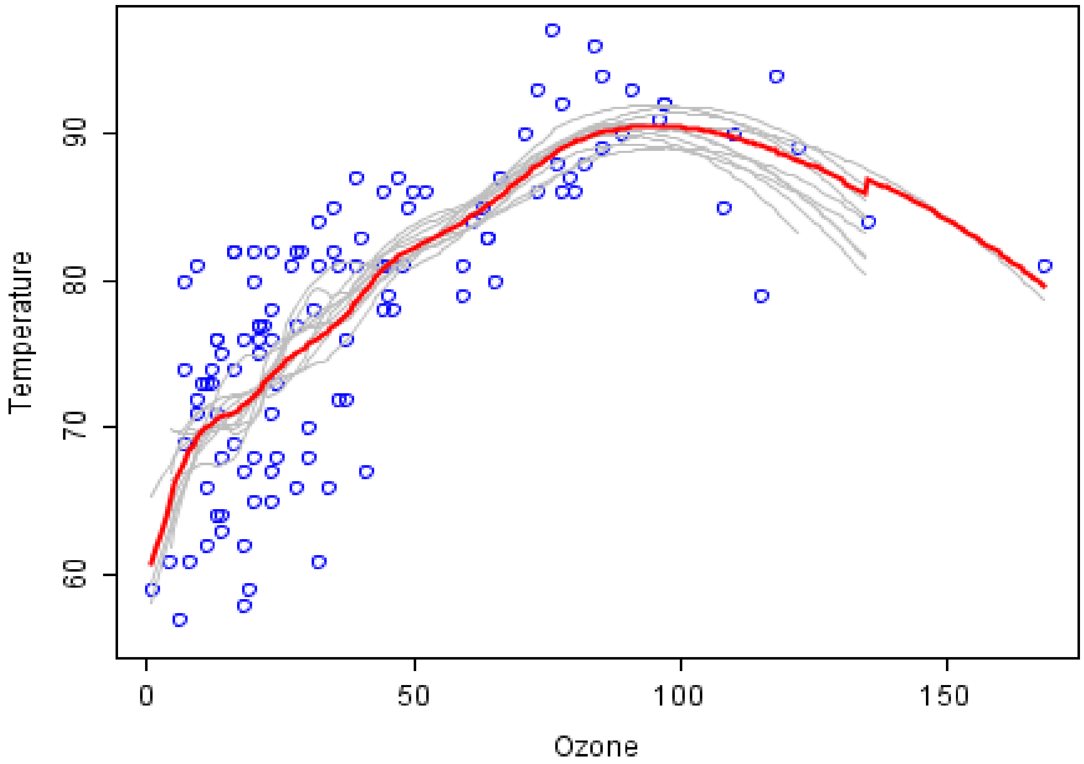
5. This procedure is repeated M times and the errors averaged to give the single tree error $\bar{e}_S$ and the bagged error $\bar{e}_B$.

Note: for instance, Z=25 and M=100.

Flow Chart of the Bagging Algorithm



BAGGING. Example with Ozone data



Measuring the relationship between Ozone concentration and Temperature using 100 iterations bagging approach

When does Bagging work?

- Learning algorithm is unstable: if small changes to the training set cause large changes in the learned classifier.
- If the learning algorithm is unstable, then Bagging almost always improves performance
- Datasets with high variance of the instances

BAGGING. Summarising (1/2)

- In statistics, **bootstrapping** is any test or metric that relies on random sampling with replacement.
- In Bagging, the multiple versions are formed by making bootstrap replicates of the learning set and using these as new learning sets.

BAGGING. Summarising (2/2)

- Bagging predictor is a method for generating multiple versions of a predictor and using these to get an aggregated predictor.
- The aggregation average over the versions when predicting a numerical outcome and does a plurality vote when predicting a class.
- Bagging is a special case of the model averaging approach.

References

- **[Breiman, 1994]** Breiman, Leo (1994). “Bagging Predictors”. Technical Report No. 421. University of California.
- **[Breiman, 1996]** Breiman, Leo (1996). "Bagging predictors". Machine Learning. 24 (2): 123–140.
- **[Efron and Tibshirani, 1994]** Efron, B.; Tibshirani, R. (1994). “An introduction to the bootstrap”. New York: Chapman & Hall.

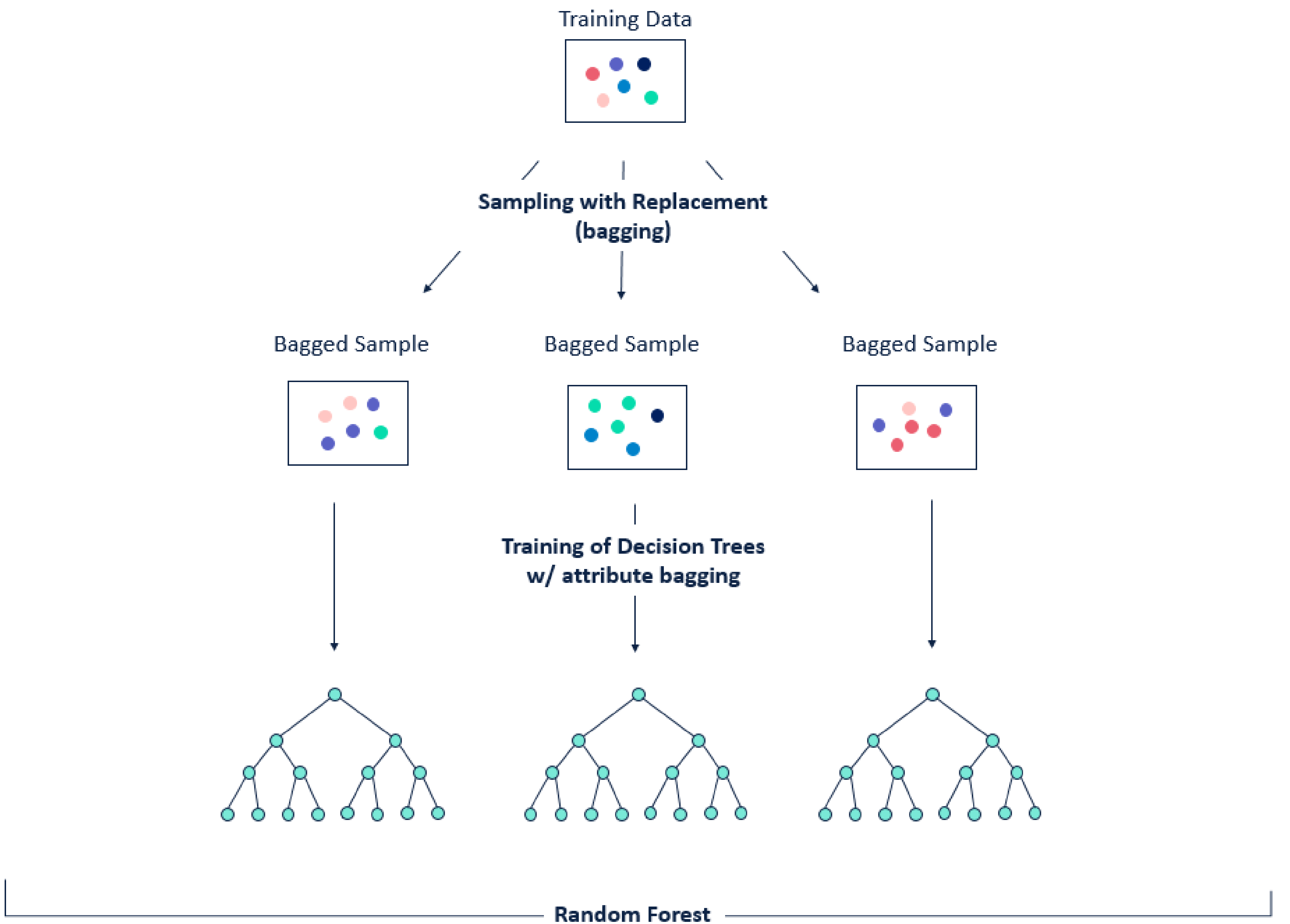
Random Forest Algorithm

Angel Berian
La Salle – Universitat Ramon Llull
2021.02.22

The Random Forests Algorithm

- **Developed by Leo Breiman (2001)**

- **Description:**
  - Given a training set S
  - For i = 1 to k do:
    - Build subset S_i by sampling with replacement from S
    - Learn tree T_i from S_i
      - At each node:
        - Choose best split from random subset of F features
      - Each tree grows to the largest extend, and no pruning
  - Make predictions according to majority vote of the set of k trees.



Source: https://community.alteryx.com/t5/Alteryx-Designer-Knowledge-Base/Seeing-the-Forest-for-the-Trees-An-Introduction-to-Random-Forest/ta-p/158062

Features of Random Forests

- It runs efficiently on large data bases.
- It can handle thousands of input variables without variable deletion.
- It gives estimates of what variables are important in the classification.
- It generates an internal unbiased estimate of the generalization error as the forest building progresses.
- It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.
- It has methods for balancing error in class population unbalanced data sets.

Features of Random Forests

- Generated forests can be saved for future use on other data.
- Prototypes are computed that give information about the relation between the variables and the classification.
- It offers an experimental method for detecting variable interactions.
- **Highly susceptible to correlation between features**