问题描述：女婿受丈母娘欢迎程度

ID	Appearance	Income	Age	Profession	是否受欢迎
1	Good	Low	Older	Steady	N
2	Good	Low	Older	Unstable	N
3	Great	Low	Older	Steady	Y
4	Ah	Good	Older	Steady	Y
5	Ah	Great	Younger	Steady	Y
6	Ah	Great	Younger	Unstable	N
7	Great	Great	Younger	Unstable	Y
8	Good	Good	Older	Steady	N
9	Good	Great	Younger	Steady	Y
10	Ah	Good	Younger	Steady	Y
11	Good	Good	Younger	Unstable	Y
12	Great	Good	Older	Unstable	Y
13	Great	Low	Younger	Steady	Y
14	Ah	Good	Older	Unstable	N

1. ID3 决策树的构建

a) 计算所有数据的信息熵

$$H(D) = -\sum_{k=1}^{K} p_k log p_k = -\frac{9}{14}\log(\frac{9}{14}) - \frac{5}{14}\log(\frac{5}{14}) = 0.94$$

其中 pk 表示第 k 类样本所占的比例，注上式中的 log(P)的对数底是 2。

b) 计算每个属性的条件熵

- 以 Appearance 为例

$$\boldsymbol{H(Y|X) = -\sum_{i=1}^{n} p_i H(Y|X = x_i), p_i = P(X = x_i)}$$

$$\mathrm{H(Great)} = -\frac{4}{4}\log_2\left(\frac{4}{4}\right) = 0$$

$$\mathrm{H(Good)} = -\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right) = 0.971$$

$$\mathrm{H(Ah)} = -\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right) = 0.971$$

$\mathrm{H(Dv|FApp)} = \mathrm{p(Great)H(Great)+p(Good)H(Good)+p(Ah)H(Ah)}$

$= 4/14 * 0 + 5/14*0.971 + 5/14*0.971$

$= 0.693$

最终得到以不同属性进行分类时的条件熵：

$$H(D|F_{App}) = 0.693 \qquad H(D|F_{Age}) = 0.789$$

$$H(D|F_{Inc}) = 0.911 \qquad H(D|F_{Job}) = 0.892$$

c) 计算每个属性进行分割时信息增益

$$(Dv|F_{App})=H(D)-H(Dv|F_{App})=0.94-0.693=0.246$$
$$(Dv|F_{Inc})=H(D)-H(Dv|F_{Inc})=0.94-0.911=0.029$$
$$(Dv|F_{Age})=H(D)-H(Dv|F_{Age})=0.94-0.798=0.151$$
$$(Dv|F_{Job})=H(D)-H(Dv|F_{Job})=0.94-0.892=0.048$$

选择信息熵增加最多的属性 Appearance 首先进行样本分割。

d) Appearace = Good 下的样本包括

ID	Appearance	Income	Age	Profession	是否受欢迎
1	Good	Low	Older	Steady	N
2	Good	Low	Older	Unstable	N
3	Good	Good	Older	Steady	N
4	Good	Great	Younger	Steady	Y
5	Good	Good	Younger	Unstable	Y

对以上数据，计算 Income，Age 和 Profession 信息增益。

H(Dv|F Inc)=2/5 * H(Dv|Low) + 2/5 * H(Dv|Good) + 1/5 * H(Dv|Great) = 0.4

H(Dv|F Age) = 3/5 * H(Dv|Older) + 2/5 * H*(Dv|Younger)= 0

H(Dv|F Profession) = 3/5 * H(Dv|Steady) + 2/5 * H(Dv|Unstable) = 0.95

$$(Dv|F_{Inc})=H(Dv)-H(Dv|F_{Inc})=0.971-0.4=0.571$$
$$(Dv|F_{Age})=H(Dv)-H(Dv|F_{Age})=0.971$$
$$(Dv|F_{Job})=H(Dv)-H(Dv|F_{Job})=0.021$$

按照 Age 进行分类，信息增益最高，所以 Appearace = Good 下的样本再按照 Age 属性进行分类

按照 Age 分类后，Age=Older 的样本都是 N，Younger 的样本都是 Y，因为分割后样本都是同一类别，所以不需要再进行分割了，Age 分类后的节点都是叶子节点。

e) Appearace = Ah 下的样本包括：

ID	Appearance	Income	Age	Profession	是否受欢迎
1	Ah	Good	Older	Steady	Y
2	Ah	Great	Younger	Steady	Y
3	Ah	Great	Younger	Unstable	N
4	Ah	Good	Younger	Steady	Y
5	Ah	Good	Older	Unstable	N

对以上数据，计算 Income，Age 和 Profession 信息增益。

H(Dv|F Inc)=2/5 * H(Dv|Great) + 3/5 * H(Dv|Good) =0.9508

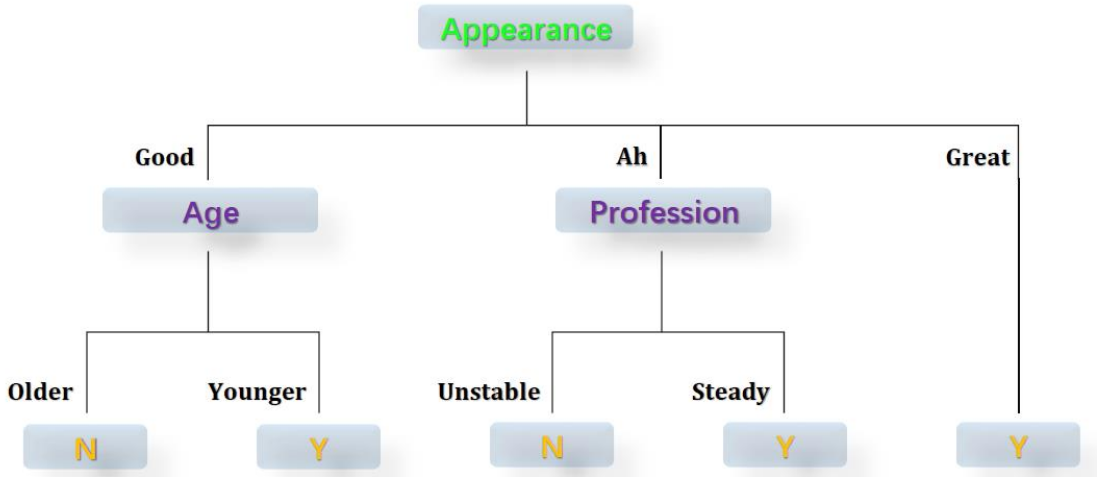
H(Dv|F Age) = 2/5 * H(Dv|Older) + 3/5 * H*(Dv|Younger)= 0.9508

H(Dv|F Profession) = 3/5 * H(Dv|Steady) + 2/5 * H(Dv|Unstable) = 0

$$(Dv|F_{Inc})=H(Dv)-H(Dv|F_{Inc})=0.971-0.9508=0.0202$$
$$(Dv|F_{Age})=H(Dv)-H(Dv|F_{Age})=0.971-0.9508=0.0202$$
$$(Dv|F_{Job})=H(Dv)-H(Dv|F_{Job})=0.971$$

因为 Profession 的信息增益最大，所以按照 Profession 进行分割。分割后的样本都各属于同一个类别，所以不需要再进行分割。.

f) Appearace = Great 下的样本都是 Y 类别的所以无需再分类。最终树的构建如下图。



2. C4.5 决策树算法和 CART 算法

- GainRatio 是什么?

$$GainRatio(D,T)=\frac{Gain(D,T)}{IV(T)}$$

$$IV(T)=-\sum_{v=1}^{V}\frac{|D^v|}{|D|}log_2\frac{|D^v|}{|D|}$$

- 为什么我们倾向于使用 GainRatio?
  - 信息增益对于取值数较多的属性有所偏好，为减少这种不利影响，C4.5 采用增益率来选择最优划分。IV(a)为属性 a 的固有值，a 可能取值越多，IV(a)越大。
- 怎样使用信息增益率进行节点划分?

- 因为增益率对于取值数较小的属性有所偏好。所以 C4.5 不是直接选择增益率最大的候选划分属性，而是先从候选划分属性中找出信息增益高于平均水平的属性，再从中选出增益率最高的。

- Gini Index?

$$Gini\,(D) = 1 - \sum_{k=1}^{N} p_k^2$$

  - Gini 值反映了从数据集 D 中随机抽取两个样本，其类别标记不一致的概率，因此 Gini 系数越小，则数据集的 D 的纯度越高。

  - a 的 Gini Index 为

$$Gini_a = \sum_{v=1}^{V} \frac{|D^v|}{|D|} Gini\,(D^v)$$

- 如何使用 Gini Index 划分节点?

  - Gini 指数在选择属性 a 后，选择使得 Gini 指数最小的属性作为当前划分的最优属性。

- Why people are likely to use C4.5 or CART rather than ID3?

  - C4.5 算法比 ID3 算法有以下改进

    - ID3 不能处理连续特征，C4.5 将连续的特征离散化。

    - 信息增益容易偏向取值较多的特征，而 C4.5 引入信息增益比的概念，来偏向于取值较多的特征的问题。

    - ID3 无法处理缺失值，但是 C4.5 可以在某些特征缺失的情况下选择划分的属性（通过将数据划分为有特征值和无特征值的数据来处理），并且在选型了划分属性后处理如何将样本划入响应子节点的问题（通过将缺失样本划入所有子节点，不过样本的权重按照各子节点的样本数量比例来分配）。

    - ID3 有过拟合的问题，C4.5 引入了正则化系数进行初步的剪枝。

  - CART 算法比 C4.5 算法又有以下改进：

    - C4.5 的剪枝算法有优化的空间，CART 通过后剪枝和交叉验证来选择最合适的决策树

- C4.5 生成的是多叉树，CART 采用二叉树模型提高运算效率
- C4.5 只能用于分类，CART 算法可以用于回归算法。
- C4.5 使用了上模型，计算对数一般比较耗时。CART 算法的 GINI INDEX 能够提高运算效率又不牺牲太多准确性。

以上就是 C4.5 算法和 CART 算法更受人们青睐的原因。