简述决策树的构建过程

- 构建根节点，将所有训练数据都放在根节点
- 选择一个最优特征，按照这一特征将训练数据集分割成子集，使得各个子集在当前条件下获得最好的分类
- 如果子集非空，或未达到停机条件，递归 1，2 步骤，直到所有训练数据子集都被正确分类或没有合适的特征为止

ID3 决策树与 C4.5 决策树的区别

按照（**简述决策树的构建过程**）中所描述的，构建决策树过程中一个关键步骤就是选择一个最优特征；而 ID3 决策树与 C4.5 决策树一个最大的区别就是在选择最优特征时所依赖的标准不同。

在 ID3 决策树中，选择最佳特征通过信息增益指标来选择，所谓信息增益可以定义为：

数据集对于某特征的信息增益 = 数据集的经验熵 - 这个特征对数据集的条件经验熵

$$g(D,A) = H(D) - H(D|A)$$

其中 D 为数据集，A 表示数据集样本上的某个特征，H（D）为数据集的经验熵；H（D|A）表示特征 A 对数据集 D 的经验条件熵。

$$H(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|} \log_2 \frac{|C_k|}{|D|}$$

其中 K 表示 D 中类别的数量，C_k表示第 k 个类别所包含样本的个数。

$$H(D|A) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \sum_{k=1}^{K} \frac{|D_{ik}|}{|D|} \log_2 \frac{|Di_k|}{|D|}$$

其中 n 表示在特征 A 上不同取值的个数，D_{ik}表示在特征上取第 i 个值且类别为 k 时的样本个数。

在 C4.5 决策树中，为了解决信息增益偏向于选择取值较多的特征的问题，选择最佳特征通过信息增益熵来选择，所谓信息增益熵可以定义为：

$$g_R(D,A)=\frac{g(D,A)}{H_A(D)}$$

其中$H_A(D)$表示，训练数据集 D 关于特征 A 的值的熵。

$$H_A(D)=-\sum_{i=1}^{n}\frac{|D_i|}{|D|}\log_2\frac{|D_i|}{|D|}$$

其中 n 表示特征 A 上取得不同值的个数，D_i表示在特征 A 上取值为第 i 个时，样本的个数。

此外 1）ID3 只能处理离散型变量，而 C4.5 通过将连续值离散化来处理连续型变量。2）ID3 没有对缺失值的处理策略，而 C4.5 通过引入有缺失值样本对所有样本的比例来处理缺失值带来的“信息增益失真”的问题。

CART 回归树构建过程

在 CART 树的构建过程中，假设决策树是一颗二叉树；决策树的生成过程就是递归地构建二叉决策树的过程。对回归树来说用，我们这里讨论一个基本形式即为平方误差最小化准则，进行特征选择，生成二叉树。

输入：训练数据集 D;

输出：回归树 f(x)

在训练数据集所在的输入空间中，递归地将每个区域划分为两个子区域并决定每个子区域上的输出值，构建二叉树：

1）选择最优切分变量（特征）j 与切分点（特征值域上的值）s, 求解

$$\min_{j,s}[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_2\in R_2(j,s)}(y_i-c_2)^2]$$

2）用选定的(j, s)划分区域并决定相应的输出值，其中$R_1(j,s)$是划分后的左子区域，而$R_2(j,s)$是划分后的右子区域；c_1是左子区域上的预测值(此区域上所有样

本真实值的均值），而c_2是右子区域上的预测值(此区域上所有样本真实值的均值)。

$$R_1(j,s) = \{x|x^{(j)} \le s\}, R_2(j,s) = \{x|x^{(j)} > s\}$$

$$\hat{c}_m = \frac{1}{N_m} \sum_{x_i \in R_m(j,s)} y_i, x \in R_m, m = 1,2$$

3）递归地对两个子区域调用步骤 1），2），直到满足停机条件。常用的停机条件有 a. 树的深度 b. 叶子区域的个数 c. 叶子区域上样本的个数等。

4）将输入空间划分为 M 个区域 R1, R2，…, RM, 生成决策树，划分的空间即为叶子节点上的空间：

$$f(x) = \sum_{m=1}^{M} \hat{c}_m I(x \in R_m)$$

大家可以对照 XGBoost 节点分裂的过程来学习 CART 回归树的生成过程。

决策树的优缺点

优点：

- 缺失值不敏感，对特征的宽容程度高，可缺失可连续可离散
- 可以计算特征重要性，且可解释性强
- 算法对数据没有强假设
- 可以解决线性及非线性问题
- 有特征选择等辅助功能

缺点：

- 处理关联性数据比较薄弱 - 重要性强且关联的特征都得到了重视
- 正负量级有偏样本的样本效果较差
- 单棵树的拟合效果欠佳，容易过拟合

决策树如何防止过拟合？说说具体方法。

我们在讨论防止机器学习过拟合的时候，通过分类的方法，大致确立了这么几个改进方向，1）数据 2）模型 3）正则化 4）训练。在这个题目中，我们重点讨论如何通过改进决策树模型来防止过拟合，当然其他几个方向对防止决策树过拟合同样适用。

通过改进模型来防止过拟合的主要思路是简化模型，使得模型能够学习样本中的共同特征（即主要特征），而摈弃个性化的特征（即次要特征）。而对树模型进行简化的方法又可以分为预剪枝（在训练过程中进行剪枝），和后剪枝（在决策树构建完成之后进行剪枝）

预剪枝的主要方法有：

1） 限制树的深度 - 当树到达一定深度的时候，停止树的生长
2） 限制叶子节点的数量
3） 规定叶子区域内最少的样本数，未达到最少样本数的叶子区域不做分裂
4） 计算每次分裂对**测试集**的准确度提升

后剪枝的核心思想是让算法生成一颗完全生长的决策树，然后最底层向上计算是否剪枝。剪枝过程将子树删除，用一个叶子节点替代，该节点的类别同样按照多数投票的原则进行判断。同样地，后剪枝也可以通过在测试集上的准确率进行判断，如果剪枝过后准确率有所提升，则进行剪枝。相比于预剪枝，后剪枝方法通常可以得到泛化能力更强的决策树，但时间开销会更大。