Decision Tree （决策树）的一些补充

1、信息增益

特征的分类能力

如果一个特征对结果影响比较大，那么就可以认为这个特征的分类能力比较大。相亲时候一般会先问收入，再问长相，然后问其家庭情况。也就是说在这边收入情况影响比较大，所以作为第一个特征判断，如果不合格那可能连后续都不用询问了。

有什么方法可以表明特征的分类能力呢？

信息增益是特征选择的一个重要指标，它定义为一个特征能够为分类系统带来多少信息，带来的信息越多，说明该特征越重要，相应的信息增益也就越大。

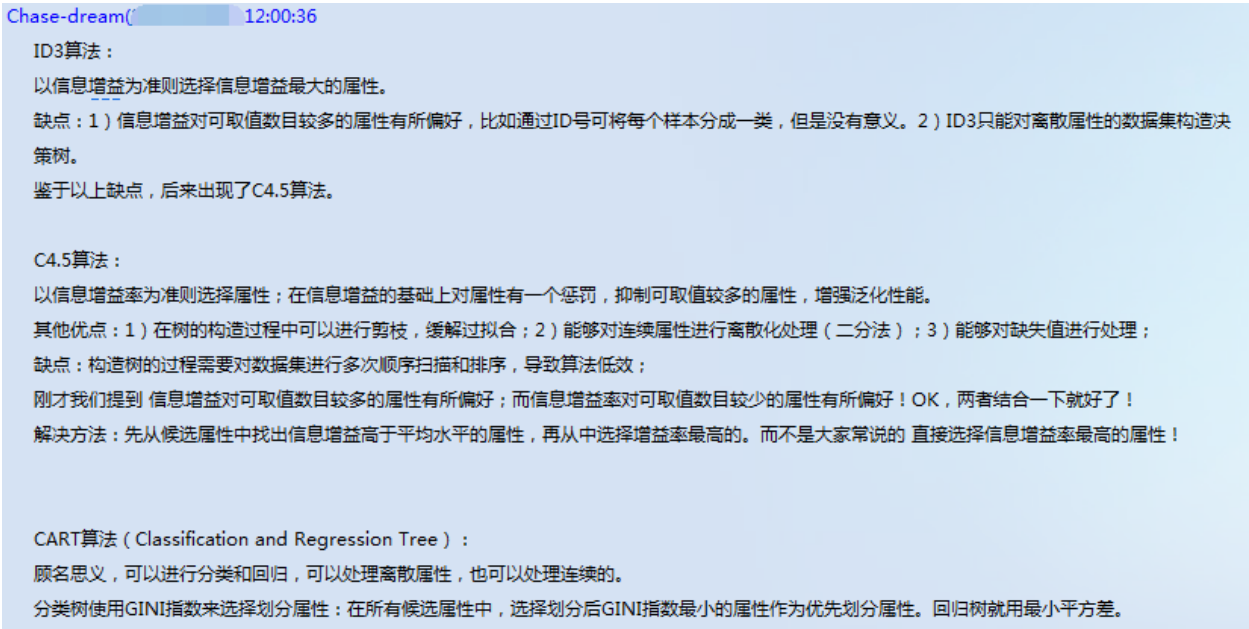
信息增益代表了在一个条件下，信息复杂度（不确定性）减少的程度。

在决策树中，我们的关键就是每次选择一个特征，特征有多个，那么到底按照什么标准来选择哪一个特征？这个问题就可以用信息增益来度量。如果选择一个特征后，信息增益最大（信息不确定性减少的程度最大），那么我们就选取这个特征。

2、决策树算法的 3 个主要部分

- 特征选择：信息增益 / 信息增益率 / GINI 指数
    - 信息增益：这也正是 ID3 算法使用的特征选择方法。
        - 熵，表示随机变量的不确定性。
        - 条件熵，在一个条件下，随机变量的不确定性。
        - 信息增益：熵 - 条件熵。 也就是说，信息增益是在一定条件下，信息不确定性减少的程度！
        - 通俗地讲，X(明天下雨)是一个随机变量，X的熵可以算出来， Y(明天阴天)也是随机变量，在阴天情况下下雨的信息熵我们如果也知道的话（此处需要知道其联合概率分布或是通过数据估计）即是条件熵。
        - 两者相减就是信息增益！原来明天下雨例如信息熵是2，条件熵是0.01（因为如果是阴天就下雨的概率很大，信息就少了），这样相减后为1.99，在获得阴天这个信息后，下雨信息不确定性减少了1.99！是很多的！所以信息增益大！也就是说，阴天这个信息对下雨来说是很重要的！
    - 信息增益率（也叫信息增益比）：这也正是 C4.5 算法使用的特征选择方法。
        - 在信息增益的基础上，加入对于属性划分的惩罚项，即除以一项 Info（公式略）。取值数目多的属性该项也会变大，避免了 ID3 中出现的这样的问题。
    - GINI 指数：GINI 描述的是纯度，与信息熵的含义相似。GINI 指数反映了数据集的纯度，值越小，纯度越高。我们在候选集合中选择使得划分后的基尼指数最小的属性作为最优划分属性。
- 决策树生成
    - ID3 算法：在决策树的各个结点上应用增益准侧进行特征选择。
        - 从根节点开始，对结点结算所有可能特征的信息增益，选择信息增益最大的特征作为结点的特征，并由该特征的不同取值构建子节点；
        - 对子节点递归地调用以上方法，构建决策树；

        - 直到所有特征的信息增益均很小或者没有特征可选时为止。
    - C4.5 算法
        - C4.5 算法与 ID3 算法的区别主要在于它在生产决策树的过程中，使用信息增益比来进行特征选择。
        - ID3 与 C4.5 的一些细节与区别：https://zhuanlan.zhihu.com/p/26760551
    - CART 算法
        - 分类与回归树（classification and regression tree，CART），与 C4.5 一样，由 ID3 算法演化而来。CART 假设决策树是一个二叉树，通过递归地二分每个特征，将特征空间划分为有限个单元，并在这些单元上确定预测的概率分布。
        - 对于回归树，采用的是平方误差最小化准则；对于分类树，采用基尼指数最小化准则。
        - 介绍的比较好的一个链接：https://zhuanlan.zhihu.com/p/30616889
    - ID3 ，C4.5 和 CART 算法之间的区别：https://www.zhihu.com/question/27205203，https://zhuanlan.zhihu.com/p/34534004
    - 群里大佬的一个分享，下面我截图出来了，这样看着比较直观并且很全面。



- 决策树的剪枝
    - 预剪枝和后剪枝
        - 预剪枝是指在决策树生成过程中，对每个结点在划分前先进行估计，若当前结点的划分不能带来决策树泛化能力（在训练时加入验证集随时进行泛化验证）的提升，则停止划分并将当前结点标记为叶节点；
        - 后剪枝则是先从训练集中生成一颗完整的树，然后自底向上对非叶节点进行考察，若该节点对应的子树替换为叶节点能够提升泛化能力，则进行剪枝将该子树替换为叶节点，否则不剪枝。后剪枝技术通常比预剪枝保留了更多的分支，它是自底向上的剪枝，因此它的欠拟合风险较小，泛化能力往往优于预剪枝，然而因为总是要完全生长一棵树，这就要花费很多时间训练了，数据集规模大、维度高时并不适用实际应用。
    - 如果对训练集建立完整的决策树，会使得模型过于针对训练数据，拟合了大部分的噪声，即出现过拟合的现象。为了避免这个问题，有两种解决的办法：
        - 1、当熵减少的量小于某一个阈值时，就停止分支的创建。这算是一种贪心算法。

        - 2、先创建完整的决策树，然后再尝试消除多余的节点，也就是采用剪枝的方法。
        - 小结：第一种方法存在一个潜在的问题，有可能某一次分支的创建不会令熵值有太大的下降，但是随后的子分支却有可能会使得熵大幅降低。因此，我们更倾向于采用剪枝的方法。
        - 剪枝方法具体如下：
            - 1、计算每个节点的熵；
            - 2、递归地从树的叶节点向上回缩，如果将某一个节点点的所有叶节点合并，能够使得其损失函数减小，则进行剪枝，将父节点变成新的叶节点；
            - 3、返回 2，直到不能继续合并。

    - 决策树剪枝算法（几个讲的比较好的链接）：
        - https://www.cnblogs.com/starfire86/p/5749334.html
        - https://blog.csdn.net/yujianmin1990/article/details/49864813
        - https://zhuanlan.zhihu.com/p/30296061
        - https://zhuanlan.zhihu.com/p/24498143

3、决策树的优缺点

- 优点
    - 易于理解和解释，甚至比线性回归更直观；
    - 与人类做决策思考的思维习惯契合；
    - 模型可以通过树的形式进行可视化展示；
    - 可以直接处理非数值型数据，不需要进行哑变量的转化，甚至可以直接处理含缺失值的数据；
- 缺点
    - 处理连续变量不好；
    - 不好处理变量之间存在许多错综复杂的关系，如金融数据分析；
    - 决定分类的因素取决于更多变量的复杂组合时；
    - 可规模性一般。

4、写的比较好的博客

- 决策树梳理：https://blog.csdn.net/ice_martin/article/details/63683657
- 决策树算法原理：https://www.cnblogs.com/nxld/p/6371453.html

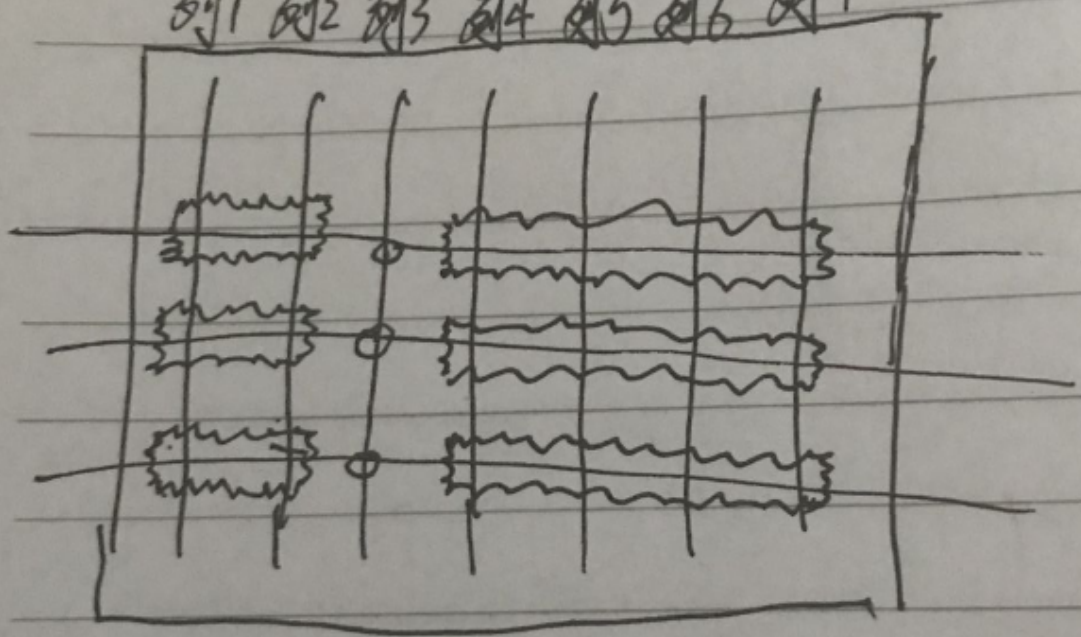
5、对 github 中的代码的图片补充

No.

Date

我们的数据集DataSet（等切分的）

列1 列2 列3 列4 列5 列6 列7



比如，现在我们的 index = 2，也就是列3

其中的 value = 3