请问（决策树、Random Forest、Booting、Adaboot）GBDT和XGBoost的区别是什么？

集成学习的集成对象是学习器。Bagging和Boosting属于集成学习的两类方法，Bagging方法有放回地采样同数量样本训练每个学习器，然后再一起集成（简单投票），Boosting方法使用全部样本（可调权重）依次训练每个学习器，迭代集成（平滑加权）

决策树属于最常用的学习器，其学习过程是从根建立树，也就是如何决策叶子节点分裂。ID3/C4.5决策树用信息熵计算最优分裂，CART决策树用基尼指数计算最优分裂，xgboost决策树使用二阶泰勒展开系数计算最优分裂。

下面提到的学习器都是决策树：

1.Bagging方法：

学习器间不存在强依赖关系，学习器可并行训练生成，集成方式一般为投票。

Random Forest属于Bagging的代表，放回抽样，每个学习器随机选择部分特征去优化。

2.Boosting方法

学习器之间存在强依赖关系，必须串行生成，集成方式为加权和。

Adaboost属于Boosting，采用指数函数损失函数替代原本分类任务的0-1损失函数。

（指数损失函数：

$$L(Y|f(X)) = exp[-yf(x)]$$

）

GBDT属于Boosting的优秀代表，对函数残差近似值进行梯度下降，用CART回归树做学习器，集成为回归模型。

XGBoost属于Boosting的集大成者，对函数残差近似值进行梯度下降，迭代时利用了二阶梯度信息，集成模型可分类可回归。由于它可在特征粒度上并行计算，结构风险和工程实现都做了很多优化，泛化性能和扩展性能都比GBDT要好。具体的优点：

- 损失函数是泰勒展开二项逼近，而不是像GBDT里的就是一阶导数，可以加快优化速度
- 将树模型的复杂度加入到正则项中，避免过拟合，泛化性能更好
- 在寻找最佳分割点时，考虑到传统的贪心算法效率较低，实现了一种近似贪心算法，用来加速和减小内存消耗，除此之外还考虑了稀疏数据集和缺失值的处理，对于特征的值有缺失的样本，XGBoost依然能自动找到其要分裂的方向。
- XGBoost支持并行处理，XGBoost的并行不是在模型上的并行，而是在特征上的并行，将特征列排序后以block的形式存储在内存中，在后面的迭代中重复使用这个结构。这个block也使得并行化成为了可能，其次在进行节点分裂时，计算每个特征的增益，最终选择增益最大的那个特征去做分割，那么各个特征的增益计算就可以开多线程进行。