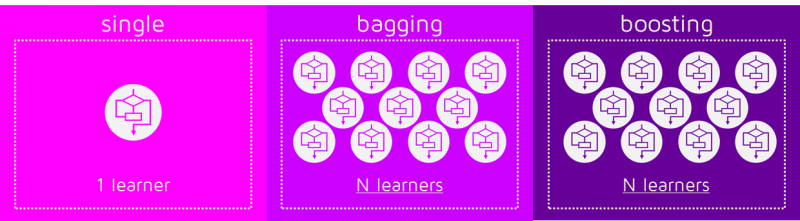
## Bagging Vs Boosting

## 1. Introduction to Ensemble Learning

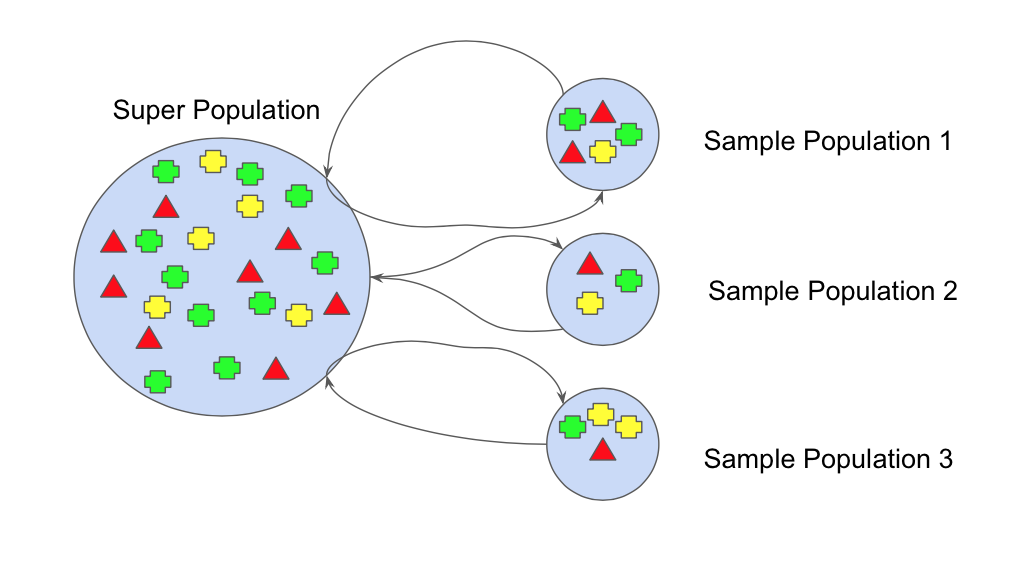
* **Bagging** and **boosting** are both ensemble learning methods in machine learning.
* **Bagging** and **boosting** are similar in that they are both ensemble techniques, where a set of weak learners are combined to create a strong learner that obtains better performance than a single one.
* Ensemble learning helps to improve machine learning model performance by combining several models. This approach allows the production of better predictive performance compared to a single model.
* The basic idea behind ensemble learning is to learn a set of classifiers (experts) and to allow them to vote. This diversification in Machine Learning is achieved by a technique called **ensemble learning**. The idea here is to train multiple models, each with the objective to predict or classify a set of results.
* **Bagging** and **boosting** are two types of ensemble learning techniques. These two decrease the variance of single estimate as they combine several estimates from different models. So the result may be a model with higher stability.
* The main causes of error in learning are due to **noise, bias and variance**. Ensemble helps to minimize these factors. By using ensemble methods, we’re able to increase the stability of the final model and reduce the errors mentioned previously.
* **Bagging helps to decrease the model’s variance.**
* **Boosting helps to decrease the model’s bias.**
* 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 as shown in the following diagram:



Before understanding bagging and boosting and how different classifiers are selected in the two algorithms, we need to first understand about **Bootstrapping.**

## Bootstrapping

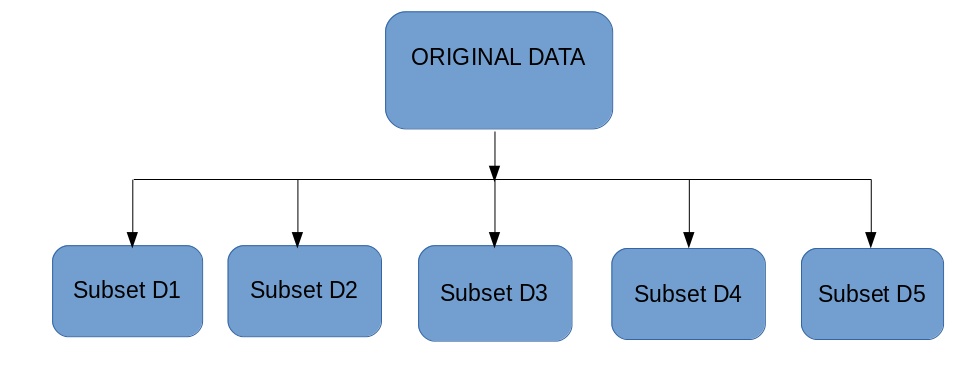
* **Bootstrap** refers to random sampling with replacement. Bootstrap allows us to better understand the bias and the variance with the dataset.
* So, **Bootstrapping** is a sampling technique in which we create subsets of observations from the original dataset with replacement. The size of the subsets is the same as the size of the original set.
* Bootstrap involves random sampling of small subset of data from the dataset. This subset can be replaced.
* The selection of all the example in the dataset has equal probability. This method can help to better understand the mean and standand deviation from the dataset.
* Let’s assume we have a sample of ‘n’ values (x) and we want an estimate of the mean of the sample. We can calculate it as follows:
* mean(x) = 1/n \* sum(x)
* Bootstrapping can be represented diagrammatically as follows:



Now, we turn our attention to bagging and boosting.

## Bagging

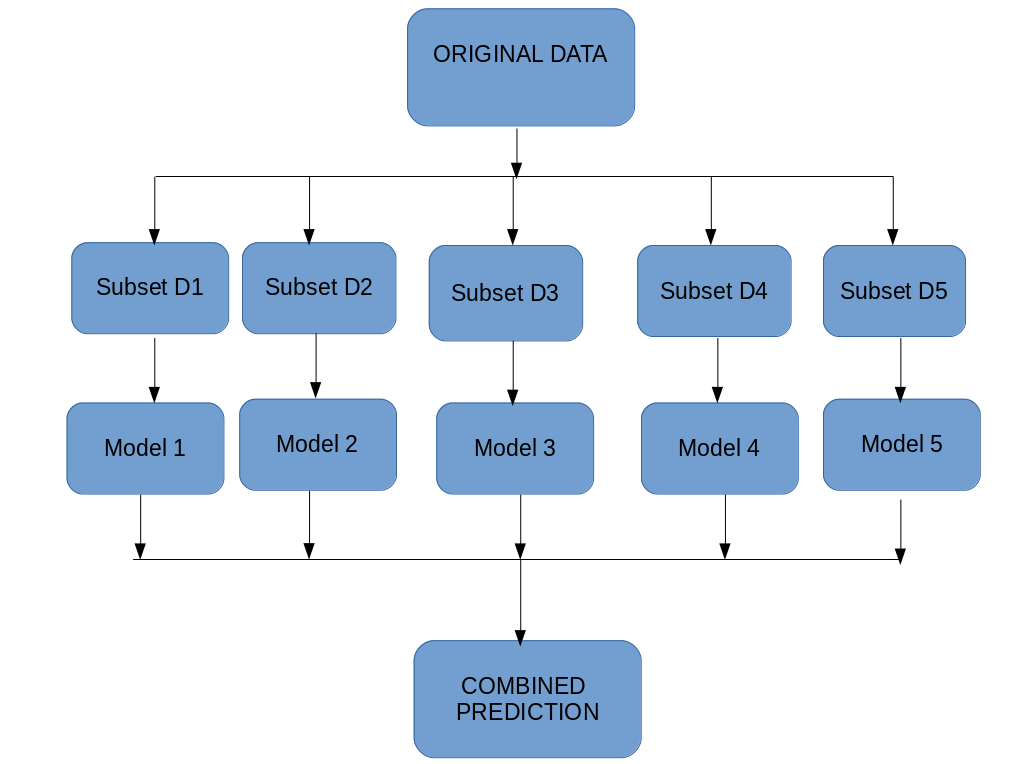
* Bagging ( or Bootstrap Aggregation), is a simple and very powerful ensemble method. Bagging is the application of the Bootstrap procedure to a high-variance machine learning algorithm, typically decision trees.
* The idea behind bagging is combining the results of multiple models (for instance, all decision trees) to get a generalized result. Now, bootstrapping comes into picture.
* Bagging (or Bootstrap Aggregating) technique uses these subsets (bags) to get a fair idea of the distribution (complete set). The size of subsets created for bagging may be less than the original set.
* It can be represented as follows:



**Bagging** works as follows:-

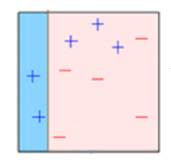
1. Multiple subsets are created from the original dataset, selecting observations with replacement.
2. A base model (weak model) is created on each of these subsets.
3. The models run in parallel and are independent of each other.
4. The final predictions are determined by combining the predictions from all the models.

Now, bagging can be represented diagrammatically as follows:

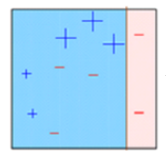


## Boosting

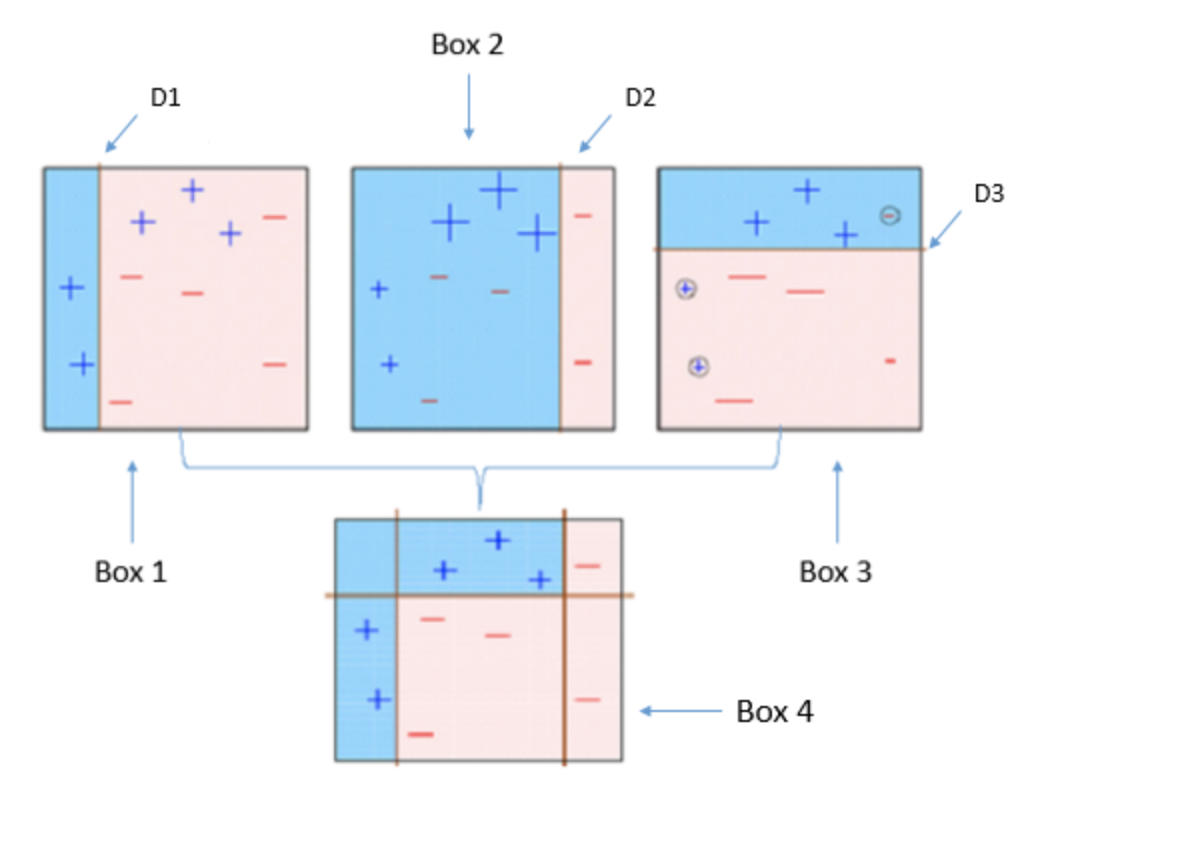
* Boosting is a sequential process, where each subsequent model attempts to correct the errors of the previous model. The succeeding models are dependent on the previous model.
* In this technique, learners are learned sequentially with early learners fitting simple models to the data and then analyzing data for errors. In other words, we fit consecutive trees (random sample) and at every step, the goal is to solve for net error from the prior tree.
* When an input is misclassified by a hypothesis, its weight is increased so that next hypothesis is more likely to classify it correctly. By combining the whole set at the end converts weak learners into better performing model.
* Let’s understand the way boosting works in the below steps.
  + A subset is created from the original dataset.
  + Initially, all data points are given equal weights.
  + A base model is created on this subset.
  + This model is used to make predictions on the whole dataset.



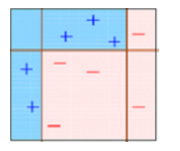
1. Errors are calculated using the actual values and predicted values.
2. The observations which are incorrectly predicted, are given higher weights. (Here, the three misclassified blue-plus points will be given higher weights)
3. Another model is created and predictions are made on the dataset. (This model tries to correct the errors from the previous model)



1. Similarly, multiple models are created, each correcting the errors of the previous model.
2. The final model (strong learner) is the weighted mean of all the models (weak learners).

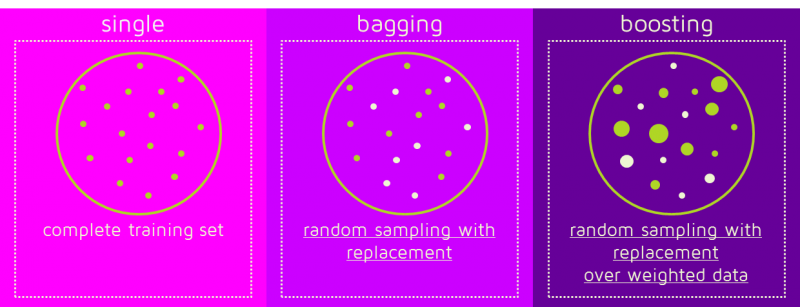


* Thus, the boosting algorithm combines a number of weak learners to form a strong learner.
* The individual models would not perform well on the entire dataset, but they work well for some part of the dataset.
* Thus, each model actually boosts the performance of the ensemble.



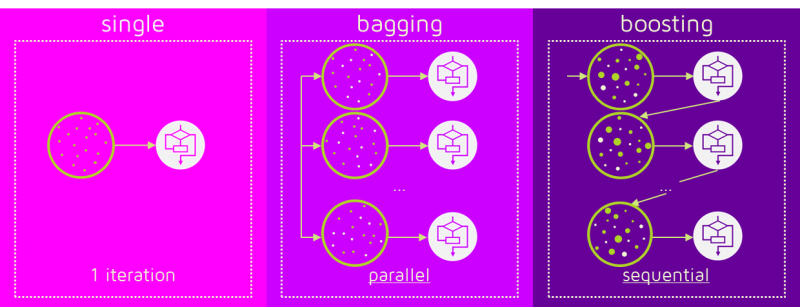
## Getting N learners for Bagging and Boosting

* 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.
* This is represented diagrammatically as follows:



## Weighted data elements

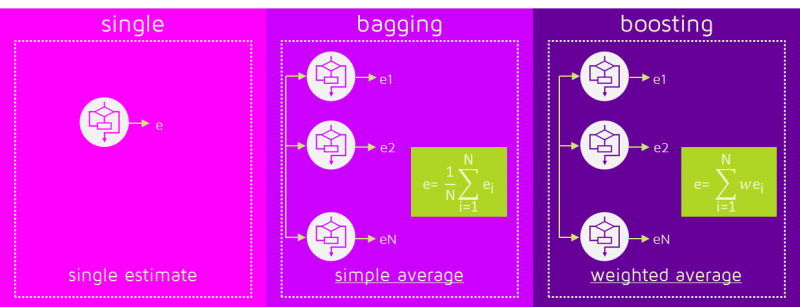
* Now, we know the main difference between the two methods.
* While the training stage is parallel for Bagging (i.e., each model is built independently), Boosting builds the new learner in a sequential way as follows:



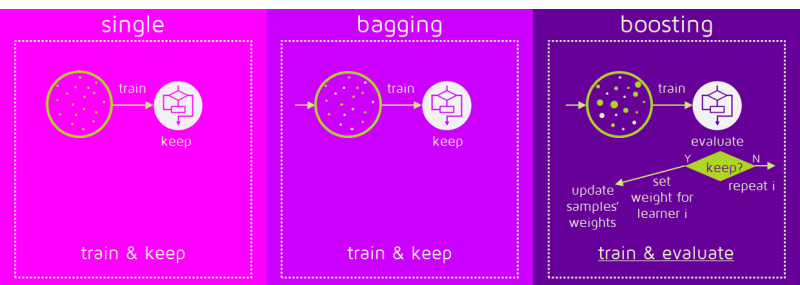
* In Boosting algorithms each classifier is trained on data, taking into account the previous classifiers’ success.
* After each training step, the weights are redistributed. Misclassified data increases its weights to emphasise the most difficult cases.
* In this way, subsequent learners will focus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

## Classification stage in action

* 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.
* This is shown diagrammatically below:



* 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.

## Selecting the best technique- Bagging or Boosting

* Now, the question may come to our mind - whether to select Bagging or Boosting for a particular problem.
* 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.

## Similarities between Bagging and Boosting

Similarities between Bagging and Boosting are as follows:-

* Both are ensemble methods to get N learners from 1 learner.
* Both generate several training data sets by random sampling.
* Both make the final decision by averaging the N learners (or taking the majority of them i.e Majority Voting).
* Both are good at reducing variance and provide higher stability.

## Differences between Bagging and Boosting

Differences between Bagging and Boosting are as follows:-

* **Bagging** is the simplest way of combining predictions that belong to the same type while **Boosting** is a way of combining predictions that belong to the different types.
* **Bagging** aims to decrease variance, not bias while **Boosting** aims to decrease bias, not variance.
* In **Baggiing** each model receives equal weight whereas in **Boosting** models are weighted according to their performance.
* In **Bagging** each model is built independently whereas in **Boosting** new models are influenced by performance of previously built models.
* In **Bagging** different training data subsets are randomly drawn with replacement from the entire training dataset. In **Boosting** every new subsets contains the elements that were misclassified by previous models.
* **Bagging** tries to solve over-fitting problem while **Boosting** tries to reduce bias.
* If the classifier is unstable (high variance), then we should apply **Bagging**. If the classifier is stable and simple (high bias) then we should apply **Boosting**.
* **Bagging** is extended to Random forest model while **Boosting** is extended to **Gradient boosting**.