1. **根据下表1所给的训练数据集，利用信息增益比（C4.5算法）生成决策树。**

**答：**特征(年龄)的信息增益比为0.052

特征(有工作)的信息增益比为0.352

特征(有自己的房子)的信息增益比为0.433

特征(信贷情况)的信息增益比为0.232

特征(有自己的房子)的信息增益比最大，选择为根节点划分特征

当有自己的房子为是

此时类别均为是，纯净，停止划分

当有自己的房子为否

特征(年龄)的信息增益比为0.164

特征(有工作)的信息增益比为1.000

特征(信贷情况)的信息增益比为0.340

特征(有工作)的信息增益比最大，选择为当前节点特征

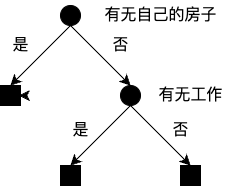
当有工作为否

此时类别均为否，纯净，停止划分

当有工作为是

此时类别均为是，纯净，停止划分

决策树如下：



1. **对下表1所给的训练数据集，根据信息增益准则选择最优特征。**

**答：**特征(年龄)的信息增益为0.083

特征(有工作)的信息增益为0.324

特征(有自己的房子)的信息增益为0.420

特征(信贷情况)的信息增益为0.363

特征(有自己的房子)的信息增益最大，选择为根节点划分特征

当有自己的房子为是

此时类别均为是，纯净，停止划分

当有自己的房子为否

特征(年龄)的信息增益为0.252

特征(有工作)的信息增益为0.918

特征(信贷情况)的信息增益为0.474

特征(有工作)的信息增益最大，选择为当前节点特征

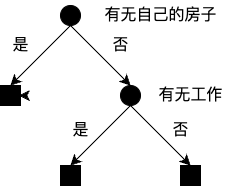
当有工作为否

此时类别均为否，纯净，停止划分

当有工作为是

此时类别均为是，纯净，停止划分

决策树如下：



1. **根据下表1所给的训练数据集，利用ID3算法生成决策树。**

**答：同2**

1. **根据下表1所给的训练数据集，利用CART算法生成决策树。**

**答：**特征(年龄)的基尼系数为0.4267

特征(有工作)的基尼系数为0.3200

特征(有自己的房子)的基尼系数为0.2667

特征(信贷情况)的基尼系数为0.2844

特征(有自己的房子)的基尼系数最小，选择为根节点划分特征

当有自己的房子为否

特征(年龄)的基尼系数为0.3148

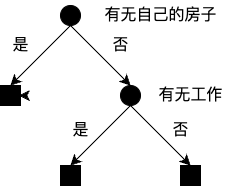
特征(有工作)的基尼系数为0

特征(信贷情况)的基尼系数为0.2222

特征(有工作)的基尼系数最小，选择为当前节点特征

其他节点纯净，停止划分（同1、2）

决策树如下：



**表1 贷款申请样本数据表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **年龄** | **有工作** | **有自己的房子** | **信贷情况** | **类别** |
| **1** | **青年** | **否** | **否** | **一般** | **否** |
| **2** | **青年** | **否** | **否** | **好** | **否** |
| **3** | **青年** | **是** | **否** | **好** | **是** |
| **4** | **青年** | **是** | **是** | **一般** | **是** |
| **5** | **青年** | **否** | **否** | **一般** | **否** |
| **6** | **中年** | **否** | **否** | **一般** | **否** |
| **7** | **中年** | **否** | **否** | **好** | **否** |
| **8** | **中年** | **是** | **是** | **好** | **是** |
| **9** | **中年** | **否** | **是** | **非常好** | **是** |
| **10** | **中年** | **否** | **是** | **非常好** | **是** |
| **11** | **老年** | **否** | **是** | **非常好** | **是** |
| **12** | **老年** | **否** | **是** | **好** | **是** |
| **13** | **老年** | **是** | **否** | **好** | **是** |
| **14** | **老年** | **是** | **否** | **非常好** | **是** |
| **15** | **老年** | **否** | **否** | **一般** | **否** |

1. **已知如下表所示的训练数据，试用平方误差损失准则生成一个二叉回归树。**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **x** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **y** | **4.50** | **4.75** | **4.91** | **5.34** | **5.80** | **7.05** | **7.90** | **8.23** | **8.70** | **9.00** |

**答：**

1. 初始节点：mean= (4.50 + 4.75 + 4.91 + 5.34 + 5.80 + 7.05 + 7.90 + 8.23 + 8.70 + 9.00) / 10 = 6.618

总平方损失：27.63 （总损失较大，开始划分节点）

1. 寻找分割点:

• 遍历可能的分割点：1.5, 2.5, ..., 9.5。

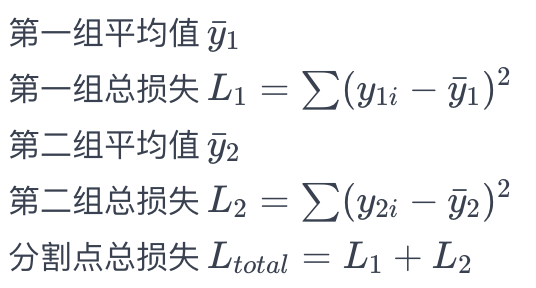
• 对每个分割点，根据下面公式分别计算左侧和右侧的平均值和平方损失。

---以x=5.5为例：

第一组（x ≤ 5.5）: {4.50, 4.75, 4.91, 5.34, 5.80}

第二组（x > 5.5）: {7.05, 7.90, 8.23, 8.70, 9.00}

计算每组y的平均值，然后计算损失：



• 计算得到的每个分割点总损失，选择总损失最小的分割点作为划分节点：

---每个分割点的总损失如下：

x=1.5的总损失：22.648；x=2.5的总损失：17.702；x=3.5 的总损失：12.193；x=4.5 的总损失：7.379；**x=5.5的总损失：3.359**；x=6.5 的总损失：5.074；x=7.5 的总损失：10.052；x=8.5的总损失：15.178；x=9.5 的总损失：21.328。在这些分割点中，x=5.5 有最小的总损失（3.359），因此它是最佳的分割点。在构建决策树时，第一步将会是以 x=5.5 作为根节点的分割条件。

1. 递归分割:

* 对于x<=5.5的左侧数据集，重复步骤b。
* 对于x>5.5的右侧数据集，也重复步骤b

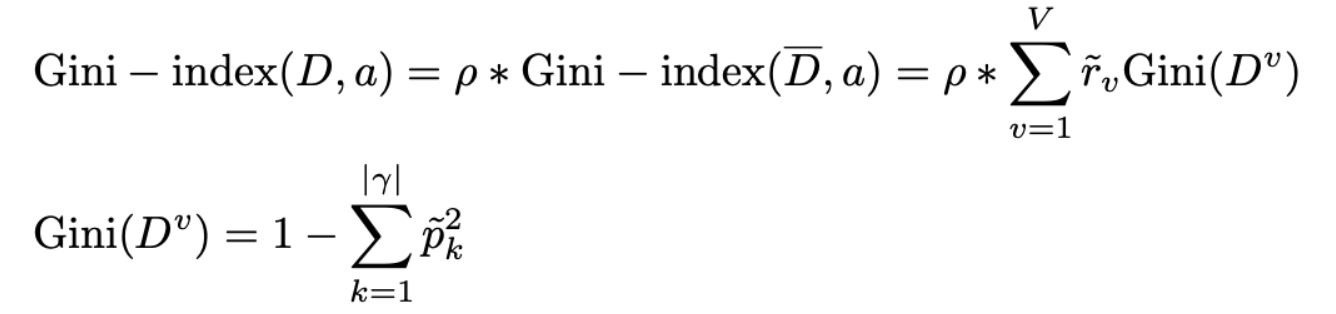
1. 终止条件:

假设我们的停止条件是每个节点至少需要包含两个数据点，那么当数据不能再分时，递归停止。(根据具体停止划分情况而定)

1. **试析使用“最小训练误差”作为决策树划分选择准则的缺陷。**

**答：**使用"最小训练误差"作为决策树划分选择准则，由于使用的是训练集数据，可能会将训练特征中的一些异常或者偶然作为模型的一部分，导致过度拟合的问题。

1. **试将4.4.2节对缺失值的处理机制推广到基尼指数的计算中去。**



1. **某公司招聘职员考查身体、业务能力、发展潜力这3项。身体分为合格1、 不合格0两级，业务能力和发展潜力分为上1、中2、下3三级。分类为合格1、不合格-1两类。已知10个人的数据，如表所示。假设弱分类器为决策树桩。试用 AdaBoost 算法学习一个强分类器。**

应聘人员情况数据表

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 身体 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 业务能力 | 1 | 3 | 2 | 1 | 2 | 1 | 1 | 1 | 3 | 2 |
| 发展潜力 | 3 | 1 | 2 | 3 | 3 | 2 | 2 | 1 | 1 | 1 |
| 分类 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | -1 | -1 |

答：编程题，采用sklearn的AdaBoostClassifier分类器，构建并训练得到强分类器（https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html）

1. **给定如表所示训练数据。假设弱分类器由 x＜v或 x＞v产生，其阈值v使该分类器在训练数据集上分类误差率最低。试用 AdaBoost 算法学习一个强分类器。**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| y | 1 | 1 | 1 | -1 | -1 | -1 | 1 | 1 | 1 | -1 |

**答：**

计算步骤：

1. 初始化权重: 初始时，每个样本的权重都是相等的，即 = ，其中是样本数量，每个样本的初始权重是 0.1。

2. 对于每一轮:

- 选择阈值最小化加权分类误差。

- 计算加权误差率 = 。

- 计算弱分类器的权重。

- 更新样本权重：对于正确分类的样本， ，对于错误分类的样本，。然后归一化所有样本权重使它们总和为 1。(这一步PPT上没有，可以实施也可以不实施)

3. 组合弱分类器: 最终的强分类器是所有弱分类器的加权组合，即sign，其中 是第轮的弱分类器。

以第一轮为例：

在第一轮中，最佳阈值是 2，方向是 "greater"（即x> 2 ），所有x值大于2的被分类为 -1，否则为 1。弱分类器的加权误差率是 0.3。

弱分类器的权重，然后更新训练样本的权重。这将为下一轮的弱分类器训练提供基础。

前五轮迭代：

这些弱分类器组合成一个强分类器，强分类器的决策基于所有弱分类器的加权投票结果。使用这些弱分类器来对新数据进行分类以检验强分类器的性能。

1. **试析随机森林为何比决策树 Bagging 集成的训练速度更快。**

答：决策树的生成过程中，最耗时的就是搜寻最优切分属性；随机森林在决策树训练过程中引入了随机属性选择，大大减少了此过程的计算量；因而随机森林比普通决策树Bagging训练速度要快。

1. **Gradient Boosting［Friedman，2001］是一种常用的Boosting 算法，试析其与 AdaBoost 的异同。**

答：Gradient Boosting和其它Boosting算法一样，通过将表现一般的数个模型（通常是深度固定的决策树）组合在一起来集成一个表现较好的模型。抽象地说，模型的训练过程是对一任意可导目标函数的优化过程。通过反复地选择一个指向负梯度方向的函数，该算法可被看做在函数空间里对目标函数进行优化。因此可以说Gradient Boosting = Gradient Descent + Boosting。和AdaBoost一样，Gradient Boosting也是重复选择一个表现一般的模型并且每次基于先前模型的表现进行调整。不同的是，AdaBoost是通过提升错分数据点的权重来定位模型的不足而Gradient Boosting是通过算梯度（gradient）来定位模型的不足。因此相比AdaBoost, Gradient Boosting可以使用更多种类的目标函数。

1. **试析 Bagging 通常为何难以提升朴素贝叶斯分类器的性能。**

答：通常情况下，Bagging 难以显著提升朴素贝叶斯分类器的性能，这是因为朴素贝叶斯分类器本身在处理特征之间存在强相关性或者特征空间较大时表现较好，而 Bagging 主要适用于降低高方差的模型，对于朴素贝叶斯这种低方差高偏差的模型，Bagging 的效果并不明显。此外，朴素贝叶斯已经假设了特征之间的独立性，而 Bagging 通过随机抽样生成的子模型可能会打破这一假设，从而影响了 Bagging 对朴素贝叶斯的性能提升。

1. **简述如何增强个体学习器的多样性。**

答：增强个体学习器的多样性是提高集成学习性能的关键。以下是几种常用的方法：

数据抽样：通过对训练数据进行抽样来生成不同的训练集，个体学习器就可以在不同的数据子集上进行训练。例如，随机森林中的每棵树都是在数据的不同随机子集上训练的。

特征抽样：对于特征丰富的数据集，可以通过随机选择部分特征来训练不同的学习器。这种方法在处理高维数据时特别有效。

使用不同的算法：集成不同类型的学习算法可以增加多样性。例如，可以将决策树、神经网络和支持向量机等不同类型的学习器组合起来。

调整超参数：即使是同一种算法，不同的超参数设置也会导致学习过程的差异，从而产生多样性。

引入随机性：在学习过程中引入随机性，如随机选择初始权重或随机决定网络的结构，也可以产生不同的学习器。