Bagging and boosting are both ensemble learning techniques used in machine learning to improve the performance of models, especially decision trees.

**Bagging**

Bagging is a technique where multiple copies of the same base learning algorithm are trained on different subsets of the training data. The subsets are created by randomly sampling the training data with replacement. This means that some instances may appear multiple times in a subset, while others may not appear at all. The key idea behind bagging is to reduce the variance of the model by averaging or voting on the predictions of the individual models.

The most common algorithm that uses bagging is the Random Forest. In a Random Forest, multiple decision trees are trained on different subsets of the data, and their predictions are aggregated to make the final prediction. This helps reduce overfitting and improves the model's generalization.

**Boosting**

Boosting is another ensemble technique, but it works differently from bagging. In boosting, multiple weak learners (often shallow decision trees) are trained sequentially, and each subsequent learner focuses on the examples that the previous learners found difficult to classify correctly. The idea is to give more weight to the instances that are misclassified in the previous rounds, thus focusing on the mistakes and trying to correct them.

The most well-known boosting algorithms are AdaBoost, Gradient Boosting, and XGBoost. These algorithms build a strong learner by combining the outputs of multiple weak learners. Boosting typically results in a model with lower bias and lower variance compared to a single weak learner.

Here's a brief summary of both techniques:

**Bagging**

1. Training multiple models in parallel on random subsets of the training data (with replacement).

2. Reduces variance by averaging or voting on the predictions of individual models.

3. Examples: Random Forest, Bagged Decision Trees.

**Boosting**

1. Training multiple models sequentially, with each model focusing on correcting the errors of the previous models.

2. Reduces both bias and variance, resulting in a strong learner.

3. Examples: AdaBoost, Gradient Boosting, XGBoost.

Bagging and boosting are ensemble techniques used to improve the performance and generalization of machine learning models. Bagging focuses on reducing variance, while boosting aims to reduce both bias and variance by giving more attention to challenging examples during the training process.

Parametric and non-parametric algorithms are two broad categories of statistical and machine learning models. They differ in how they make assumptions about the underlying data distribution and the number of parameters used in the model.

**Parametric Algorithms**

Parametric algorithms make strong assumptions about the underlying data distribution. They assume that the data follows a specific mathematical distribution or functional form, such as a normal distribution or a linear relationship. These algorithms have a fixed number of parameters that are determined during the training process, and these parameters define the model.

Examples of parametric algorithms include:

**1. Linear Regression**: Assumes a linear relationship between the input features and the target variable, with parameters like slope and intercept.

**2. Logistic Regression**: Assumes a logistic function to model binary classification problems.

**3. Naive Bayes**: Assumes that features are conditionally independent given the class labels and uses probability distributions for classification.

Parametric models are computationally efficient and often require less data to train. However, they may not capture complex data patterns that do not conform to the assumed distribution.

**Non-Parametric Algorithms**

Non-parametric algorithms make fewer assumptions about the underlying data distribution. These algorithms are more flexible and can adapt to a wide range of data patterns. They do not have a fixed number of parameters; the number of parameters depends on the size of the training data. As more data is collected, non-parametric models can become more complex.

Examples of non-parametric algorithms include:

**1. K-Nearest Neighbors (KNN):** Makes predictions based on the k nearest data points in the training set without assuming any specific functional form.

**2. Decision Trees:** Divide the feature space into regions without assuming any specific data distribution.

**3. Support Vector Machines (SVM):** While SVMs are often used with linear kernels (parametric), they can also be used with non-linear kernels to make them non-parametric.

Non-parametric models can capture complex data patterns and are suitable for situations where the underlying data distribution is not well-known or does not follow a simple parametric form. However, they may require more data to generalize well and can be computationally expensive.

Parametric algorithms make strong assumptions about data distributions and have a fixed number of parameters, while non-parametric algorithms are more flexible and adapt to data patterns without specific distribution assumptions and can have a variable number of parameters based on the data size. The choice between parametric and non-parametric models depends on the nature of the data and the problem you are trying to solve.