

Karar Ağaçları

Giriş

Doç. Dr. İlhan AYDIN

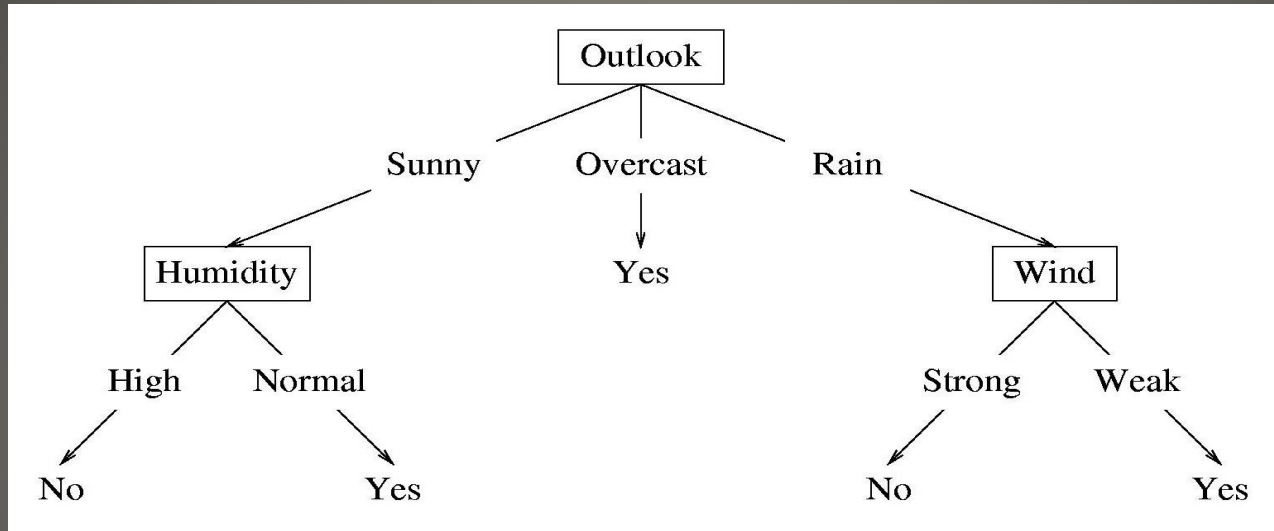
- Fonksiyon Yaklaşımı
 - Sorun Ayarı
 - Olası örnekler kümesi: X
 - Olası etiketler kümesi: Y
 - Bilinmeyen hedef fonksiyon: $f: X \rightarrow Y$
 - Fonksiyon hipotezi seti: $H = \{h \mid h: X \rightarrow Y\}$
 - Girdi
 - Bilinmeyen hedef fonksiyon f eğitim örnekleri
 - $\{x_i, y_i\}_{i=1}^n \{x_1, y_1, \dots, x_n, y_n\}$
 - f 'ye en iyi yaklaşan hipotez $h \in H$

- Örnek Veri Seti
 - Sütunlar özellikleri belirtir x_i
 - Satırlar etiketli örnekleri belirtir $\{x_i, y_i\}$
 - Sınıf etiketi, bir tenis oyununun oynanıp oynanmadığını göstermektedir.
 - $\{x_i, y_i\}$

Predictors				Response
Outlook	Temperature	Humidity	Wind	Class
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

Karar Ağaçları

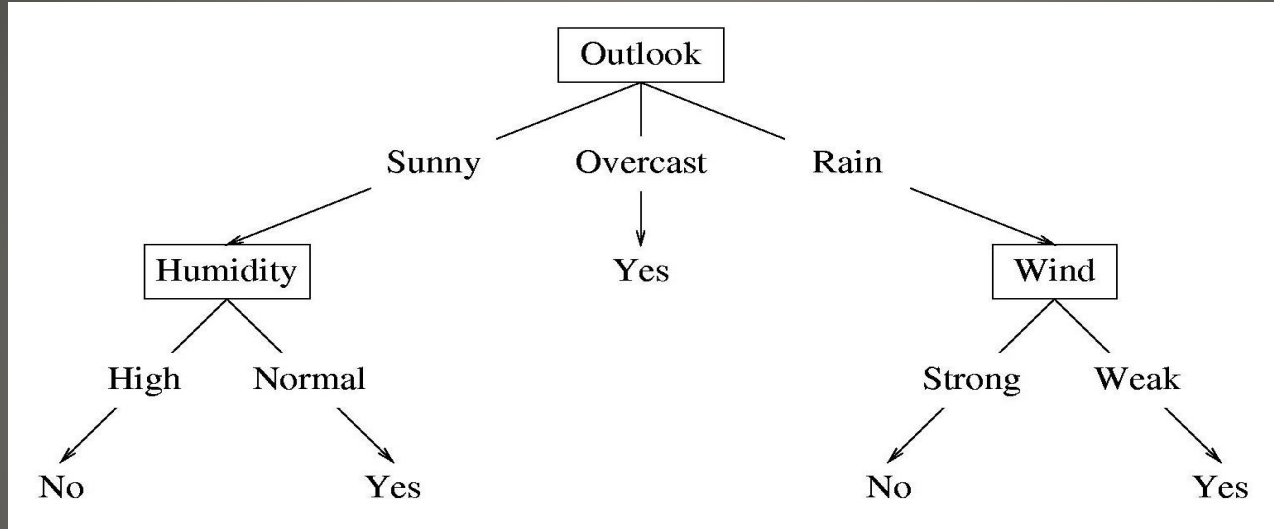
- Veriler için olası bir karar ağacı



- Her dahili düğüm: bir x_i niteliğini test edin
- Bir düğümdeki her dal: x_i için bir değer seç
- Her yaprak düğümü: Y 'yi tahmin et ($p(Y | x \in leaf)$)

Karar Ağaçları

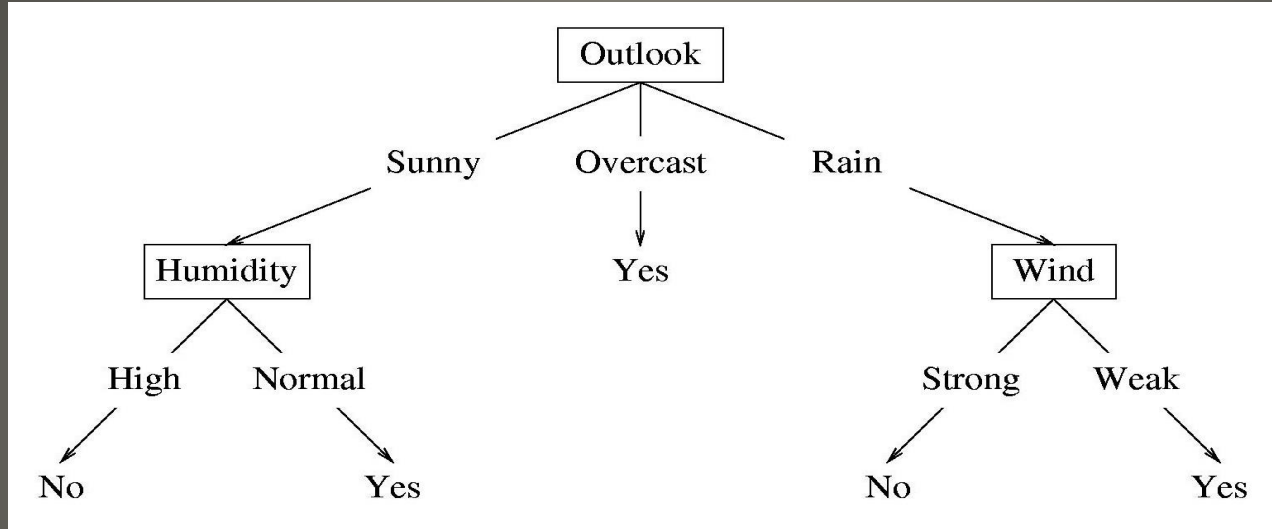
- Veriler için olası bir karar ağacı



- Hangi tahminde bulunurduk?
- <outlook=sunny, temperature=hot, humidity=high, wind=weak> ?

Karar Ağaçları

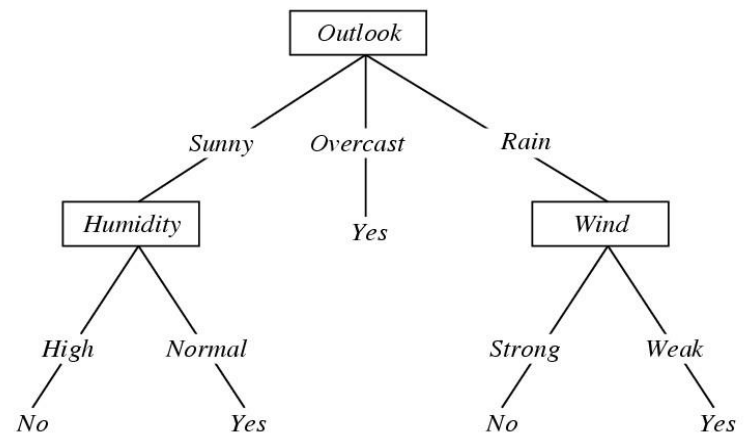
- Veriler için olası bir karar ağacı



- Özellikler sürekli ise, dahili düğümler bir özelliğin değerini bir eşığe karşı test edebilir.

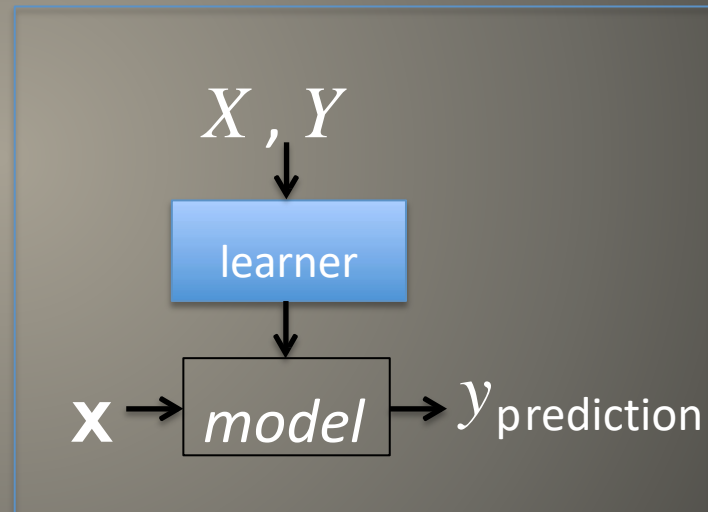
Karar Ağaçları

- Veriler için olası bir karar ağacı
 - Sorun Ayarı
 - Olası örnekler kümesi x_i
 - X 'teki her x örneği bir özellik vektörüdür
 - Ör: $\langle \text{Humidity}=\text{low}, \text{Wind}=\text{weak}, \text{Outlook}=\text{rain}, \text{Temp}=\text{hot} \rangle$
 - Bilinmeyen hedef fonksiyon $f: X \rightarrow Y$
 - Y ayrık değerlidir
 - Fonksiyon hipotezleri seti $H = \{h \mid h: X \rightarrow Y\}$
 - Her hipotez h bir karar ağacıdır
 - Ağaçlar x 'i yaprağa göre sıralar, bu da y 'yi atar



Karar Ağaçları

- (Toplu) Makine Öğreniminin Aşamaları
 - Verilen: etiketli eğitim verileri $X, Y = \{x_i, y_i\}_{i=1}^n$
 - Her $x_i \sim D(X)$ 'i $y_i = f_{target}(x_i)$ ile varsayar
- Model Eğitimi:
 - $model \leftarrow classifier.train(X, Y)$
- Modeli yeni verilere uygula
 - Verilen yeni etiketlenmemiş örnek Her $x \sim D(X)$
 - $y_{prediction} \leftarrow model.predict(x)$



Karar Ağaçları

- Örnek Uygulama: Sezaryen Riskini Öngören Bir Ağaç:

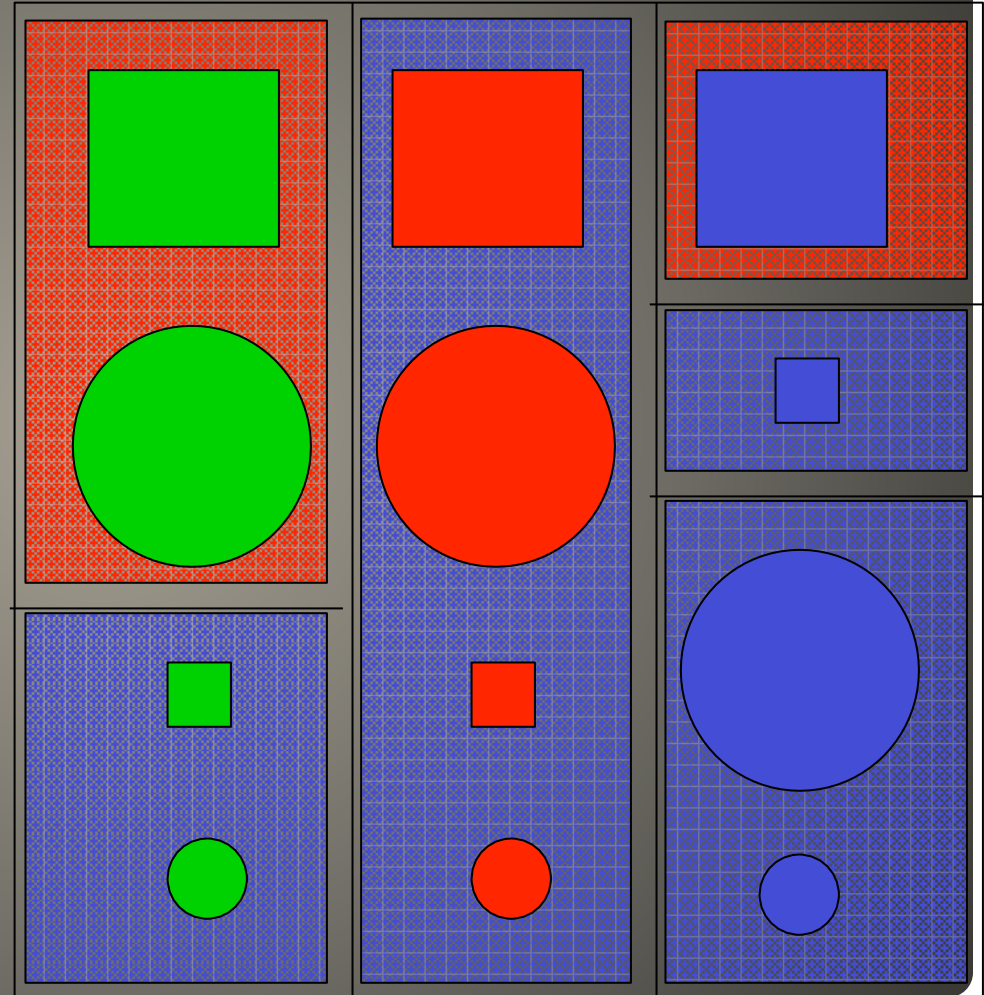
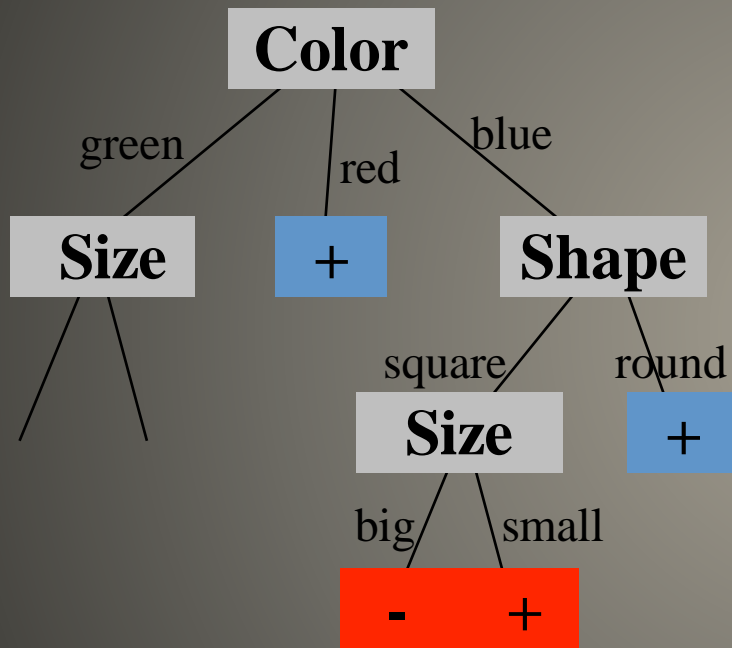
Learned from medical records of 1000 women

Negative examples are C-sections

```
[833+,167-] .83+ .17-  
Fetal_Presentation = 1: [822+,116-] .88+ .12-  
| Previous_Csection = 0: [767+,81-] .90+ .10-  
| | Primiparous = 0: [399+,13-] .97+ .03-  
| | Primiparous = 1: [368+,68-] .84+ .16-  
| | | Fetal_Distress = 0: [334+,47-] .88+ .12-  
| | | | Birth_Weight < 3349: [201+,10.6-] .95+ .05-  
| | | | Birth_Weight >= 3349: [133+,36.4-] .78+ .22-  
| | | Fetal_Distress = 1: [34+,21-] .62+ .38-  
| Previous_Csection = 1: [55+,35-] .61+ .39-  
Fetal_Presentation = 2: [3+,29-] .11+ .89-  
Fetal_Presentation = 3: [8+,22-] .27+ .73-
```

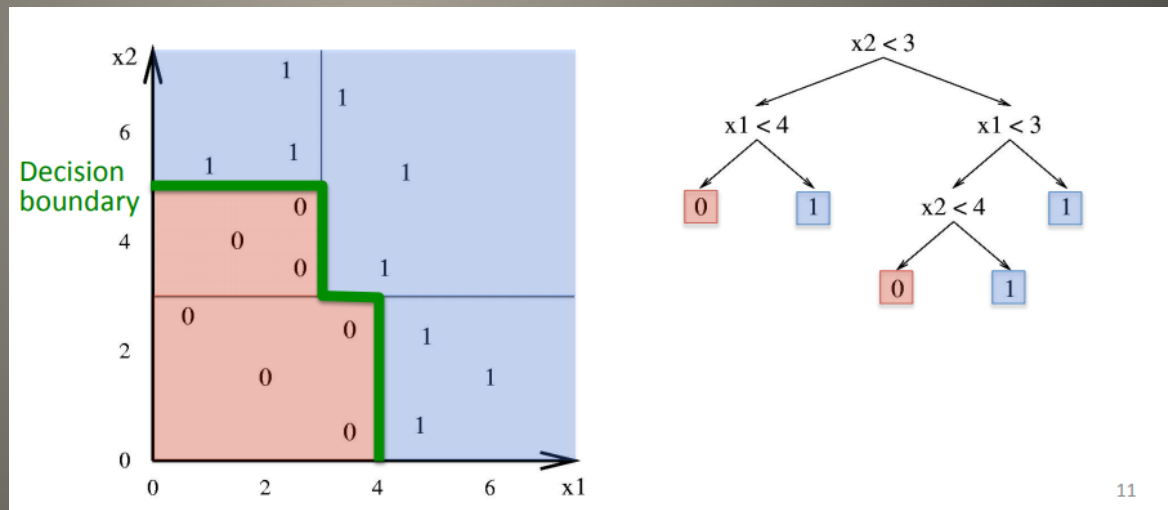
Karar Ağaçları

- Karar Ağacı Kaynaklı Bölme



Karar Ağaçları

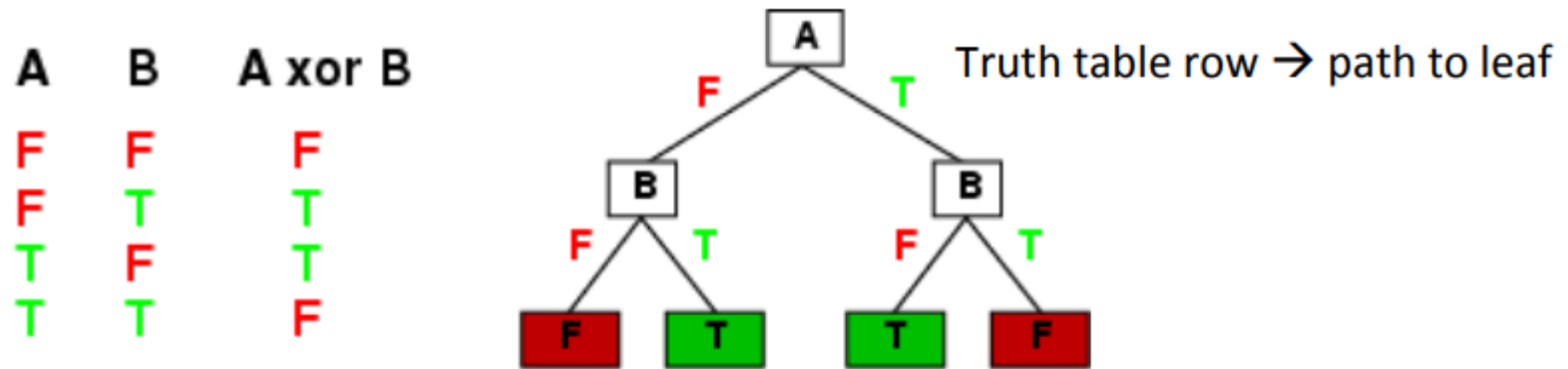
- Karar Ağacı – Karar Sınırı:
 - Karar ağaçları, özellik uzayını eksen paralel (hiper) dikdörtgenlere böler
 - Her dikdörtgen bölge bir etiketle etiketlenir
 - veya etiketler üzerinde bir olasılık dağılımı



Karar Ağaçları

- **Dışavurumculuk**

- Karar ağaçları, girdi özneliliklerinin herhangi bir Boolean fonksiyonunu temsil edebilir.



- En kötü durumda, ağaç katlanarak çok sayıda düğüm gerektirecektir.

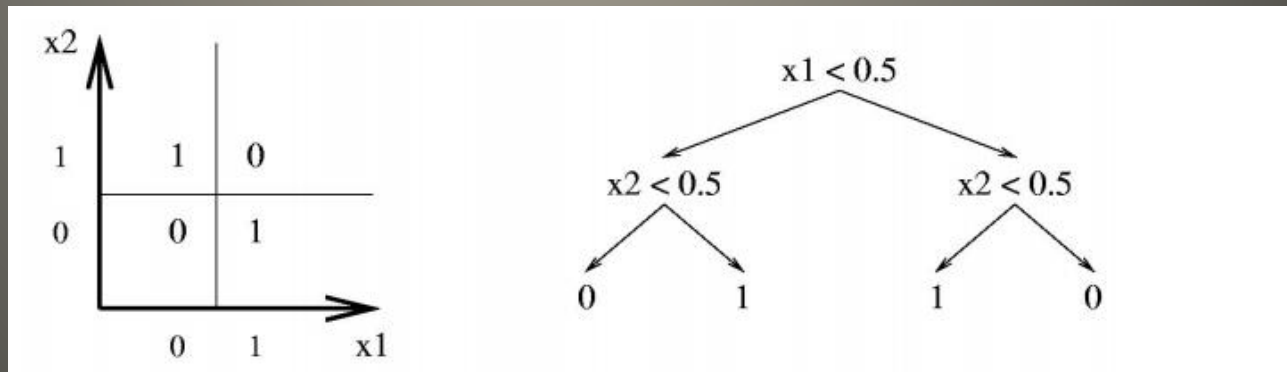
Karar Ağaçları

- **Dışavurumculuk**

- Karar ağaçlarının değişken boyutlu bir hipotez alanı vardır.

- #düğümler (veya derinlik) arttıkça, hipotez uzay büyür
 - Derinlik 1 ("karar güdüsü"): herhangi bir boolean değerini temsil edebilir
 - Derinlik 2: iki özelliğin herhangi bir boolean fn'si; bazıları üç özellik içerir (e.g., $(x_1 \wedge x_2) \vee (\neg x_1 \wedge \neg x_3)$)

Örnek:



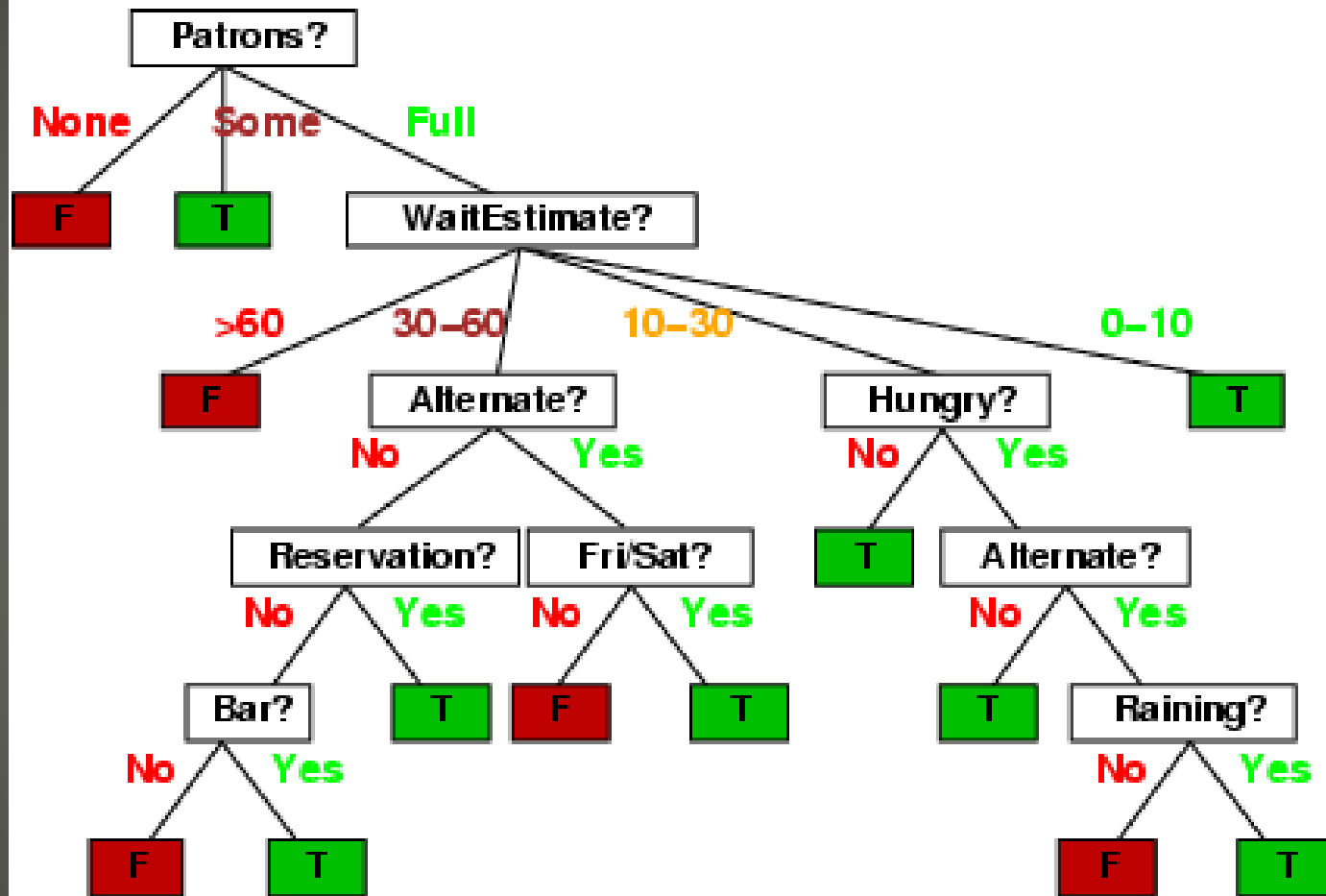
Karar Ağaçları

- Örnek
- Bir restoranda bir masa için bekleyip beklememe konusunda bir patronun kararını modelleyin

Example	Attributes										Target
	<i>Alt</i>	<i>Bar</i>	<i>Fri</i>	<i>Hun</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	<i>Wait</i>
X_1	T	F	F	T	Some	\$\$\$	F	T	French	0–10	T
X_2	T	F	F	T	Full	\$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	\$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	\$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	\$\$\$	F	T	French	>60	F
X_6	F	T	F	T	Some	\$\$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	\$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	\$\$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	\$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	\$\$\$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	\$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	\$	F	F	Burger	30–60	T

7,000 olası vaka

Karar Ağaçları



Karar Ağaçları

- Tercih yanlılığı: Ockham'ın Usturası
 - Ockhamlı William tarafından belirtilen ilke (1285-1347)
 - “ non sunt multiplicanda entia praeter necessitatem ”
 - varlıklar gerekliliğin ötesinde çoğaltılmamalıdır
 - AKA Occam'ın Usturası, Ekonomi Yasası veya Parsimony Yasası
- Fikir: En basit tutarlı açıklama en iyisidir
- Bu nedenle, tüm eğitim örneklerini doğru bir şekilde sınıflandıran en küçük karar ağacı en iyisidir.
 - Kanıtlanabilir en küçük karar ağacını bulmak NP-zordur
 - ...Öyleyse, eğitim örnekleriyle tutarlı en küçük ağacı oluşturmak yerine, oldukça küçük bir ağaç oluşturun.

Karar Ağaçları

- Yukarıdan Aşağıya Temel Algoritma Karar Ağaçlarının İndüksiyonu
- düğüm = karar ağacının kökü
- Ana program
 - $A \leftarrow$ sonraki düğüm için “en iyi” karar niteliği.
 - Düğüm için karar özneliği olarak A atayın.
 - A'nın her değeri için, düğümün yeni bir soyundan oluşturun
 - Eğitim örneklerini yaprak düğümlerine göre sıralama
 - Eğitim örnekleri mükemmel bir şekilde sınıflandırılmışsa, durun. Aksi takdirde, yeni yaprak düğümleri üzerinde özyineleme.
- Hangi özelliğin en iyi olduğunu nasıl seçeriz?

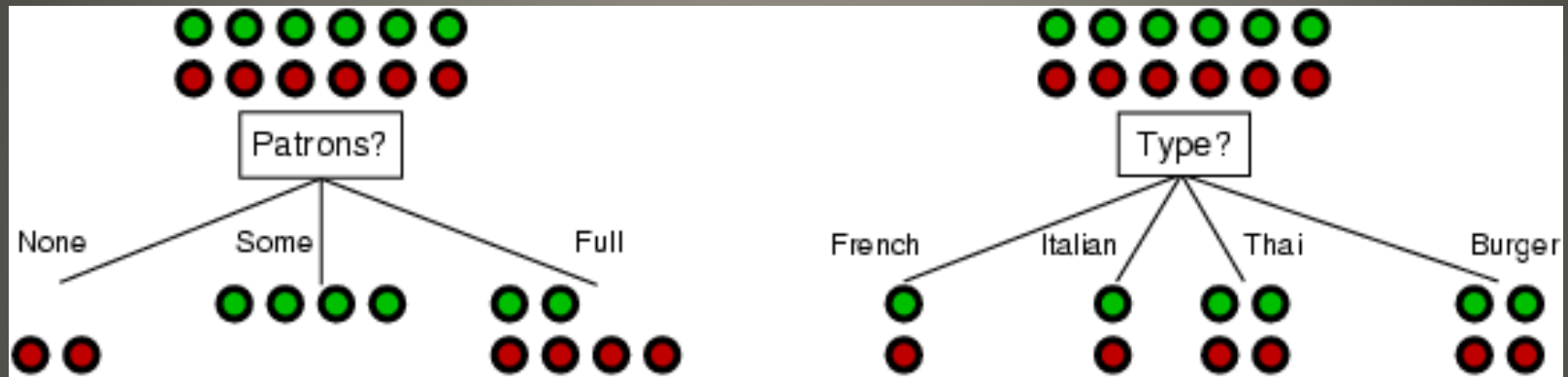
Karar Ağaçları

- En İyi Niteliği Seçmek
- Ana problem: belirli bir örnek kümesini bölmek için hangi Niteliğin seçilmesi
- Bazı olasılıklar:
 - Random: Herhangi bir niteliği rastgele seçin
 - Least-Values: Mümkün olan en az sayıda değeri olan özniteliği seçin.
 - Most-Values: Mümkün olan en fazla sayıda değere sahip özelliği seçin
 - Max-Gain: Beklenen en büyük bilgi kazancına sahip özniteliği seçin
 - alt ağacının alt ağacının beklenen en küçük boyutuyla sonuçlanan öznitelik
- ID3 algoritması, en iyi özniteliği seçmek için Maks-Kazanç yöntemini kullanır.

Karar Ağaçları

- Öznitelik Seçme

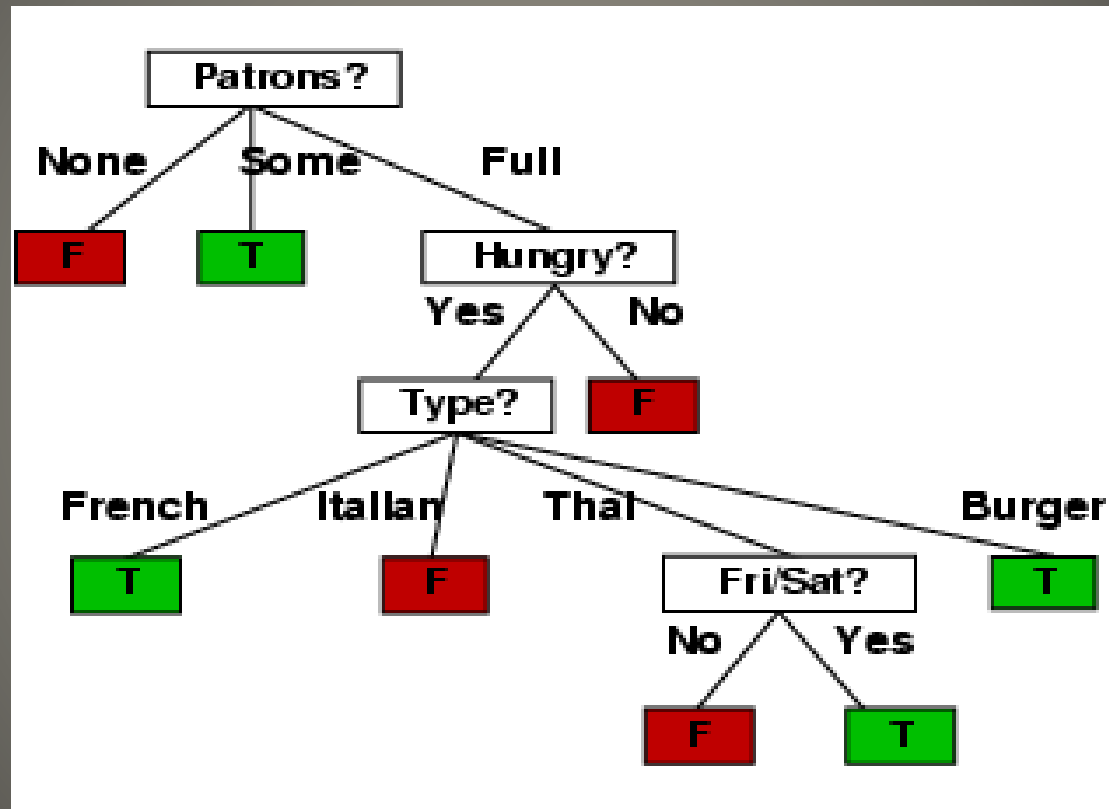
- Fikir: iyi bir nitelik, örnekleri (ideal olarak) "tümü olumlu" veya "tümü olumsuz" olan alt kümelere böler



- Hangi bölüm daha bilgilendirici: Kullanıcılar? veya Tür?

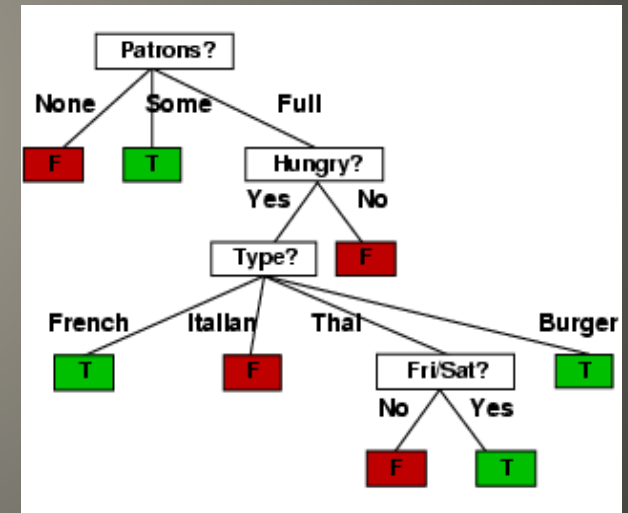
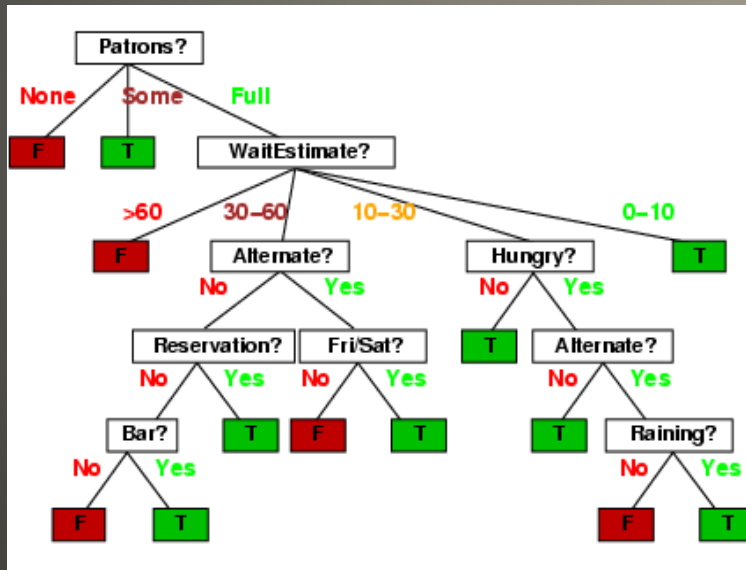
Karar Ağaçları

- ID3 Kaynaklı Karar Ağacı



Karar Ağaçları

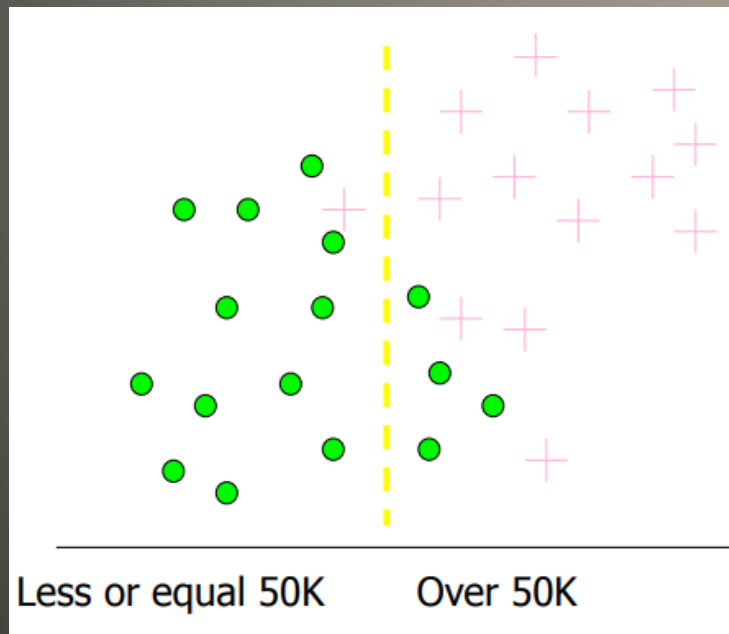
- İki Karar Ağacını Karşılaştırın



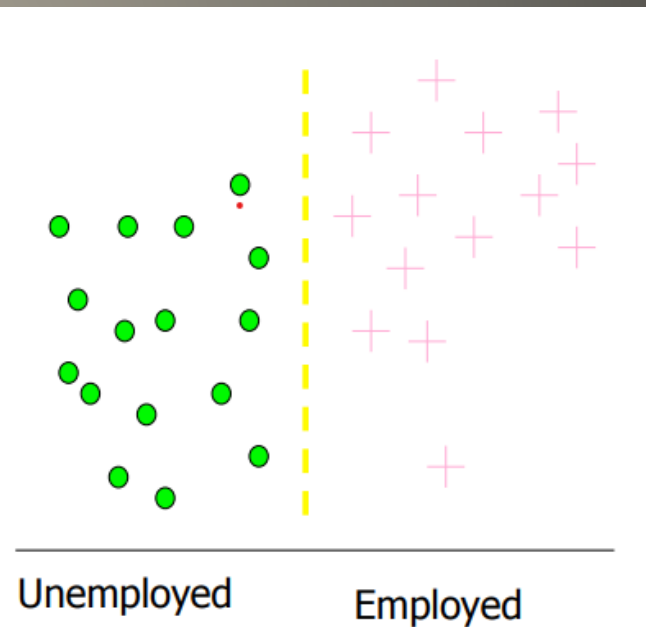
Karar Ağaçları

- Bilgi Kazanımı
 - Hangi test daha bilgilendirici?

Bakiyenin 50K'yı aşıp
aşmadığına göre bölün

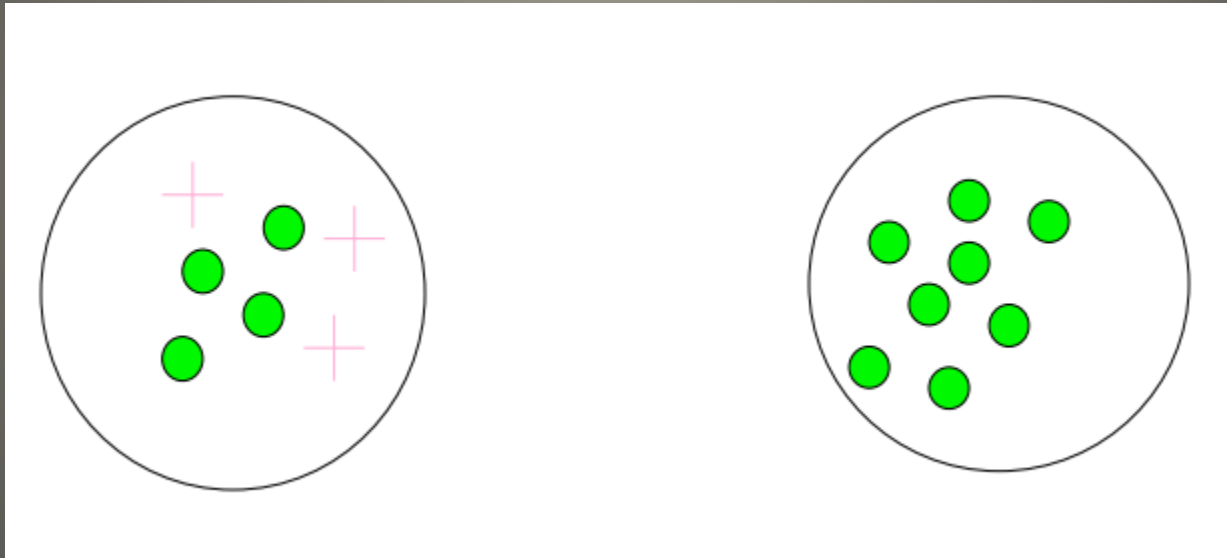


Başvuranın istihdam edilip
edilmediğine göre bölün



Karar Ağaçları

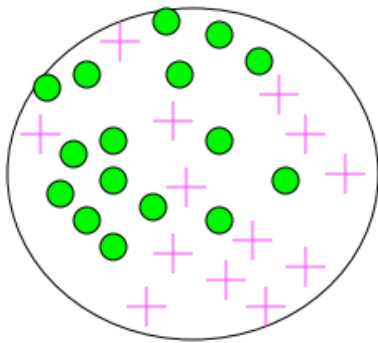
- Bilgi Kazanımı
 - Safsızlık/Entropi (gayri resmi)
 - Bir grup örnekte kirlilik seviyesini ölçer



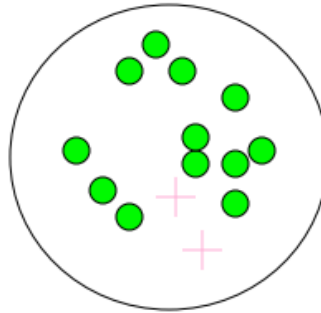
Karar Ağaçları

- Kirlilik (impure)

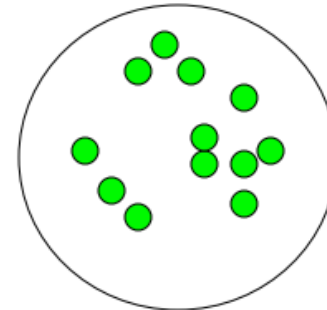
Very impure group



Less impure



**Minimum
impurity**



Karar Ağaçları

- Entropi: safsızlığı ölçmenin yaygın bir yoludur.

- Rastgele bir X değişkeninin entropisi $H(X)$

$$H(X) = - \sum_{i=1}^n P(X = i) \log_2 P(X = i)$$

of possible
values for X

- $H(X)$, rastgele çizilmiş bir X değerini kodlamak için gereken beklenen bit sayısıdır (en verimli kod altında)

Karar Ağaçları

- Entropi: safsızlığı ölçmenin yaygın bir yoludur

- Rastgele bir X değişkeninin entropisi $H(X)$

$$H(X) = - \sum_{i=1}^n P(X = i) \log_2 P(X = i)$$

of possible values for X

- $H(X)$, rastgele çizilmiş bir X değerini kodlamak için gereken beklenen bit sayısıdır (en verimli kod altında)
- Neden? Bilgi Teorisi
 - En verimli kod, $X=i$ mesajını kodlamak için $-\log_2 P(X=i)$ bitlerini atar
 - Bu nedenle, bir rastgele X 'i kodlamak için beklenen bit sayısı:

$$\sum_{i=1}^n P(X = i) (-\log_2 P(X = i))$$

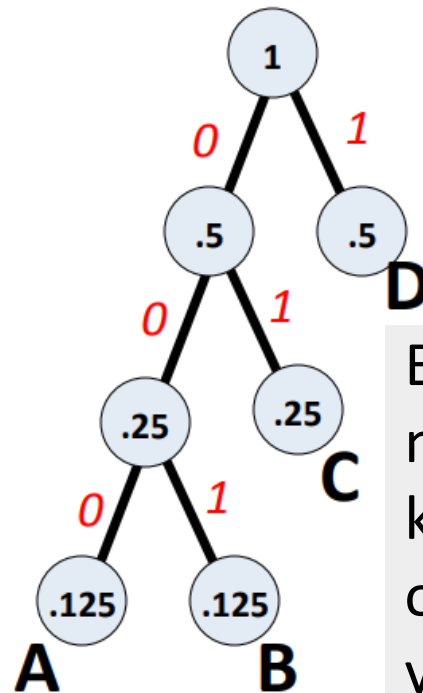
Karar Ağaçları

- Huffman Kodu Örneği
 - 1952'de MIT öğrencisi David Huffman, bir ev ödevi yaparken, tüm sembollerin olasılıklarının $1/2$ 'nin integral kuvvetleri olduğu durumda ideal olan zarif bir kodlama şeması tasarladı.
 - Bir Huffman kodu aşağıdaki şekilde oluşturulabilir:
 - Tüm sembolleri gerçekleşme olasılığına göre sıralayın
 - Yeni bir bileşik sembol oluşturmak için en düşük olasılığa sahip iki sembolü art arda birleştirin; sonunda, her düğümün altındaki tüm düğümlerin olasılığı olduğu bir ikili ağaç oluşturacağız.
 - Her düğümde yönü fark ederek, her yaprağa giden bir yol izleyin

Karar Ağaçları

- Huffman Kodu Örneği

M	P
A	.125
B	.125
C	.25
D	.5



M	code	length	prob	
A	000	3	0.125	0.375
B	001	3	0.125	0.375
C	01	2	0.250	0.500
D	1	1	0.500	0.500

average message length

1.750

Bu olasılık dağılımına sahip birçok mesaja (A,B,C veya D) bu kodu kullanırsak, zaman zaman içinde ortalama bit/mesaj 1,75'e yaklaşmalıdır.

Karar Ağaçları

- İki Sınıflı Durumlar

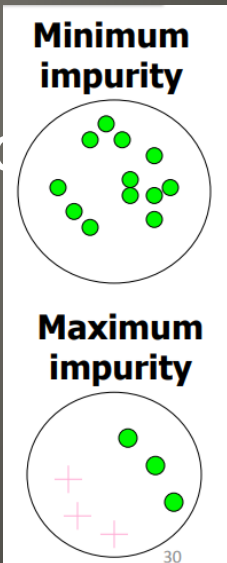
$$\text{Entropy } H(x) = - \sum_{i=1}^n P(x = i) \log_2 P(x = i)$$

- Hepsinin bulunduğu grubun entropisi nedir?örnekler aynı sınıfa mı ait?

- entropy = $-1 \log_2 1 = 0$
- öğrenme için iyi bir eğitim seti değil

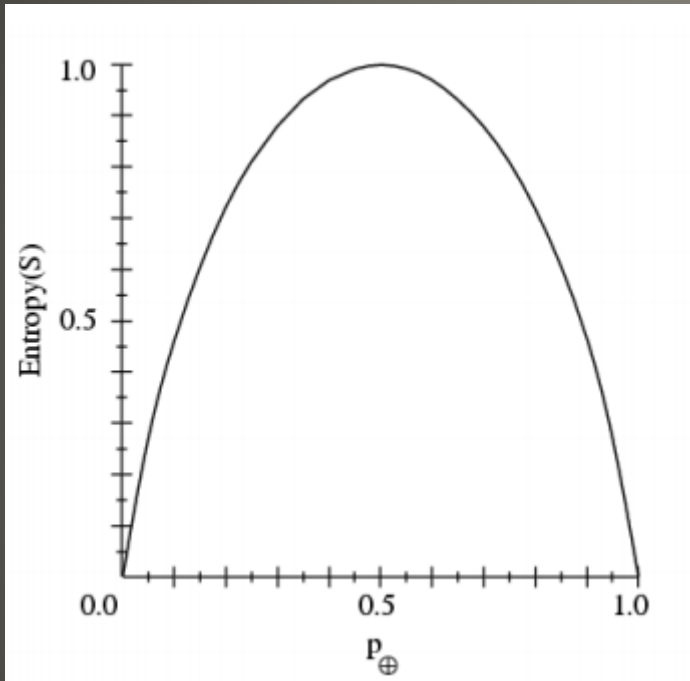
- Her iki sınıfta da %50'si olan bir grubun entropisi nedir?

- entropy = $-0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$
- öğrenme için iyi bir eğitim seti



Karar Ağaçları

• Örnek Entropi



- S örnek bir eğitim örneğidir
- P_+ , S'deki pozitif örneklerin oranıdır
- P_- , S'deki negatif örneklerin oranıdır
- Entropi, S'nin saflığını ölçer

$$H(S) \equiv -p_+ \log_2 p_+ - p_- \log_2 p_-$$

Karar Ağaçları

- **Bilgi Kazanımı (Information Gain)**
 - Belirli bir eğitim öznitelik vektörleri kümesindeki hangi öz niteliğin öğrenilecek sınıflar arasında ayırım yapmak için en yararlı olduğunu belirlemek istiyoruz.
 - Bilgi kazanımı, özellik vektörlerinin belirli bir öz niteliğinin ne kadar önemli olduğunu bize söyler.
 - Bir karar ağacının düğümlerindeki öz niteliklerin sırasına karar vermek için kullanacağız.

Karar Ağaçları

- Entropiden Bilgi Kazancına
 - Rastgele bir X değişkeninin entropisi $H(X)$

$$H(X) = - \sum_{i=1}^n P(X = i) \log_2 P(X = i)$$

Karar Ağaçları

- Entropiden Bilgi Kazancına

- Rastgele bir X değişkeninin entropisi $H(X)$

$$H(X) = - \sum_{i=1}^n P(X = i) \log_2 P(X = i)$$

- Spesifik koşullu entropi $Y=v$ verilen X 'in $H(X/Y=v)$:

$$H(X|Y = v) = - \sum_{i=1}^n P(X = i|Y = v) \log_2 P(X = i|Y = v)$$

Karar Ağaçları

- Entropiden Bilgi Kazancına

- Rastgele bir X değişkeninin entropisi $H(X)$

$$H(X) = - \sum_{i=1}^n P(X = i) \log_2 P(X = i)$$

- Spesifik koşullu entropi $Y=v$ verilen X 'in $H(X|Y=v)$:

$$H(X|Y = v) = - \sum_{i=1}^n P(X = i|Y = v) \log_2 P(X = i|Y = v)$$

- Y verilen X 'in koşullu entropisi $H(X|Y)$:

$$H(X|Y) = \sum_{v \in \text{values}(Y)} P(Y = v) H(X|Y = v)$$

Karar Ağaçları

- Entropiden Bilgi Kazancına

- Rastgele bir X değişkeninin entropisi $H(X)$

$$H(X) = - \sum_{i=1}^n P(X = i) \log_2 P(X = i)$$

- Spesifik koşullu entropi $Y=v$ verilen X 'in $H(X|Y=v)$:

$$H(X|Y = v) = - \sum_{i=1}^n P(X = i|Y = v) \log_2 P(X = i|Y = v)$$

- Y verilen X 'in koşullu entropisi $H(X|Y)$:

$$H(X|Y) = \sum_{v \in \text{values}(Y)} P(Y = v) H(X|Y = v)$$

- X ve Y 'nin karşılıklı bilgileri (diğer adıyla Bilgi Kazanımı):

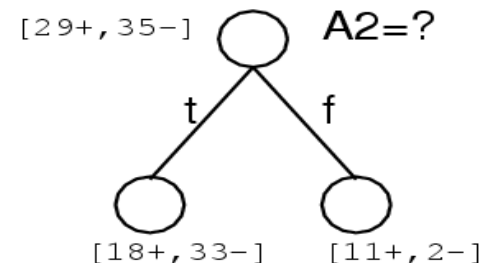
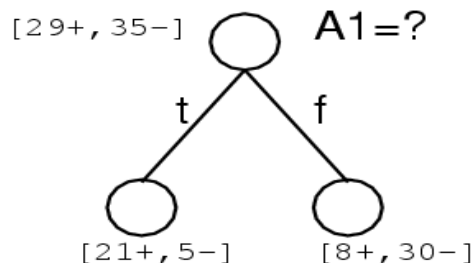
$$I(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

Karar Ağaçları

- **Bilgi Kazanımı**

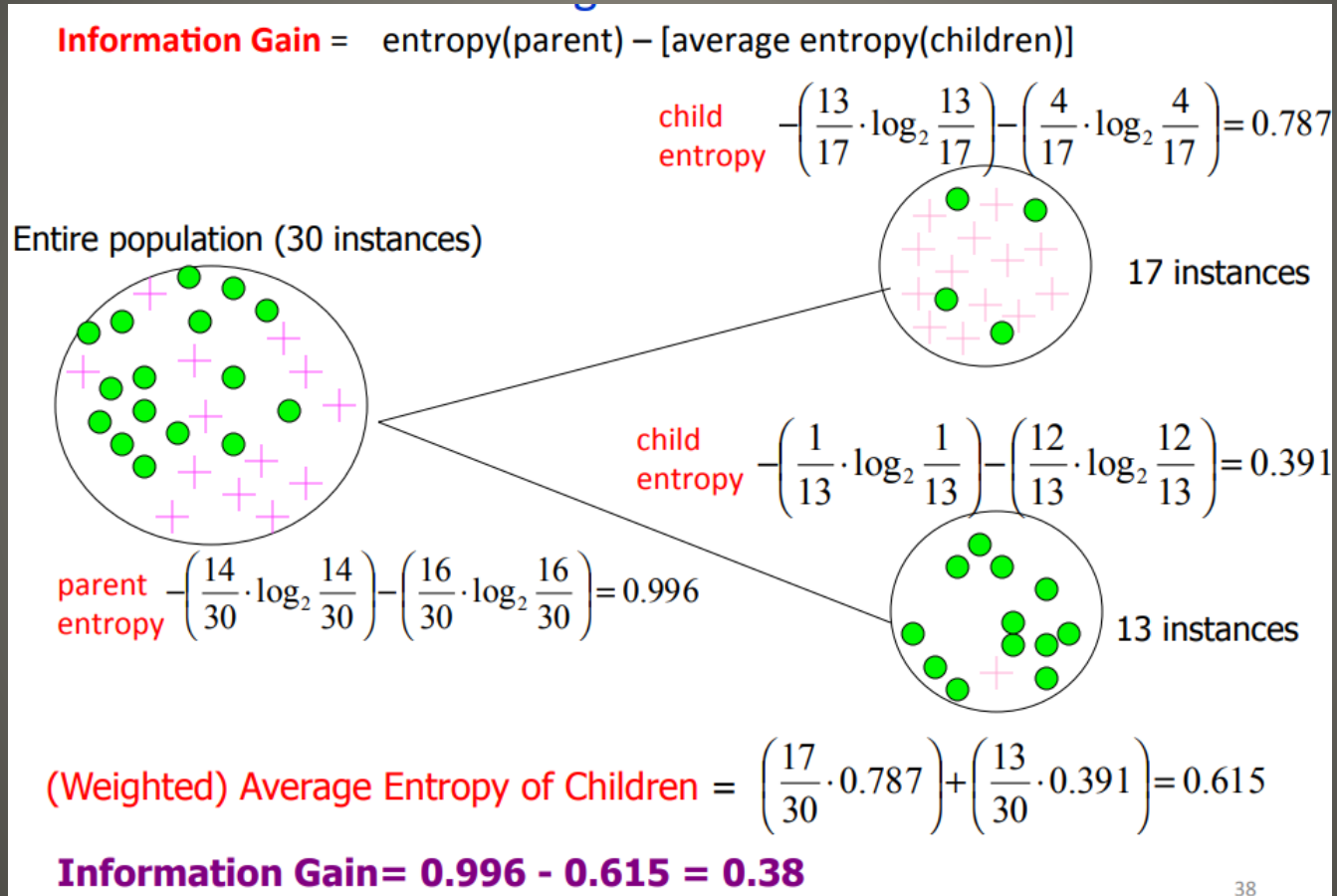
- Bilgi Kazancı, girdi özelliği A ile hedef değişken Y arasındaki karşılıklı bilgidir.
- Bilgi Kazancı, A değişkenine göre sıralama nedeniyle veri örneği S için hedef değişken Y'nin entropisinde beklenen azalmadır.

$$Gain(S, A) = I_S(A, Y) = H_S(Y) - H_S(Y|A)$$



Karar Ağaçları

- Bilgi Kazancının Hesaplanması



- Entropi Tabanlı Otomatik Karar Ağacı Yapımı

Eğitim Seti X

$x_1 = (f_{11}, f_{12}, \dots, f_{1m})$

$x_2 = (f_{21}, f_{22}, \dots, f_{2m})$

.

.

$x_n = (f_{n1}, f_{n2}, \dots, f_{nm})$

Düğüm 1 Hangi özellik kullanılmalıdır?

- Quinlan, ID3 sisteminde bilgi kazancını ve daha sonra her ikisi de entropiye dayalı kazanç oranını önerdi.

Karar Ağaçları

- Eğitim Örnekleri

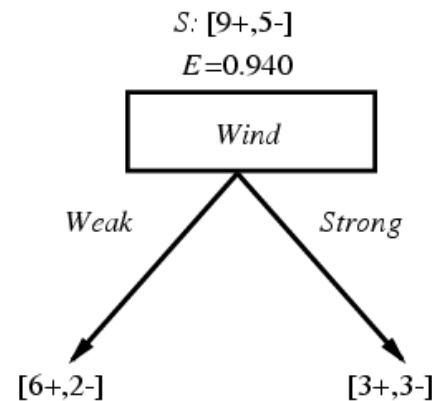
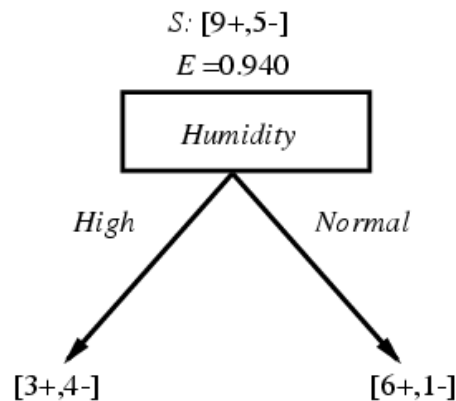
Day	Outlook	Temperature	Humidity	Wind	PlayTenn
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Karar Ağaçları

- Sonraki Özelliği Seçme

Selecting the Next Attribute

Which attribute is the best classifier?

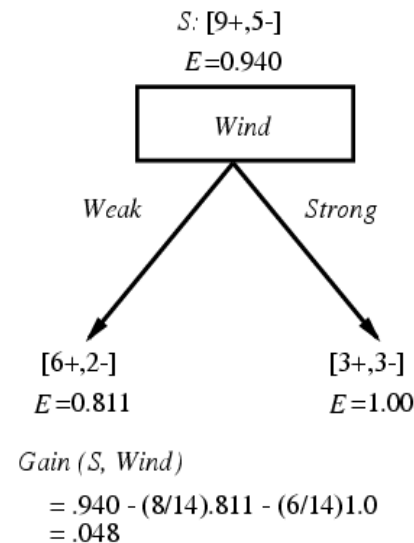
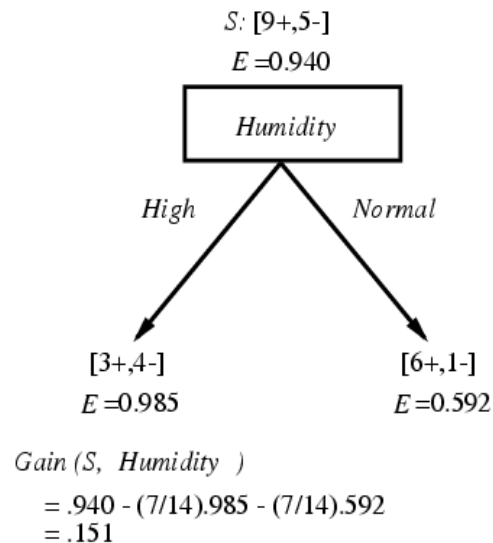


Karar Ağaçları

- Sonraki Özelliği Seçme

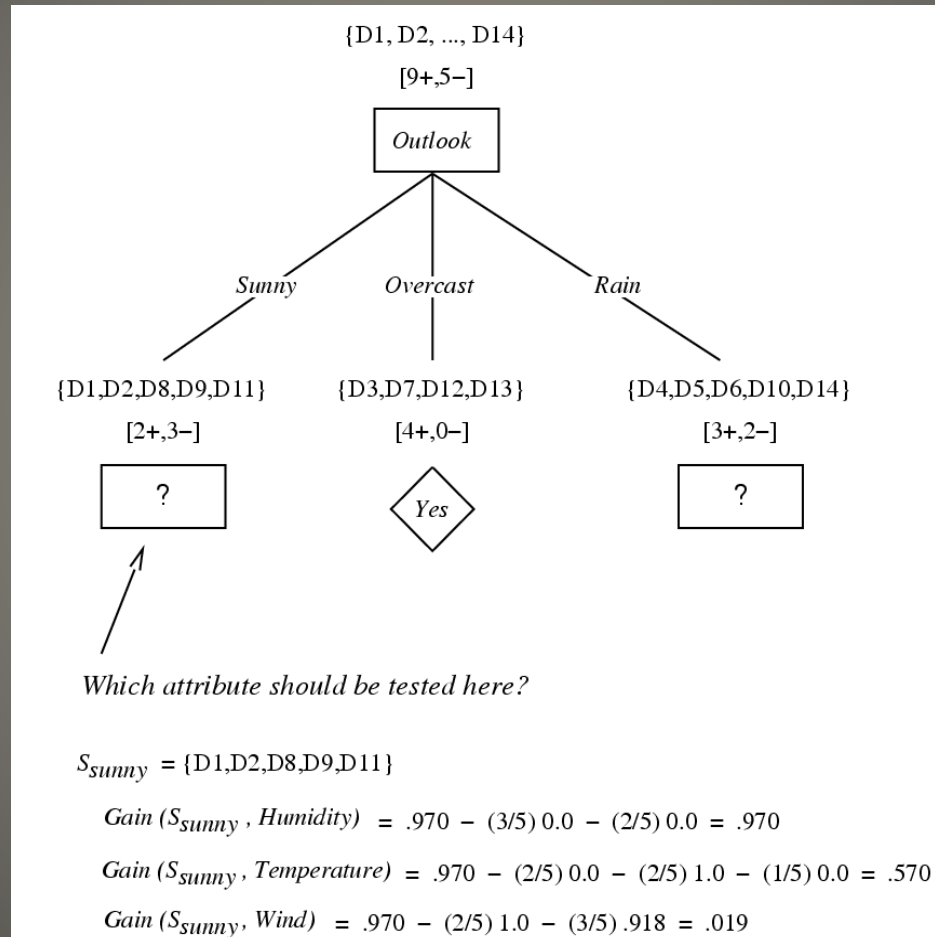
Selecting the Next Attribute

Which attribute is the best classifier?



Karar Ağaçları

- Sonraki Özelliği Seçme

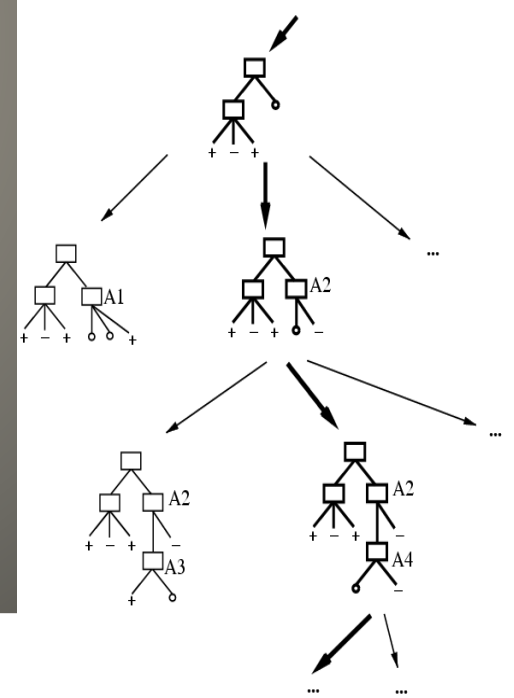


Karar Ağaçları

- Karar Ağacı Uygulaması
 - <http://webdocs.cs.ualberta.ca/~aixplore/learning/DecisionTrees/Applet/DecisionTreeApplet.html>

Karar Ağaçları

- Hangi Ağaç Çıktı Olmalı?
 - ID3, karar ağaçları alanında buluşsal arama yapar
 - Kabul edilebilir en küçük ağaçta durur. Neden?
 - Occam'ın usturası: verilere uyan en basit hipotezi tercih edin



Karar Ağaçları

- ID3 algoritması, kategorik olmayan bir dizi C_1, C_2, \dots, C_n , sınıf özneliği C ve bir eğitim seti T kaydı verilen bir karar ağacı oluşturur.

```
function ID3(R:input attributes, C:class attribute,  
S:training set) returns decision tree;  
    If S is empty, return single node with value Failure;  
    If every example in S has same value for C, return  
    single node with that value;  
    If R is empty, then return a single node with most  
    frequent of the values of C found in examples S;  
    # causes errors -- improperly classified record  
    Let D be attribute with largest Gain(D,S) among R;  
    Let {dj | j=1,2, ..., m} be values of attribute D;  
    Let {Sj | j=1,2, ..., m} be subsets of S consisting of  
    records with value dj for attribute D;  
    Return tree with root labeled D and arcs labeled  
    d1..dm going to the trees ID3(R-{D},C,S1) . . .  
    ID3(R-{D},C,Sm);
```

Karar Ağaçları

- Ne kadar iyi çalışıyor?
- Birçok vaka çalışması, karar ağaçlarının en az insan uzmanlar kadar doğru olduğunu göstermiştir.
 - Göğüs kanserini teşhis etmek için yapılan bir araştırma, insanların örneklerin %65'ini doğru bir şekilde sınıflandırmasını sağladı; karar ağacı %72 doğru olarak sınıflandırılmıştır.
 - British Petroleum, daha önceki bir kural tabanlı uzman sistemin yerini alan açık deniz petrol platformları için gaz-yağ ayrımı için bir karar ağacı tasarladı.
 - Cessna, 90.000 örnek ve örnek başına 20 özellik kullanarak bir uçak uçuş kontrolörü tasarladı.

Karar Ağaçları