**Bagging and Boosting are both ensemble methods in Machine Learning, but what’s the key behind them?**

* bagging on weak learners which exhibit high variance and low bias, (NO PRECISO, EXACTO)
* boosting on low variance and high bias is observed. (PRECISO, NO EXACTO)

Bagging and Boosting are similar in that they are both **ensemble techniques**, where a set of weak learners are combined to create a strong learner that obtains better performance than a single one. So, let’s start from the beginning:

**What is an ensemble method?**

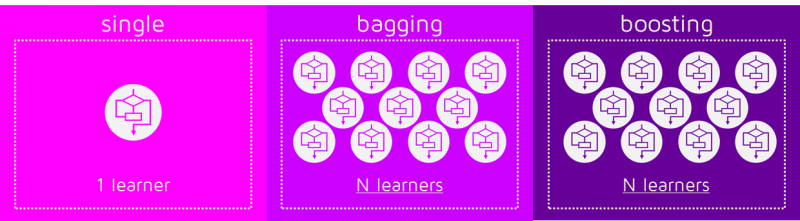
To train **multiple models** using the same learning algorithm.

The ensembles take part in a bigger group of methods, called **multiclassifiers,** where a set of hundreds or thousands of learners with a common objective are fused together to solve the problem.

The second group of multiclassifiers contain **the hybrid methods**. They use a set of learners too, but they can be trained using different learning techniques. Stacking is the most well-known. If you want to learn more about Stacking, you can read my previous post, “**[Dream team combining classifiers](https://quantdare.com/dream-team-combining-classifiers/" \t "_blank)**“.

The main causes of error in learning are due to **noise, bias and variance**. Ensemble helps to minimize these factors. These methods are designed to improve the stability and the accuracy of Machine Learning algorithms. Combinations of multiple classifiers decrease variance, especially in the case of unstable classifiers, and may produce a more reliable classification than a single classifier.

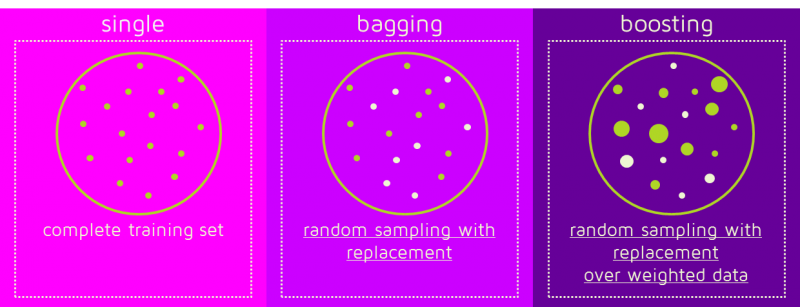
To use Bagging or Boosting you must select a base learner algorithm. For example, if we choose a classification tree, Bagging and Boosting would consist of a pool of trees as big as we want.



**How do Bagging and Boosting get N learners?**

Bagging and Boosting get N learners by generating additional data in the training stage. N new training data sets are produced by **random sampling with replacement** from the original set. By sampling with replacement some observations may be repeated in each new training data set.

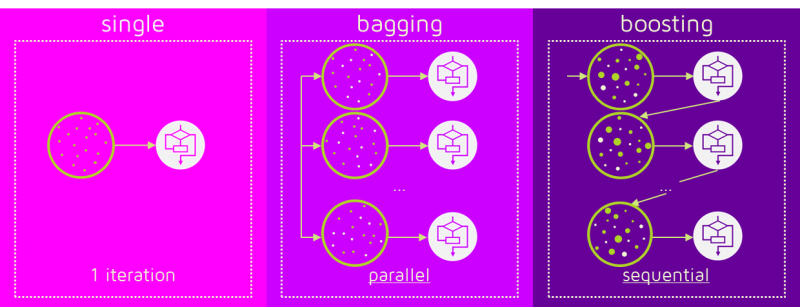
In the case of Bagging, any element has the same probability to appear in a new data set. However, for Boosting the observations are weighted and therefore some of them will take part in the new sets more often:



These multiple sets are used to train the same learner algorithm and therefore different classifiers are produced.

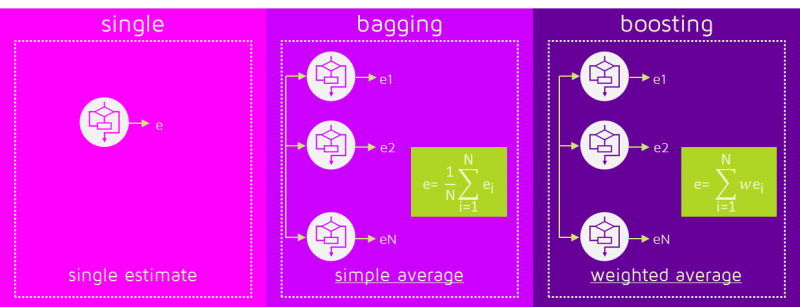
**Why are the data elements weighted?**

At this point, we begin to deal with the main difference between the two methods. While the training stage is parallel for Bagging (i.e., each model is built independently), Boosting builds the new learner in a sequential way

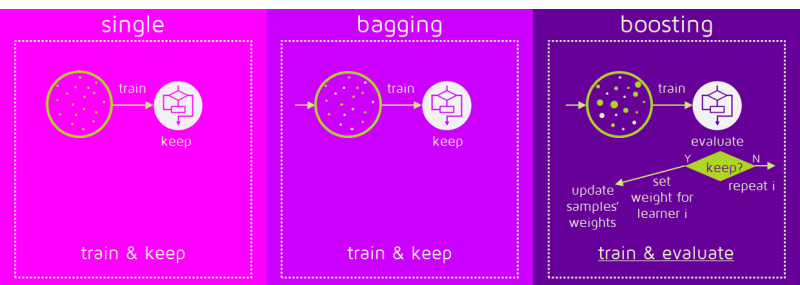
:

In Boosting algorithms each classifier is trained on data, taking into account the previous classifiers’ success. After each training step, the weights are redistributed. M**isclassified data increases its weights** to emphasise the most difficult cases. In this way, subsequent learners will focus on them during their training.

**How does the classification stage work?**

To predict the class of new data we only need to **apply the N learners to the new observations.** In Bagging the result is obtained by averaging the responses of the N learners (or majority vote). However, Boosting assigns a second set of weights, this time for the N classifiers, in order to take a **weighted average** of their estimates.

In the Boosting training stage, the algorithm allocates weights to each resulting model. A learner with good a classification result on the training data will be assigned a higher weight than a poor one. So when evaluating a new learner, Boosting needs to keep track of learners’ errors, too. Let’s see the differences in the procedures:



Some of the Boosting techniques include an extra-condition to keep or discard a single learner. For example, in AdaBoost, the most renowned, an error less than 50% is required to maintain the model; otherwise, the iteration is repeated until achieving a learner better than a random guess.

The previous image shows the general process of a Boosting method, but several alternatives exist with different ways to determine the weights to use in the next training step and in the classification stage. Click here if you like to go into detail: **[AdaBoost](https://en.wikipedia.org/wiki/AdaBoost" \t "_blank)**, **[LPBoost](https://en.wikipedia.org/wiki/LPBoost" \t "_blank)**, **[XGBoost](http://arxiv.org/pdf/1603.02754v1.pdf" \t "_blank)**, **[GradientBoost](https://en.wikipedia.org/wiki/Gradient_boosting" \t "_blank)**, **[BrownBoost](https://en.wikipedia.org/wiki/BrownBoost" \t "_blank)**.

**Which is the best, Bagging or Boosting?**

There’s not an outright winner; it depends on the data, the simulation and the circumstances.

Bagging and Boosting decrease the variance of your single estimate as they combine several estimates from different models. So the result may be a model with **higher stability**.

If the problem is that the single model gets a very low performance, Bagging will rarely get a **better bias**. However, Boosting could generate a combined model with lower errors as it optimises the advantages and reduces pitfalls of the single model.

By contrast, if the difficulty of the single model is **over-fitting**, then Bagging is the best option. Boosting for its part doesn’t help to avoid over-fitting; in fact, this technique is faced with this problem itself. For this reason, Bagging is effective more often than Boosting.

**To sum up:**

|  |  |  |
| --- | --- | --- |
| **Similarities** |  | **Differences** |
| Both are ensemble methods to get N learners from 1 learner… |  | … but, while they are built independently for Bagging, Boosting tries to add new models that do well where previous models fail. |
| Both generate several training data sets by random sampling… |  | … but only Boosting determines weights for the data to tip the scales in favor of the most difficult cases. |
| Both make the final decision by averaging  the N learners (or taking the majority of them)… |  | … but it is an equally weighted average for Bagging and a weighted average for Boosting, more weight to those with better performance on training data. |
| Both are good at reducing variance and provide higher stability… |  | … but only Boosting tries to reduce bias. On the other hand, Bagging may solve the over-fitting problem, while Boosting can increase it. |