简述一下随机森林算法的原理

随机森林算法是 Bagging 集成框架下的一种算法，它同时对训练数据和特征采用随机抽样的方法来构建更加多样化的基模型。随机森林具体的算法步骤如下：

1. 假如有 N 个样本，则有放回的随机选择 N 个样本(每次随机选择一个样本，然后返回继续选择)。这选择好了的 N 个样本用来训练一个决策树，作为决策树根节点处的样本。
2. 当每个样本有 M 个属性时，在决策树的每个节点需要分裂时，随机从这 M 个属性中选取出 m 个属性，满足条件 m << M。然后从这 m 个属性中采用某种策略（比如说信息增益）来选择 1 个属性作为该节点的分裂属性。
3. 决策树形成过程中每个节点都要按照步骤 2 来分裂（很容易理解，如果下一次该节点选出来的那一个属性是刚刚其父节点分裂时用过的属性，则该节点已经达到了叶子节点，无须继续分裂了）。一直到不能够再分裂为止。注意整个决策树形成过程中没有进行剪枝。
4. 按照步骤 1~3 建立大量的决策树，这样就构成了随机森林了。

随机森林的随机性体现在哪里？

随机森林的随机性体现在每颗树的训练样本是随机的，树中每个节点的分裂属性集合也是随机选择确定的。

1. **随机采样**：随机森林在计算每棵树时，从全部训练样本（样本数为 n）中选取一个**可能有重复的、大小同样为 n** 的数据集进行训练（即 booststrap 采样）。
2. **特征选取的随机性**：在每个节点随机选取**所有特征的一个子集**，用来计算最佳分割方式。

随机森林算法的优缺点？

优点

--- 特征和数据的随机抽样 ---

1. 它可以处理很高维度（特征很多）的数据，并且不用降维，无需做特征选择
2. 如果有很大一部分的特征遗失，仍可以维持准确度。
3. 不容易过拟合
4. 对于不平衡的数据集来说，它可以平衡误差。
5. 可以判断出不同特征之间的相互影响（类似于控制变量法）

--- 树模型的特性---

6. 它可以判断特征的重要程度

--- 算法结构 ---

7. 训练速度比较快，容易做成并行方法
8. 实现起来比较简单

缺点

1. 随机森林已经被证明在某些噪音较大的分类或回归问题上会过拟合。（决策树的学习本质上进行的是决策节点的分裂，依赖于训练数据的空间分布）
2. 对于有不同取值的属性的数据，取值划分较多的属性会对随机森林产生更大的影响，所以随机森林在这种数据上产出的属性权值是不可信的

随机森林为什么不能用全样本去训练 m 棵决策树？

随机森林的基学习器是同构的，都是决策树，如果用全样本去训练 m 棵决策树的话；基模型之间的多样性减少，互相相关的程度增加，不能够有效起到减少方差的作用；对于模型的泛化能力是有害的。

随机森林和 GBDT 的区别？

1. 随机森林采用的 bagging 思想，而 GBDT 采用的 boosting 思想。
2. 组成随机森林的树可以并行生成；而 GBDT 只能是串行生成。
3. 随机森林对异常值不敏感；GBDT 对异常值非常敏感。
4. 随机森林对训练集一视同仁；GBDT 对训练集中预测错误的样本给予了更多关注。
5. 随机森林是通过减少模型方差提高性能；GBDT 是通过减少模型偏差提高性能。
6. 对于最终的输出结果而言，随机森林采用多数投票等；而 GBDT 则是将所有结果累加起来，或者加权累加起来。
7. 组成随机森林的树可以是分类树，也可以是回归树；而 GBDT 只能由回归树组成。

附加题

CV:RNN 中使用 ReLU 可以解决梯度消失问题吗？

这个问题可以参考: https://www.cnblogs.com/itmorn/p/13285206.html

NLP:Transformer 中使用多头注意力的好处是什么

这个问题可以参考: https://blog.csdn.net/qq_27590277/article/details/106264120

BI:最近邻问题在推荐系统的应用场景是什么?具体算法有哪些?

https://blog.csdn.net/qq_36298178/article/details/82228474?utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromMachineLearnPai2%7Edefault-2.baidujs&dist_request_id=1331647.22921.16184828813940069&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromMachineLearnPai2%7Edefault-2.baidujs

以上可以参考