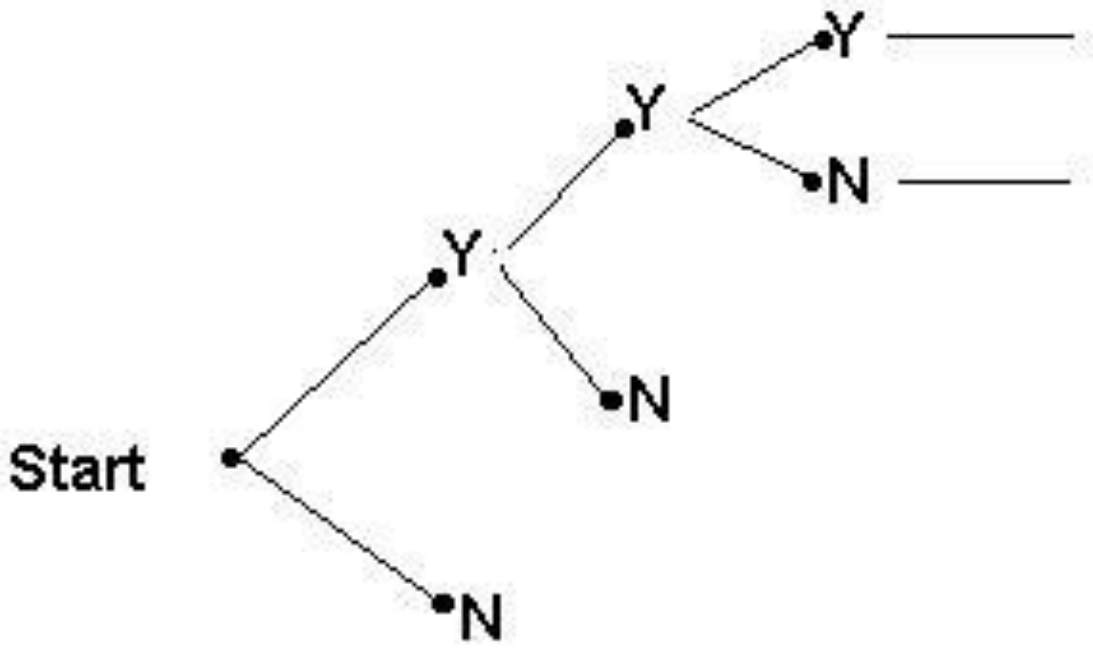
Makine Öğrenmesi

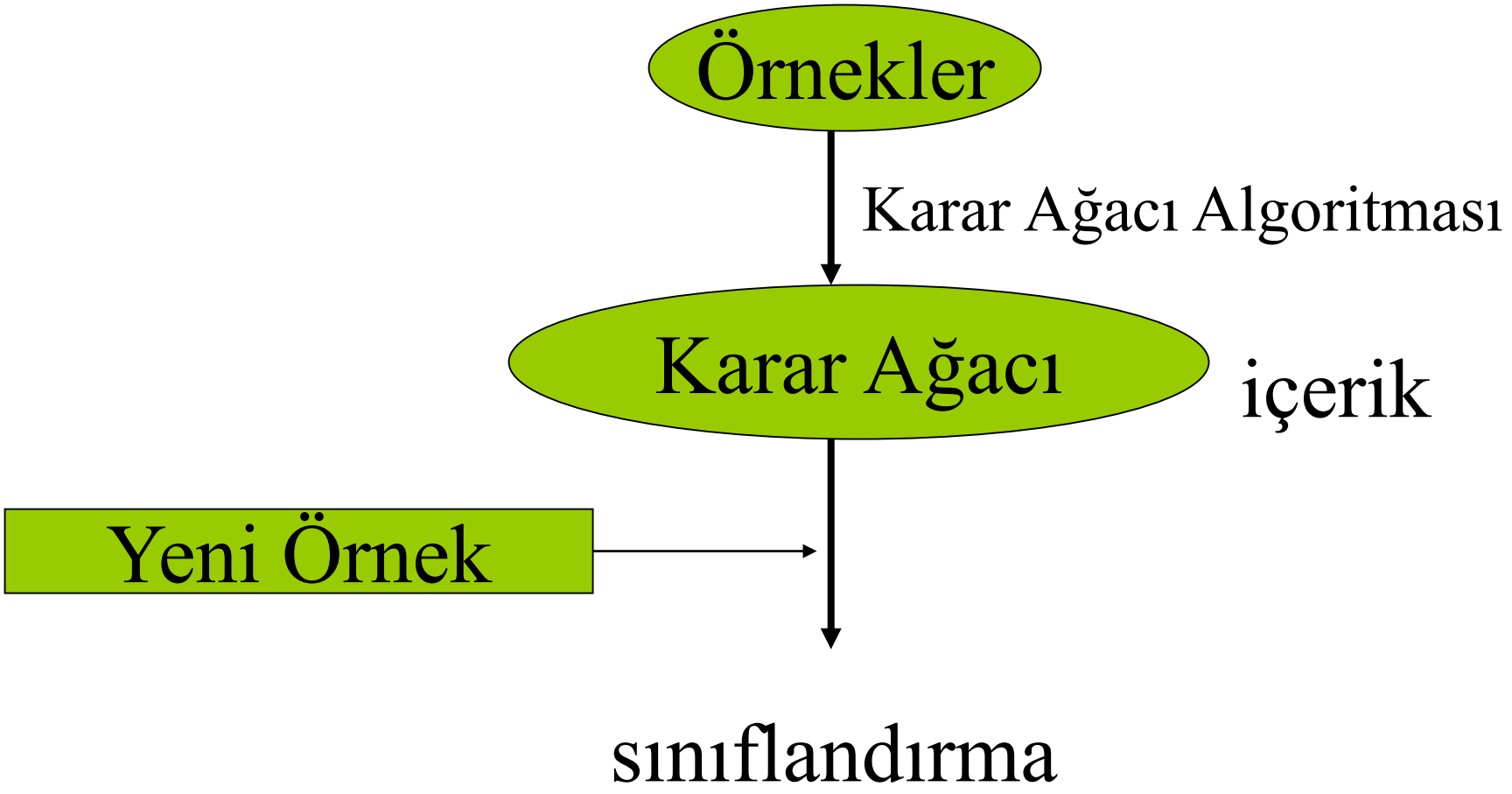
Karar Ağaçları

Doç. Dr. İlhan AYDIN

Karar Ağaçları (Decision trees)

Karar Ağaçları, Tree yapısında olup bir olayın sonuçlandırılmasında sorunun cevabına göre hareket ederler.




Örnekler
Karar Ağacı Algoritması
Karar Ağacı
içerik
Yeni Örnek
sınıflandırma

- Karar Ağacı algoritması 2 aşamadan oluşmaktadır:
  - Ağacı oluşturma
    - En başta bütün öğrenme kümesi ağaçtadır.
  - Ağacı budama
    - Öğrenme kümesindeki gürültülü verilerden oluşan ve test kümesinde hataya neden olan dallar silinir.

- **Entropy'e Dayalı Algoritmalar**

  ID3, C 4.5

- **Sınıflandırma ve Regresyon Ağaçları (CART)**

  Twoing, Gini

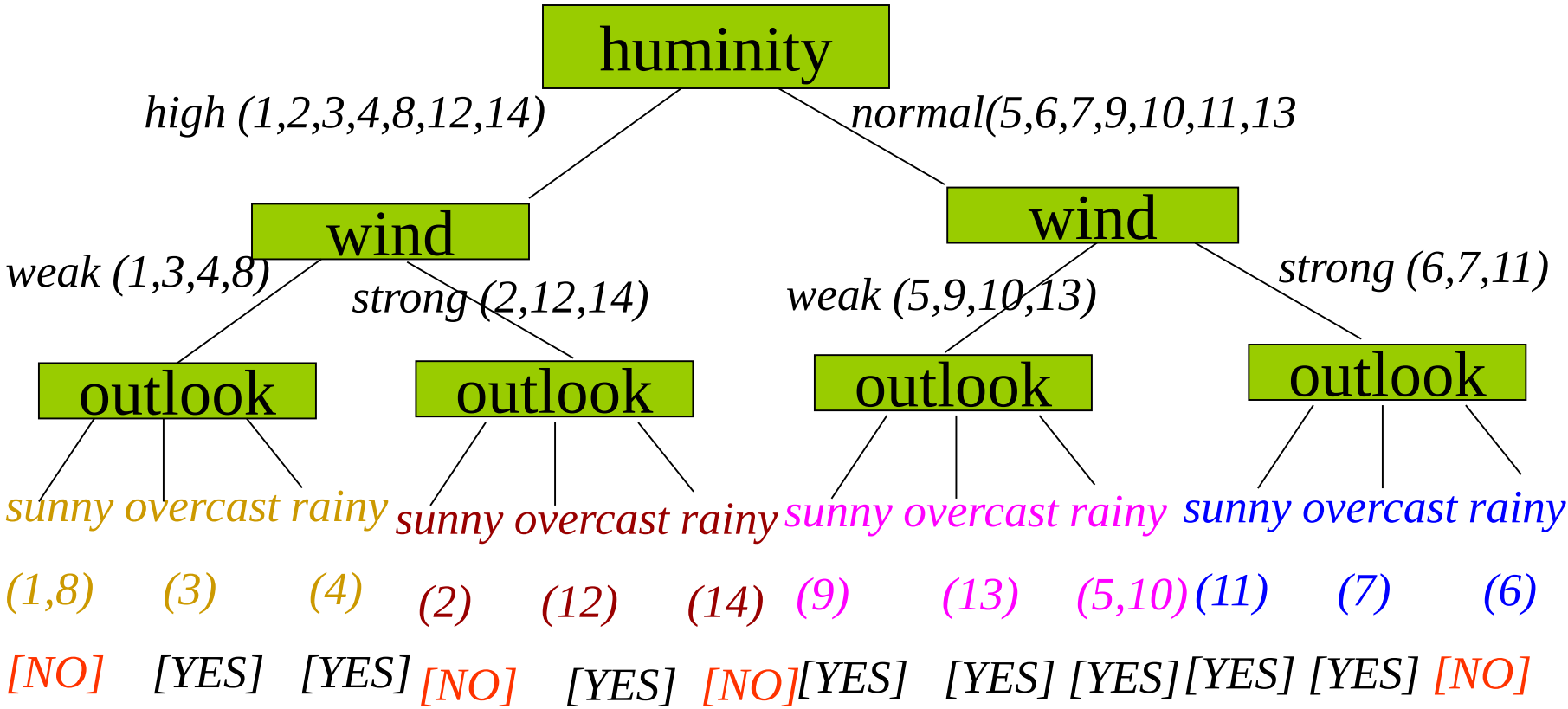
- **Bellek Tabanlı Sınıflandırma Algoritmaları**

  K-EnYakın Komşu

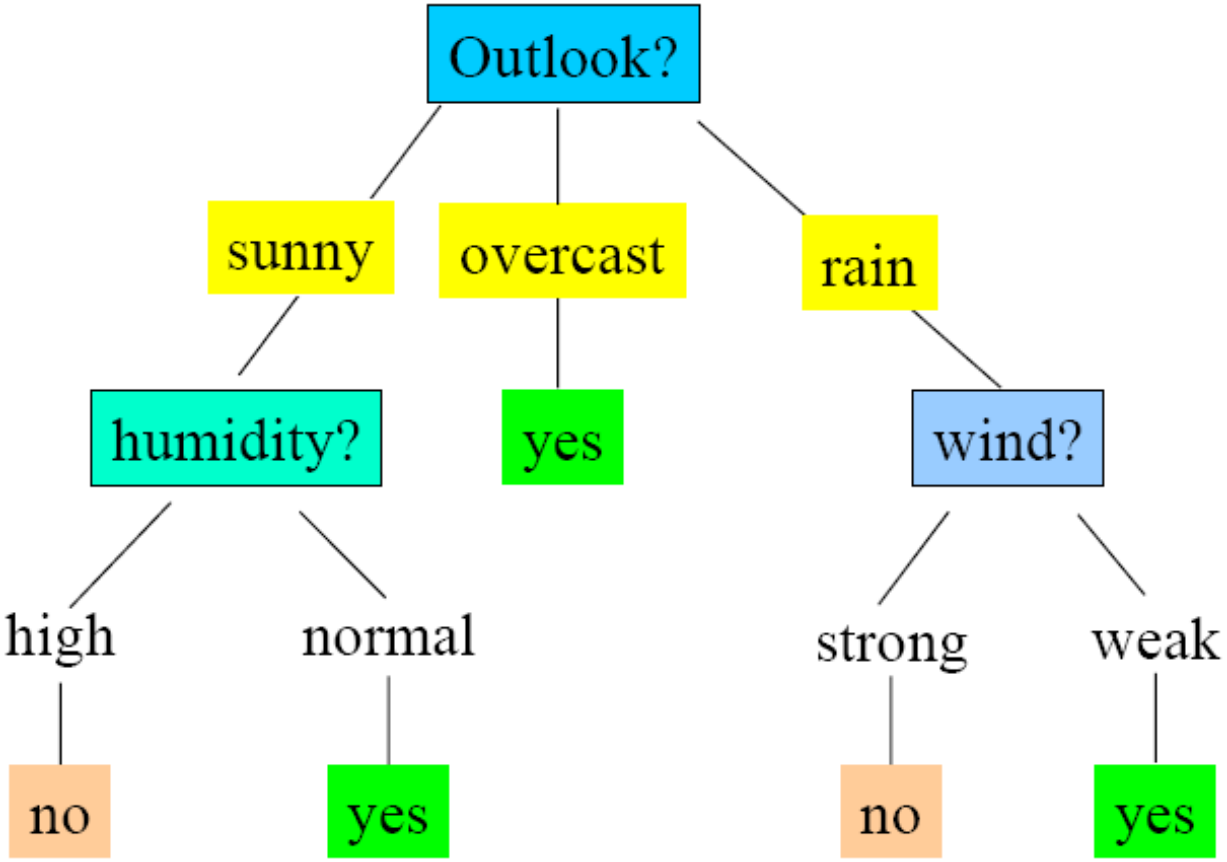
Örnek Karar Ağacı : Play Tennis ? (hava tenis oynamaya uygun mu?)

Day	Outlook	Temperature	Huminity	Wind	PlayTennis
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

Özelliklerden biri kök seçilir (outlook, temp, huminity, wind)



veya


Outlook?
sunny
overcast
rain
humidity?
yes
wind?
high
normal
strong
weak
no
yes
no
yes

Karar Ağacı iyi bir çözümdür

ancak

optimum değildir

optimum bir karar ağacının oluşturulması için bir kuralın olması gerekir

Bilgi kazancı ölçümü: Entropi

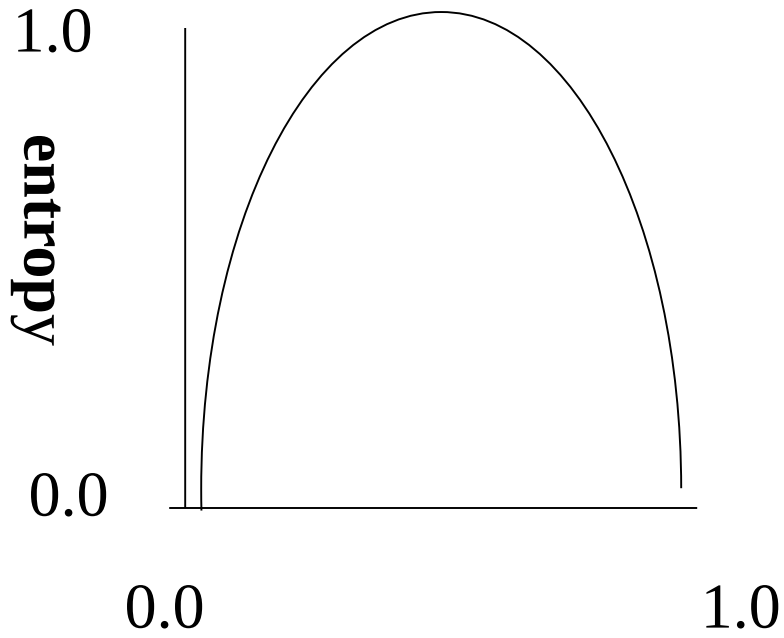
- Entropy rastgeleliğin, belirsizliğin ve beklenmeyen durumun ortaya çıkma olasılığını gösterir.
  - Sınıflandırmada
    - örneklerin tümü aynı sınıfa ait ise entropy=0
    - örnekler sınıflar arasında eşit dağılmış ise entropi=1
    - örnekler sınıflar arasında rastgele dağılmış ise 0<entropi<1

Information Gain
(Bilgi Kazancı- maksimum kazanç)

$$\text{Entropy}(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

Bütün örnekler aynı sınıfa ait ise E(S)=0 (homojen)

Bütün örnekler sınıflara eşit dağılmış ise E(S)=1 (heterojen)



Karar Ağacı oluşturma algoritması (ID 3) Adım:1

- Karar ağacının hangi kararı alacağı belirlenir.
  - Örnek veri setinde tenis oynamaya gidilip gidilmeyeceğine (play tennis) karar verilecektir.

Sistemin Entropy si hesaplanır Adım:2

$$E(s_1,s_2,\ldots,s_m) = -\sum_{i=1}^{m} \frac{s_i}{s} \log_2 \frac{s_i}{s}$$

m *sınıf sayısı*

14 tane örnek 9 tane YES 5 tane NO

$$\text{Entropy}(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

$$\text{Entropy}(S) \equiv -(9/14)\log_2(9/14) - (5/14)\log_2(5/14)$$

$$\text{Entropy}(S) \equiv 0.940$$

Adım :3

- Ağacın en üstünde yani kökte/root ta konumlanacak nitelik (özellik) belirlenir.

  Neye göre belirlenir?

- Bilgi kazancı (information gain) en yüksek olan **ÖZELLİK** ağacın en üstünde konumlandırılır.

Adım : 4

- Bilgi kazancı (information gain) nasıl hesaplanır?

A özelliğinin, S örneği için kazancı (information gain)

$$Gain(S,A) \equiv Entropy(S) - \Sigma\ P(v)\ Entropy(S(v))$$

v: Values of A

$$P(v) \equiv |S(v)| / |S|$$

S:[9+,5-]

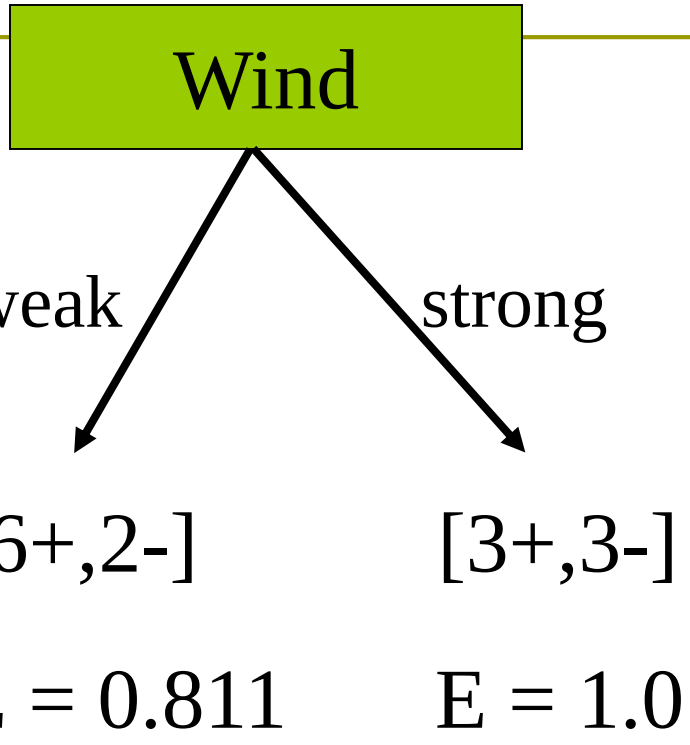
E = 0.940

Gain(S,wind) = ?

Gain(S,huminity) = ?

Gain(S,temperature) = ?

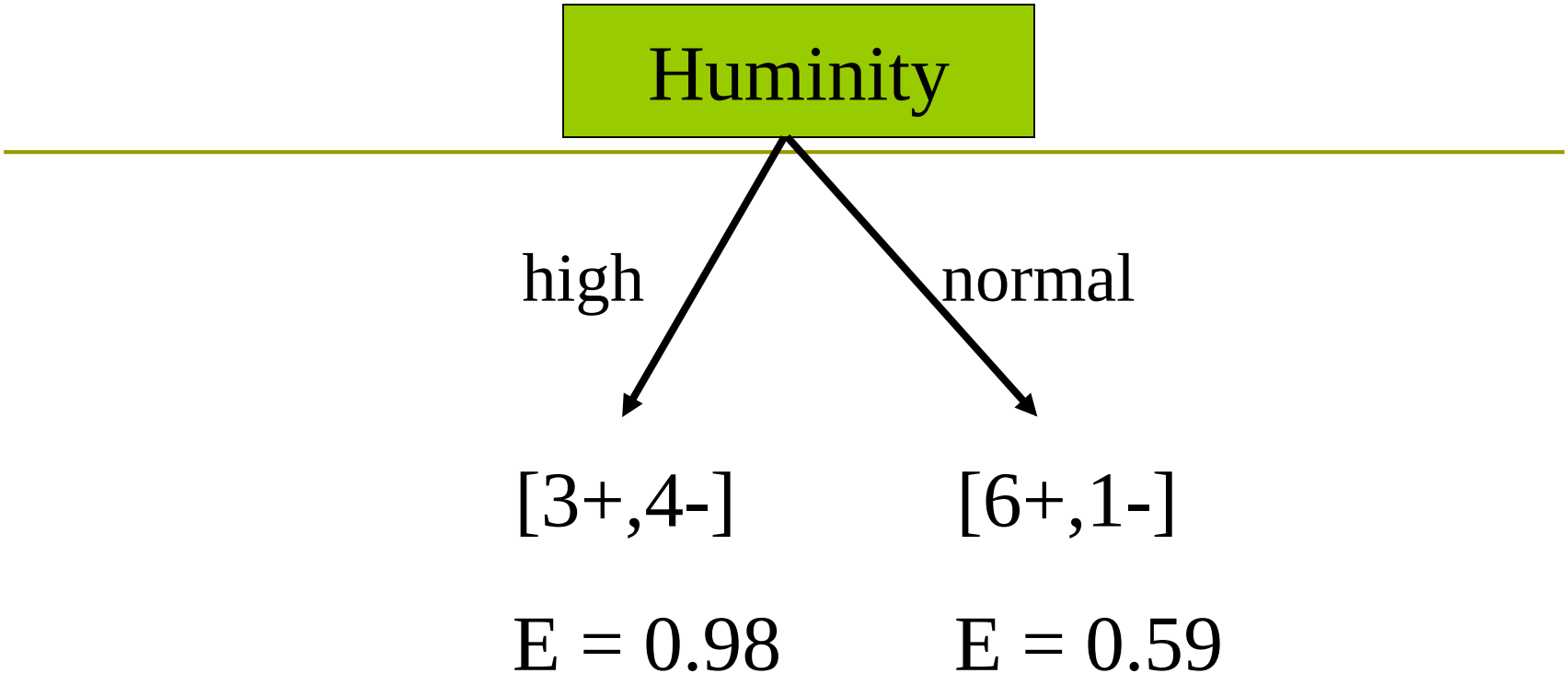
Gain(S,outlook) = ?



Gain(S,wind) = 0.940 – [(8/14)[-(6/8)log$_2$(6/8) – (2/8)log$_2$(2/8)]]

–[(6/14)[-(3/6)log$_2$(3/6) – (3/6)log$_2$(3/6)]]

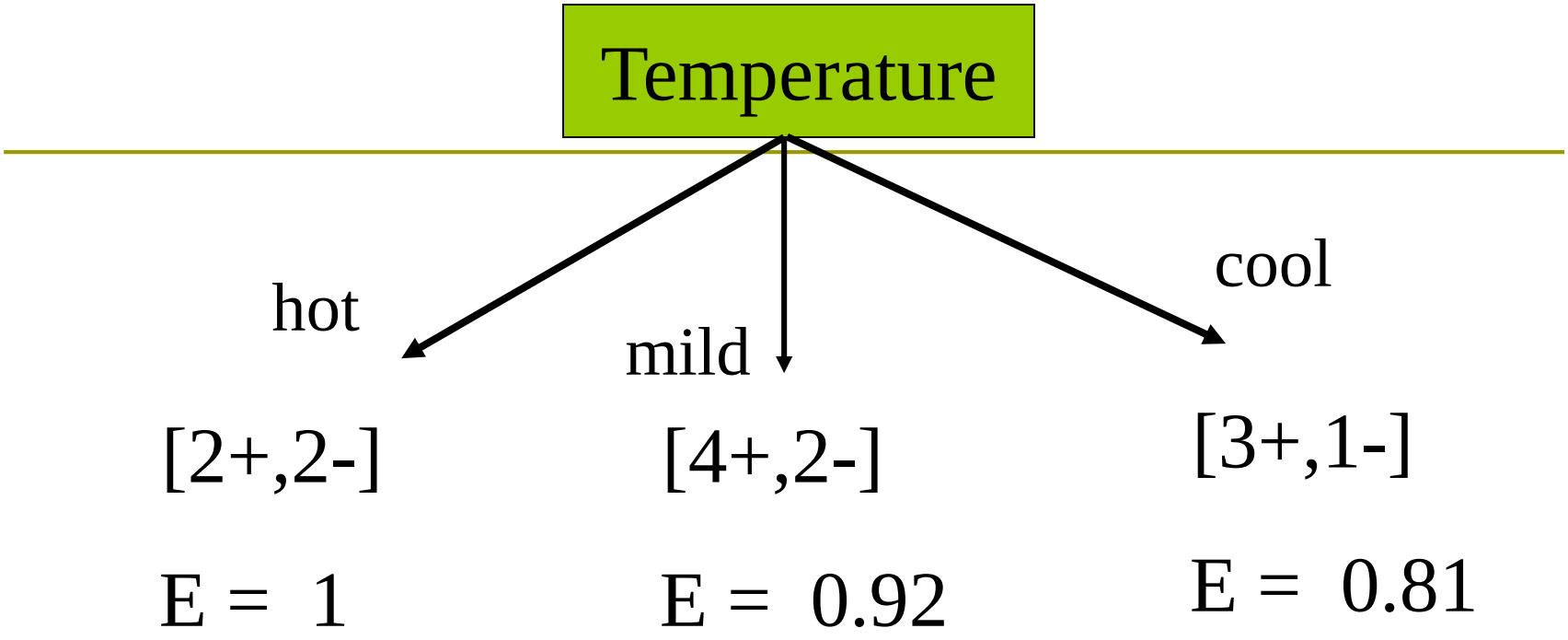
= 0.048



Gain(S,huminity) = 0.940 – [(7/14)[-(3/7)log$_2$(3/7) – (4/7)log$_2$(4/7)]]

–[(7/14)[-(6/7)log$_2$(6/7) – (1/7)log$_2$(1/7)]]

= 0.15



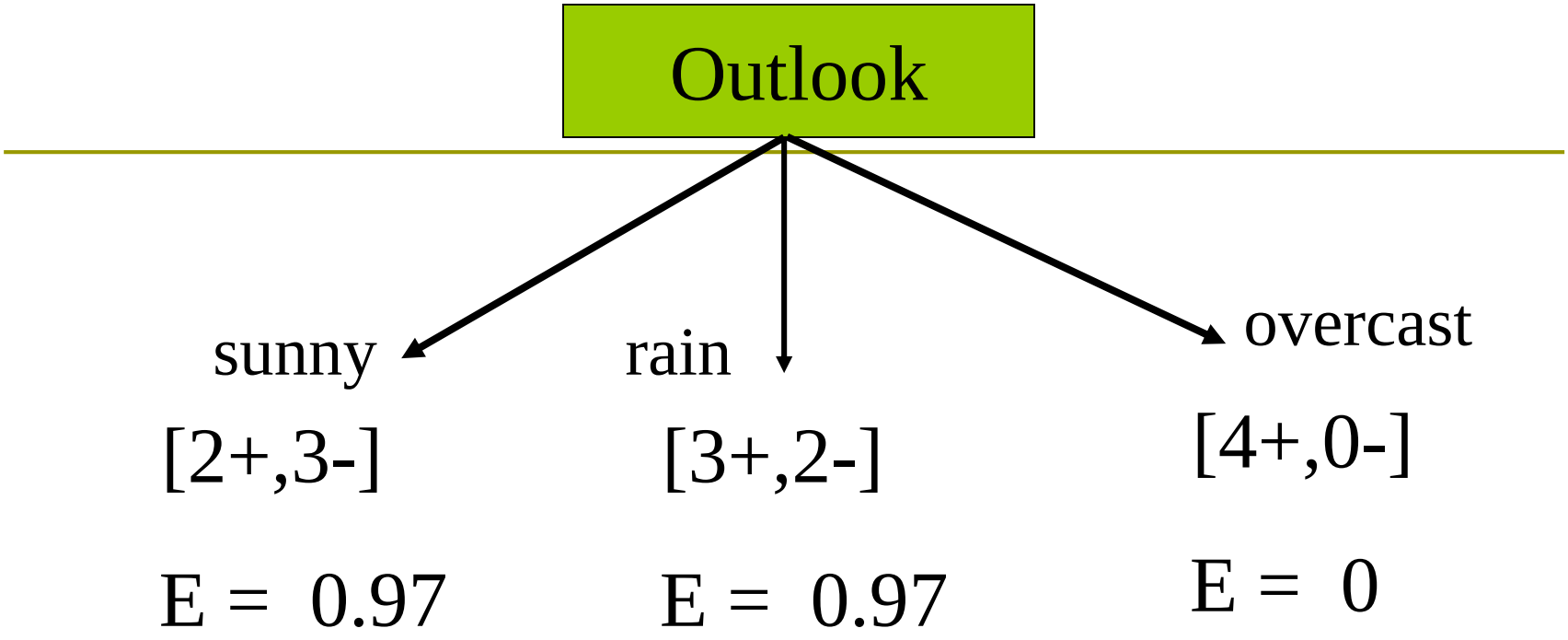
Gain(S,temperature) = 0.940

$- [(4/14)[-(2/4)\log_2(2/4) - (2/4)\log_2(2/4)]]$

$-[(6/14)[-(4/6)\log_2(4/6) - (2/6)\log_2(2/6)]]$

$-[(4/14)[-(3/4)\log_2(3/4) - (1/4)\log_2(1/4)]]$

= 0.027



Gain(S,temperature) = 0.940

$- [(5/14)[-(2/5)\log_2(2/5) - (3/5)\log_2(3/5)]]$

$-[(5/14)[-(3/5)\log_2(3/5) - (2/5)\log_2(2/5)]]$

$-[(4/14)[-(4/4)\log_2(4/4) - (0/4)\log_2(0/4)]]$

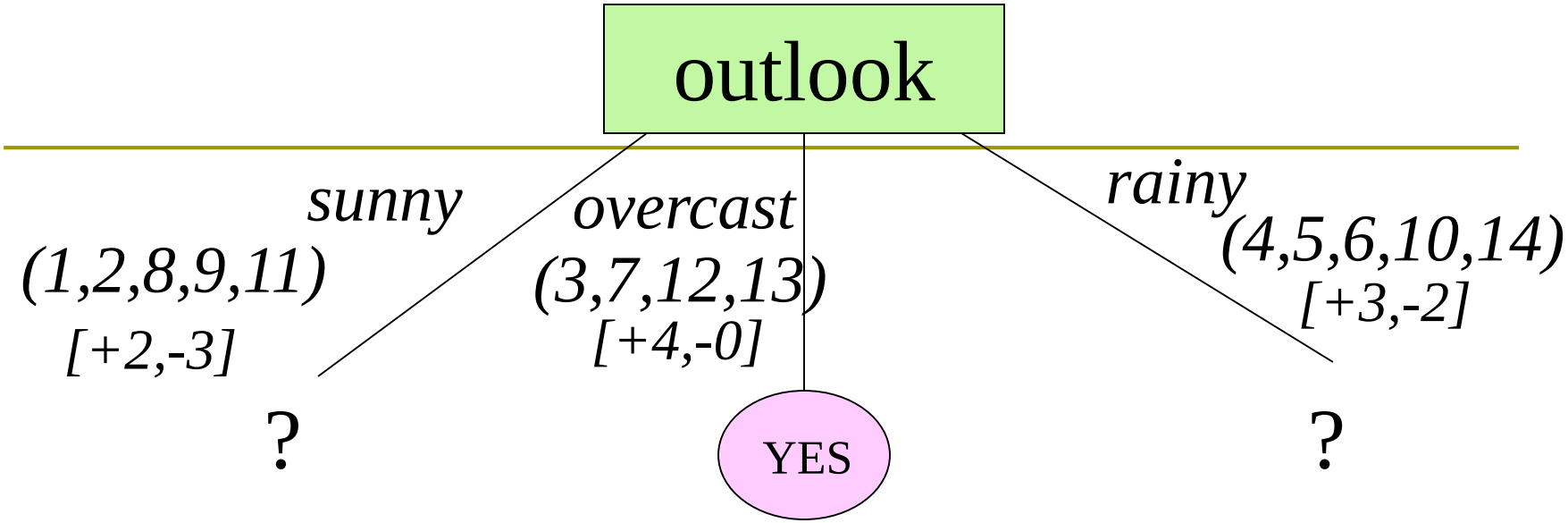
= 0.246

Gain(S,wind) = 0.048

Gain(S,huminity) = 0.15

Gain(S,temperature) = 0.027

Gain(S,outlook) = 0.246



S_{sunny}=[+2,-3]

$E(sunny) = -(2/5)\log_2(2/5) - (3/5)\log_2(3/5) = 0.97$

$Gain(S_{sunny,huminity}) = ?$

$Gain(S_{sunny,temp}) = ?$

$Gain(S_{sunny,wind}) = ?$

$$Gain(S_{sunny,huminity}) = 0.97$$

$$-(2/5)[-(2/2)\log_2(2/2)-(0/2)\log_2(0/2)]$$

$$-(3/5)[-(0/3)\log_2(0/3)-(3/3)\log_2(3/3)]$$

$$= 0.97$$

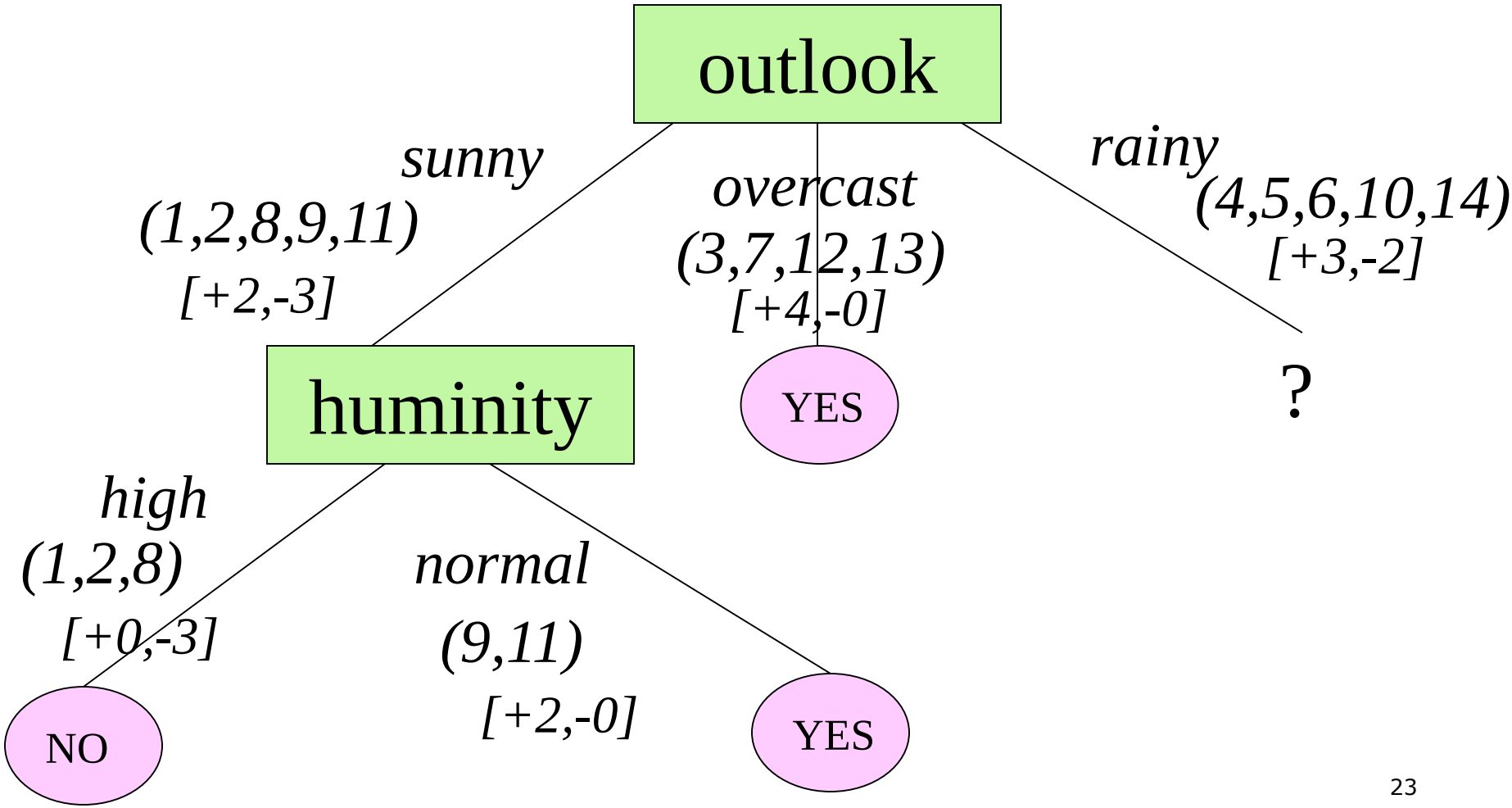
$$Gain(S_{sunny,wind}) = 0.97$$

$$-(3/5)[-(1/3)\log_2(1/3)-(2/3)\log_2(2/3)]$$

$$-(2/5)[-(1/2)\log_2(1/2)-(1/2)\log_2(1/2)]$$

$$= 0.019$$

$$Gain(S_{sunny,temp}) = 0.57$$


outlook
sunny
(1,2,8,9,11)
[+2,-3]
overcast
(3,7,12,13)
[+4,-0]
rainy
(4,5,6,10,14)
[+3,-2]
huminity
YES
?
high
(1,2,8)
[+0,-3]
normal
(9,11)
[+2,-0]
NO
YES

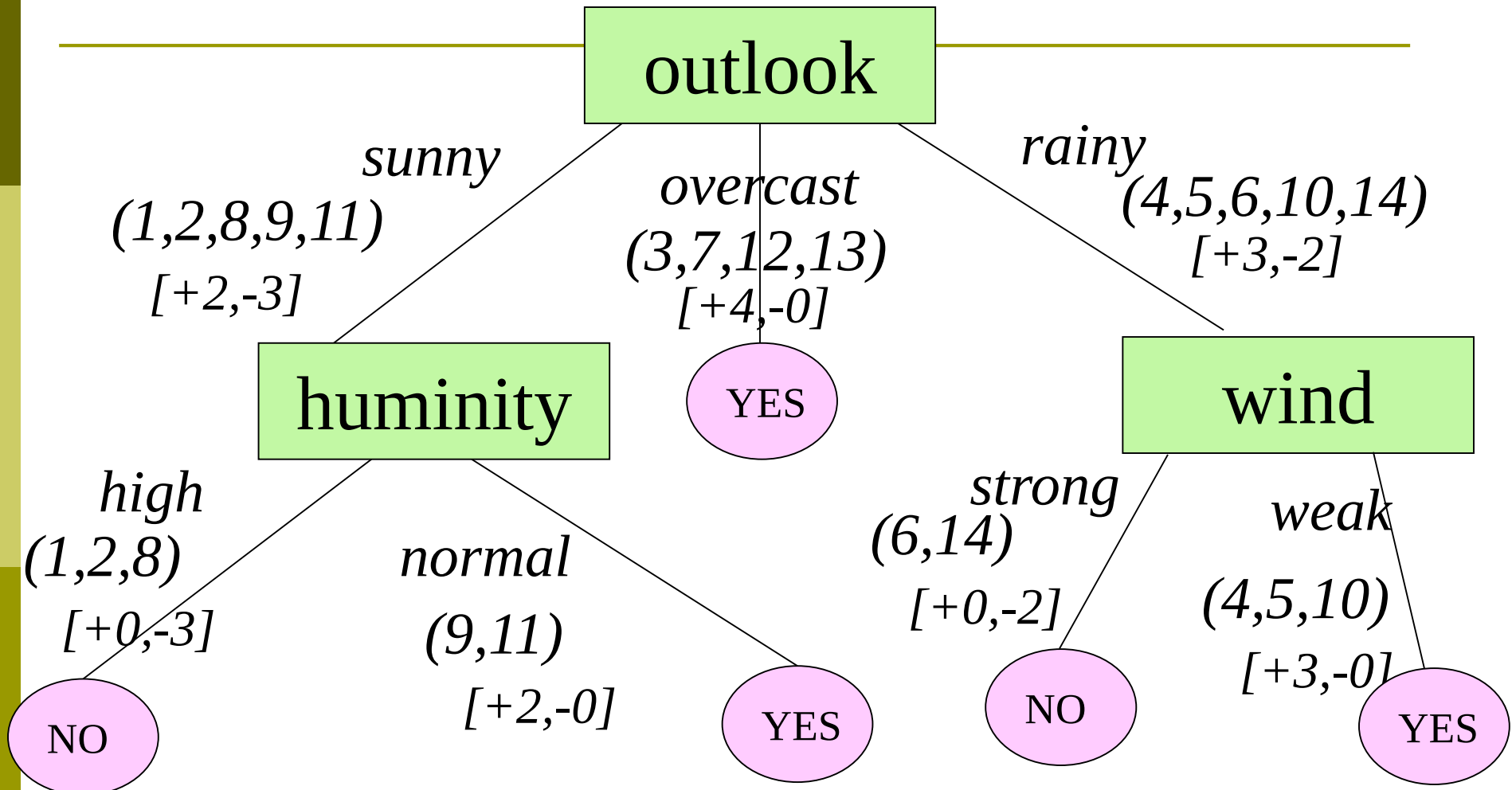
Aynı işlem

$$Gain(S_{rainy,huminity}) = ?$$

$$Gain(S_{rainy,temp}) = ?$$

$$Gain(S_{rainy,wind}) = ?$$

bulmak için yapılır.


outlook
sunny
(1,2,8,9,11)
[+2,-3]
overcast
(3,7,12,13)
[+4,-0]
rainy
(4,5,6,10,14)
[+3,-2]
huminity
YES
wind
high
(1,2,8)
[+0,-3]
normal
(9,11)
[+2,-0]
strong
(6,14)
[+0,-2]
weak
(4,5,10)
[+3,-0]
NO
YES
NO
YES

Karar Ağacı kullanarak sınıflandırma

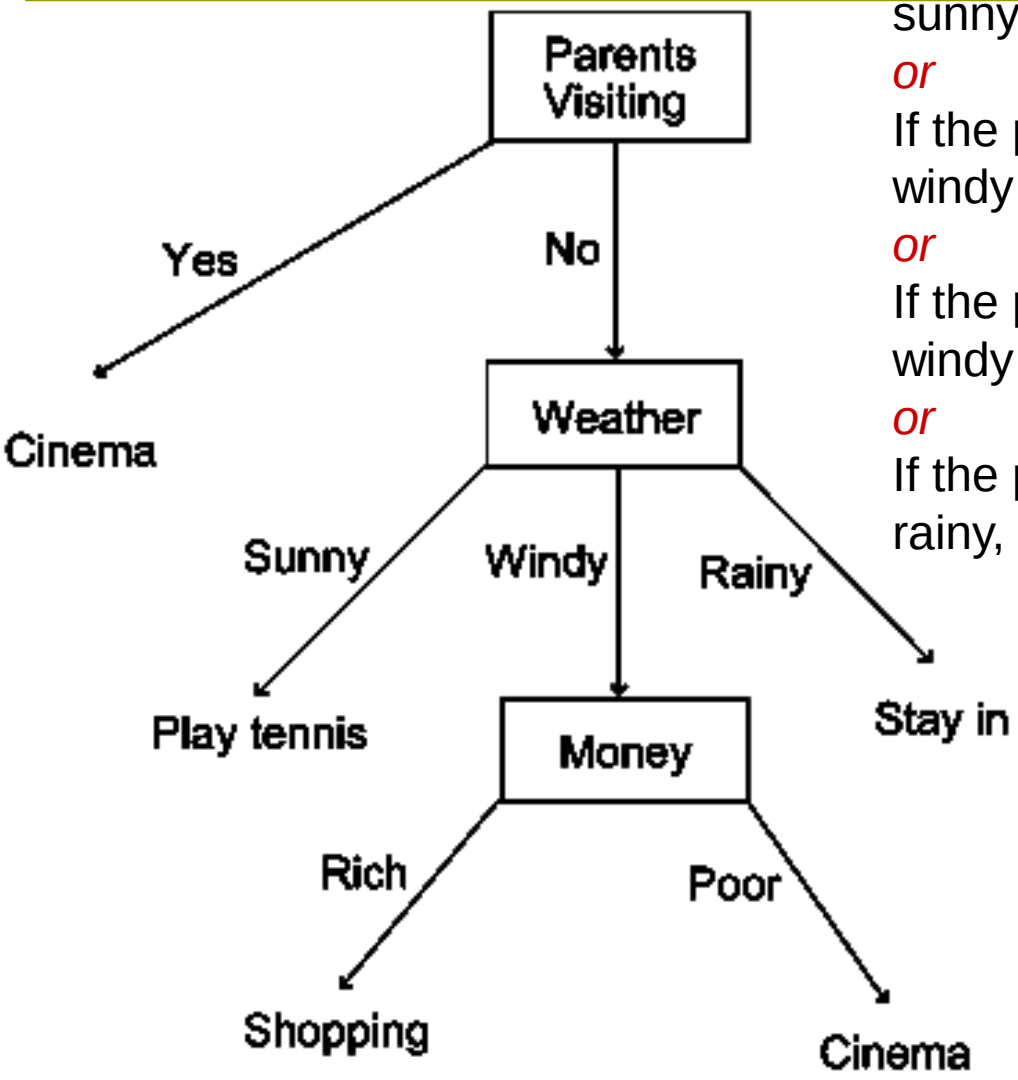
- Avantajları:
  - Karar ağacı oluşturmak zahmetsizdir
  - Küçük ağaçları yorumlamak kolaydır
  - Anlaşılabilir kurallar oluşturulabilinir
  - Sürekli ve ayrık nitelik değerleri için kullanılabilir

Karar Ağacı kullanarak sınıflandırma

- **Dezavantajları:**
  - Sürekli nitelik değerlerini tahmin etmekte çok başarılı değildir
  - Sınıf sayısı fazla ve öğrenme kümesi örnekleri sayısı az olduğunda model oluşturma çok başarılı değildir
  - Zaman ve yer karmaşıklığı öğrenme kümesi örnekleri sayısına, nitelik sayısına ve oluşan ağacın yapısına bağlıdır
  - Hem ağaç oluşturma karmaşıklığı hem de ağaç budama karmaşıklığı fazladır

Örnek: Karar Ağacı oluşturma

weekend	weather	parent	money	decision
w1	sunny	yes	rich	cinema
w2	sunny	no	rich	tennis
w3	windy	yes	rich	cinema
w4	rainy	yes	poor	cinema
w5	rainy	no	rich	stay in
w6	rainy	yes	poor	cinema
w7	windy	no	poor	cinema
w8	windy	no	rich	shopping
w9	windy	yes	rich	cinema
w10	sunny	no	rich	tennis



If the parents are visiting, then go to the cinema

or

If the parents are not visiting *and* it is sunny, then play tennis

or

If the parents are not visiting *and* it is windy *and* you're rich, then go shopping

or

If the parents are not visiting *and* it is windy *and* you're poor, then go to cinema

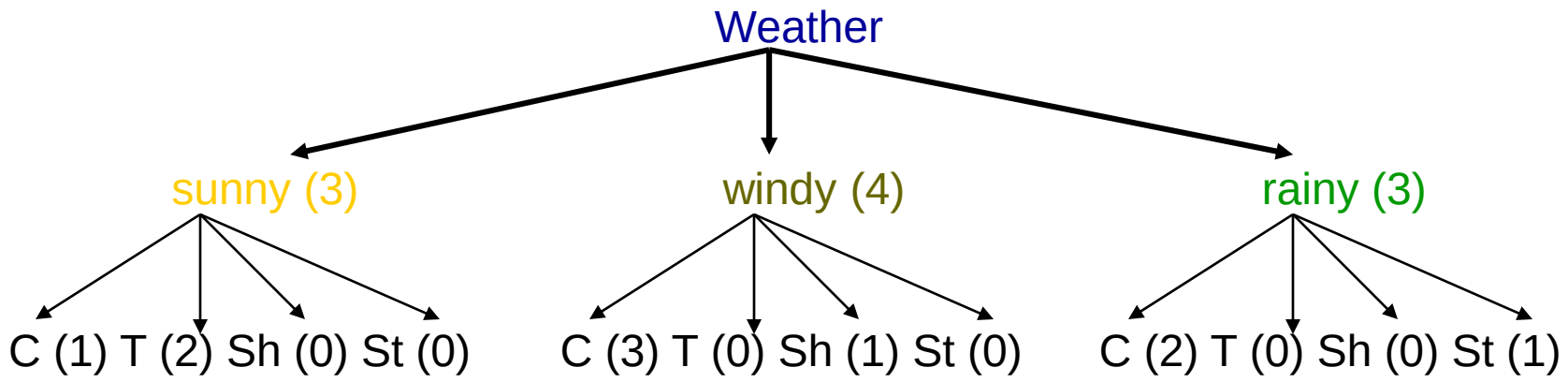
or

If the parents are not visiting *and* it is rainy, then stay in.

Önce sistemin Entropy si hesaplanır.

$$\text{Entropy}(S) = -p_{cinema}\log_2(p_{cinema}) - p_{tennis}\log_2(p_{tennis}) - p_{shopping}\log_2(p_{shopping}) - p_{stay\_in}\log_2(p_{stay\_in})$$
$$= -(6/10) * \log_2(6/10) - (2/10) * \log_2(2/10) - (1/10) * \log_2(1/10) - (1/10) * \log_2(1/10)$$
$$= -(6/10) * -0.737 - (2/10) * -2.322 - (1/10) * -3.322 - (1/10) * -3.322$$
$$= 0.4422 + 0.4644 + 0.3322 + 0.3322 = 1.571$$

Köke yerleştirilecek özellik belirlenir (weather, parents, money ?)

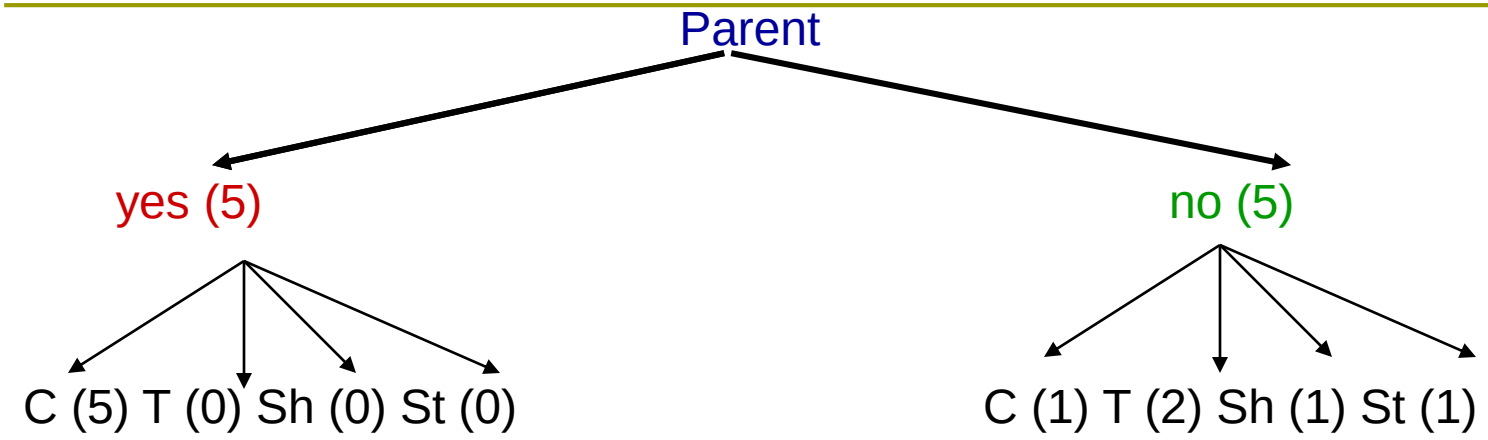


Gain(S,weather) = 1.571 – [(3/10)[-(1/3)log$_2$(1/3) – (2/3)log$_2$(2/3)]]

–[(4/10)[-(3/4)log$_2$(3/4) – (1/4)log$_2$(1/4)]]

–[(3/10)[-(1/3)log$_2$(1/3) – (2/3)log$_2$(2/3)]]

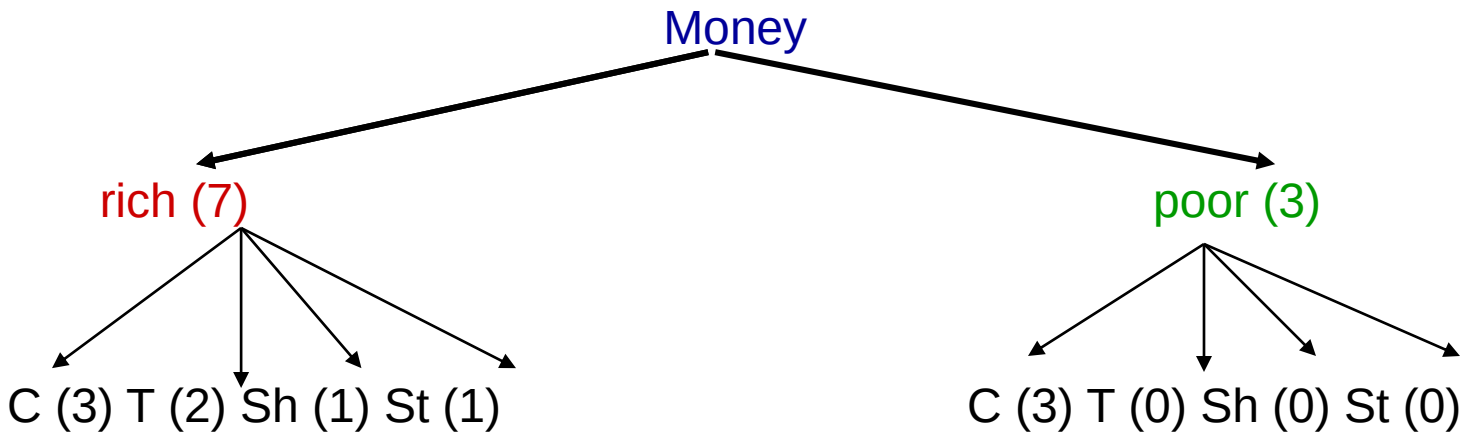
= 0.70



Gain(S,parent) = 1.571 – [(5/10)[-(1/5log$_2$(1/5) – (2/5)log$_2$(2/5) – (1/5)log$_2$(1/5) – (1/5)log$_2$(1/5)]]

–[(5/10)[-(5/5)log$_2$(5/5) – 0]]

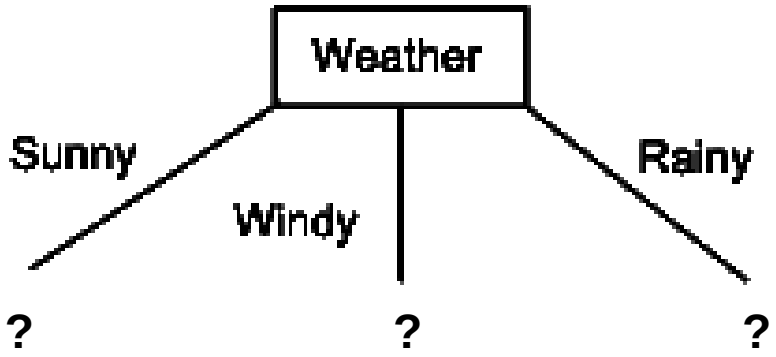
= 0.61



Gain(S,money) = 1.571 – [(7/10)[-(3/7log$_2$(3/7) – (2/7)log$_2$(2/7) – (1/7)log$_2$(1/7) – (1/7)log$_2$(1/7)]]

–[(3/10)[-(3/3)log$_2$(3/3) – 0]]

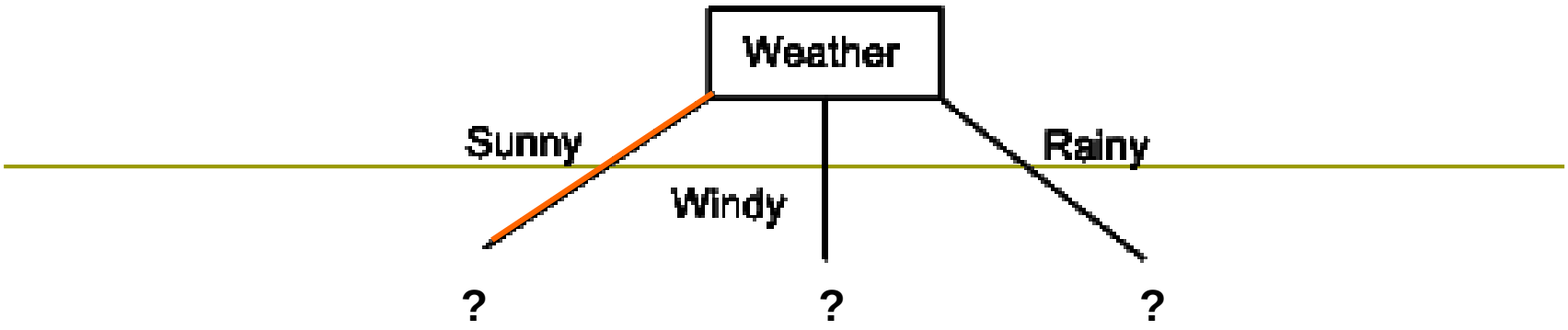
= 0.2816



S_{sunny} = {W1, W2, W10}. W1, W2 ve W10, sırasıyla Cinema, Tennis ve Tennis.

S_{windy} = {W3, W7, W8, W9}. W3, W7, W8 ve W9 sırasıyla Cinema, Cinema, Shopping ve Cinema.

S_{rainy} = {W4, W5, W6}. W4, W5 ve W6 sırasıyla Cinema, Stay in ve Cinema.



Sunny'nin altına gelecek özelliği belirlemek için Gain($S_{sunny, parents}$) ve Gain($S_{sunny, money}$) hesaplanmalıdır.

Önce, Entropy(S_{sunny}) değeri bilinmelidir.

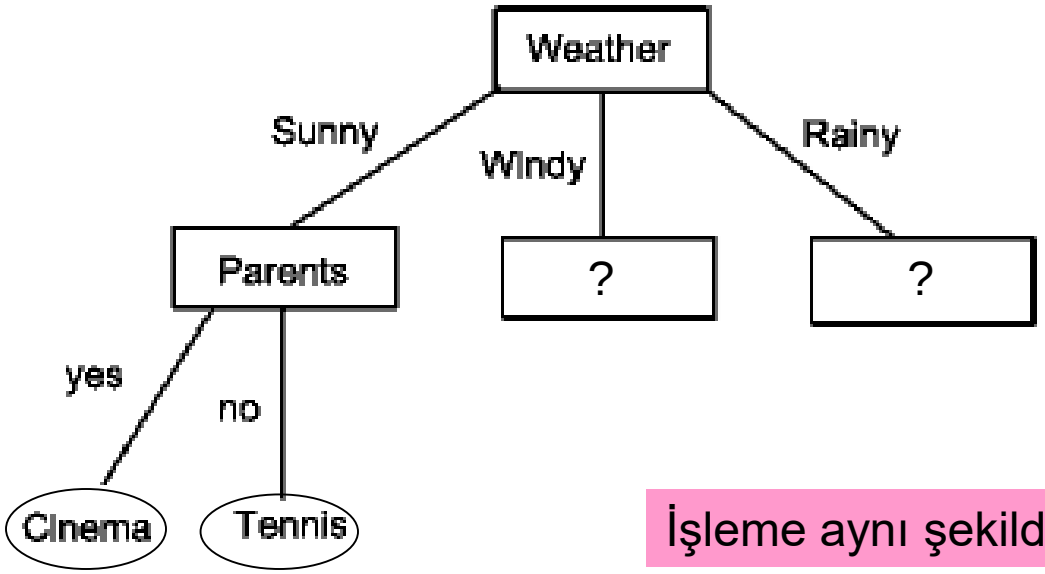
Weekend	Weather	Parents	Money	Decision
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W10	Sunny	No	Rich	Tennis

$E(sunny) = -(1/3)\log_2(1/3) - (2/3)\log_2(2/3) = 0.918$

Sunny'nin altına gelecek özelliği belirlemek için
Gain($S_{sunny, parents}$) ve Gain($S_{sunny, money}$) ?

$Gain(S_{sunny,parents}) = 0.918-(1/3)[-(1/1)\log_2(1/1)] - (2/3)[-(2/2)\log_2(2/2)]= 0.918$

$Gain(S_{sunny,money}) = 0.918-(3/3)[-(1/3)\log_2(1/3)-(2/3)\log_2(2/3)]= 0$



İşleme aynı şekilde devam edilir

C4.5 Algoritması

• Quinlan'ın ID3 algoritması yine aynı kişi tarafından genişletilerek C4.5 adını almıştır.

• ID3 algoritmasında bir özellik sayısal değerlere sahip ise sonuç alınamamaktadır. Bu yüzden C4.5 algoritması geliştirilmiştir.

• Sayısal değerler ile çalışılırken bir eşik değeri belirlenir. Bu eşik değeri bulunurken özelliğin değerleri sıralanır ve $[v_i, v_{i+1}]$ aralığının orta noktası alınır, ve bu değer t eşik değeri olarak belirlenir. Ve özellik değeri bu t eşik değerinden büyük veya küçük eşit olmak üzere ikiye ayrılır.

Bilinmeyen Nitelik Değeri

- C4.5 kayıp verilere sahip örneklerde bir düzeltme faktörü kullanır.
- H(X) ve H(X,T) değerleri hesaplanırken, bilinen niteliklere sahip örnekler alınmıştır.
- F faktörü kullanılarak kazanç ölçütü düzeltilir.

F= Veri tabanında değeri bilinen niteliğe sahip örneklerin sayısı / Veri tabanındaki tüm örneklerin sayısı

Kazanç (X) =F(H(T) – H(X,T))