**Ensemble Models**

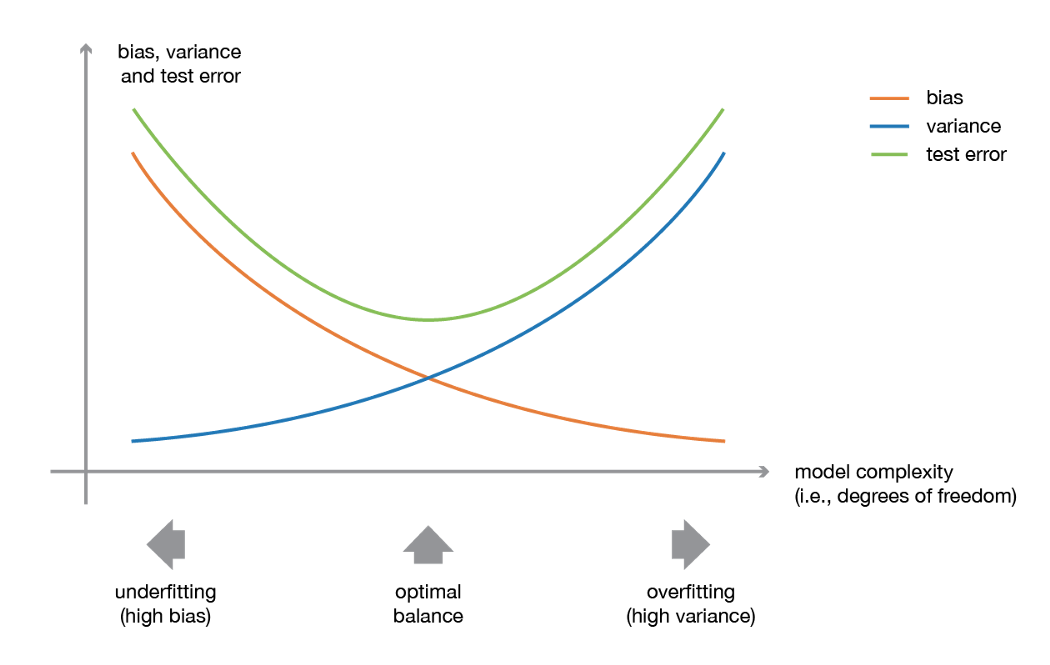
**What are ensemble methods?**

Ensemble learning is a machine learning paradigm where multiple models (often called “weak learners”) are trained to solve the same problem and combined to get better results. The main hypothesis is that when weak models are correctly combined, we can obtain more accurate and/or robust models.

**Single weak learner**

In machine learning, no matter if we are facing a classification or a regression problem, the choice of the model is extremely important to have any chance to obtain good results. This choice can depend on many variables of the problem: quantity of data, dimensionality of the space, distribution hypothesis…

A low bias and a low variance, although they most often vary in opposite directions, are the two most fundamental features expected for a model. Indeed, to be able to “solve” a problem, we want our model to have enough degrees of freedom to resolve the underlying complexity of the data we are working with, but we also want it to have not too much degrees of freedom to avoid high variance and be more robust. This is the well-known bias-variance trade-off.



**Major kinds of meta-algorithms that aims at combining weak learners:**

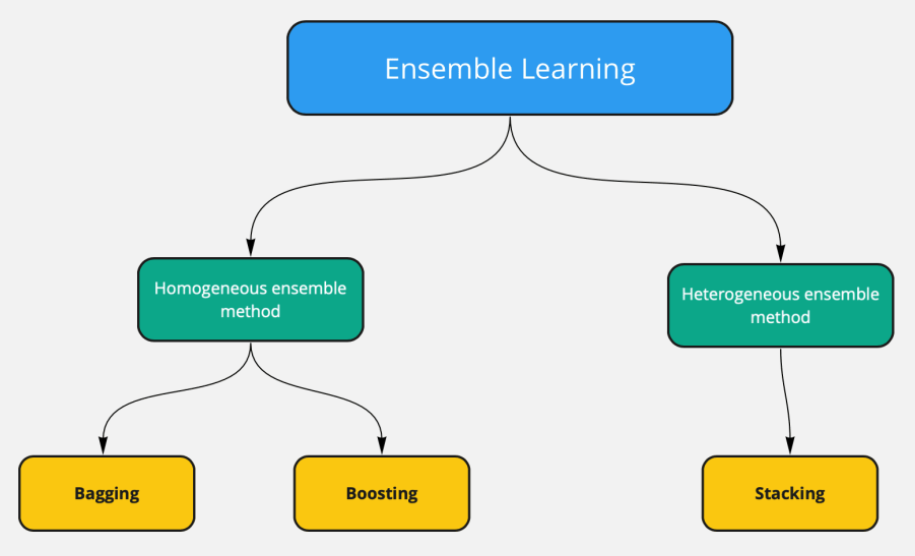
**Bagging** (**B**ootstrap & **Agg**regat**ing**), that often considers homogeneous weak learners, learns them independently from each other in parallel and combines them following some kind of deterministic averaging process

**Boosting**, that often considers homogeneous weak learners, learns them sequentially in a very adaptative way (a base model depends on the previous ones) and combines them following a deterministic strategy

**Stacking**, that often considers heterogeneous weak learners, learns them in parallel and combines them by training a meta-model to output a prediction based on the different weak models’ predictions.

**Basic Ensemble Techniques**, (1) Max Voting (2) Averaging (3) Weighted Average

**Max Voting:** The max voting method is generally used for classification problems. In this technique, multiple models are used to make predictions for each data point. The predictions by each model are considered as a ‘vote’. The predictions which we get from the majority of the models are used as the final prediction.

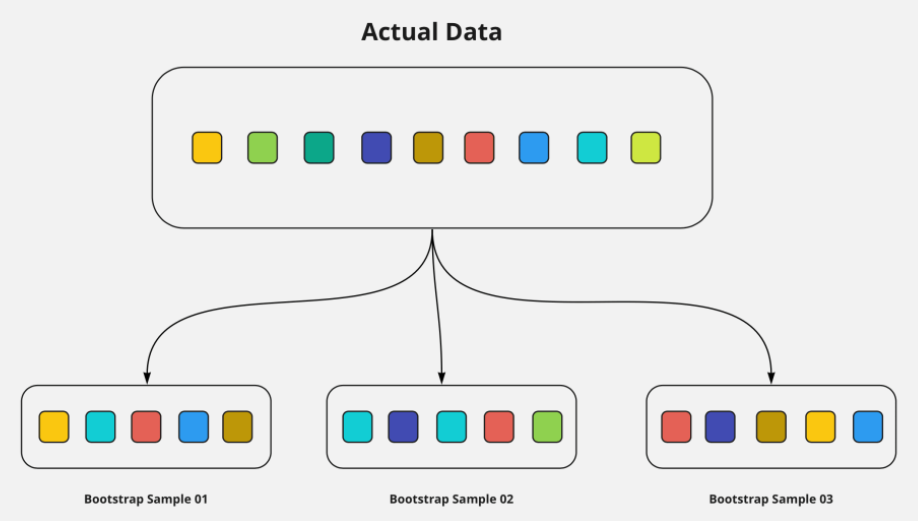


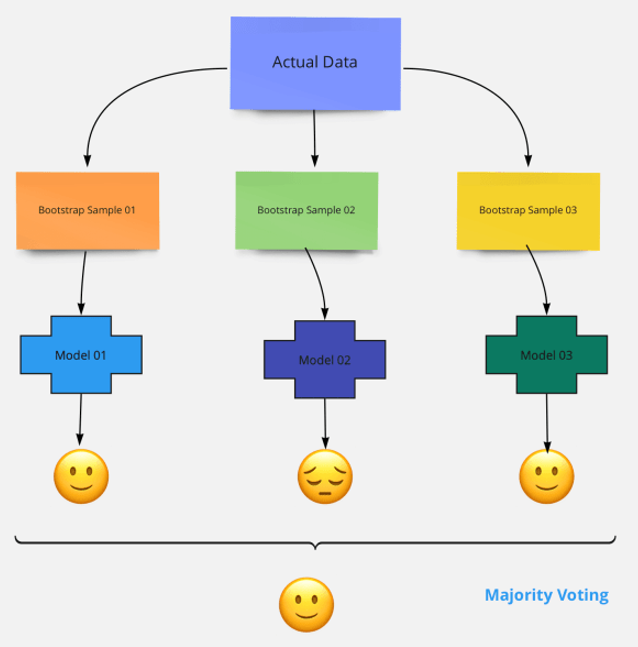
The primary question is "**what advantage will we get with ensemble learning?**"

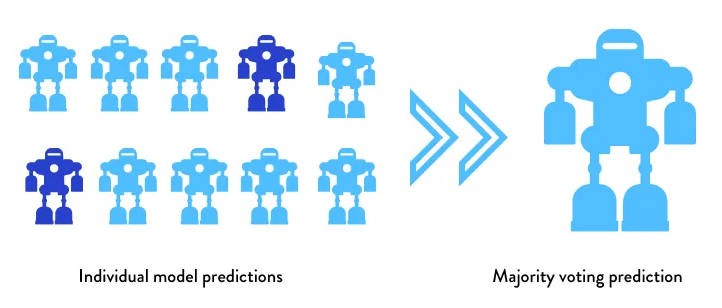
In single model approach, if the build model is having **high bias or high variance,** we will be limited to that. Even though we are having methods to handle **high bias or high variance**. Still if the final is facing any of the bias or variance issues, we can’t do anything.

Whereas if we build multiple models, we can **reduce** the high variance and high bias issue by averaging all models. If the individual models are having high bias, then when we build multiple models, the high bias will average out. The same is true for high variance cases too.

**In short:** The ensemble learning means instead of building a single model for prediction. We will build multiple machine learning models; we call these models as weak learners. A combination of all weak learners makes the strong learner, which generalizes to predict all the target classes with a decent amount of accuracy.







**Pros:** Bagging helps in reducing the overfitting. As we are averaging all the models’ outputs using the majority voting approach.

**Cons:** For regression models the predicted value won’t be optimised. if any one of the models is deviating more as the output value will be average of all the models.

**Bagging Algorithms**

Below are the list algorithms that fall under bagging.

* Random forest algorithm

**What is Boosting?**

In boosting all the individual models will build one after the other. Each model output will pass as input to the next model along with next model bootstrap sample data.

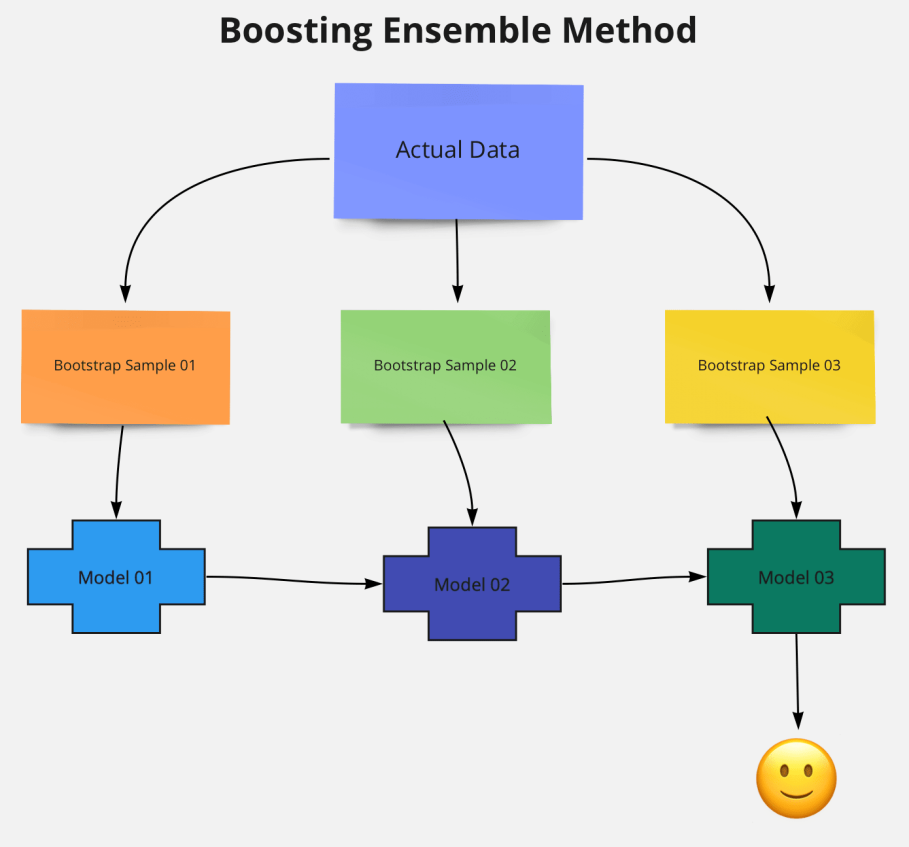
In bagging the models are built parallel so we don’t know what the error of each model is. Whereas in boosting once the first model built, we know the error of that model. So, when we pass this first model to the next model the intention is to reduce the error further. In some boosting algorithm, each model has to reduce a minimum of 50% of error.

Pros:

* Boosting reduces the bias, as each model tries to reduce the errors of the previous model in the sequential chain.
* We can use multiple loss functions to reduce the error in each sequential model.

Cons:

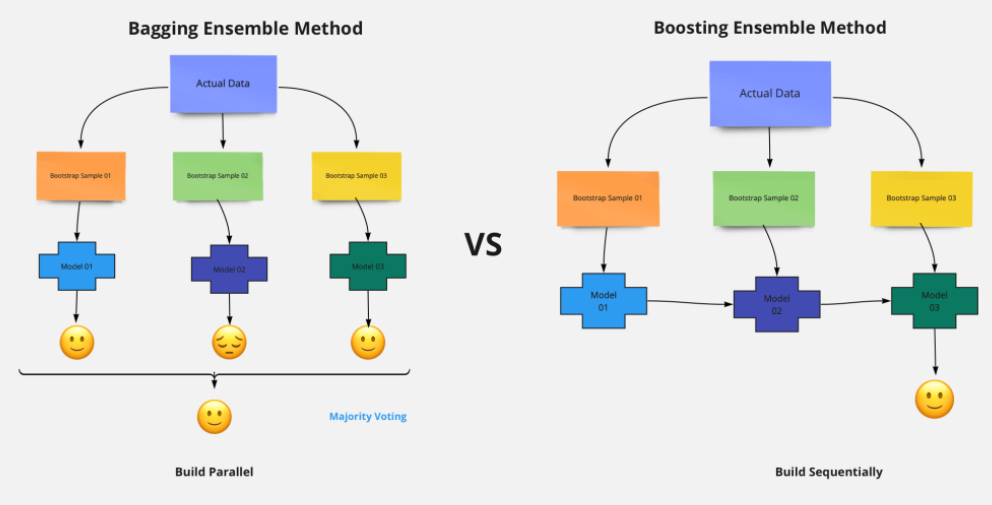
* Boosting method tends to over-fitting, As the models build sequentially, all the models try to reduce the train error.
* We need special care in turning the hyper-parameters.



**Boosting Algorithms**

Below is the list of algorithms that fall under boosting.

* Adaboost algorithm
* Xgbm algorithm
* LightGBM
* CatBoost
* LPBoost
* GradientBoost
* BrownBoost



**Which one is better? Boosting Or Bagging?**

Now the final question - Which one to choose for modelling then?

This depends on the problem. Sometimes for selecting the final method we need to have a look at each method's advantages and disadvantages.

Let’s say if the individual models are getting low model performance, then in the bagging the combination of all the low performance models will lead to the low performing model.

Whereas if the individual models are overfitting, then the final model with the boosting method will lead to an overfitting model, in such case we can use the bagging method.

So, the final conclusion is we don’t have any hard rule for which method to use but in most cases bagging methods will outperform well than the boosting methods. The main problem with boosting methods is that they tend to overfit the data.