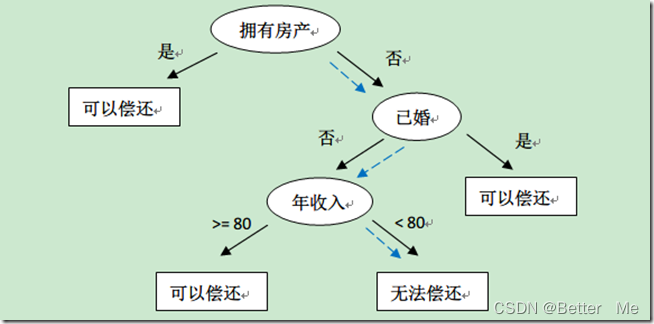
**决策树及决策树的划分依据（ID3、C4.5、CART）**

**一、决策树是什么？**

* 决策树是一种基于树状结构的机器学习算法，用于解决分类和回归问题。它是一种自上而下的递归分割方法，通过对特征空间的递归划分来构建一个树形模型，用于进行预测和决策。
* 在决策树中，每个内部节点表示对某个特征的测试，每个分支代表该特征的一个取值，而每个叶节点表示一个类别标签或一个回归值。决策树的构建过程是通过对特征空间进行递归划分，使得每个叶节点包含尽可能纯的样本（分类问题）或使得每个叶节点的回归值尽可能接近目标值（回归问题）。
* 决策树的构建过程可以根据不同的算法和指标进行选择划分特征的策略，常用的包括ID3、C4.5、CART等算法。这些算法根据特征的信息增益、信息增益比、基尼指数等指标来选择最优的划分特征。

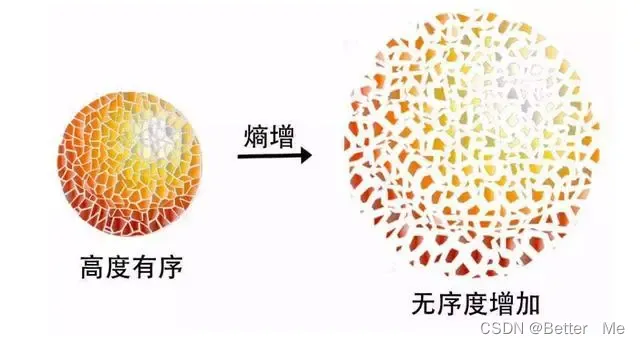
比如，给别人借钱判断他是否有偿还能力



**二、决策树划分依据**

**1.熵**

物理学上，**熵Entropy**是“混乱”程度的量度



**系统越有序，熵值越低；系统越混乱，熵值越高**。

1948年香农提出了**信息熵**的概念。

* 信息理论：
  + **从信息的完整性上进行的描述**：

当系统的有序状态一致时，数据越集中的地方熵值越小，数据越分散的地方熵值越大。

* + **从信息的有序性上进行的描述**：

当数据量一致时，系统越有序，熵值越低；系统越混乱，熵值越大。

[信息熵](https://so.csdn.net/so/search?q=%E4%BF%A1%E6%81%AF%E7%86%B5&spm=1001.2101.3001.7020)时度量样本集合纯度最常用的一种指标。

* 信息熵公式：  
  H ( X ) = − Σ P ( x ) ∗ l o g 2 ( P ( x ) ) H(X) = - Σ P(x) \* log2(P(x))*H*(*X*)=−Σ*P*(*x*)∗*log*2(*P*(*x*))  
  其中，H(X)表示随机变量X的信息熵，P(x)表示随机变量X取值为x的概率。Σ表示对所有可能取值x求和。log2表示以2为底的对数运算。

**2.决策树划分依据-------信息增益（ID3）**

**信息增益**：以某特征划分数据集前后的熵的差值。熵可以表示样本集合的不确定性，熵越大，样本的不稳定性就越大。因此可以**使用划分数据集前后集合熵的差值来衡量使用当前特征对于样本集合D划分效果的好坏**。

具体计算信息熵的步骤如下：

1. 计算划分前目标变量（分类标签）的信息熵，即初始的不确定性。
2. 对于每个特征，计算它在划分后对目标变量的条件熵。条件熵是在已知某个特征取值的情况下，目标变量的平均不确定性。
3. 用划分后的条件熵值对划分前的信息熵进行减法运算，得到信息增益。信息增益越大，表示使用该特征进行划分能够更好地减少不确定性，即特征对分类任务的贡献更大。

具体的信息增益计算公式如下：

* 信息增益 = 初始信息熵 - 条件熵

条件熵计算公式：  
H ( Y ∣ X ) = Σ P ( x ) ∗ H ( Y ∣ x ) H(Y|X) = Σ P(x) \* H(Y|x)*H*(*Y*∣*X*)=Σ*P*(*x*)∗*H*(*Y*∣*x*)  
其中，H(Y|X)表示在给定特征X的条件下，目标变量Y的条件熵；P(x)表示特征X取值为x的概率；H(Y|x)表示在特征X取值为x的条件下，目标变量Y的熵。

在决策树的构建过程中，**选择信息增益最大的特征作为当前节点的划分特征**，以最大程度地减少不确定性。然而，信息增益偏向于选择具有较多取值的特征，因此，在实际应用中，可以使用信息增益比等指标来对信息增益进行修正，以避免对具有较多取值的特征的过度偏好。

* 案例：

如下图，第一列为论坛号码，第二列为性别，第三列为活跃度，最后一列用户是否流失

要解决的问题：**性别和活跃度两个特征，哪个对用户流失影响更大？**



通过计算信息增益可以解决这个问题，统计上右表信息。

其中positive为正样本(已流失)，negative为负样本(未流失)，下面计算三个熵

* + 初始信息熵：
    - 整体类别信息熵  
      H ( Y ) = − Σ P ( y ) ∗ l o g 2 ( P ( y ) ) = − （（ 5 / 15 ） ∗ l o g 2 ( 5 / 15 ) + ( 10 / 15 ) ∗ l o g 2 ( 10 / 15 ) ) = 0.9182 H(Y) = - Σ P(y) \* log2(P(y)) = -（（5/15）\*log2(5/15)+(10/15)\*log2(10/15)) = 0.9182*H*(*Y*)=−Σ*P*(*y*)∗*log*2(*P*(*y*))=−（（5/15）∗*log*2(5/15)+(10/15)∗*log*2(10/15))=0.9182
    - 性别属性信息熵  
      H ( Y ∣ x 11 ) = − （（ 3 / 8 ） ∗ l o g 2 ( 3 / 8 ) + ( 5 / 8 ) ∗ l o g 2 ( 5 / 8 ) ） = 0.9543 H(Y|x11) = -（（3/8）\*log2(3/8)+(5/8)\*log2(5/8)） = 0.9543*H*(*Y*∣*x*11)=−（（3/8）∗*log*2(3/8)+(5/8)∗*log*2(5/8)）=0.9543

H ( Y ∣ x 12 ) = − （（ 2 / 7 ） ∗ l o g 2 ( 2 / 7 ) + ( 5 / 7 ) ∗ l o g 2 ( 5 / 7 ) ） = 0.8631 H(Y|x12) = -（（2/7）\*log2(2/7)+(5/7)\*log2(5/7)） = 0.8631*H*(*Y*∣*x*12)=−（（2/7）∗*log*2(2/7)+(5/7)∗*log*2(5/7)）=0.8631

H(Y|x11)表示男性信息熵，H(Y|x12)表示女性信息熵

* + - 活跃度属性信息熵  
      H ( Y ∣ x 21 ) = − （ 0 + ( 6 / 6 ) ∗ l o g 2 ( 6 / 6 ) ） = 0 H(Y|x21) = -（0 + (6/6)\*log2(6/6)） = 0*H*(*Y*∣*x*21)=−（0+(6/6)∗*log*2(6/6)）=0

H ( Y ∣ x 22 ) = − （（ 1 / 5 ） ∗ l o g 2 （ 1 / 5 ） + ( 4 / 5 ) ∗ l o g 2 ( 4 / 5 ) ） = 0.7219 H(Y|x22) = -（ （1/5）\* log2（1/5）+ (4/5)\*log2(4/5)） = 0.7219*H*(*Y*∣*x*22)=−（（1/5）∗*log*2（1/5）+(4/5)∗*log*2(4/5)）=0.7219

H ( Y ∣ x 23 ) = − ( ( 4 / 4 ) ∗ l o g 2 ( 4 / 4 ) + 0 ） = 0 H(Y|x23) = -((4/4)\*log2(4/4) + 0） = 0*H*(*Y*∣*x*23)=−((4/4)∗*log*2(4/4)+0）=0

H(Y|x21)表示活跃度高的信息熵，H(Y|x22)表示活跃度中的信息熵，H(Y|x23)表示活跃度低的信息熵

* + 条件信息熵：
    - 性别属性的条件信息熵  
      H ( Y ∣ X 1 ) = Σ P ( x 1 ) ∗ H ( Y ∣ x 1 ) = （（ 8 / 15 ） ∗ 0.9543 + （ 7 / 15 ） ∗ 0.8631 ） = 0.9112 H(Y|X1) = Σ P(x1) \* H(Y|x1) = （（8/15）\* 0.9543 + （7/15）\* 0.8631） = 0.9112*H*(*Y*∣*X*1)=Σ*P*(*x*1)∗*H*(*Y*∣*x*1)=（（8/15）∗0.9543+（7/15）∗0.8631）=0.9112  
      H(Y|X1)表示性别属性的条件信息熵
    - 活跃度属性的条件信息熵  
      H ( Y ∣ X 2 ) = Σ P ( x 2 ) ∗ H ( Y ∣ x 2 ) = （（ 6 / 15 ） ∗ 0 + （ 5 / 15 ） ∗ 0.7219 + （ 4 / 15 ） ∗ 0 ） = 0.2406 H(Y|X2) = Σ P(x2) \* H(Y|x2) = （（6/15）\* 0 + （5/15）\* 0.7219 + （4/15）\*0） = 0.2406*H*(*Y*∣*X*2)=Σ*P*(*x*2)∗*H*(*Y*∣*x*2)=（（6/15）∗0+（5/15）∗0.7219+（4/15）∗0）=0.2406
  + 信息增益：
    - 性别属性的信息增益  
      信息增益 = 初始信息熵 ( 类别信息熵 ) − 条件熵 = 0.9182 − 0.9112 = 0.007 信息增益 = 初始信息熵(类别信息熵) - 条件熵 =0.9182-0.9112=0.007信息增益=初始信息熵(类别信息熵)−条件熵=0.9182−0.9112=0.007
    - 活跃度属性的信息增益  
      信息增益 = 初始信息熵 ( 类别信息熵 ) − 条件熵 = 0.9182 − 0.2406 = 0.6776 信息增益 = 初始信息熵(类别信息熵) - 条件熵 =0.9182-0.2406=0.6776信息增益=初始信息熵(类别信息熵)−条件熵=0.9182−0.2406=0.6776

根据以上信息增益可以看出，**活跃度对用户流失的影响比较大**。在做特征选择或者数据分析的时候，应该重点考察活跃度这个指标

**3.决策树划分依据-------信息增益率（C4.5**[算法](https://edu.csdn.net/course/detail/40020?utm_source=glcblog&spm=1001.2101.3001.7020)**）**

在上面的介绍中，我们有意忽略了“编号”这一列，若把“编号”也作为一个候选划分属性，则根据信息增益公式可计算出它的信息增益率为0.9182，远大于其他候选划分属性。

实际上，**信息增益准则对可取值数目较多的属性有所偏好**，为减少这种偏好可能带来的不利影响，著名的**C4.5决策树算法不直接使用信息增益，而是使用“增益率”来选择最优划分属性**。

**增益率**：增益率是用前面的信息增益Gain(Y,x)和属性x对应的“固有值”的比值来共同定义的。  
G a i n r a t i o ( Y , x ) = G a i n ( Y , x ) / I V ( x ) Gain\_ratio(Y,x)=Gain(Y,x)/IV(x)*Gainr*​*atio*(*Y*,*x*)=*Gain*(*Y*,*x*)/*IV*(*x*)  
其中，IV是指属性分裂信息度量  
I V ( x ) = − Σ P ( x ) ∗ l o g 2 ( P ( x ) ) IV(x)=- ΣP(x) \* log2(P(x))*IV*(*x*)=−Σ*P*(*x*)∗*log*2(*P*(*x*))

* 案例：

以上面案例为例，计算性别和活跃度两个属性的信息增益率

* + 计算属性分裂信息度量
    - 性别属性分裂信息度量  
      I V ( x 1 ) = − Σ P ( x 1 ) ∗ l o g 2 ( P ( x 1 ) ) = − ( ( 8 / 15 ) ∗ l o g 2 ( 8 / 15 ) + ( 7 / 15 ) ∗ l o g 2 ( 7 / 15 ) ） = 0.9968 IV(x1)=- ΣP(x1) \* log2(P(x1)) = -((8/15)\*log2(8/15)+(7/15)\*log2(7/15)） =0.9968*IV*(*x*1)=−Σ*P*(*x*1)∗*log*2(*P*(*x*1))=−((8/15)∗*log*2(8/15)+(7/15)∗*log*2(7/15)）=0.9968
    - 活跃度属性分裂信息度量  
      I V ( x 2 ) = − Σ P ( x 2 ) ∗ l o g 2 ( P ( x 2 ) ) = − ( ( 6 / 15 ) ∗ l o g 2 ( 6 / 15 ) + ( 5 / 15 ) ∗ l o g 2 ( 5 / 15 ) + ( 4 / 15 ) ∗ l o g 2 ( 4 / 15 ) ) = 1.5656 IV(x2)=- ΣP(x2) \* log2(P(x2)) = -((6/15)\*log2(6/15)+(5/15)\*log2(5/15)+(4/15)\*log2(4/15)) =1.5656*IV*(*x*2)=−Σ*P*(*x*2)∗*log*2(*P*(*x*2))=−((6/15)∗*log*2(6/15)+(5/15)∗*log*2(5/15)+(4/15)∗*log*2(4/15))=1.5656
  + 计算信息增益率
    - 性别属性信息增益率  
      G a i n r a t i o ( Y , x 1 ) = G a i n ( Y , x 1 ) / I V ( x 1 ) = 0.007 / 0.9968 = 0.00702 Gain\_ratio(Y,x1)=Gain(Y,x1)/IV(x1) =0.007/0.9968 =0.00702*Gainr*​*atio*(*Y*,*x*1)=*Gain*(*Y*,*x*1)/*IV*(*x*1)=0.007/0.9968=0.00702
    - 活跃度属性信息增益率  
      G a i n r a t i o ( Y , x 2 ) = G a i n ( Y , x 2 ) / I V ( x 2 ) = 0.6776 / 1.5656 = 0.4328 Gain\_ratio(Y,x2)=Gain(Y,x2)/IV(x2) =0.6776/1.5656 =0.4328*Gainr*​*atio*(*Y*,*x*2)=*Gain*(*Y*,*x*2)/*IV*(*x*2)=0.6776/1.5656=0.4328

活跃度的信息增益率更高一些，所以在构建决策树时，优先选择。

通过这种方式，在选取节点的过程中，我们可以降低取值较多的属性的选取偏好。

* C4.5算法流程
* while(当前节点“不纯”)：
* 1.计算当前节点的类别熵（计算类别的信息熵）
* 2.计算当前阶段的属性熵（计算属性信息熵和属性条件信息熵）
* 3.计算各属性的信息增益（类别的信息熵-属性的条件信息熵）
* 4.计算各个属性的分裂信息度量
* 5.计算各个属性的信息增益率
* end while
* 当前阶段设置为叶子节点
* 为什么是用C4.5要好？
  + 用信息增益率来选择属性

克服了用信息增益来选择属性时偏向选择值多的属性的不足

* + 采用了一种后剪枝方法

避免树的高度无节制的增长，避免过度拟合数据

* + 对于缺失值的处理

对于处理缺少属性值，C4.5的策略是给它的每个可能值赋予一个概率。通过概率来给缺失的属性值赋值

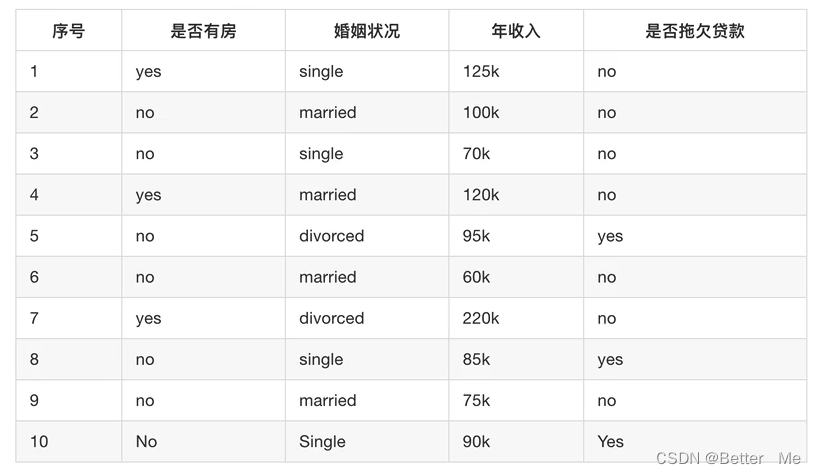
**4.决策树划分依据-------基尼值和基尼指数（CART）**

* **基尼值**： 衡量的是从一个随机样本中随机选取两个样本，其类别标签不一致的概率。基尼值越小，表示节点的纯度越高，样本的类别越趋于一致。

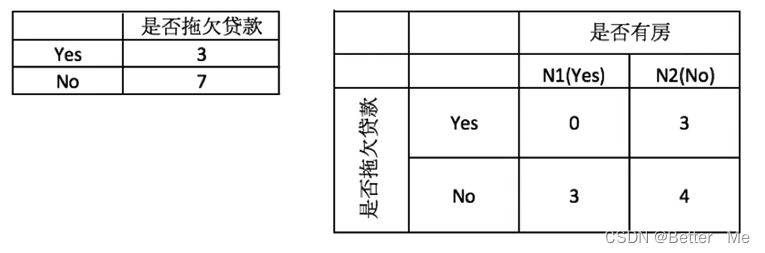
对于一个分类问题，假设有K个类别，每个类别的概率为 p1, p2, …, pK。则基尼值的计算公式如下：  
G i n i = 1 − Σ p i 2 Gini = 1 - Σ pi^2*Gini*=1−Σ*pi*2  
其中，pi表示第i个类别的概率。

* **基尼指数**：是基于基尼值而定义的一个指标，用于度量节点的不纯度。对于一个节点，假设该节点有M个类别，每个类别的样本数分别为 n1, n2, …, nM。基尼指数的计算公式如下：  
  G i n i I n d e x = Σ ( n / N ) ∗ G i n i i GiniIndex = Σ (n/N) \* Gini\_i*GiniIndex*=Σ(*n*/*N*)∗*Ginii*​  
  其中，n表示每个类别的样本数量，N表示总样本数量，Gini\_i表示每个子节点的基尼值。
* 案例：

根据下图列表，按照基尼指数的划分依据，做出决策树



* + 对数据集非序列标号属性(是否有房，婚姻状况，年收入)分别计算它们的Gini指数，**取Gini指数最小的属性作为决策树的根节点属性**。
  + 第一次大循环
  + 根节点的Gini值为：  
    G i n i ( 是否拖欠贷款 ) = 1 − ( ( 3 / 10 ) 2 + ( 7 / 10 ) 2 ) = 0.42 Gini(是否拖欠贷款) = 1 - ((3/10)²+(7/10)²)=0.42*Gini*(是否拖欠贷款)=1−((3/10)2+(7/10)2)=0.42
  + 当根据是否有房来进行划分时，Gini指数为：



G i n i ( 左子节点 ) = 1 − [ ( 0 / 3 ) 2 + ( 3 / 3 ) 2 ] = 0 Gini(左子节点) = 1 - [(0/3)²+(3/3)²]=0*Gini*(左子节点)=1−[(0/3)2+(3/3)2]=0

G i n i ( 右子节点 ) = 1 − [ ( 3 / 7 ) 2 + ( 4 / 7 ) 2 ] = 0.4898 Gini(右子节点) = 1 - [(3/7)²+(4/7)²]=0.4898*Gini*(右子节点)=1−[(3/7)2+(4/7)2]=0.4898

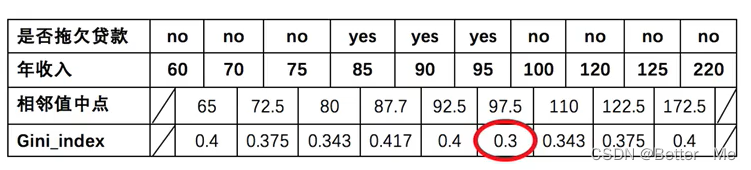
G i n i I n d e x ( 是否有房 ) = （ 3 / 10 ） ∗ 0 + （ 7 / 10 ） ∗ 0.4898 = 0.343 GiniIndex(是否有房) = （3/10）\*0 + （7/10）\* 0.4898=0.343*GiniIndex*(是否有房)=（3/10）∗0+（7/10）∗0.4898=0.343

* + 若按照婚姻状况属性来划分，属性婚姻状况有三个可能的取值{married,single,divorced}，则有三种分类情况：
    - 分类1为，married/{single,divorced}：  
      G i n i I n d e x ( 婚姻状况 ) = （ 4 / 10 ） ∗ 0 + （ 6 / 10 ） ∗ [ 1 − ( 3 / 6 ) 2 − ( 3 / 6 ) 2 ] = 0.3 GiniIndex(婚姻状况) = （4/10）\* 0 + （6/10）\* [1-(3/6)²-(3/6)²]=0.3*GiniIndex*(婚姻状况)=（4/10）∗0+（6/10）∗[1−(3/6)2−(3/6)2]=0.3
    - 分类2为，single/{married,divorced}：  
      G i n i I n d e x ( 婚姻状况 ) = （ 4 / 10 ） ∗ 0.5 + （ 6 / 10 ） ∗ [ 1 − ( 1 / 6 ) 2 − ( 5 / 6 ) 2 ] = 0.367 GiniIndex(婚姻状况) = （4/10）\* 0.5 + （6/10）\* [1-(1/6)²-(5/6)²]=0.367*GiniIndex*(婚姻状况)=（4/10）∗0.5+（6/10）∗[1−(1/6)2−(5/6)2]=0.367
    - 分类3为，divorced/{married,single}：  
      G i n i I n d e x ( 婚姻状况 ) = （ 2 / 10 ） ∗ 0 + （ 8 / 10 ） ∗ [ 1 − ( 2 / 8 ) 2 − ( 6 / 8 ) 2 ] = 0.4 GiniIndex(婚姻状况) = （2/10）\* 0 + （8/10）\* [1-(2/8)²-(6/8)²]=0.4*GiniIndex*(婚姻状况)=（2/10）∗0+（8/10）∗[1−(2/8)2−(6/8)2]=0.4

对比计算结果，根据婚姻状况属性来划分根节点时取Gini指数最小的分组作为划分结果，即分类1：married/{single,divorced}

* + 同理，按年收入属性来划分：

对于年收入属性为数值型属性，首先需要将数据按升序排序，然后从小到大依次用相邻值的中间值作为分割将样本划分为两组。比如当面对年收入为60和70这两个值时，我们算得其中间值为65。以中间值65作为分割点求出Gini指数。



由上图可以看出，年薪按97.5划分时Gini指数最小。

根据以上计算可知，三个属性划分根节点的Gini指数最小的有两个：年收入和婚姻状况属性，它们Gini指数都是0.3。此时，选取首先出现的属性married作为第一次划分。

第二次大循环

* + 接下来，采用同样的方法，分别计算剩下属性，其中根节点的Gini值为：  
    G i n i ( 是否拖欠贷款 ) = 1 − ( 3 / 6 ) 2 − ( 3 / 6 ) 2 = 0.5 Gini(是否拖欠贷款) = 1-(3/6)²-(3/6)²=0.5*Gini*(是否拖欠贷款)=1−(3/6)2−(3/6)2=0.5
  + 对于是否有房属性，可得：  
    G i n i I n d e x ( 是否有房 ) = （ 2 / 6 ） ∗ 0 + （ 4 / 6 ） ∗ [ 1 − ( 3 / 4 ) 2 − ( 1 / 4 ) 2 ] = 0.25 GiniIndex(是否有房) = （2/6）\* 0 + （4/6）\* [1-(3/4)²-(1/4)²]=0.25*GiniIndex*(是否有房)=（2/6）∗0+（4/6）∗[1−(3/4)2−(1/4)2]=0.25
  + 对于年收入属性：

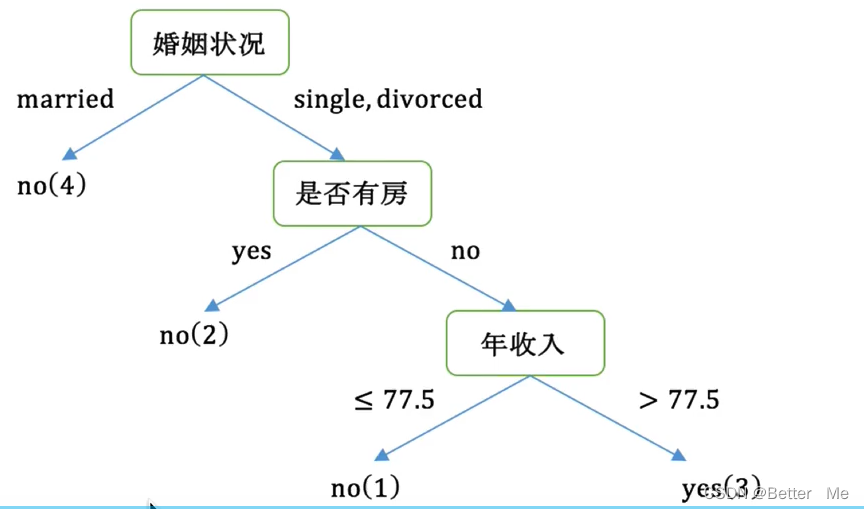


由上图可以看出，年薪按92.5划分时Gini指数最小。

所以这轮选取是否有房作为第二次划分。

第三次大循环

选取年收入作为第三次划分，最终的决策树如下所示：



* **CART的算法流程**：
* while(当前节点“不纯”):
* 1.遍历每个变量的每一种分割方式，找到最好的分割点
* 2.分割成两个节点N1和N2
* end while
* 每个节点足够“纯”为止

**三、决策树划分依据小结**

**1.常见决策树的启发函数比较**

| **名称** | **提出时间** | **分支方式** | **备注** |
| --- | --- | --- | --- |
| ID3 | 1975 | 信息增益 | ID3只能对离散属性的数据集构成决策树 |
| C4.5 | 1993 | 信息增益率 | 优化后解决了ID3分支过程中总喜欢偏向选择值较多的属性 |
| CART | 1984 | Gini指数 | 可以进行分类和回归，可以处理离散属性，也可以处理连续属性 |

**2.ID3算法**

存在的缺点：

1. ID3算法在选择根节点和各内部节点中的分支属性时，**采用信息增益作为评价标准**。信息增益的缺点时倾向于选择取值较多的属性，在有些情况下这类属性可能不会提供太多有价值的信息。
2. ID3算法**只能对描述属性为离散型属性的数据集构造决策树**。

**3.C4.5算法**

较于ID3做出的改进：

1. 用信息增益率来选择属性
2. 可以处理连续数值型属性
3. 采用了一种后剪枝方法
4. 对于缺失值的处理

C4.5的优缺点：

* 优点：

产生的分类规则易于理解，准确率较高

* 缺点：
  + 在构造树的过程中，需要对数据集进行多次的顺序扫描和排序，因而导致算法的低效。
  + C4.5只适合于能够驻留于内存的数据集，当训练集大的无法在内存容纳时程序无法运行。

**4.CART算法**

CART算法相比C4.5算法的分类方法，采用了简化的[二叉树](https://edu.csdn.net/course/detail/40020?utm_source=glcblog&spm=1001.2101.3001.7020)模型，同时特征选择采用了近似的基尼指数来简化计算。

**C4.5不一定时二叉树，但CART一定是二叉树**。

**5.多变量决策树(multi-variate decision tree)**

无论是ID3、C4.5、CART，在做特征选择的时候都是选择最优的一个特征来做分类决策，但是大多数，**分类决策不应该是由某一个特征决定的，而是应该由一组特征决定的**。这时候就需要多变量决策树来解决问题，这个算法的代表是OC1

**6.如何评估分割点的好坏？**

如果一个分割点可以将当前的所有节点分为两类，使得每一类都很“纯”，也就是同一类的记录较多，那么就是一个好分割点（简单来说，就是分割点的左右子树最好是叶子节点）

**构建决策树采用贪心算法，只考虑当前纯度差最大的情况作为分割点**

**四、**决策树算法**API**

* class sklearn.tree.DecisionTreeClassifier(criterion=“gini”, max\_depth=None, random\_state=None)
  + criterion
    - 特征选择标准
    - “gini"或者"entropy”，前者代表基尼值，后者代表信息增益。默认值是"gini"，即CART算法
  + min\_samples\_split
    - 内部节点再划分所需最小样本数
    - 这个值限制 子树继续划分的条件，如果某节点的样本数少于min\_samples\_split，则不会继续再尝试选择最优特征来继续划分。默认值是2，如果样本量不大，则不需要修改这个值；如果样本量数量级非常大，则需要增大这个值。
  + min\_samples\_leaf
    - 叶子节点最少样本数
    - 这个值限制了叶子节点最少的样本数，如果某叶子节点数目小于样本数，则会和兄弟节点一起被剪枝。默认值是1，如果样本量不大，则不需要修改这个值；如果样本量大，则建议增大这个值。
  + max\_depth
    - 决策树最大深度
    - 默认可以不输入，如果不输入的话，决策树在建立子树的时候不会限制子树的深度。一般来说，数据少或者特征少的时候可以不管这个值。如果模型样本量多，特征也多的情况下，建议限制决策树的深度。
  + random\_state
    - 随机数种子