XGBoost 是 Boosting 算法的其中一种。Boosting 算法的思想是将许多弱分类器集成在一起形成一个强分类器。

Boosting 集成学习是由多个相关联的决策树联合决策，即不同的决策树根据不同的权重联合预测出最后的结果，且每个决策树是独立的。

XGBoost 的目标是希望建立 K 个回归树，使得树群的预测值尽量接近真实值，而且有尽量大的泛化能力，其目标函数为：

$$obj(\theta)=\sum_{i}^{n}L(y_i,\hat{y}_i)+\sum_{k=1}^{K}\Omega(f_k)$$

其中，式中第二项表示决策树的复杂度，其表达式为：

$$\Omega(f)=\gamma T+\frac{1}{2}\lambda\|w\|^2$$

可以看到，XGBoost 的目标函数在损失函数的基础上加上了表示模型复杂度的正则项，正则化项同样包含两部分，T 表示叶子结点的个数，w 表示叶子节点的分数。γ 可以控制叶子结点的个数，λ 可以控制叶子节点的分数不会过大，防止过拟合。

众所周知，回归树主要有两个参数需要解决。第一个是选取哪个特征作为分裂节点，第二个是节点的预测值。对于这两个参数的选取，XGBoost 使用了和 CART 回归树一样的想法，利用贪婪算法，只考虑这个节点的样本，遍历所有特征的所有特征划分点，不同的是使用上式目标函数值作为评价函数。这种思路使得 XGBoost 可以并行化，对于同层节点计算分裂点时候可以多线程并行，训练速度更快。

如何求得最优的目标，即求得最小的损失函数，XGBoost 使用了二次函数最优化的方法，对于那些损失函数不为二次的，使用泰勒公式展开，将其近似于二次函数。

XGBoost 算法的其他优点：设计了针对稀疏数据的处理方法。进行交叉验证，方便选择更好的参数等等。