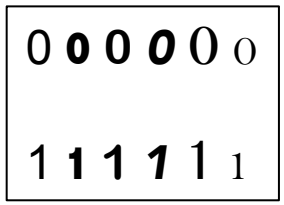
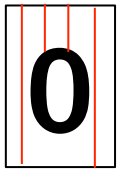
REMINDER

Classification: Digit classification



- Represent with appropriate hand-crafted or systemic or automatically extracted features

$$\begin{bmatrix} width \\ height \\ width\,/\,height \\ pixeldensity \\ concavity \\ \dots \end{bmatrix}$$

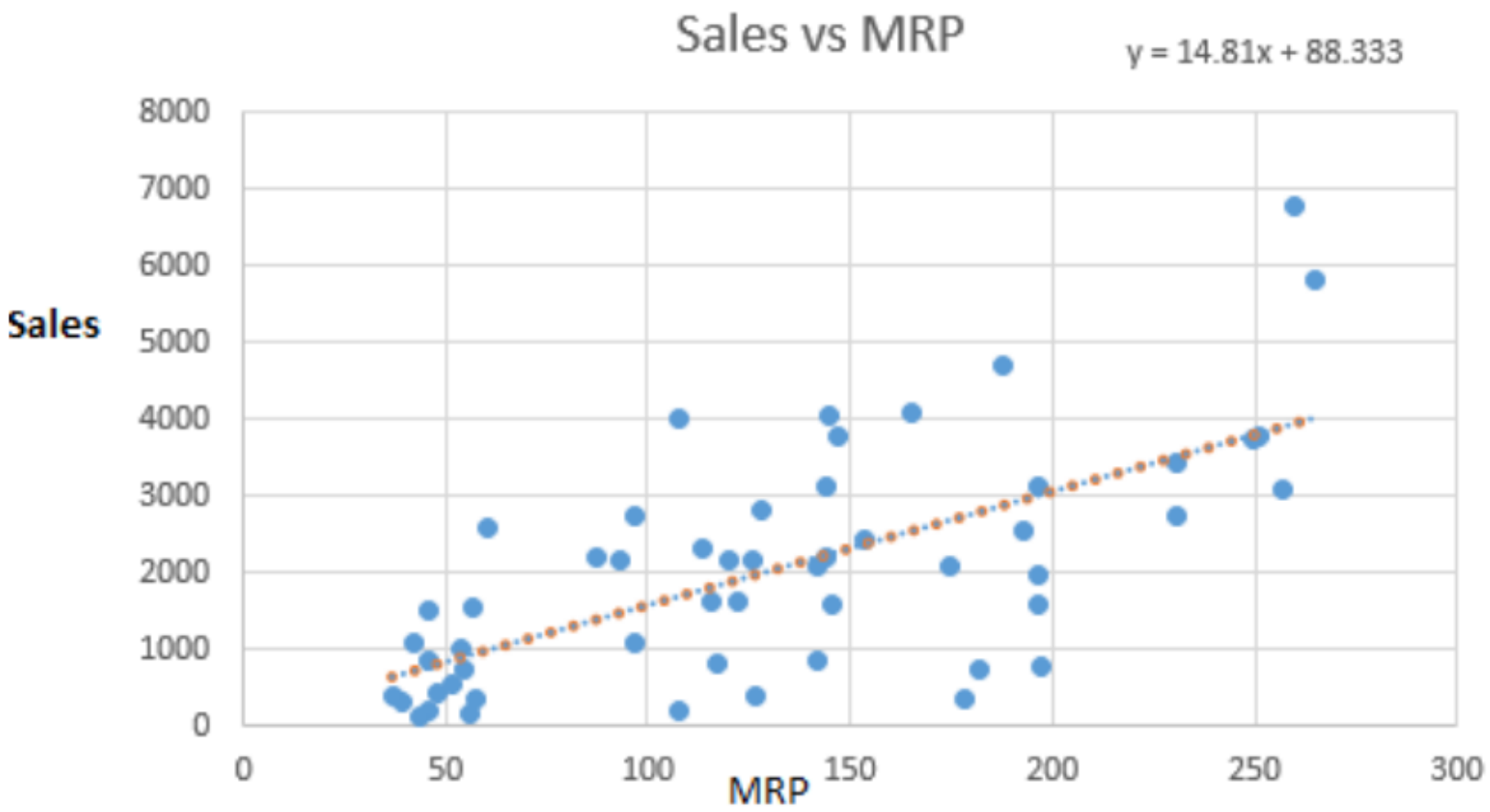


Regression: Applicant scoring

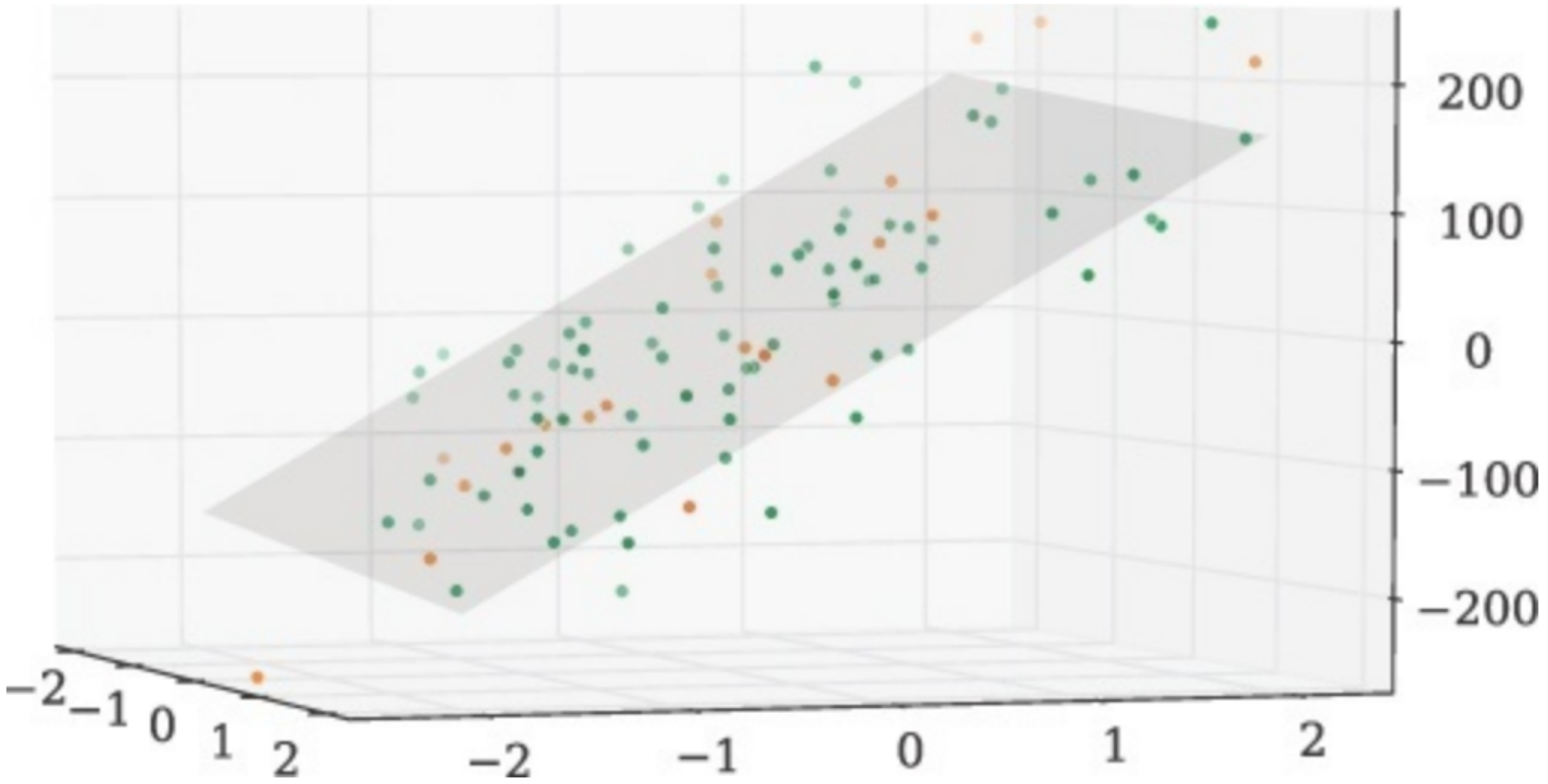
- Automatically assign a score to a credit card applicant.

$$\begin{bmatrix} age \\ salary \\ current\_debt \\ credit\_amount \end{bmatrix}$$

- score = $\theta_0+\theta_1$*age+θ_2*salary+θ_3*current_debt+θ_4*credit_amount


Sales vs MRP
y = 14.81x + 88.333
Sales
8000
7000
6000
5000
4000
3000
2000
1000
0
0
50
100
150
200
250
300
MRP

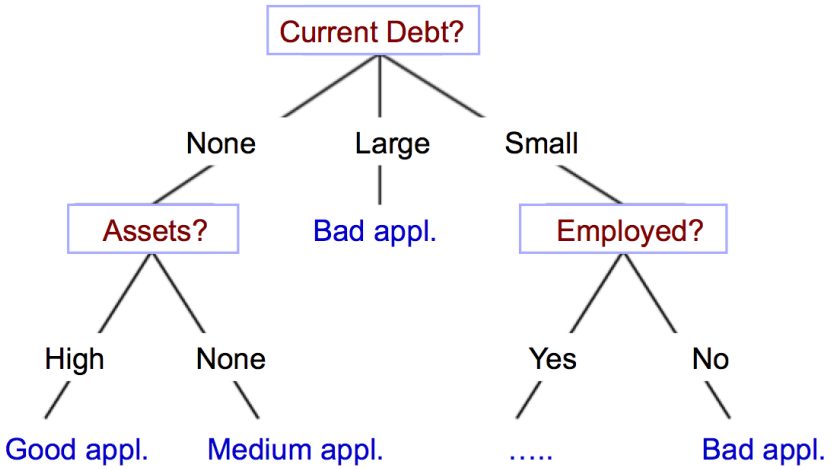
Multiple Linear Regression

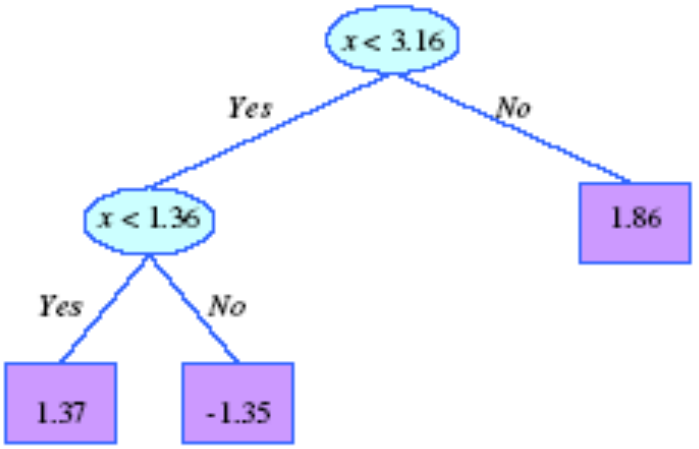


Decision Tree Learning

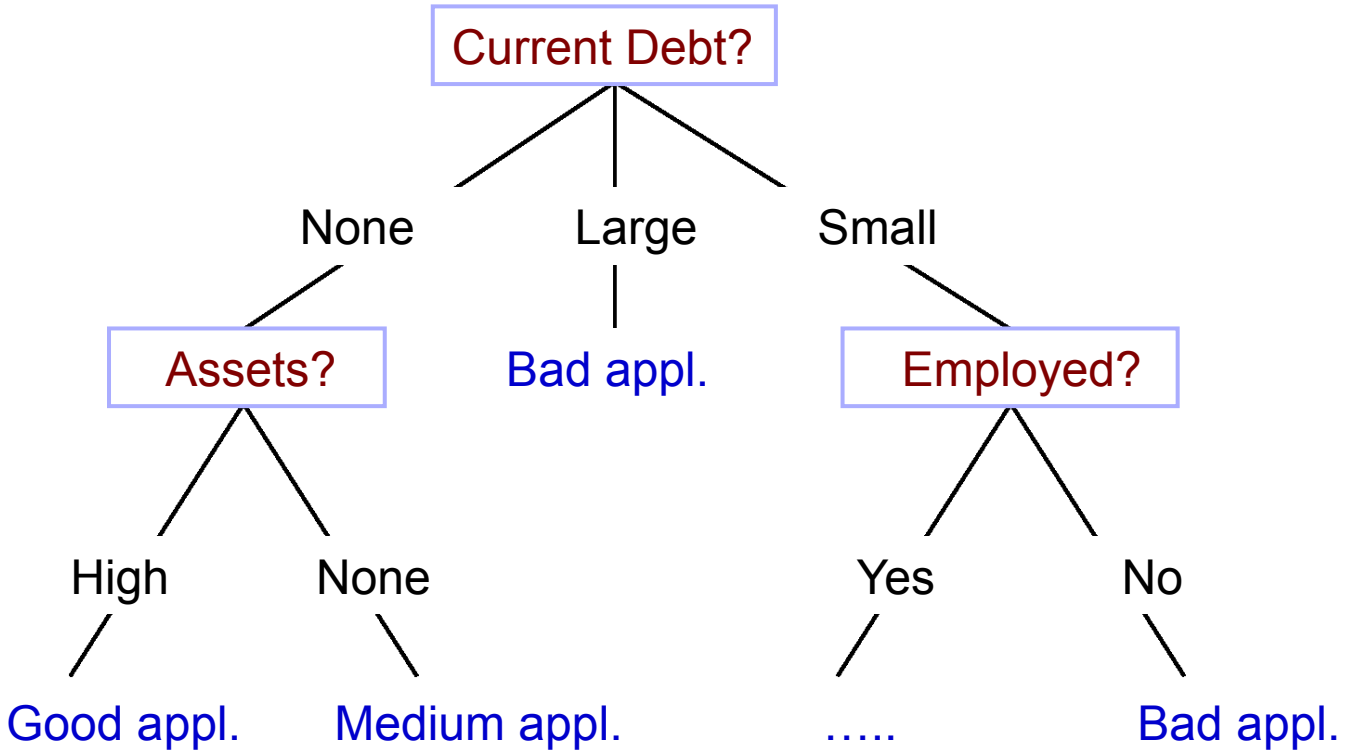
Decision Trees

- En çok kullanılan ve pratik makine öğrenmesi yöntemlerinden biri
- Hem sınıflandırma, hem regresyon problemlerine uygun
- Rasgele Ormanlar veya Gradyan Artırımlı Karar Ağaçları gibi ileri makine öğrenmesi algoritmalarının temel öğesi



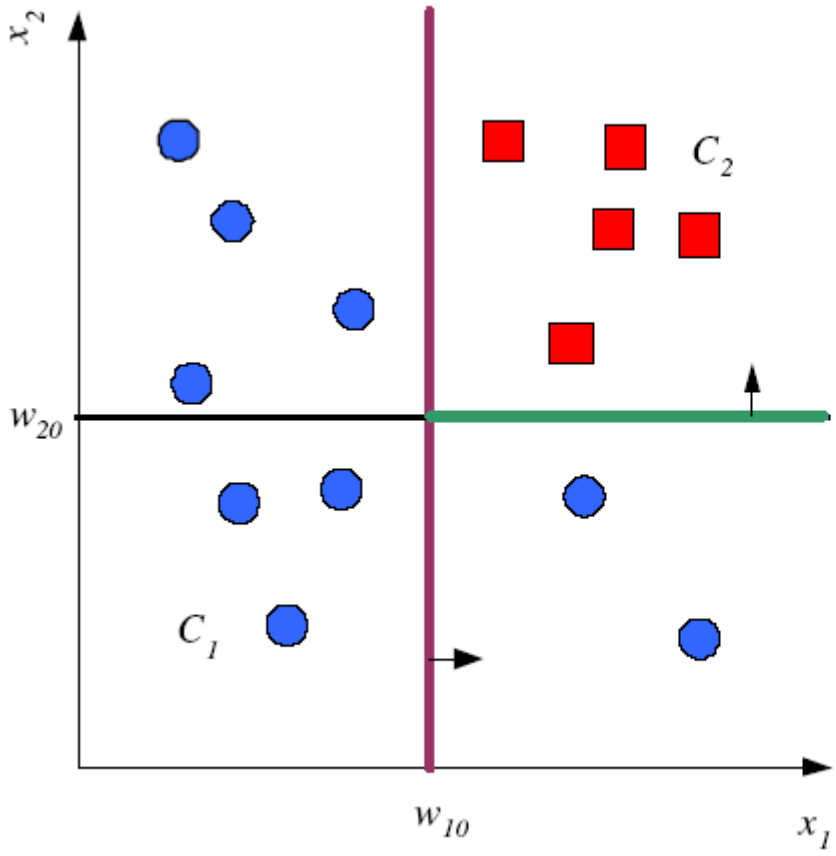


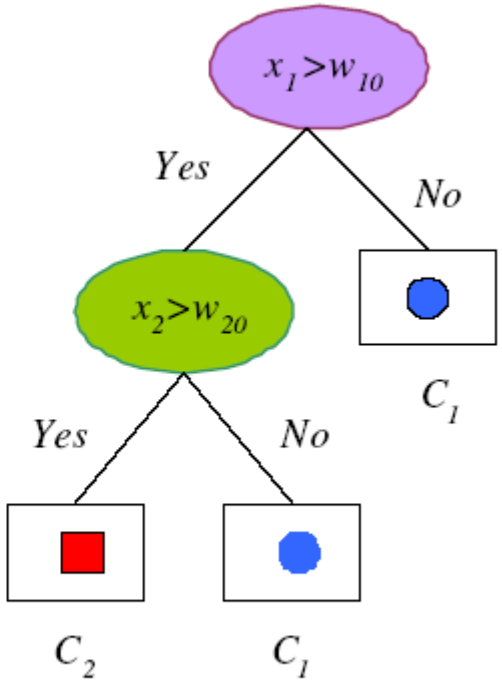
“Good credit applicant?” için Karar Ağacı



- Her iç düğüm bir test/soru
- Her dal (branch) testin bir sonucu/cevabı
- Her uç düğüm bir karar/etiket♪

Karar Bölgeleri (Decision Regions)

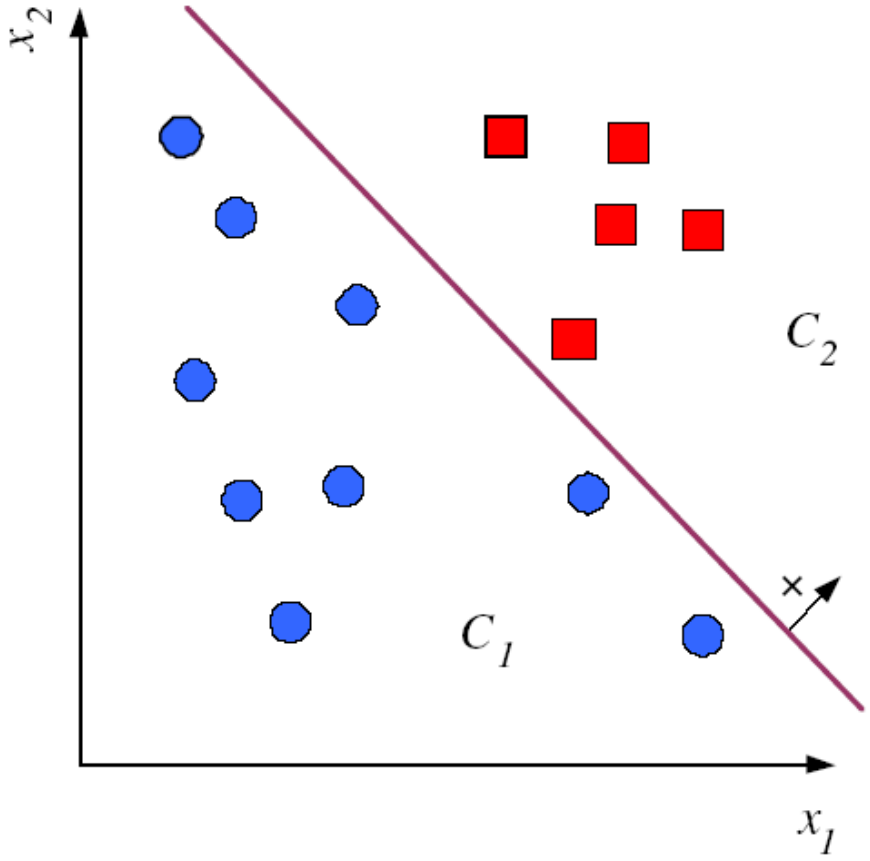


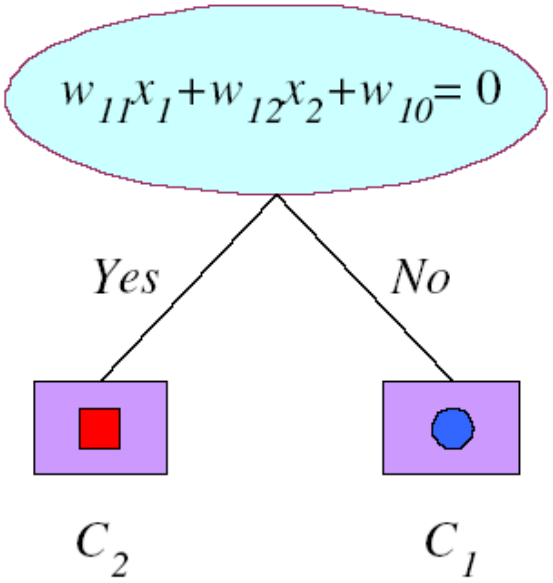


- İç düğümler
  - Tek değişkenli:
    - Ayrık (Discrete) x_i : *kaç değer alabiliyorsa, o kadar dallanma*
    - Sürekli (Continuous) x_i : Eşik değeri ile ikili dallanma: $x_i > \theta$

Birden çok değişkene bakan ağaçlar (Multivariate Trees)

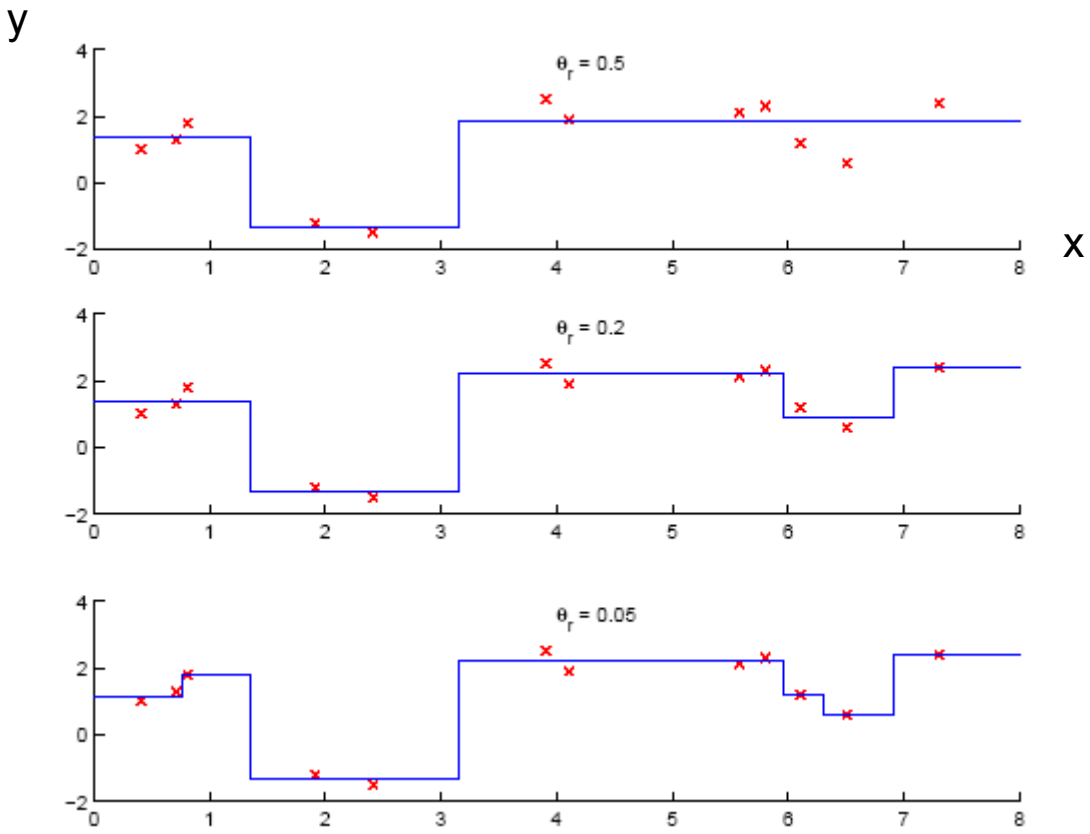
Genelde az kullanılır



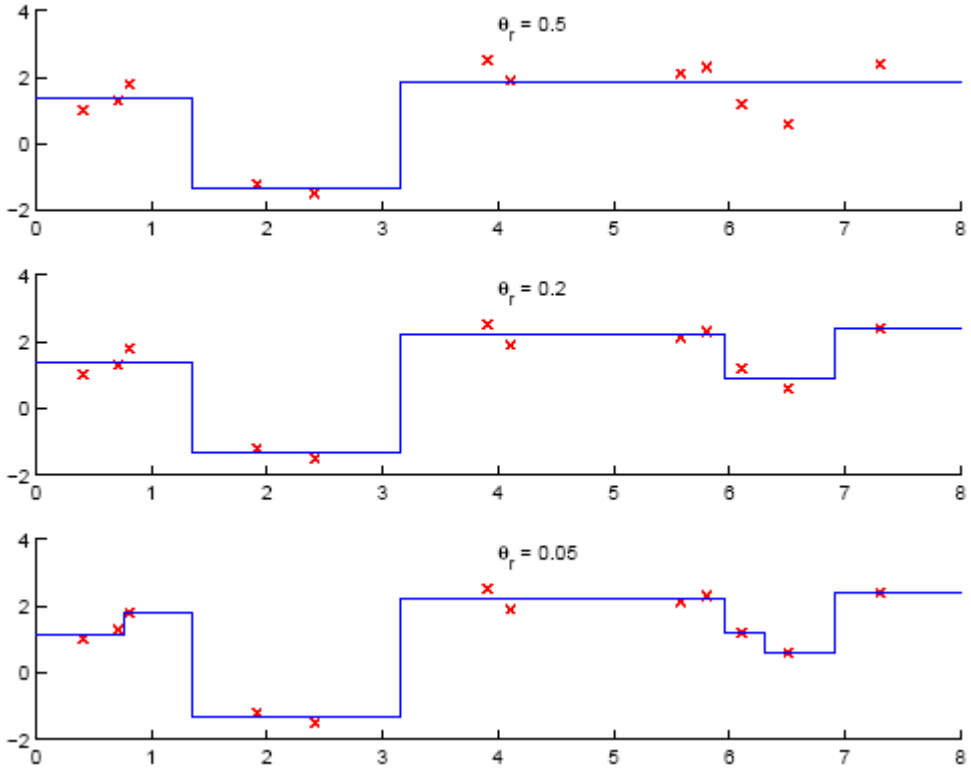


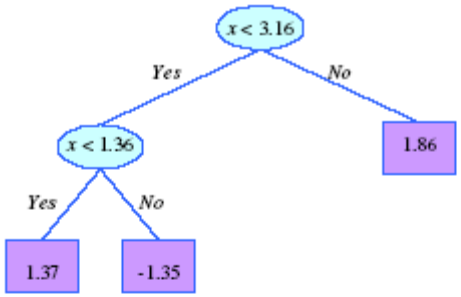
Karar Ağaçları ile Regresyon

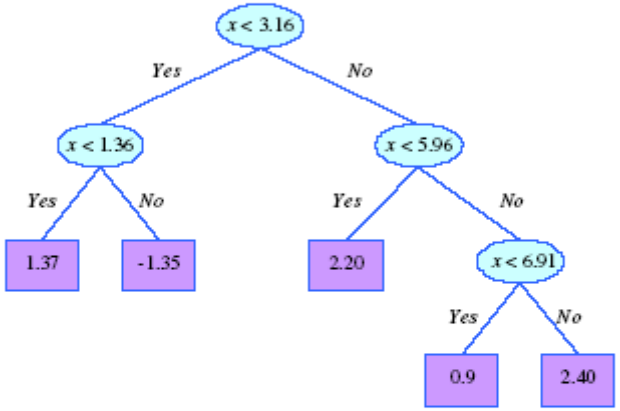
- Elimizdeki kırmızı noktalar veriler olsun. Biz de x ve y arasındaki fonksiyonu öğrenmeye çalışıyor olalım.

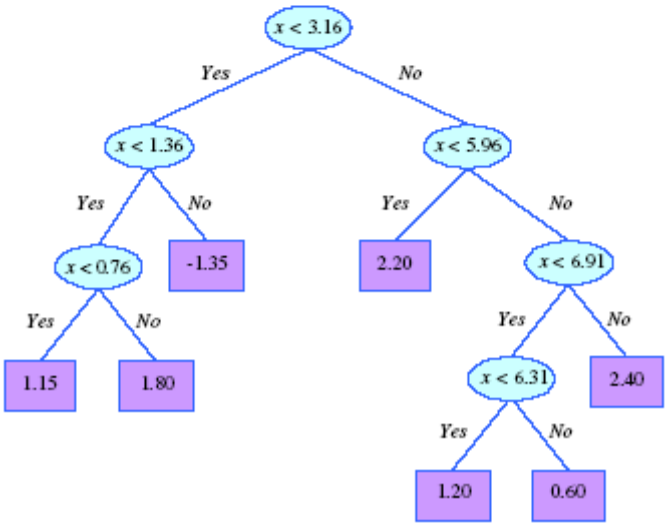


Hangisi doğru ağaç?



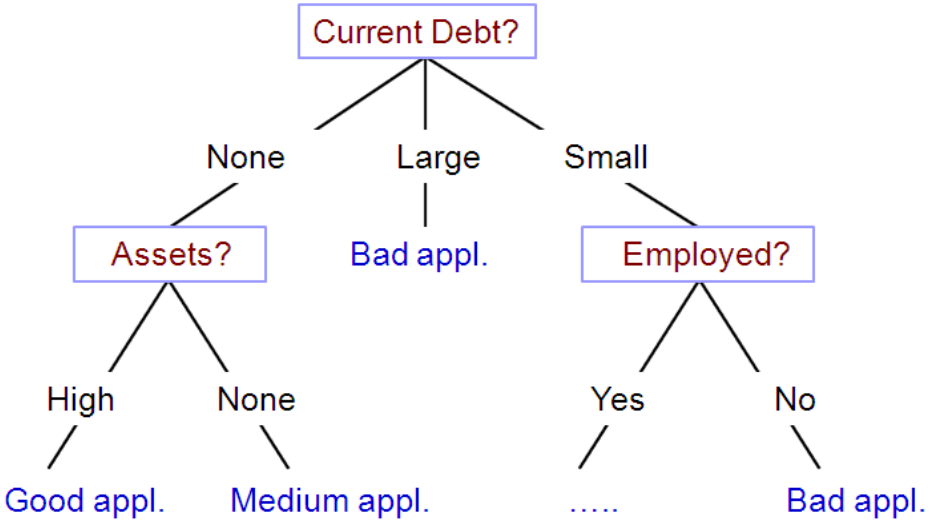






- We do not know which tree is better, one simple way to select the model (tree) is to use validation set performance.

Karar Ağacı

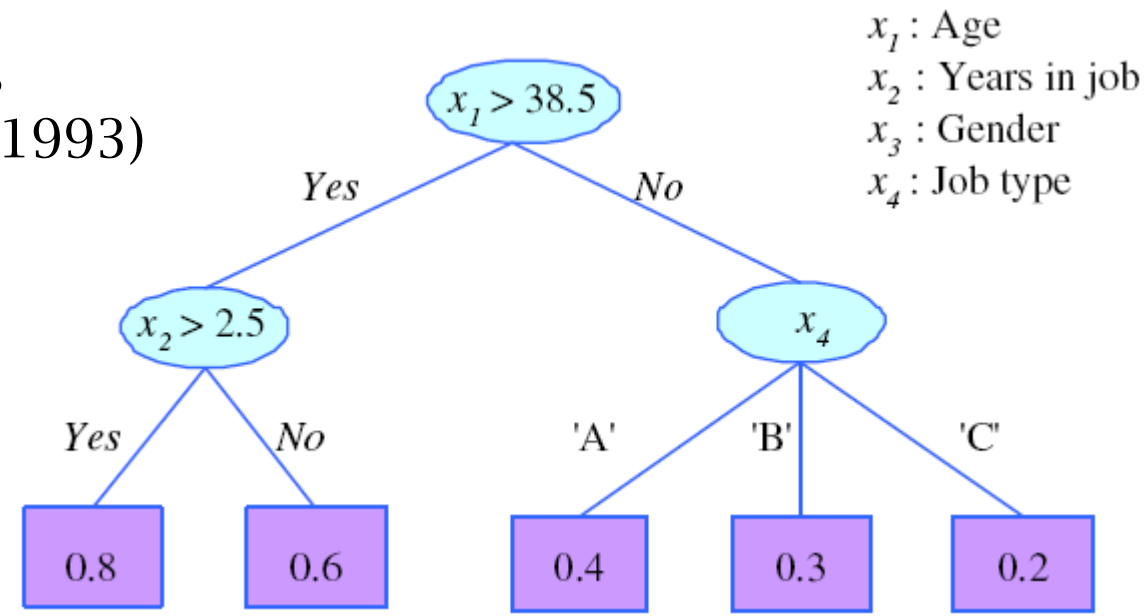


If (Current Debt= Large) **OR** (Current Debt= Small **AND** Employed? = No)
then **Bad Applicant**

- .... OR (... AND ... AND ...) OR (... AND ...) OR ...

Rule Extraction from Trees

C4.5Rules
(Quinlan, 1993)



R1: IF (age>38.5) AND (years-in-job>2.5) THEN y =0.8
R2: IF (age>38.5) AND (years-in-job≤2.5) THEN y =0.6
R3: IF (age≤38.5) AND (job-type='A') THEN y =0.4
R4: IF (age≤38.5) AND (job-type='B') THEN y =0.3
R5: IF (age≤38.5) AND (job-type='C') THEN y =0.2

Karar Ağacı Öğrenmesi

- Verilen bir eğitim kümesi için, onu sıfır hata ile öğrenen pek çok karar ağacı bulunabilir.

- Bu ağaçların en küçüğünü bulmak NP-complete bir problemdir (Quinlan 1986), bu yüzden yerel arama algoritmaları ile “iyi” ağaçlar bulmaya yöneliyoruz.

ID3 Karar Ağacı Öğrenme Algoritması

- Öğrenme **açgözlü (greedy)** ve **tekrarlamalı** (**recursive)** ilerler (Breiman et al, 1984; Quinlan, 1986, 1993)

- Her iç düğümde;
  - Sorgulayacak bir öznitelik seç
  - Cevaplara göre dallanmayı gerçekleştir
  - Tekrarla

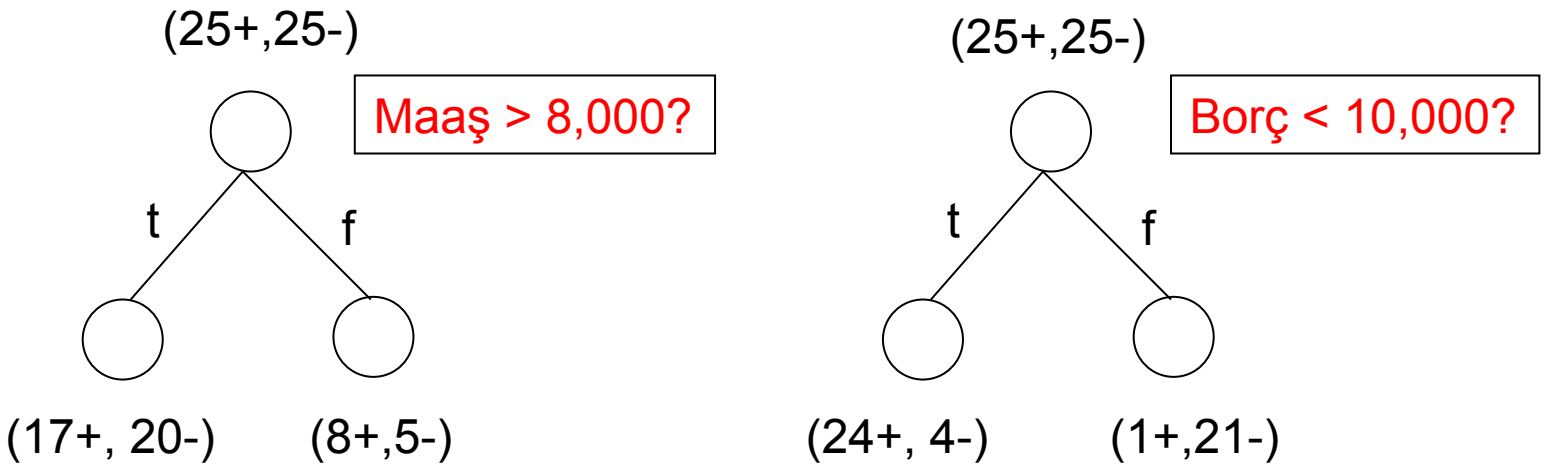
- Peki öznitelik seçimi nasıl olmalı?
  - ...

Top-Down Induction of Decision Trees

Main loop:

1. $A \leftarrow$ the "best" decision attribute for next *node*
2. Assign A as decision attribute for *node*
3. For each value of A, create new descendant of *node*
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes

- Diyelim elimizde 25+, 25- eğitim örneği (training sample) var. Bu örneklere A1'e veya A2'ye göre böldüğümüzde ortaya çıkan tablo aşağıdaki gibi.



- Soru: Hangi ağaçta "kalan belirsizlik" daha azdır?

Entropi

- **Belirsizlik Ölçütü**
- Bir rasgele değişken X'in entropisi:

$$H[X] = -\sum_{X=x} p(x) \log_2 p(x)$$

Entropy of a **Binary** Random Variable

- Example: Assume at an internal node in the decision tree, there are 25 samples of good applicants and 75 samples of bad applicants. What is the entropy of the GoodApplicant random variable at that node?

25/100 -> p= 0.25

75/100 -> 1-p= 0.75

Entropy = $0.25 \times (-\log_2 0.25) + 0.75 \times (-\log_2 0.75) = 0.81$

Note: $\log_2 0.25 = \log_2 2^{-2} = -2$

Use of Entropy in Determining the Next Test/Question

- We will use the entropy of the remaining tree as our measure to prefer one attribute over another.

- We will choose the attribute that brings us the biggest information gain, or equivalently, results in a tree with the lower weighted entropy.

Information Gain

$$Gain(S, A) = \text{expected reduction in entropy due to sorting on } A$$

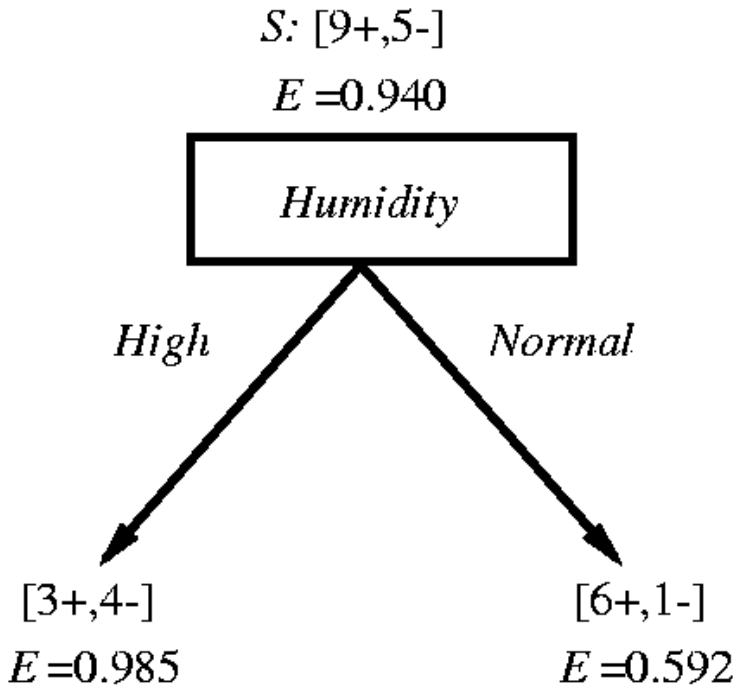
$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

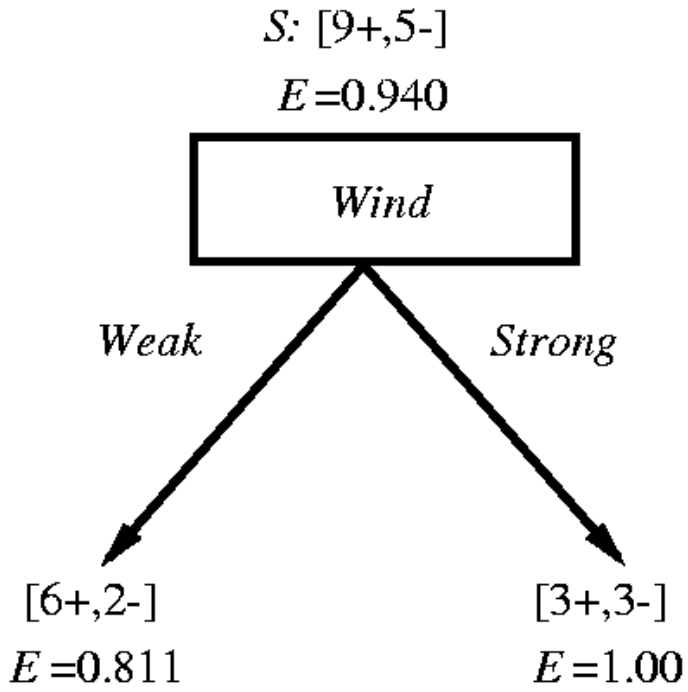
Please note: Gain is [what you started with – Remaining entropy].
So we can simply choose the tree with the **smallest remaining entropy**!

Training Examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
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

Which attribute is the best classifier?





Gain (S, Humidity)

= .940 - (7/14).985 - (7/14).592
= .151

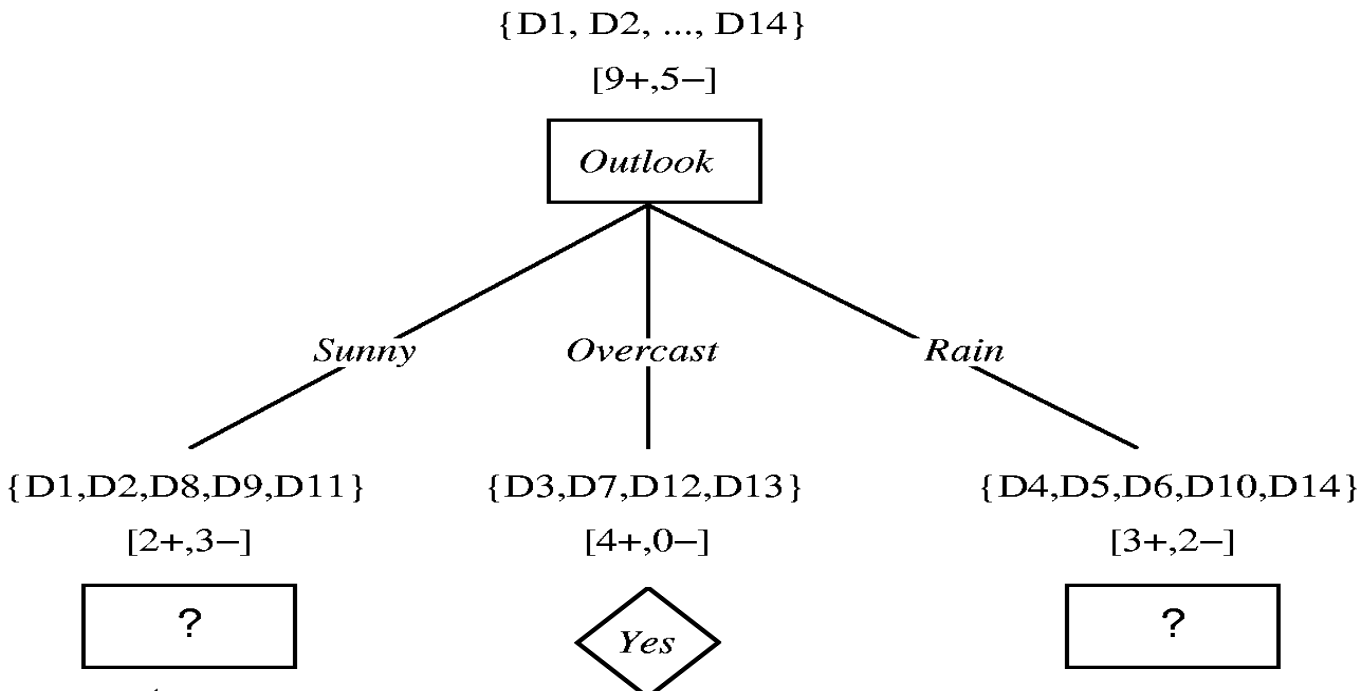
Gain (S, Wind)

= .940 - (8/14).811 - (6/14)1.0
= .048

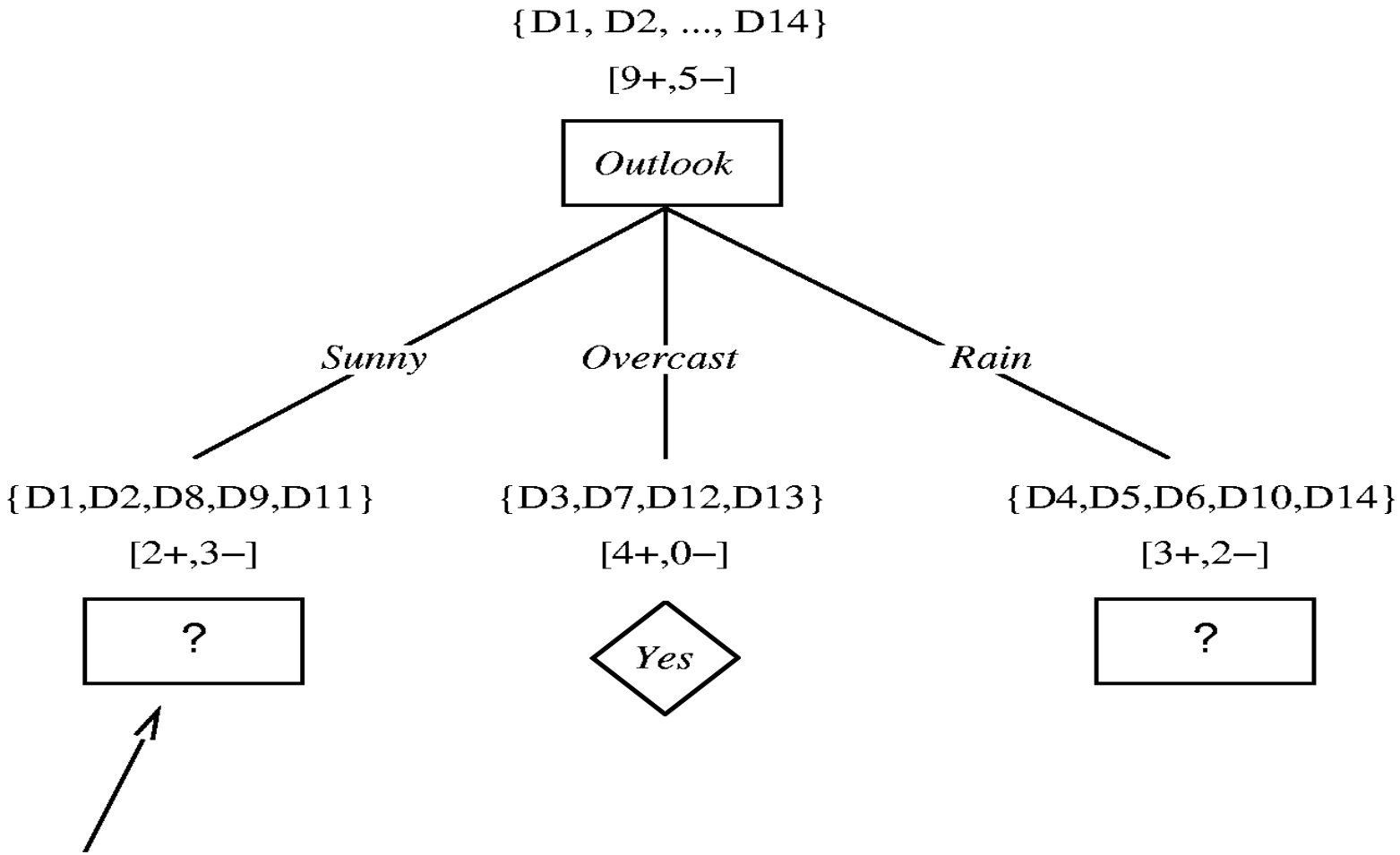
We would select the Humidity attribute to split the root node as it has a higher Information Gain.

Selecting the Next Attribute

- Computing the information gain for each attribute, we selected the *Outlook* attribute as the first test, resulting in the following partially learned tree:



- We can repeat the same process recursively, until Stopping conditions are satisfied.



Which attribute should be tested here?

S_{sunny} = {D1,D2,D8,D9,D11}

Gain (S_{sunny}, *Humidity*) = .970 − (3/5) 0.0 − (2/5) 0.0 = .970

Gain (S_{sunny}, *Temperature*) = .970 − (2/5) 0.0 − (2/5) 1.0 − (1/5) 0.0 = .570

Gain (S_{sunny}, *Wind*) = .970 − (2/5) 1.0 − (3/5) .918 = .019

Until stopped:

- Select one of the unused attributes to partition the remaining examples at each non-terminal node using only the training samples associated with that node

Stopping criteria:

- each leaf-node contains examples of one type
- algorithm ran out of attributes
- …

Other Issues With Decision Trees

Continuous Values

Missing Attributes

…

Continuous Valued Attributes

- Create a discrete attribute to test continuous variables

  Temperature = 82:5

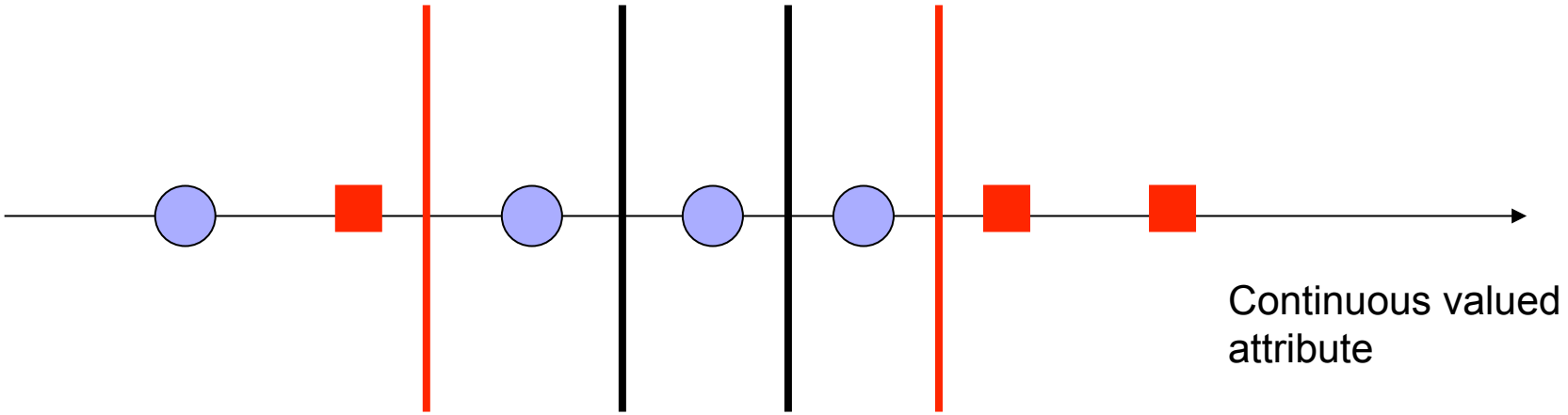
  (*Temperature* > 72:3) = *t; f*

- *How to find the threshold?*

Temperature:	40	48	60	72	80	90
PlayTennis:	No	No	Yes	Yes	Yes	No

Incorporating continuous-valued attributes

- Where to cut?
  - We can show that the threshold is always the the transitions (shown in red boundaries between the two classes)



Handling attributes with differing costs

- Sometimes, some attribute values are more expensive or difficult to prepare.
  - medical diagnosis, BloodTest has cost $150
- In practice, it may be desired to postpone acquisition of such attribute values until they become necessary.
- To this purpose, one may modify the attribute selection measure to penalize expensive attributes.
  - Tan and Schlimmer (1990) $\frac{Gain^2(S,A)}{Cost(A)}$
  - Nunez (1988) $\frac{2^{Gain(S,A)} - 1}{(Cost(A)+1)^w}, w \in [0,1]$

Handling training examples with missing attribute values

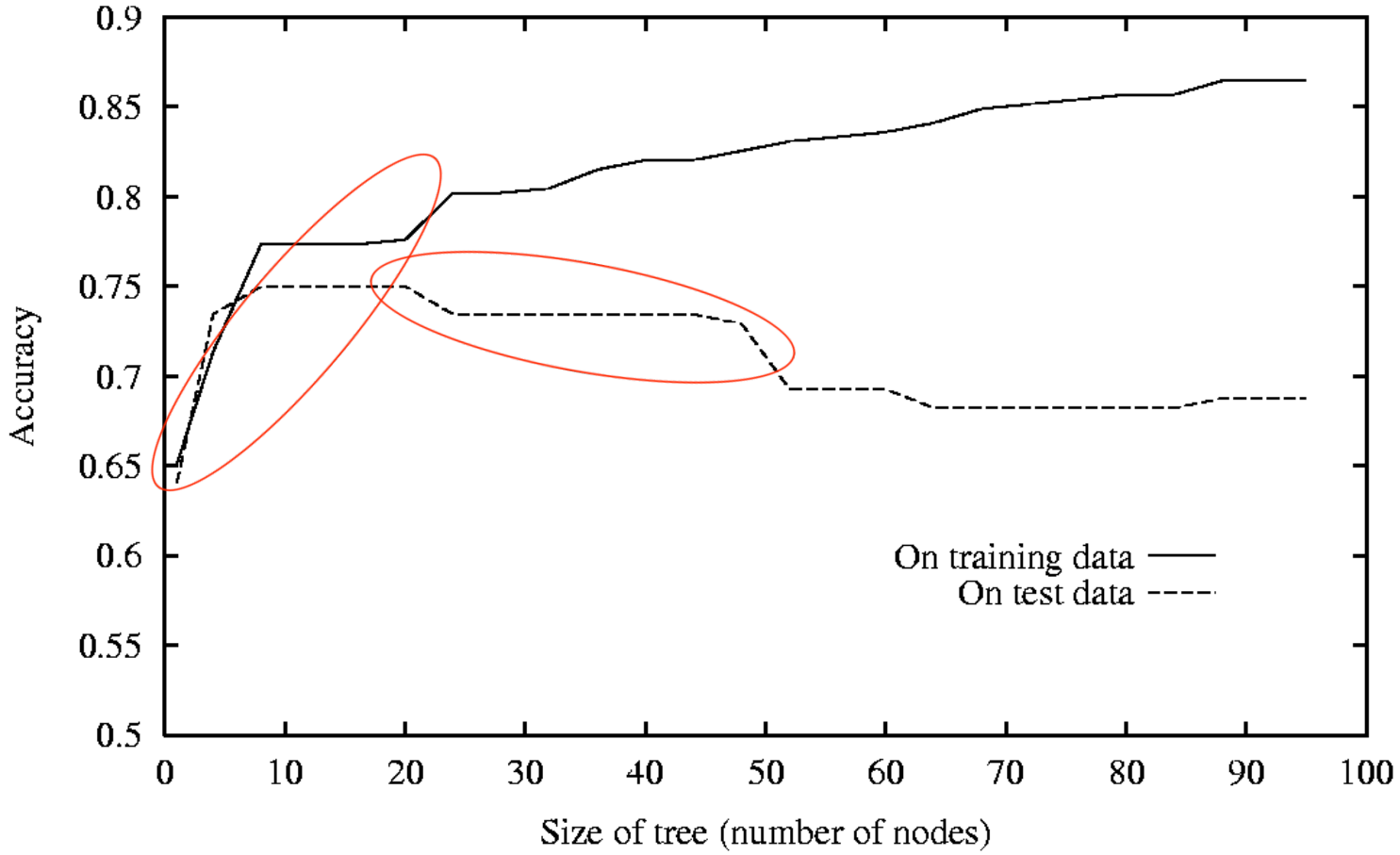
- What if an example x is missing the value an attribute A?

- Simple solution:
  - Use the most common value among examples at node n.
  - Or use the most common value among examples at node n that have classification c(x)

- More complex, probabilistic approach
  - Assign a probability to each of the possible values of A based on the observed frequencies of the various values of A
  - Then, propagate examples down the tree with these probabilities.
  - The same probabilities can be used in classification of new instances (used in C4.5)

Overfitting (Aşırı Uyum)

Makine öğrenmesinde tipik gözlem: Modelin karmaşıklığı (complexity) arttıkça, eğitim setinin öğrenilmesi kolaylaşır; ama genelleme geneleme başarısı bir yerden sonra düşer.

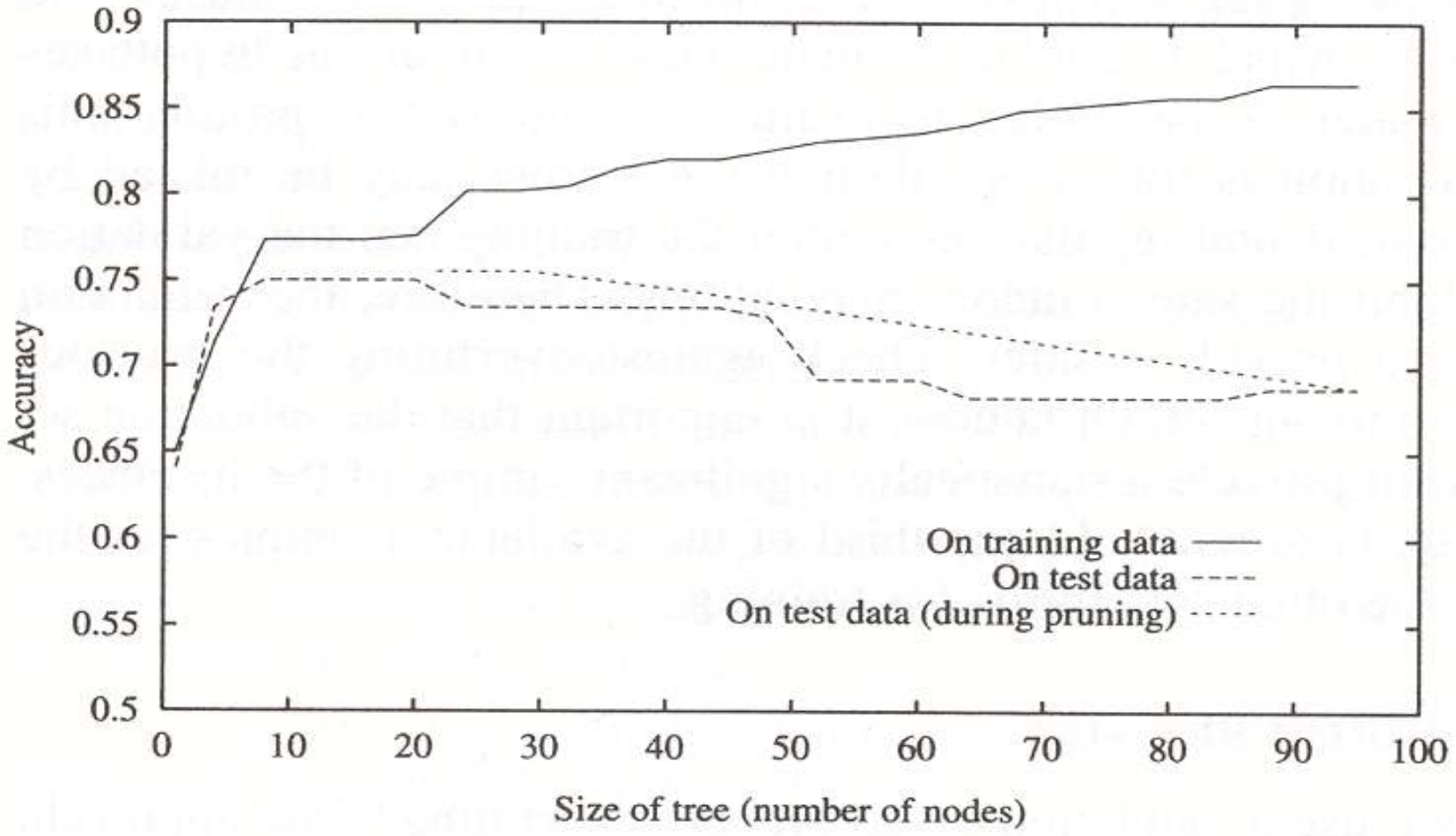


Aşırı Uyum Nasıl Engellenir?

1. **Eğitimi erken durdurma**
   - Örn. bölünecek bir uç düğüme en az k eleman düşmelidir diyerek.
2. **Budama** (tercih)
   - Ağacı tam büyütüp, sonunda bazı dallarını budamak. Alt ağacı budanan düğüme, alt ağaçtaki en yaygın etiket verilir.

- Önceden budama (büyümeyi durdurma) veya sonradan budama yapılacaksa da, en “iyi” ağacı nasıl seçeceğimiz sorusu önemlidir:
  - Ayrı bir sınama kümesi üzerinde test gerekli.
  - Eğitim kümesinde budamanın etkisi sınansa ne olur?

Reduced error pruning



Strengths and Advantages of Decision Trees

- Rule extraction from trees
  - A decision tree can be used for feature extraction (e.g. seeing which features are useful)
- Interpretability: human experts may verify and/or discover patterns
- It is a compact and fast classification method