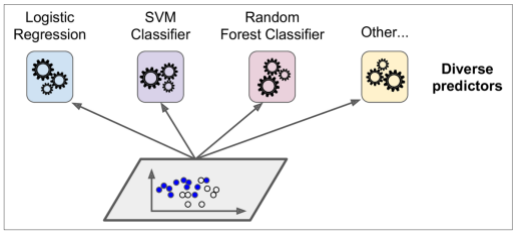
7. Ensemble Learning and Random Forests

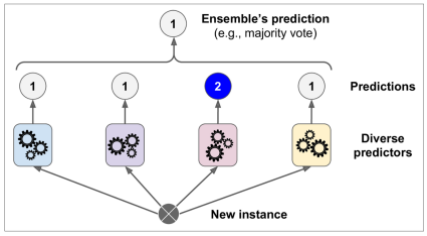
Suppose you pose a complex question to thousands of random people, then aggregate their answers. In many cases you will find that this aggregated answer is better than an expert's answer. This is called the wisdom of the crowd. Similarly, if you aggregate the predictions of a group of predictors (such as classifiers or regressors), you will often get better predictions than with the best individual predictor. A group of predictors is called an ensemble; thus, this technique is called Ensemble Learning, and an Ensemble Learning algorithm is called an Ensemble method.

Example of Ensemble Learning:

You can train a group of Decision Tree classifiers, each on a different random subset of the training set. To make predictions, you obtain the predictions of all the individual trees, then predict the class that gets the most votes, this is called Random Forest.

Voting Classifiers



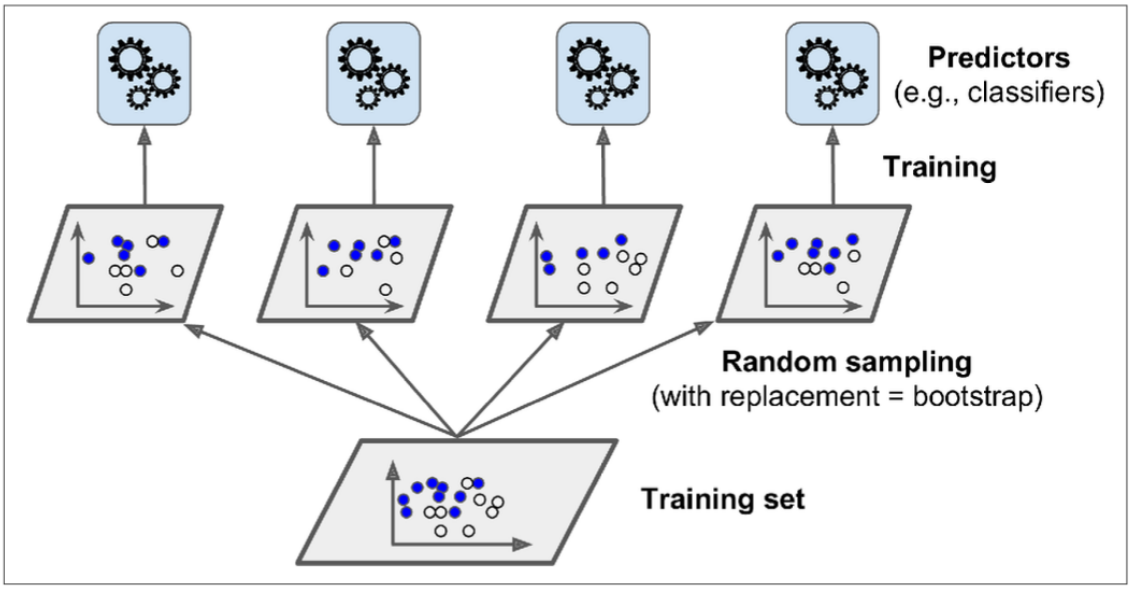


As we can see here the one that gets the most votes wins. This brought into high number of classifiers works very well, as using a coin that flips 51% of the times X, at the long run will be most of the times X. Ensemble methods work best when the predictors are as independent from one another as possible. One way to get diverse classifiers is to train them using very different algorithms. This increases the chance that they will make very different types of errors, improving the ensemble's accuracy.

Hard vs Soft voting

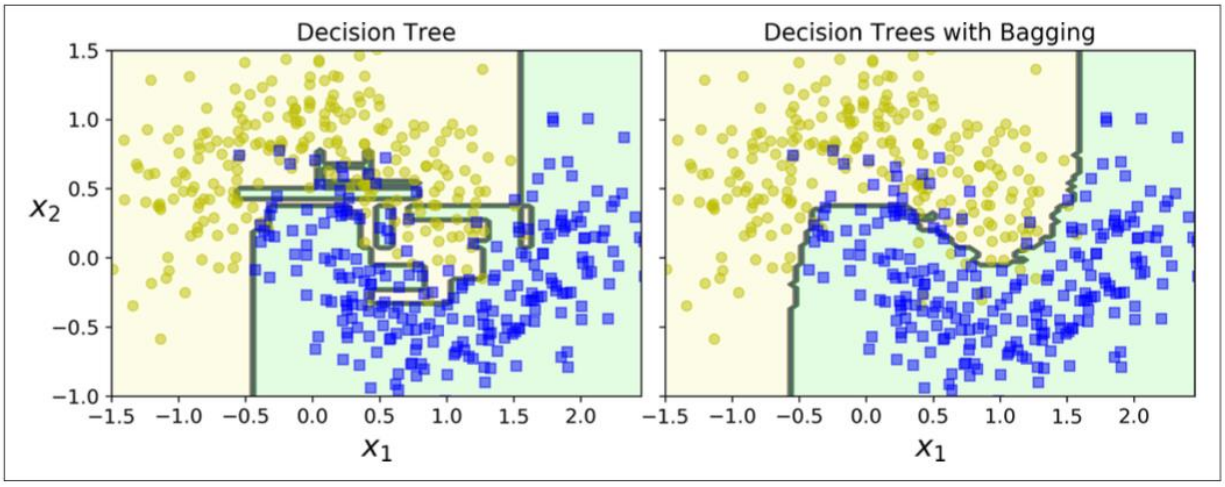
Hard voting is like a straightforward majority voting system without considering the confidence levels of classifiers, while soft voting considers the probability estimates of each class, potentially leading to more weighted and informed decisions. Soft voting is generally preferred when the individual models provide reliable probability estimates.

Bagging & Pasting



1. **Bagging (Bootstrap Aggregating):**
    - With Replacement: In bagging, each model is trained on a random subset of the training data sampled with replacement. This means that the same training instance can appear more than once in the same training subset for a particular model.
    - Implication: Because of sampling with replacement, bagging introduces more diversity in the subsets each model is trained on, as some instances may be repeated, and others left out entirely.

2. **Pasting:**
    - Without Replacement: Pasting, on the other hand, involves sampling without replacement. Each model is still trained on a random subset of the training data, but any given instance can appear in only one subset for a model.
    - Implication: Pasting ensures that the subsets used for training different models are more distinct compared to bagging, as there are no repeated instances in a single model's training set.



Use cases

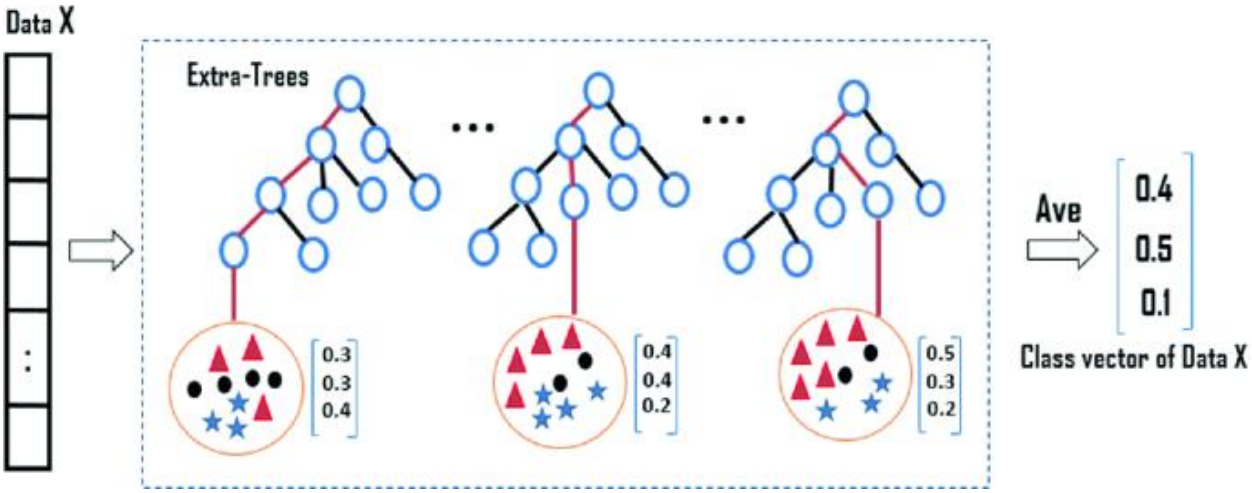
- Use Bagging: When you have a large dataset and models prone to overfitting. It's beneficial for reducing variance and improving model generalization.
- Use Pasting: When working with smaller datasets where you want to ensure that the models consider all available data equally, and you're concerned about over-representing certain instances.

Out-of-Bag Evaluation & Random Patches and Random Subspaces

Out-of-Bag Evaluation provides an efficient way to estimate model performance without a separate validation set, while Random Patches and Random Subspaces introduce methods to create more diverse sets of predictors by sampling features as well as instances, which can be especially useful in high-dimensional datasets.

Random Forests vs Extra Trees

Both Random Forests and Extra-Trees are ensemble methods that use multiple decision trees, the key difference lies in how the trees are constructed. Random Forests look for the best split among a subset of features, whereas Extra-Trees use random splits for both features and thresholds. This makes Extra-Trees generally faster but potentially slightly more biased, with lower variance compared to Random Forests.



Feature Importance

Random Forests make it easy to measure the relative importance of each feature, based on the impurity and usage of each node when splitting

Boosting

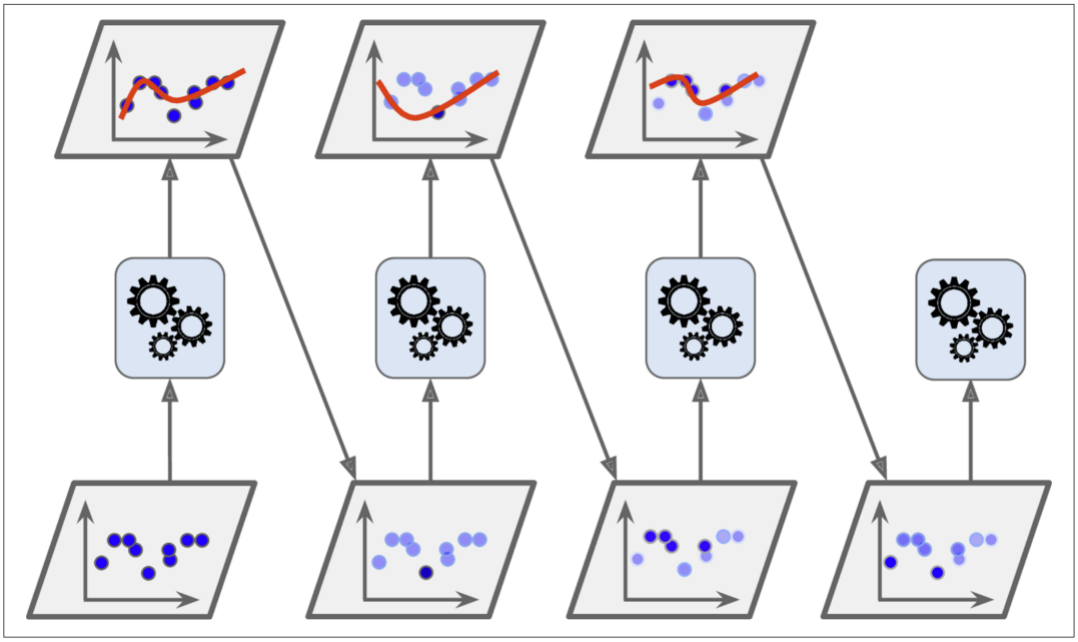
Boosting is an ensemble technique that combines multiple weak learners (typically simple models) to form a stronger learner. The key idea is to train predictors sequentially, each trying to correct its predecessor.
How It Works: The subsequent models focus more on the training instances that the previous models misclassified. This approach aims to convert a set of weak learners into a single strong learner.

AdaBoost (Adaptive Boosting)

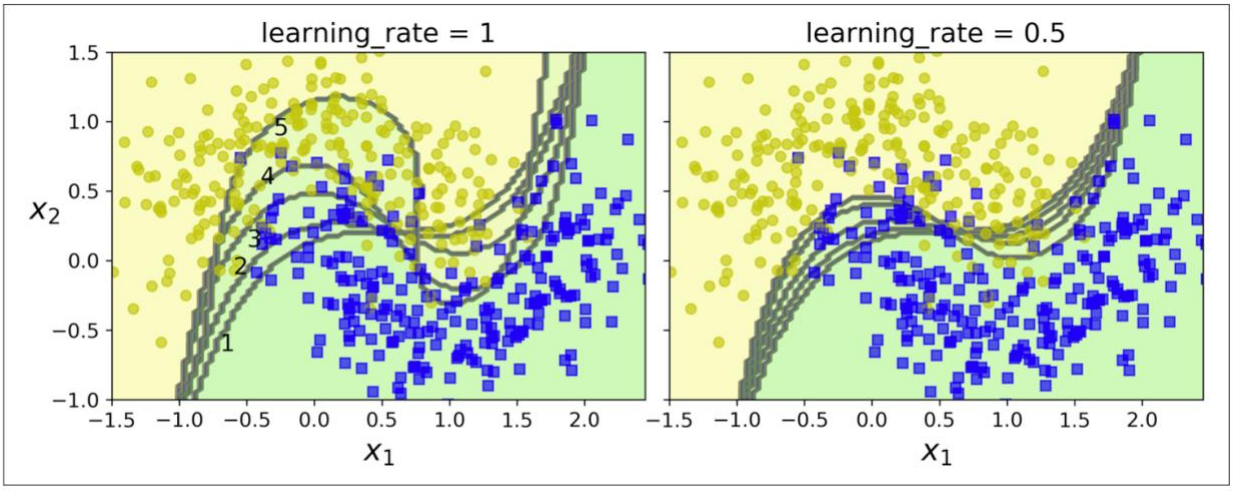
AdaBoost is one of the most popular boosting methods. It works by assigning higher weights to the training instances that the preceding models misclassified.

Training Process:

1. Initial Weights: All training instances start with equal weights.
2. Sequential Training: A model is trained, and the weights of misclassified instances are increased.
3. Weighted Errors: Each model is assigned a weight based on its accuracy; models with lower error have higher weights.
4. Final Prediction: The predictions are made by the weighted vote of all the models.



There is one important drawback to this sequential learning technique: it cannot be parallelized (or only partially), since each predictor can only be trained after the previous predictor has been trained and evaluated. As a result, it does not scale as well as bagging or pasting.



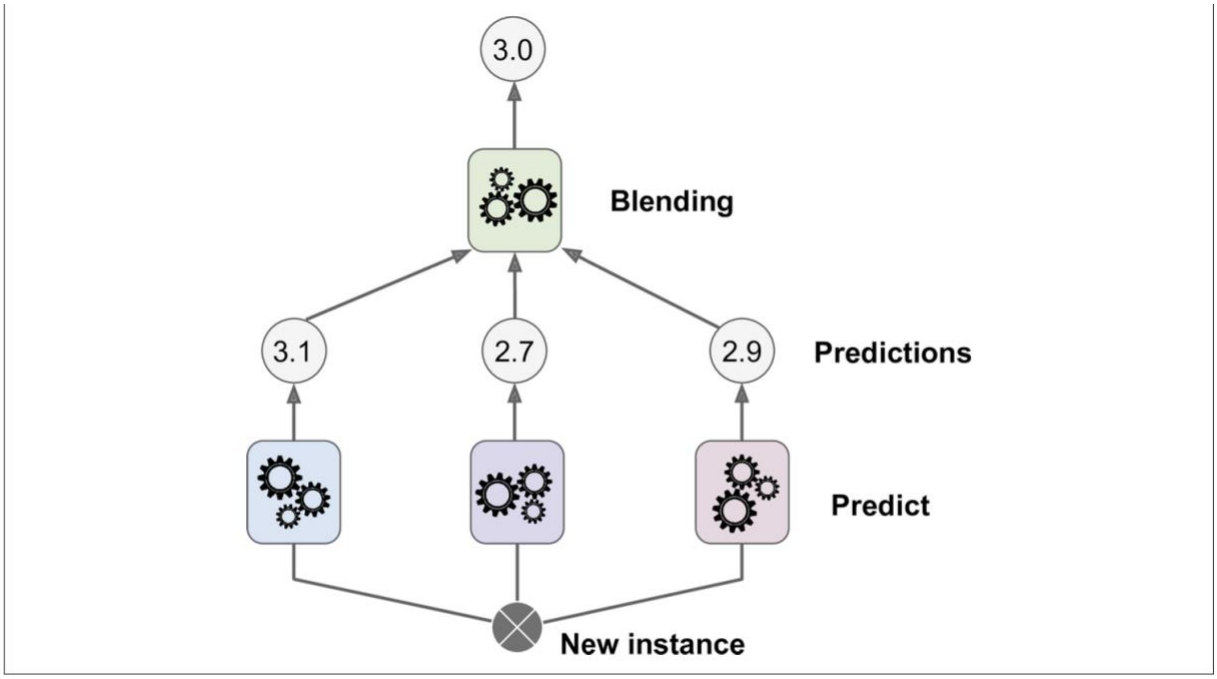
Gradient Boosting

Gradient Boosting works by sequentially adding predictors to an ensemble, each one correcting its predecessor. However, instead of tweaking the instance weights at every iteration like AdaBoost, Gradient Boosting tries to fit the new predictor to the residual errors made by the previous predictor.

Training Process:

1. Initial Model: Train a model on the data.
2. Residual Errors: Train a second model on the residual errors of the first model.
3. Repeat: Continue adding models, each correcting the residual errors of the combined ensemble.
4. Final Model: The final prediction is a sum of the predictions from all models.

Stacking (Stacked Generalization)



Stacking is another ensemble technique where instead of using trivial functions (like hard voting) to aggregate the predictions of all predictors in an ensemble, it trains a model to perform this aggregation.

How It Works:

1. Base Models: First, multiple base models (like decision trees, SVMs, etc.) are trained on the data.
2. Meta-Model: The predictions of all base models are used as inputs, and the actual target value as the output to train a new model (the meta-learner or meta-model).
3. Final Prediction: This meta-model makes the final prediction, effectively learning the best way to combine the outputs of the base models.

Advantages:

- Flexibility: It can combine different types of models, harnessing their individual strengths.
- Performance: Often provides better predictive performance compared to individual models or simple ensemble techniques.