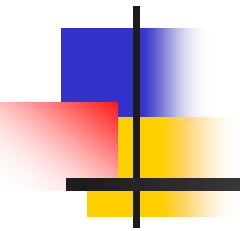


Veri Madenciliği

Sınıflandırma ve Tahmin

Sınıflandırma vs. Tahmin

- Sınıflandırma (Classification)
  - Kategorik sınıf etiketlerini öngörme
  - Bir model oluşturur ve veriyi sınıflandırır
    - Öğrenme kümesi (training set)
    - Sınıf etiketleri (class labels) biliniyor
  - Sınıfı bilinmeyen veriler (sınama kümesi) yaratılan modele göre sınıflandırılır
- Tahmin, öngörü (Prediction)
  - Sürekli değere sahip fonksiyonları modeller
    - Bu modele göre eksik yada bilinmeyen değerleri tahmin eder

Sınıflandırma – Amaç/Yöntem

- Sınıflandırma: Ayrık değişkenlerin hangi kategoride olduklarını diğer nitelikleri kullanarak tahmin etme
- Girdi: öğrenme kümesi (Training set)
  - Ayrık nesnelerden oluşur
  - Her nesne niteliklerden oluşur, niteliklerden biri sınıf bilgisidir (sınıf etiketi)
- Yöntem:
  - Öğrenme kümesi kullanılarak bir model oluşturulur
  - Bulunan modelin başarımı belirlenir
- Çıktı:
  - Sınıf etiketi belli olmayan nesneler oluşturulan model kullanılarak mümkün olan en iyi şekilde sınıflara atanır

Uygulama Alanları

- Kredi başvurusu değerlendirme
- Kredi kartı harcamasının sahtekarlık olup olmadığına karar verme
- Hastalık teşhisi
- DNA üzerinden akrabalık teşhisi
- Metinleri konularına göre ayırma
- Kullanıcı davranışları belirleme

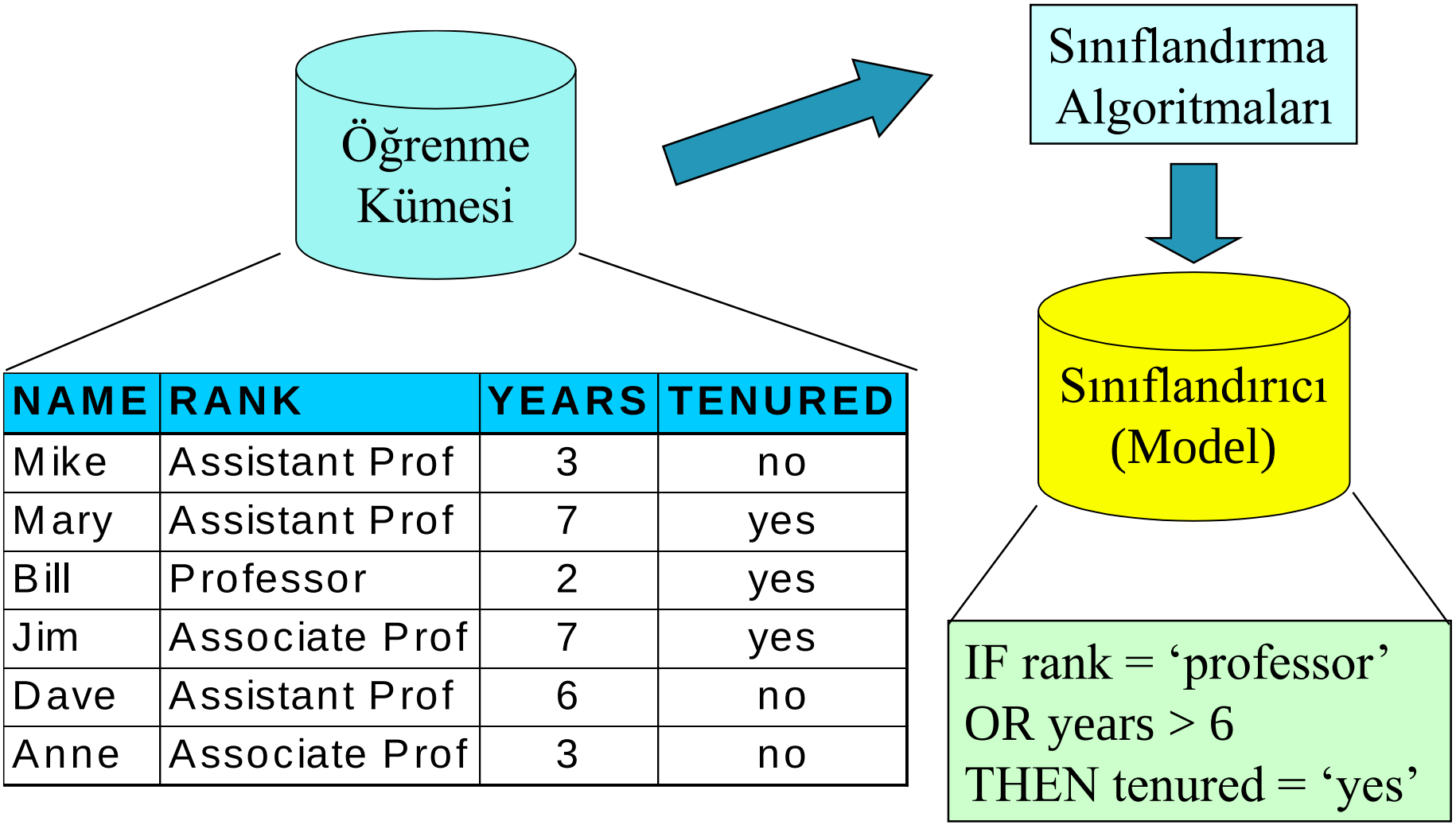
Veri Önişleme

- Veri Dönüşümü
  - Ayrıklaştırma (Discretisation)
    - Sayısal yaş → {çocuk, genç, orta-yaş, yaşlı}
    - Derece olarak sıcaklık →{soğuk,serin,ılık,sıcak}
  - Normalizasyon
    - [-1,1], [0,1]…
- Veri temizleme
  - Gürültü azaltma (Noise reduction)
  - Gereksiz nitelik silme

Sınıflandırma — İki Aşamalı

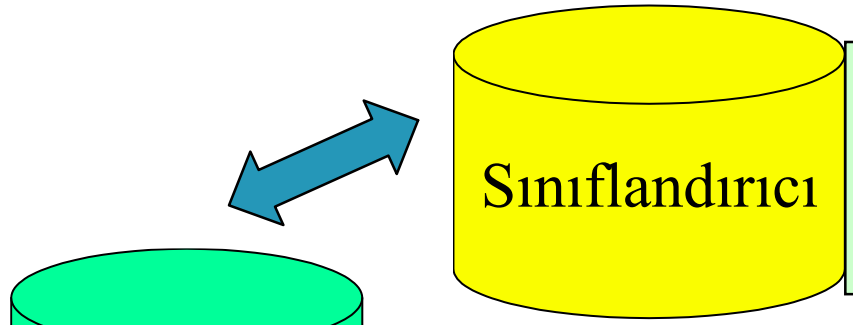
- Model oluşturma: önceden belli sınıfların ifade yöntemi
  - Her nesnenin sınıf etiketi olarak tanımlanan niteliğinin belirlediği bir sınıfta olduğu varsayılır
  - Model oluşturmak için kullanılan nesnelerin oluşturduğu veri kümesi öğrenme kümesi (training set) olarak tanımlanır
  - Model farklı şekillerde ifade edilebilir: if – else kuralları, karar ağaçları, matematiksel formuller
- Modeli kullanma: gelecek bilinmeyen verileri sınıflandırma
  - Modelin başarımı (doğruluğu) belirlenir
    - Sınıf etiketi bilinen bir sınama kümesi örneği model kullanılarak belirlenen sınıf etiketiyle karşılaştırılır
    - Modelin doğruluğu, doğru sınıflandırılmış örneklerinin toplam sınama kümesine oranı olarak belirlenir
  - Sınama kümesi ile öğrenme kümesi bağımsız olmalı
  - Eğer doğruluk oranı kabul edilebilir ise model sınıflandırma için kullanılır

1. Adım: Model Oluşturma

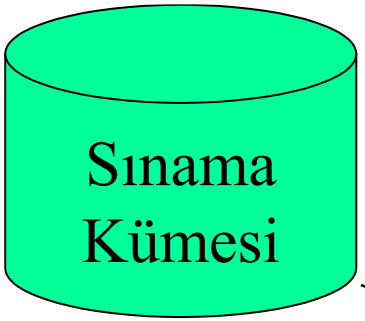


NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

2. Adım: Doğruluk değerlendirme



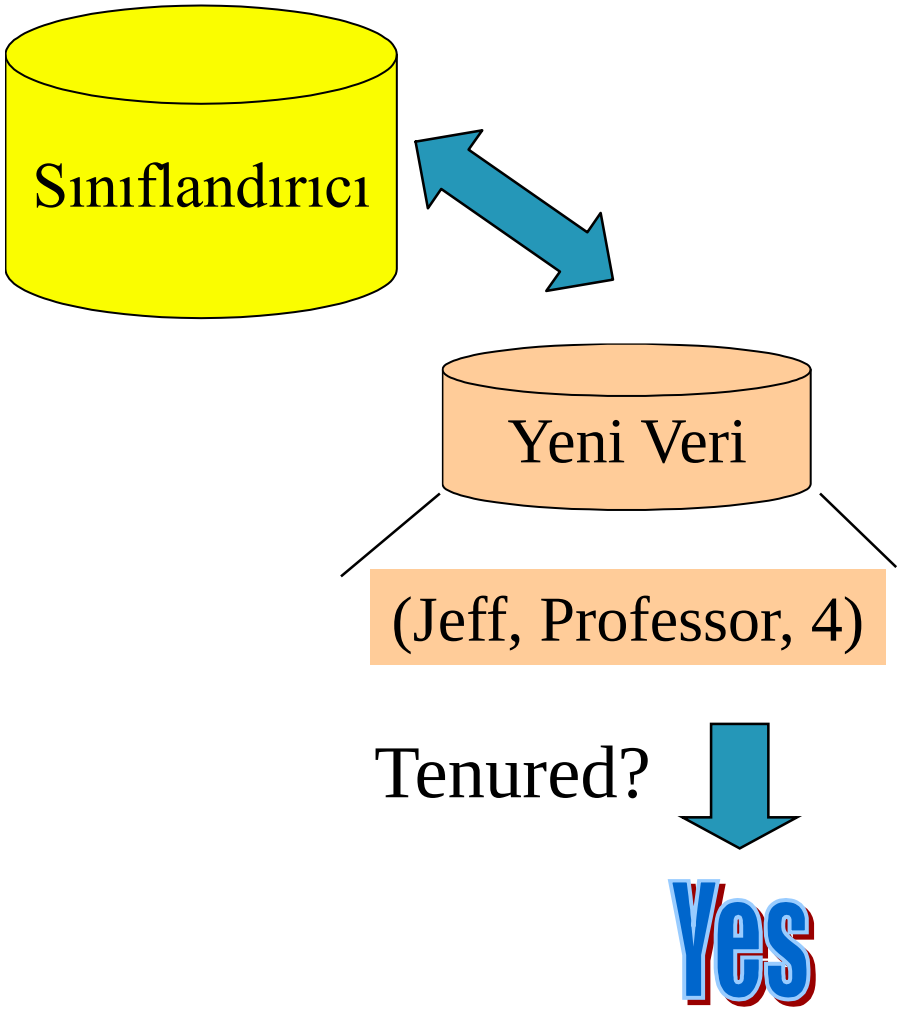
IF rank = ‘professor’
OR years > 6
THEN tenured = ‘yes’



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

75% doğruluk

Modeli Kullanma



Gözetimli vs. Gözetimsiz Öğrenme

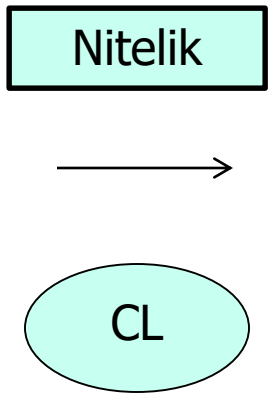
- Gözetimli Öğrenme (Supervised learning) (sınıflandırma)
  - Gözetim: Öğrenme verisindeki sınıfların sayısı ve hangi nesne hangi sınıfta biliniyor
  - Yeni veri öğrenme kümesine bağlı olarak sınıflandırılıyor
- Gözetimsiz (Unsupervised) öğrenme (demetleme)
  - Sınıf sayısı yada hangi nesne hangi sınıfta bilinmiyor
  - Verilen ölçüm değerlerine göre yeni sınıflar yaratılıyor

Sınıflandırma Metodunu Değerlendirme

- Doğruluk
  - Sınıflandırıcı doğruluğu:sınıf etiketlerinin doğruluğu
  - Tahmin doğruluğu: verinin değerinin tahmin doğruluğu
- Hız
  - Modeli oluşturma süresi
  - Sınıflandırma yapma süresi
- Kararlılık:
  - verinin gürültülü yada eksik olması durumunda iyi sonuç vermesi
- Ölçeklenebilirlik: büyük boyutlu verilerle çalışabilmesi
- Anlaşılabilir olması:
  - Kullanıcı tarafından yorumlanabilir olması

Karar Ağaçları

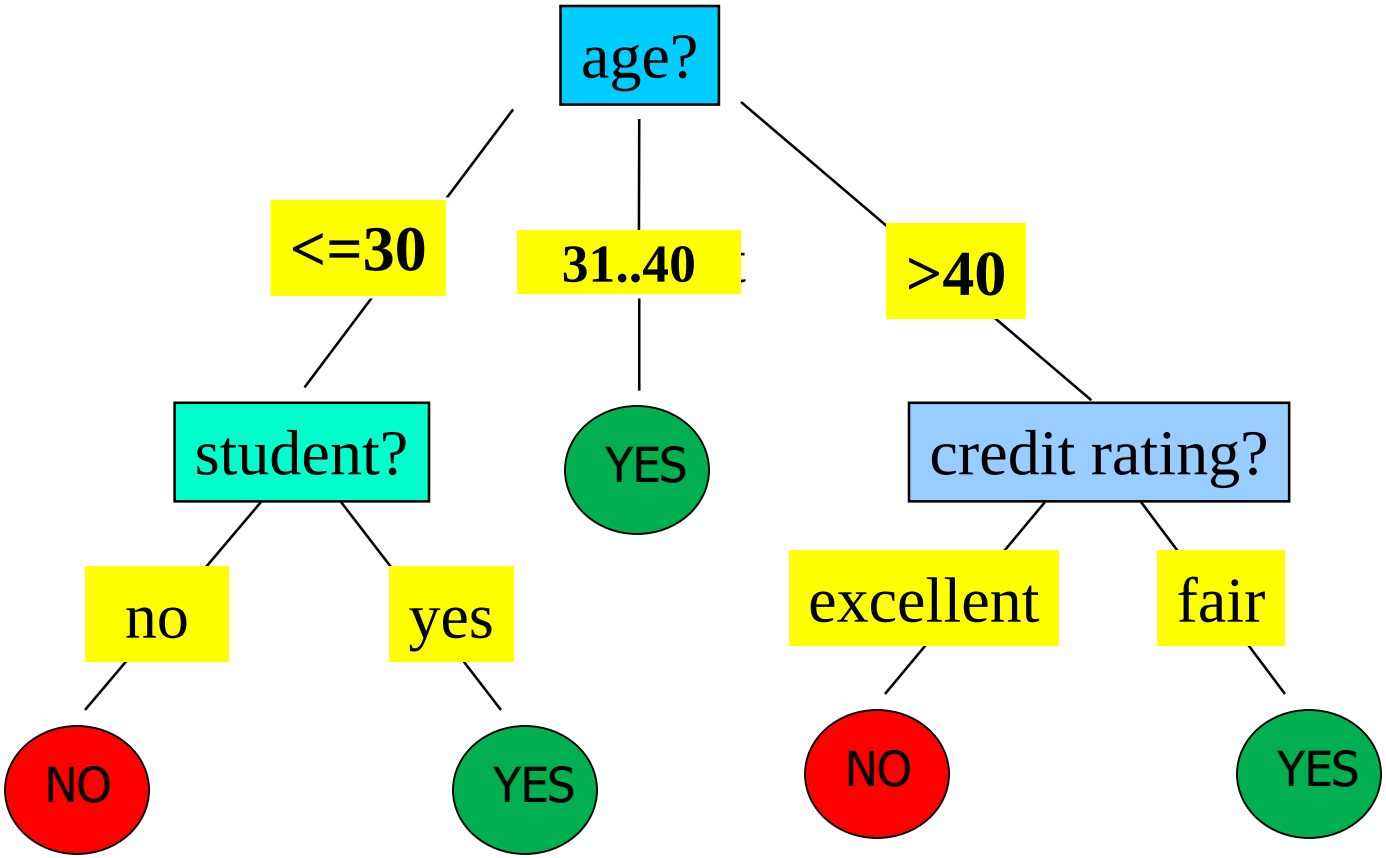
- Akış diagramı şeklinde ağaç yapısı
  - Her ara düğüm -> nitelik sınaması
  - Dallar -> sınama sonucu
  - Yapraklar -> sınıflar



Karar Ağacı: Öğrenme Kümesi

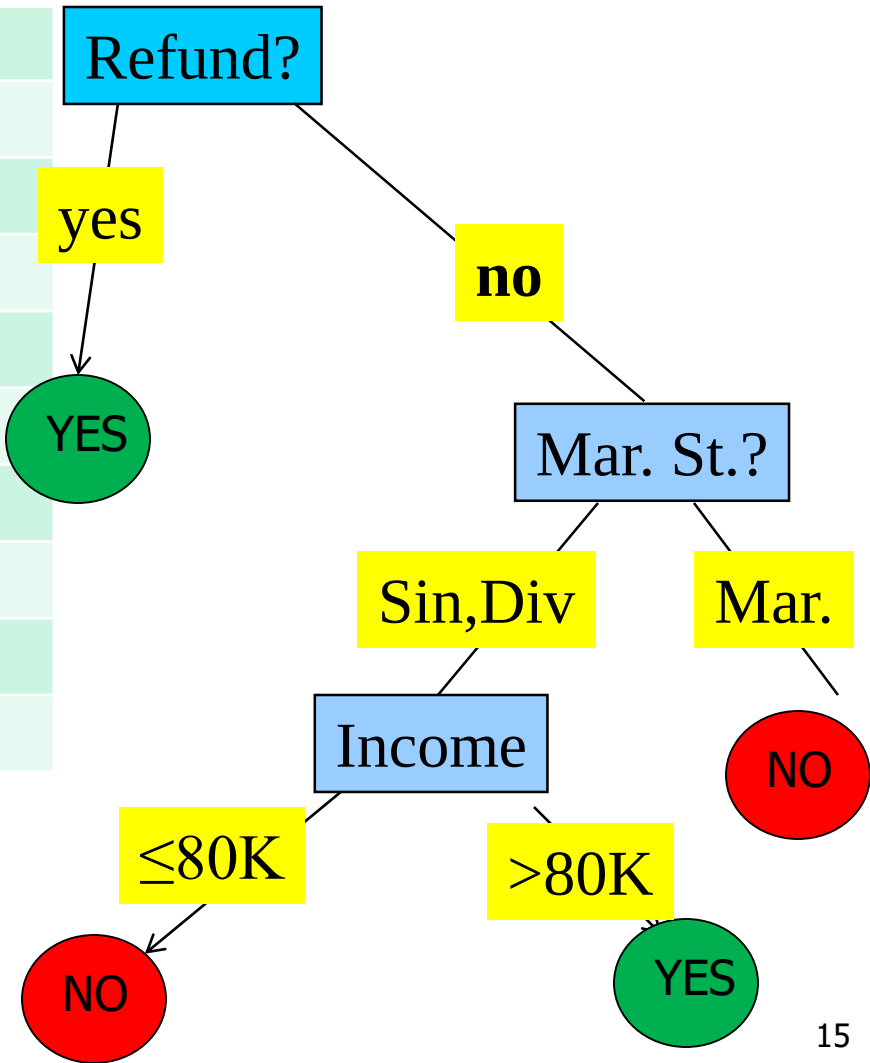
age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Çıktı: "*bilgisayar_alır*" için bir karar ağacı



Örnek 2

id	Refund	Marital St.	Income	Cheat
1	Y	Single	125K	Y
2	N	Married	100K	N
3	N	Single	70K	N
4	Y	Married	120K	Y
5	N	Divorced	95K	Y
6	N	Married	60K	N
7	Y	Divorced	220K	Y
8	N	Single	85K	Y
9	N	Married	75K	N
10	N	Single	90K	Y



Veriye uyan birden fazla karar ağacı türetilebilir

Sınıflandır: Y,Div,75K
N,Sin,55K

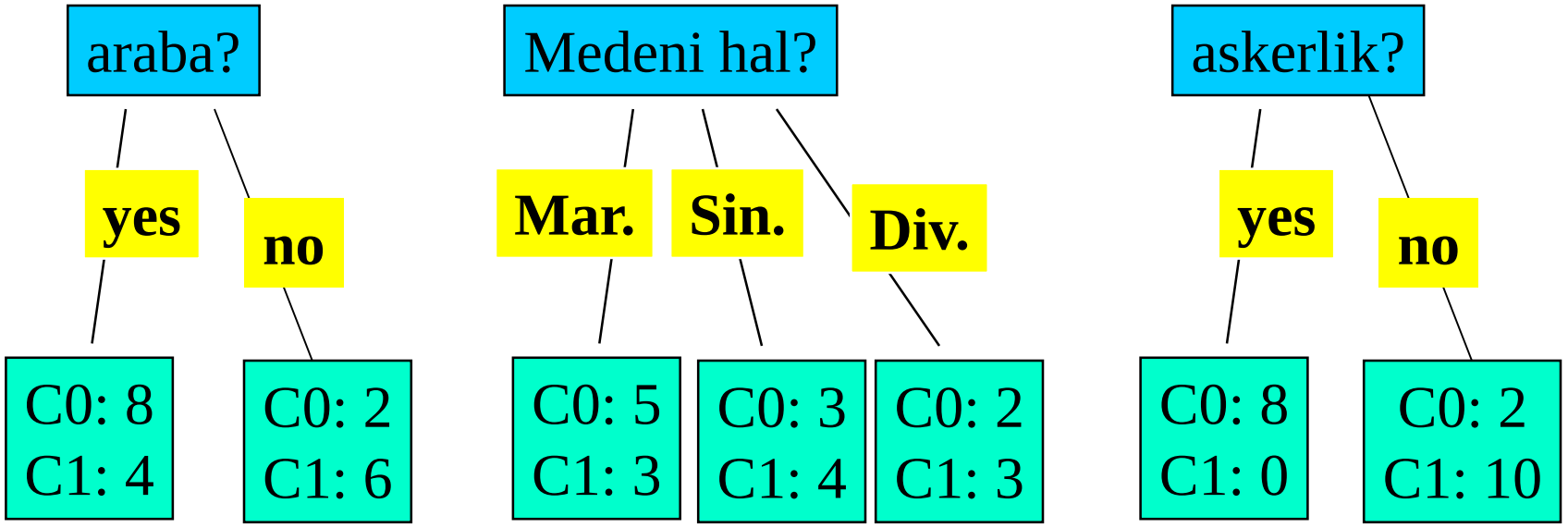
Karar Ağacı oluşturma

- Temel algoritma (açgözlü (greedy) algoritma)
  - Ağaç top-down recursive divide-and-conquer bir yaklaşımla oluşturulur
  - Ağaç bütün verinin oluşturduğu tek bir düğümle başlıyor
  - Nitelikler kategorik (eğer sürekli nitelikler varsa önceden ayrıştır)
  - Eğer örneklerin hepsi aynı sınıfa aitse düğüm yaprak olarak sonlanıyor ve sınıf etiketini alıyor
  - Örnekleri sınıflara en iyi bölecek olan nitelik seçiliyor?
    - Hüristik yada istatistiksel değerler (e.g., information gain)
- Sonlanma koşulları
  - örneklerin hepsi aynı sınıfa ait
  - örnekleri bölecek nitelik kalmamış – çoğunluk oylaması (majority voting) ile yapraktaki sınıf belirlenir
  - kalan niteliklerin değerini taşıyan örnek yok

En İyi Nitelik Seçimi

- Orjinal Veri
  - 10 in C0 (erkek)
  - 10 in C1 (kadın)

Açgözlü yöntem:
Homojen dağılım daha iyi
Sonuç verir
• Çoğu (hepsi) aynı sınıfta



Nitelik Seçimi – Bilgi Kazancı (Information Gain)

- entropy temeline dayanır– belirsizlik

$$Entropy = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- Eğer tüm örnekler aynı sınıfa aitse belirsizlik yoktur dolayısıyla entropy 0 dır
- Eğer uniform dağıldıysa her sınıf eşit olasılıkla mümkündür ve entropy 1dir
- Diğer durumlarda 0< entropy < 1
- Amaç: Entropiyi (belirsizliği) en aza indirecek niteliği seç

Nitelik Seçme Ölçütü: Bilgi Kazanımı Information Gain (ID3)

- Bilgi kazanımı en yüksek olan nitelik seçilir
- p_i D öğrenme kümesindeki bir varlığın C_i sınıfına ait olma olasılığı olsun, $p_i = |C_{i,D}|/|D|$ olarak hesaplanır
- D içindeki bir varlığı sınıflandırmak için gerekli bilgi (D nin entropisi):

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- D kümesi A niteliğine göre v parçaya bölündükten sonra D'yi sınıflandırmak için gerekli bilgi

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

- A niteliğine göre bölünmeden dolayı bilgi kazancı

$$Gain(A) = Info(D) - Info_A(D)$$

Attribute Selection: Information Gain

- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$$

$$Info_{age}(D) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694$$

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971

$\frac{5}{14} I(2,3)$ demek "age <=30" grubunda 5 örnek var toplamda 14tu, bunlar: 2 yes - 3 no

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

Benzer şekilde,

$$Gain(income) = 0.029$$

$$Gain(student) = 0.151$$

$$Gain(credit\_rating) = 0.048$$

Sürekli Değerlerli Verilerde Bilgi Ölçümü

- A niteliği sürekli değere sahip olsun
- A için en iyi bölme noktası (*best split point*) hesaplanmalı
  - Değerleri küçükten büyüğe sırala
  - İki komşu değerin orta noktası olası bölme noktasıdır
    - $(a_i+a_{i+1})/2$ is the midpoint between the values of a_i and a_{i+1}
  - A niteliği için *minimum gerekli bilgi* (entropi) gerektiren bölme noktası seçilir
  - Split:
  - D1 alt kümesi D içinde A ≤ bölme-noktası olanlar, ve
  - D2 alt kümesi D içinde A > bölme-noktası olanlar

Kazanım Oranı (Gain Ratio)(C4.5)

- Bilgi kazanımı metodu çok çeşitli değerlere sahip nitelikleri seçme eğilimdedir
- Bu problemin çözümünde C4.5 (ID3 ten geliştirilmiş) kazanım oranı kullanılır (normalization to information gain)

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

  - GainRatio(A) = Gain(A)/SplitInfo(A)
- Ex.

$$SplitInfo_A(D) = -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right) = 0.926$$

  - gain_ratio(income) = 0.029/0.926 = 0.031
- En yüksek kazanım oranına sahip nitelik seçilir

Gini index (CART, IBM IntelligentMiner)

- Herzaman binary (ikili) agac uretir
- D kümesi n sınıfdan örnekler içeriyorsa, gini index, gini(D) şu şekilde ifade edilir (p_j j sınıfının D kümesinde görülme sıklığıdır)

$$gini(D) = 1 - \sum_{j=1}^{n} p_j^2$$

- D kümesi A niteliğine göre ikiye D1 ve D2 olarak bölünürse, gini index *gini*(*D*) şu şekilde ifade edilir

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

- Kusurdaki azalma (Reduction in Impurity)

$$\Delta gini(A) = gini(D) - gini_A(D)$$

- En küçük $gini_{split}(D)$ye sahip nitelik (yada en fazla kusur azaltan) bölme noktası olarak seçilir (*tüm olası bölme noktaları tektek denenmelidir*)

Örnek: Gini index

- 9 kişi buys_computer = "yes" and 5 kişi ise "no" sınıfında

$$gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

- Income niteliği kullanarak D'yi 2ye bolduk diyelim: 10 kişi D_1 {medium, high} ve 4 kişi D_2 de {low}

$$gini_{income \in \{low\}}(D) = \left(\frac{10}{14}\right) Gini(D_1) + \left(\frac{4}{14}\right) Gini(D_2)$$

$$= \frac{10}{14}\left(1 - \left(\frac{6}{10}\right)^2 - \left(\frac{4}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2\right)$$

$$= 0.450$$

$$= Gini_{income \in \{high\}}(D)$$

- $gini_{\{low,medium\}}$ = 0.442 olarak hesaplanır
- {low,med} – {high} bolmek {Low} – {med,high} bolmekten daha iyi cunku ginisplit daha dusuk yani Δgini degeri daha yuksek

Ölçüm Yöntemlerinin Karşılaştırması

- Her üç yöntemde iyi sonuç verir ancak,
  - Information gain:
    - Cok ceşitli değerler (multivalued) alan nitelikleri secme eğilimindedir
  - Gain ratio:
    - Bir parçanın diğerinden daha küçük olduğu dengesiz bölmeler yapma eğiliminde
  - Gini index:
    - Cok ceşitli değerler (multivalued) alan nitelikleri secme eğilimindedir
    - Sınıf sayısı fazla ise sorun yaşayabiliyor
    - Böldüğü her iki grupta yaklaşık boyutlarda olma eğilimindedir

Aşırı Öğrenme ve Ağaç budama

- Aşırı öğrenme: Yaratılan karar ağacı öğrenme kümesine fazla bağlı olabilir
  - Çok fazla dal, gürültü ve sapan veriler nedeniyle anormallikler
  - Yeni verilerde düşük doğruluk
- Aşırı öğrenmeyi engelliyen iki yöntem
  - Ön budama (Prepruning): Ağaç yaratırken erken dur – eğer bölme belli bir sınır değerden kötü kazanç sağlıyorsa bölme
    - Sınır değeri belirlemek kolay değil
  - Son budama (Postpruning): Tüm ağacı yarattıktan sonra kötü kısımları aşama aşama buda
    - Öğrenme kümesinden başka ikinci bir öğrenme kümesi kullanarak en iyi budama noktaları belirlenir

Karar Ağaçlarında Aşırı Öğrenme

- Öğrenme kümesinin küçük, gürültülü olması, eksik veri içermesi
- Çözüm budama (pruning)
  - En güvenilmez dalları buda
  - Çoğunluk oylaması
  - Önbudama (prepruning)
  - Sonbudama (Postpruning)