Decsison Tree—IGain&IGainRate

c1p

setwd("d:/data")  
a<-read.csv("tennis.csv",header = T)  
a

## NO. Outlook Temperature Humidity Wind Play  
## 1 1 Sunny Hot High Weak No   
## 2 2 Sunny Hot High Strong No   
## 3 3 Overcast Hot High Weak Yes   
## 4 4 Rain Mild High Weak Yes   
## 5 5 Rain Cool Normal Weak Yes   
## 6 6 Rain Cool Normal Strong No   
## 7 7 Overcast Cool Normal Strong Yes   
## 8 8 Sunny Mild High Weak No   
## 9 9 Sunny Cool Normal Weak Yes   
## 10 10 Rain Mild Normal Weak Yes   
## 11 11 Sunny Mild Normal Strong Yes   
## 12 12 Overcast Mild High Strong Yes   
## 13 13 Overcast Hot Normal Weak Yes   
## 14 14 Rain Mild High Strong No

信息增益是衡量信息前后之间熵的变化 熵是指一个系统的信息熵越高就越无序，信息熵越低就越有序，信息熵越高，使其有序所要消耗的信息量就越大。

X={X1,X2,…Xn} 对应每一项的概率是P={P1,P2….Pn}

熵计算方法

信息增益是指新的划分方法后熵和原来熵的变化

IG(X,Y)=E(X)-E(Y)

PS:关于划分方法，以上面的表格为例，Play中YES,NO的划分方法

原始Play这列划分只是对应YES、NO

Play原始的熵计算是

E(Play)= =0.940286

对用Outlook列对Play进行划分后的熵计算是

(Outlook中有三类Overcast、Rain、Sunny)

E(Overcast)=

E(Rain)=

E(Sunny)=

IGain(Outlook)== 0.2467498

计算每一列的信息增益

=

=

=

IGain(Temperature)== 0.02922257

…

计算后发现信息增益最大的列是Outlook，选择Outlook作为根，也就是第一个分叉部分

信息增益比率

## NO. Outlook Temperature Humidity Wind Play  
## 1 1 Sunny Hot High Weak No   
## 2 2 Sunny Hot High Strong No   
## 3 3 Overcast Hot High Weak Yes   
## 4 4 Rain Mild High Weak Yes   
## 5 5 Rain Cool Normal Weak Yes   
## 6 6 Rain Cool Normal Strong No   
## 7 7 Overcast Cool Normal Strong Yes   
## 8 8 Sunny Mild High Weak No   
## 9 9 Sunny Cool Normal Weak Yes   
## 10 10 Rain Mild Normal Weak Yes   
## 11 11 Sunny Mild Normal Strong Yes   
## 12 12 Overcast Mild High Strong Yes   
## 13 13 Overcast Hot Normal Weak Yes   
## 14 14 Rain Mild High Strong No

假设表中NO.也是一列有用的信息列种每一项都不重复计算NO.列的E(NO)

E(NO)=

=0

该列的信息增益比最大。但这种划分效果极差

为了解决这个问题，提出信息增益比。

**信息增益比公式：**

**IGainR=IGain/I**

IGain就是信息增益

I公式是I=E(X)= 就是求熵

之前的求熵是

这个熵是关于Play这列，的熵

但这个I是这列的熵

列：

IGainRate(Outlook)= IGain(Outlook)/